NASA CR-134880

EFFECT OF NOZZLE NONLINEARITIES UPON NONLINEAR STABILITY OF LIQUID PROPELLANT ROCKET MOTORS

BY

M.S. Padmanabhan
E.A. Powell
B.T. Zinn

GEORGIA INSTITUTE OF TECHNOLOGY

prepared for

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

NASA Lewis Research Center
Grant NGR 11-002-179
Richard J. Priem, Project Manager

NOTICE

This report was prepared as an account of Government-sponsored work. Neither the United States, nor the National Aeronautics and Space Administration (NASA), nor any person acting on behalf of NASA:

A.) Makes any warranty or representation, expressed or implied, with respect to the accuracy, completeness, or usefulness of the information contained in this report, or that the use of any information, apparatus, method, or process disclosed in this report may not infringe privately-owned rights; or

B.) Assume any liabilities with respect to the use of, or for damages resulting from the use of, any information, apparatus, method or process disclosed in this report.

As used above, "person acting on behalf of NASA" includes any employee or contractor of NASA, or employee of such contractor, to the extent that such employee or contractor of NASA or employee of such contractor prepares, disseminates, or provides access to any information pursuant to his employment or contract with NASA, or his employment with such contractor.

Requests for copies of this report should be referred to

National Aeronautics and Space Administration
Scientific and Technical Information Facility
P. O. Box 33
College Park, Md. 20740

1. Report No.
NASA CR-134880

2. Government Accession No.

3. Recipient's Catalog No.

4 Title and Subtitle
Effect of Nozzle Nonlinearities Upon Nonlinear Stability of Liquid Propellant Rocket Motors

5 Report Date
October, 1975

6. Performing Organization Code

7. Author(s)
M. S. Padmanabhan, E. A. Powell, and B. T. Zinn

8 Performing Organization Report No.

9. Performing Organization Name and Address
Georgia Institute of Technology
Atlanta, Georgia 30332

10 Work Unit No

11. Contract or Grant No.
NGR 11-002-179

12. Sponsoring Agency Name and Address
National Aeronautics and Space Administration
Washington, D. C. 20546

13 Type of Report and Period Covered
Contractor Report

14. Sponsoring Agency Code

15. Supplementary Notes
Project Manager, Richard J. Priem, Chemical Propulsion Division, NASA Lewis Research Center, Cleveland, Ohio.

16 Abstract
A three-dimensional, nonlinear nozzle admittance relation is developed by solving the wave equation describing finite-amplitude oscillatory flow inside the subsonic portion of a choked, slowly-convergent axisymmetric nozzle. This nonlinear nozzle admittance relation is then used as a boundary condition in the analysis of nonlinear combustion instability in a cylindrical liquid rocket combustor. In both nozzle and chamber analyses solutions are obtained using the Galerkin method with a series expansion consisting of the first tangential, second tangential, and first radial modes. Using Crocco's time-lag model to describe the distributed unsteady combustion process, combustion instability calculations are presented for different values of the following parameters: (1) time-lag, (2) interaction index, (3) steady-state Mach number at the nozzle entrance, and (4) chamber length-to-diameter ratio. In each case, limit-cycle pressure amplitudes and waveforms are shown for both linear and nonlinear nozzle admittance conditions. These results show that when the amplitudes of the second tangential and first radial modes are considerably smaller than the amplitude of the first tangential mode the inclusion of nozzle nonlinearities has no significant effect on the limiting amplitude and pressure waveforms.

17. Key Words (Suggested by Author(s))
Combustion Instability
Liquid Rockets
Exhaust Nozzles
Method of Weighted Residuals

18. Distribution Statement
Unclassified - unlimited

19 Security Classif. (of this report)	20. Security Classif (of this page)	21 No. of Pages	22 Price*
Unclassified	Unclassified	141	$3.00

*For sale by the National Technical Information Service, Springfield, Virginia 22151

FOREWORD

The research described herein, which was conducted at Georgia Institute of Technology, was supported by NASA Grant No. NGR-11-002-179. The work was done under the management of the NASA Project Manager, Dr. Richard J. Priem, Chemical Rockets Division, NASA-Lewis Research Center.

ABSTRACT

A three-dimensional, nonlinear nozzle admittance relation is developed by solving the wave equation describing finite-amplitude oscillatory flow inside the subsonic portion of a choked, slowly-convergent axisymmetric nozzle. This nonlinear nozzle admittance relation is then used as a boundary condition in the analysis of nonlinear combustion instability in a cylindrical liquid rocket combustor. In both nozzle and chamber analyses solutions are obtained using the Galerkin method with a series expansion consisting of the first tangential, second tangential, and first radial modes. Using Crocco's time-lag model to describe the distributed unsteady combustion process, combustion instability calculations are presented for different values of the following parameters: (1) time-lag, (2) interaction index, (3) steady-state Mach number at the nozzle entrance, and (4) chamber length-to-diameter ratio. In each case, limit-cycle pressure amplitudes and waveforms are shown for both linear and nonlinear nozzle admittance conditions. These results show that when the amplitudes of the second tangential and first radial modes are considerably smaller than the amplitude of the first tangential mode the inclusion of nozzle nonlinearities has no significant effect on the limiting amplitude and pressure waveforms.

TABLE OF CONTENTS

Page

TABLE OF CONTENTS (Continued)

LIST OF ILLUSTRATIONS

Figure Page

LIST OF ILLUSTRATIONS (Continued)

Figure Page

SUMMARY

Recently, a three-dimensional, nonlinear nozzle admittance relation has been developed. In this analysis, the wave equation for an axisymmetric, choked nozzle was solved using the Galerkin method with an approximating series solution for the velocity potential perturbation which was compatible with recent nonlinear combustion instability theories. Assuming that the amplitude of the fundamental mode is considerably larger than the amplitudes of the remaining modes in the series expansion, nonlinear admittance coefficients were determined as a function of the frequency and amplitude of the fundamental mode.

The nonlinear nozzle theory was then applied in the analysis of nonlinear combustion instability in a cylindrical combustor with uniform injection of propellants at one end and a slowly converging nozzle at the other end. The distributed unsteady combustion process was described by means of Crocco's time-lag model. The Galerkin method was used to determine the behavior of the pressure perturbation in the rocket combustor, where the nonlinear nozzle admittance relation was used as the boundary condition at the nozzle end of the chamber. In these computations, a three-mode series expansion consisting of the first tangential (1T), second tangential (2T), and first radial (1R) modes was used. Since the amplitude and frequency of the 1T mode upon which the nonlinear nozzle admittances depend are not known a priori, an iterative solution technique was used.

Combustion instability calculations have been made for different values of the following parameters: (1) time-lag, (2) interaction index, (3) steady state Mach number at the nozzle entrance, and (4) chamber length-to-diameter ratio. In each case limit-cycle pressure amplitudes and waveforms were obtained with both the linear and nonlinear nozzle admittances. These results show that under the assumptions of the analysis the effect of nozzle nonlinearities can be safely neglected in nonlinear stability calculations.

INTRODUCTION

Various aerospace propulsion devices, such as liquid and solid propellant rocket motors and air breathing jet engines, are often subject to combustion instabilities which are detrimental to the performance and safety of operation of these devices. In order to design stable engines, capabilities for a priori determination of the linear and nonlinear characteristics of the instability and the range of operating conditions for which these engines are dynamically stable must be acquired. In order to perform such an analysis, the behavior of the exhaust nozzle under oscillatory flow conditions must be understood. In particular, it is necessary to know how a wave generated in the combustion chamber is partially transmitted and partially reflected at the nozzle entrance. The information is usually expressed as a boundary condition (usually referred to as a Nozzle Admittance Relation) that must be satisfied at the nozzle entrance.

Before such a boundary condition can be derived, the nature of the wave motion inside the nozzle must be investigated. The behavior of oscillations in a converging-diverging supercritical nozzle was first treated by Tsien[1] who considered the case in which the oscillation of the incoming flow is one-dimensional and isothermal. Crocco[2,3] extended Tsien's work to cover the more general cases of non-isothermal one- and three-dimensional oscillations. The analyses of Tsien and Crocco are both restricted to small-amplitude (i.e., linear) oscillations. More recently, a nonlinear nozzle theory has been developed by Zinn and Crocco [4,5,6] who extended the previous linear theories to the investigation of the behavior of finite-amplitude waves.

In recent studies conducted by Zinn, Powell, and Lores, theories were developed which describe the nonlinear behavior of longitudinal[7,8] and transverse[9,10] instabilities in liquid-propellant rocket chambers with quasi-steady nozzles. These theories have now been extended to situations in which the instabilities are three-dimensional and the rocket combustors are attached to conventional nozzles[11]. All of these theories have successfully predicted the transient behavior, nonlinear waveforms, and limit-cycle amplitudes of longitudinal and tangential instabilities in unstable motors.

In order to assess the importance of nozzle nonlinearities upon the nonlinear stability characteristics of various propulsion devices, a new nonlinear nozzle theory is needed for the following reasons. First, the nonlinear analysis of Zinn[5,6] is mathematically complicated and requires considerable computer time. For this reason, Zinn's analysis has never been used to perform actual computations of the wave structure in the nozzle or the nonlinear nozzle response. Secondly, the nonlinear nozzle admittance relation developed by Zinn is not compatible with the recently developed nonlinear combustion theories (see References 7 through 11). Consequently, a linear nozzle boundary condition or a short nozzle (quasi-steady) assumption had to be used in all of the nonlinear combustion instability theories developed to date. The use of a linear nozzle boundary condition in these nonlinear theories was justified by assuming that under the conditions of moderate amplitude oscillations and small mean flow Mach number the effect of nozzle nonlinearities is of higher order and can be neglected. Thus a nonlinear nozzle analysis is needed to determine the validity of this assumption. Furthermore, in the case of transverse instabilities the "linear" nozzle has been known to exert a destabilizing effect; in these cases it is especially important to know how nonlinearities affect the nozzle behavior.

Thus a nonlinear nozzle admittance relation has been developed and has been applied as a boundary condition in the recently-developed nonlinear combustion instability theories. The development of this theory, its application in the chamber stability analysis, and typical results for liquid-propellant rockets will be described in the following sections.

SYMBOLS

$A_p(\varphi)$	axially dependent amplitude functions in Eq. (4)
$B_p(t)$	time dependent amplitude functions in Eq. (18)
$B_N(\tilde{\Phi}')$	nozzle boundary residual (see Eq. (10))
b_p	complex axial acoustic eigenvalue
c	dimensionless sonic velocity, c^*/c^*_o

$E_N(\tilde{\Phi}')$	residual of Eq. (2)
$E_c(\tilde{\Phi}')$	residual of Eq. (17)
i	imaginary unit, $\sqrt{-1}$
J_m	Bessel function of the first kind, order m
k_p	multiple of fundamental frequency
m	azimuthal mode number
n	pressure interaction index
p	dimensionless pressure, $\gamma p^*/\rho_o^* c_o^{*2}$
r	dimensionless radial coordinate, r^*/r_c^*
r_c^*	chamber radius
S_{mn}	dimensionless transverse mode acoustic frequency
t	dimensionless time, $\frac{t}{(r_c^*/c_o^*)}$
u	dimensionless axial velocity, u^*/c_o^*
Y_p	linear admittance for the pth mode
z	dimensionless axial coordinate, z^*/r_c^*
γ	specific heat ratio
Γ_p	nonlinear admittance for the pth mode
ζ_p	linear admittance function
θ	azimuthal coordinate
ρ	dimensionless density, ρ^*/ρ_o^*
τ	dimensionless pressure sensitive time lag, $\frac{\tau^*}{(r_c^*/c_o^*)}$

φ	steady state potential function
Φ	velocity potential
ψ	steady state stream function
ω	dimensionless frequency

Subscripts:

e	evaluated at the nozzle entrance
n	radial mode number
r, i	real and imaginary parts of a complex quantity, respectivel
w	evaluated at the nozzle wall
o	stagnation quantity
$\varphi,\psi,r,\theta,z,t$	partial differentiation with respect to $\varphi,\psi,r,\theta,z,$ or t, respectively

Superscripts:

$(\ )'$	perturbation quantity
$(\bar{~})$	steady state quantity
$(\ )^*$	dimensional quantity, complex conjugate
$(\tilde{~})$	approximate solution

NOZZLE ANALYSIS

The development of the nonlinear nozzle theory is described in detai in Refs. (12) and (13), therefore only a brief summary will be given in this section.

Development of the Nozzle Wave Equation

As in the Zinn-Crocco analysis,[5,6] finite-amplitude, periodic oscillations were assumed to occur inside the slowly convergent, subsonic portion of an axisymmetric nozzle operating in the supercritical range. The flow in the nozzle was assumed to be adiabatic and inviscid and to have no body forces or chemical reactions. The fluid was also assumed to be calorically perfect. Under the further assumption of isentropic and irrotational flow the continuity and momentum equations were combined to obtain the following equation which describes the behavior of the velocity potential:

$$\nabla^2\Phi - \Phi_{tt} = 2\nabla\Phi\cdot\nabla\Phi_t + (\gamma-1)\ \Phi_t\ \nabla^2\Phi \qquad (1)$$

$$+ \frac{\gamma-1}{2}\ (\nabla\Phi\cdot\nabla\Phi)\ \nabla^2\Phi + \frac{1}{2}\ \nabla\Phi\cdot\nabla(\nabla\Phi\cdot\nabla\Phi)$$

These equations are consistent with those used in the second-order nonlinear combustion instability theory developed by Powell, Zinn, and Lores (see References 7 and 10).

A nozzle wave equation was obtained from Eq. (1) by expressing the velocity potential as the sum of a steady state and a perturbation (i.e. $\Phi = \bar{\Phi} + \Phi'$), introducing the (φ,ψ,θ) coordinate system used by Zinn and Crocco[5,6] (see Figure 1), assuming a slowly convergent nozzle and one-dimensional mean flow, and neglecting third order nonlinear terms. This wave equation is given by:

$$E_N\left(\Phi'\right) = f_1(\varphi)\Phi'_{\varphi\varphi} - f_2(\varphi)\Phi'_{\varphi} + f_3(\varphi)\left[2\left(\psi\Phi'_{\psi\psi} + \Phi'_{\psi}\right) + \frac{1}{2\psi}\ \Phi'_{\theta\theta}\right] \qquad (2)$$

$$- 2\ \Phi'_{\varphi t} + f_4(\varphi)\Phi'_t - \frac{1}{\bar{u}^2}\ \Phi'_{tt}$$

$$- \left\{2\ \Phi'_{\varphi}\ \Phi'_{\varphi t} + \frac{4\rho}{\bar{u}}\ \psi\Phi'_{\psi}\ \Phi'_{\psi t} + \frac{\bar{\rho}}{\bar{u}\psi}\ \Phi'_{\theta}\ \Phi'_{\theta t}\right.$$

$$+ (\gamma+1)\bar{u}^2 \, \Phi'_\varphi \, \Phi'_{\varphi\varphi} + 2 \, \bar{\rho}\bar{u} \, \psi\Phi'_\psi \, \Phi'_{\psi\varphi} + \frac{\bar{\rho}\bar{u}}{2\psi} \, \Phi'_\theta \, \Phi'_{\theta\varphi}$$

$$+ f_5(\varphi) \, (\Phi'_\varphi)^2 + f_6(\varphi) \, \psi(\Phi'_\psi)^2 + f_6(\varphi) \, \frac{1}{4\psi} \, (\Phi'_\theta)^2$$

$$+ (\gamma-1) \, \Phi'_{\varphi\varphi} \, \Phi'_t - f_4(\varphi) \, \Phi'_\varphi \, \Phi'_t$$

$$+ (\gamma-1) \, \frac{\bar{\rho}}{\bar{u}} \left[2 \left(\psi\Phi'_{\psi\psi} + \Phi'_\psi \right) + \frac{1}{2\psi} \, \Phi'_{\theta\theta} \right] \Phi'_t$$

$$+ (\gamma-1)\bar{\rho}\bar{u} \left[2 \left(\psi\Phi'_{\psi\psi} + \Phi'_\psi \right) + \frac{1}{2\psi} \, \Phi'_{\theta\theta} \right] \Phi'_\varphi \Big\} = 0$$

where

$$f_1(\varphi) = \bar{c}^2 - \bar{u}^2 \qquad (3)$$

$$f_2(\varphi) = \frac{1}{\bar{c}^2} \, \frac{d\bar{u}^2}{d\varphi}$$

$$f_3(\varphi) = \frac{\rho\bar{c}^2}{\bar{u}}$$

$$f_4(\varphi) = - \frac{(\gamma-1)}{2\bar{c}^2} \, \frac{d\bar{u}^2}{d\varphi}$$

$$f_5(\varphi) = \frac{3}{2} \left[1 + \frac{\gamma-1}{2} \, \frac{\bar{u}^2}{\bar{c}^2} \right] \frac{d\bar{u}^2}{d\varphi}$$

$$f_6(\varphi) = \frac{\rho}{2\bar{u}} \left[1 - (2-\gamma) \, \frac{\bar{u}^2}{\bar{c}^2} \right] \frac{d\bar{u}^2}{d\varphi}$$

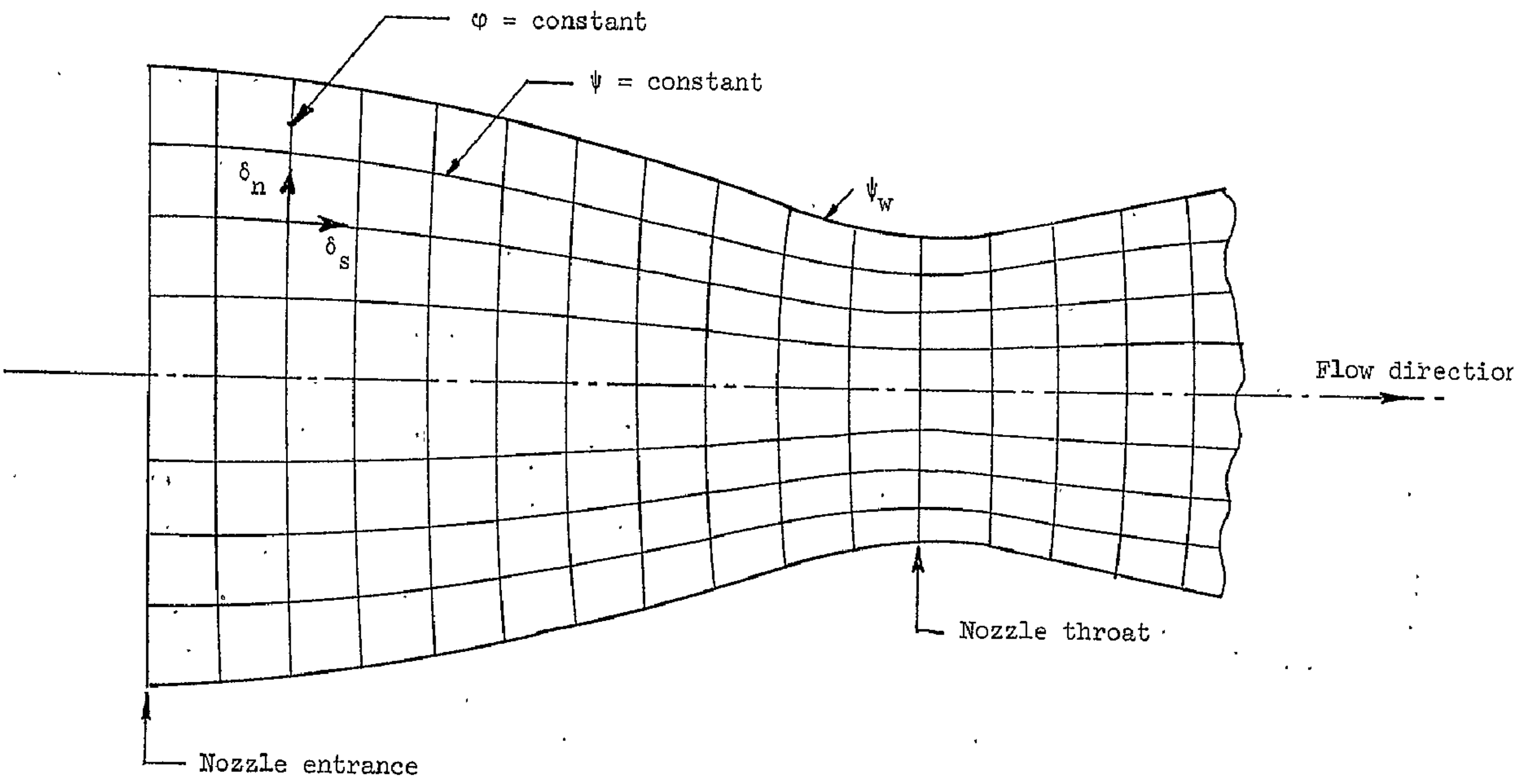


Figure 1. Coordinate System used for the Solution of the Oscillatory Nozzle Flow.

Method of Solution

In the nonlinear combustion instability theories developed by Powell and Zinn (see Refs. 7 - 11) the governing equations were solved by means of an approximate solution technique known as the Galerkin Method, which is a special case of the Method of Weighted Residuals[14,15]. In these investigations it was shown that the Galerkin Method could be successfully applied in the solution of nonlinear combustion instability problems; its application was straightforward and it required relatively little computation time. Thus the Galerkin Method was also used in the nozzle analysis to determine the nonlinear nozzle admittance relation.

The first step in using the Galerkin Method in the solution of the wave equation (i.e., Eq. (2)) was to express the velocity potential, Φ', as an approximating series expansion. The structure of this series expansion was guided by the experience gained in the nonlinear nozzle admittance studies performed by Zinn and Crocco (see Ref. 5) as well as in the nonlinear combustion instability analyses of Powell and Zinn (see Ref. 10). Thus the velocity potential was expressed as follows:

$$\tilde{\Phi}' = \sum_{p=1}^{N} \left\{A_p(\varphi)\ \cos(m\theta) J_m\left[S_{mn}\left(\frac{\psi}{\psi_w}\right)^{\frac{1}{2}}\right] e^{ik_p \omega t}\right\} \tag{4}$$

where the functions $A_p(\varphi)$ are unknown complex functions of the axial variable φ, and θ- and ψ-dependent eigenfunctions were determined from the first-order (i.e., linear) solutions by Zinn[5]. For each value of the index p, there corresponds the mode numbers $m(p)$ and $n(p)$ as well as the number k_p. This correspondence is illustrated in the table below for a three-term expansion consisting of the first tangential (1T), second tangential (2T), and first radial (1R) modes.

Table 1. Three-Mode Expansion

p	m(p)	n(p)	k_p	Mode
1	1	1	1	1T
2	2	1	2	2T
3	0	1	2	1R

In the time-dependence, ω is the fundamental frequency which must be specified and the integer k_p gives the frequency of the higher harmonics. The values of k_p for the various modes appearing in Eq. (4) were determined from the results of the nonlinear combustion instability analysis of Powell and Zinn[10]. For example it was found that, due to nonlinear coupling between modes, the 2T and 1R modes oscillated with twice the frequency of the 1T mode. Thus in Eq. (4) $k_1 = 1$ and $k_2 = k_3 = 2$. The amplitudes and phases of the various modes depend on the axial location (i.e., φ) in the nozzle through the unknown functions $A_p(\varphi)$.

Next the assumed series expansion for Φ' (i.e., Eq. (4)) was substituted into the wave equation (i.e., Eq. (2)) to form the residual, $E_N(\tilde{\Phi}')$. According to the Galerkin method, the residual $E_N(\tilde{\Phi}')$ was required to satisfy the following orthogonality conditions:

$$\int_o^T \int_S E_N(\tilde{\Phi}')\ e^{-ik_j\omega t} \cos m\theta\ J_m\left[S_{mn}\left(\frac{\psi}{\psi_w}\right)^{\frac{1}{2}}\right] dSdt = 0 \qquad (5)$$

$$j = 1, 2, \ldots N$$

where N is the number of terms in the series expansions of the dependent variables. The weighting functions in Eq. (5) correspond to the assumed time and space dependences of the terms that appear in the series expansion.

The time integration is performed over one period of oscillation, $T = 2\pi/\omega$, while the spatial integration is performed over any surface of φ = constant in the nozzle (in Eq. (5) dS indicates an incremental area on this surface).

Evaluating the spatial and temporal integrals in Eq. (5) yielded a system of N nonlinear, second order, coupled, complex ordinary differential equations to be solved for the complex amplitude functions $A_p(\varphi)$. Unfortunately these equations were not quasi-linear; that is, the highest order derivatives appeared in the nonlinear terms. This greatly complicated the numerical solution of these equations, thus an additional approximation was made to obtain a quasi-linear system of equations.

This additional approximation was based on the well-known fact that most transverse instabilities behave like the first tangential (1T) mode. Based on the results of the recent nonlinear combustion instability theory[11], it was assumed that the amplitude of the 1T mode was considerably larger than the amplitudes of the remaining modes in the series solution. Through an order of magnitude analysis correct to second order, the original non-quasilinear system of equations was reduced to the following linear inhomogeneous system of equations:

$$H_1(\varphi)\frac{d^2A_1}{d\varphi^2} + M_1(\varphi)\frac{dA_1}{d\varphi} + N_1(\varphi)A_1(\varphi) = 0 \tag{6}$$

$$H_p(\varphi)\frac{d^2A_p}{d\varphi^2} + M_p(\varphi)\frac{dA_p}{d\varphi} + N_p(\varphi)A_p(\varphi) = I_p\left\{A_1, \frac{dA_1}{d\varphi}, \frac{d^2A_1}{d\varphi^2}\right\}$$

$$p = 2, 3, \ldots N$$

where

$$H_p(\varphi) = \bar{u}^2(\bar{c}^2 - \bar{u}^2) \tag{7}$$

$$M_p(\varphi) = -\bar{u}^2\left[\frac{1}{\bar{c}^2}\;\frac{d\bar{u}^2}{d\varphi} + 2ik_p\omega\right]$$

$$N_p(\varphi) = \left[-\frac{S_p^2}{2\psi_w}\;\bar{\rho}\bar{u}\bar{c}^2 - \frac{\gamma-1}{2}\,ik_p\omega\;\frac{\bar{u}^2}{\bar{c}^2}\;\frac{d\bar{u}^2}{d\varphi} + k_p^2\omega^2\right]$$

and I_p are inhomogeneous terms which are functions of φ and the amplitude of the 1T mode, $A_1(\varphi)$.

It can be seen that the above equations are decoupled with respect to the 1T mode; that is, the solution for A_1 can be obtained independently of the amplitudes of the other modes. Thus to second order the nozzle nonlinearities do not affect the 1T mode. On the other hand, the nozzle nonlinearities influence the amplitudes of the higher modes (i.e., A_2 and A_3) by means of the inhomogeneous terms in the equations for the higher modes.

Derivation of Admittance Relations

It has been shown (see Refs. (12) and (13)) that the solution of Eq. (6) can be expressed as the sum of a homogeneous solution $A_p^{(h)}$ and a particular solution of the inhomogeneous equation $A_p^{(i)}$ as follows:

$$A_p(\varphi) = K_1A_p^{(h)}(\varphi) + A_p^{(i)}(\varphi) \tag{8}$$

Using this result a nonlinear admittance relation to be used as a boundary condition in nonlinear combustion instability analyses was derived. Noting that the velocity potential $\tilde{\Phi}'$ given by Eq. (5) is a summation of partial potentials Φ_p' where

$$\bar{\Phi}_p' = A_p(\varphi)\ \cos(m\theta)\ J_m\left[S_{mn}\left(\frac{\psi}{\psi_w}\right)\right] e^{ik_p\omega t} \tag{9}$$

a nozzle admittance relation can be written for each of the partial potentials. This is done by introducing Eq. (8) into Eq. (9), taking partial derivatives with respect to z and t and eliminating K_1 between the resulting equations. The resulting admittance relations are given by:

$$B_N(\bar{\Phi}') = \frac{\partial\bar{\Phi}_p'}{\partial z} + \gamma Y_p \frac{\partial\bar{\Phi}_p'}{\partial t} \tag{10}$$

$$+ \bar{u}_e \bar{c}_e^2 \left\{\cos(m\theta)\ J_m\left[S_{mn}\left(\frac{\psi}{\psi_w}\right)^{\frac{1}{2}}\right] e^{ik_p\omega t}\right\} \Gamma_p = 0 \ ,$$

where

$$Y_p = \left(\frac{i\bar{u}_e}{\gamma k_p\omega}\right) \frac{1}{A_p^{(h)}} \frac{dA_p^{(h)}}{d\varphi} \qquad p = 1,2,\ldots N \tag{11}$$

$$\Gamma_p = \frac{1}{\bar{c}^2 A_p^{(h)}}\left[A_p^{(i)} \frac{dA_p^{(h)}}{d\varphi} - A_p^{(h)} \frac{dA_p^{(i)}}{d\varphi}\right] \qquad p = 2,3,\ldots N \tag{12}$$

Equation (10) is the nonlinear nozzle admittance relation to be used as the boundary condition at the nozzle entrance plane in nonlinear stability analyses of rocket combustors. The quantities Yp and Γp are, respectively, the linear and nonlinear admittance coefficients for the pth mode. The nonlinear admittance, Γ_p, represents the effect of nozzle nonlinearities upon the nozzle response, and it is zero when nonlinearities are absent (i.e., for the 1T mode).

It can easily be shown that when the Mach number at the nozzle entrance is small, Eq. (10) can be expressed, correct to second order, as:

$$U_p - Y_p P_p = - \bar{u}_e \bar{c}_e^2 \Gamma_p \tag{13}$$

where U_p and P_p are the φ-dependent amplitudes of the axial velocity and pressure perturbations respectively.

In order to use the admittance relation (Eq. (10) or Eq. (13)) in combustion instability analysis, the admittance coefficients Y_p and Γ_p must be determined for the nozzle under consideration. The equations governing these quantities are readily derived from Eqs. (6) using the definition of Γ_p (i.e., Eq. (12) to obtain:

$$H_p \frac{d\zeta_p}{d\varphi} = - M_p \zeta_p - N_p - H_p \zeta_p^2 \tag{14}$$

$$H_p \frac{d\Gamma_p}{d\varphi} = \left(- H_p \zeta_p + \frac{\gamma - 1}{2\bar{c}^2} \frac{d\bar{u}^2}{d\varphi} H_p - M_p \right) \Gamma_p - \frac{I_p}{\bar{c}^2} \tag{15}$$

where

$$\zeta_p = \frac{1}{A_p^{(h)}} \frac{dA_p^{(h)}}{d\varphi} \tag{16}$$

Calculation of the Nozzle Response

To obtain the nozzle response for any specific nozzle, Eqs. (14) and (15) are solved in the following manner. As pointed out earlier, the nonlinear terms vanish for the 1T mode (i.e., $\Gamma_1 = 0$, $I_1 = 0$) and it is only necessary to solve Eq. (14) to obtain ζ_1 (and hence Y_1) at the nozzle entrance. Since Eq. (14) does not depend on the higher modes, it can be solved independently for ζ_1. Once ζ_1 has been determined both Eqs. (14)

and (15) must be solved for the other modes. In order to do this, the amplitude $A_1(\varphi)$ must be determined since Eq. (15) depends on $A_1(\varphi)$ and its derivatives through $I_p(\varphi)$. Once $\zeta_1(\varphi)$ is known, $A_1(\varphi)$ is determined by numerically integrating Eq. (16) where the constant of integration is determined by the specified value of the pressure amplitude $|p_1|$ (of the 1T mode) at the nozzle entrance. The value of A_1 thus found is introduced into Eq. (15) which is then solved for Γ_p.

Since Eqs.(14) and (15) are first order ordinary differential equations, the numerical integration of these equations must start at some initial point where the initial conditions are known, and terminate at the nozzle entrance where the admittance coefficients Y_p and Γ_p are needed. Since the equations are singular at the throat, the integration is initiated at a point that is located a short distance upstream of the throat. The needed initial conditions are obtained by expanding the dependent variables in a Taylor series about the throat ($\varphi = 0$).

In Eqs. (14) and (15), the quantities H_p, M_p, N_p and I_p are functions of the steady-state flow variables in the nozzle and these must be computed before performing the numerical integration to obtain ζ_p and Γ_p. For a specified nozzle profile, the steady-state quantities are computed by solving the quasi-one-dimensional isentropic steady-state equations for the nozzle flow. Figure 2 shows the nozzle profile used in these computations. All of the length variables have been non-dimensionalized with respect to the radius of the combustion chamber to which the nozzle is attached, and hence $r_c = 1$. At the throat r_{th} is fixed by the Mach number at the nozzle entrance plane. The nozzle profile is smooth and is completely specified by r_{cc}, r_{ct} and θ_1, which are respectively the radius of curvature at the chamber, radius of curvature at the throat and slope of the central conical section. The steady-state equations are integrated using equal steps in steady-state potential φ by beginning at the throat and continuing to the nozzle entrance where the radius of the wall equals 1.

A computer program, NOZADM, has been developed to numerically solve Eqs. (14) - (16) and calculate the linear and nonlinear nozzle admittances. A computer code and description of this program is given in Appendix A.

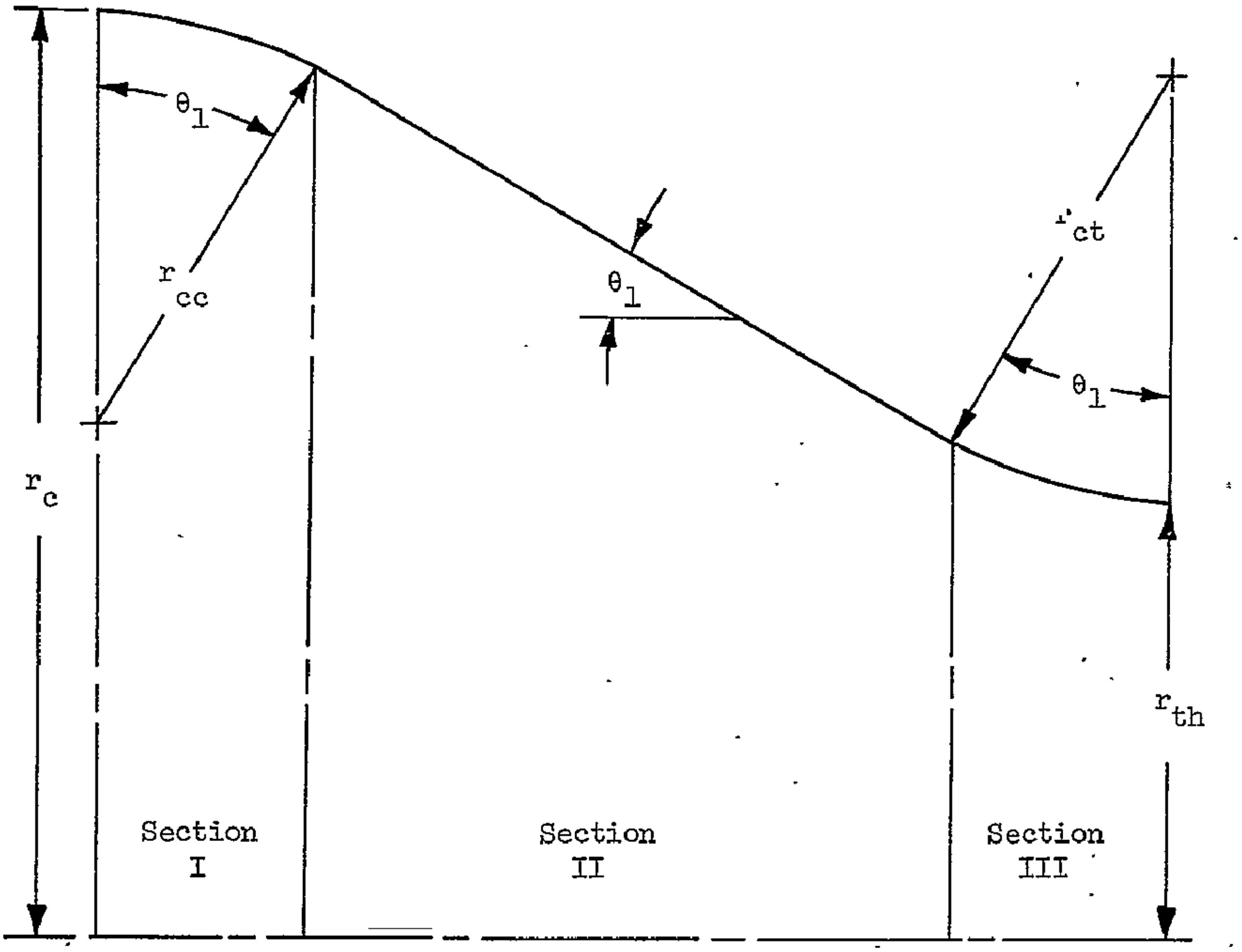


Figure 2. Nozzle Profile Used in Calculating Admittances.

COMBUSTION INSTABILITY ANALYSIS

Combustion Chamber Model

This section describes the application of the nonlinear nozzle admittance theory developed in the previous section to the analysis of combustion instability in a liquid-propellant rocket combustor. A cylindrical combustor with uniform injection of propellants at one end and a slowly-convergent nozzle at the other end was considered. The liquid propellant rocket motor that was analyzed is shown in Figure 3. The analysis of such a motor for a linear nozzle response is given in Ref. (11).

The oscillatory flow in the combustion chamber is described by the three-dimensional, second-order, potential theory developed in Ref. (11). In this theory the velocity potential Φ must satisfy the following nonlinear partial differential equation:

$$
\begin{aligned}
E_c(\Phi') = {} & \Phi'_{rr} + \frac{1}{r}\Phi'_r + \frac{1}{r^2}\Phi'_{\theta\theta} + \Phi'_{zz} - \Phi'_{tt} \\
& - 2\Phi'_r\Phi'_{rt} - \frac{2}{r^2}\Phi'_\theta\Phi'_{\theta t} - 2\Phi'_z\Phi'_{zt} \\
& - (\gamma-1)\Phi'_t(\Phi'_{rr} + \frac{1}{r}\Phi'_r + \frac{1}{r^2}\Phi'_{\theta\theta} + \Phi'_{zz}) \\
& - 2\bar{u}\Phi'_{zt} - (\gamma+1)\Phi'_t\frac{d\bar{u}}{dz} \\
& + \gamma n\frac{d\bar{u}}{dz}\left[\Phi'_t(r,\theta,z,t) - \Phi'_t(r,\theta,z,t-\bar{\tau})\right] = 0
\end{aligned}
\tag{17}
$$

where Crocco's time-lag $(n - \tau)$ model is used to describe the distributed unsteady combustion process. In the present analysis the linear nozzle boundary condition used in the previous analysis (see Eq. (2) of Ref. 11) was replaced by the nonlinear admittance condition given by Eq. (10).

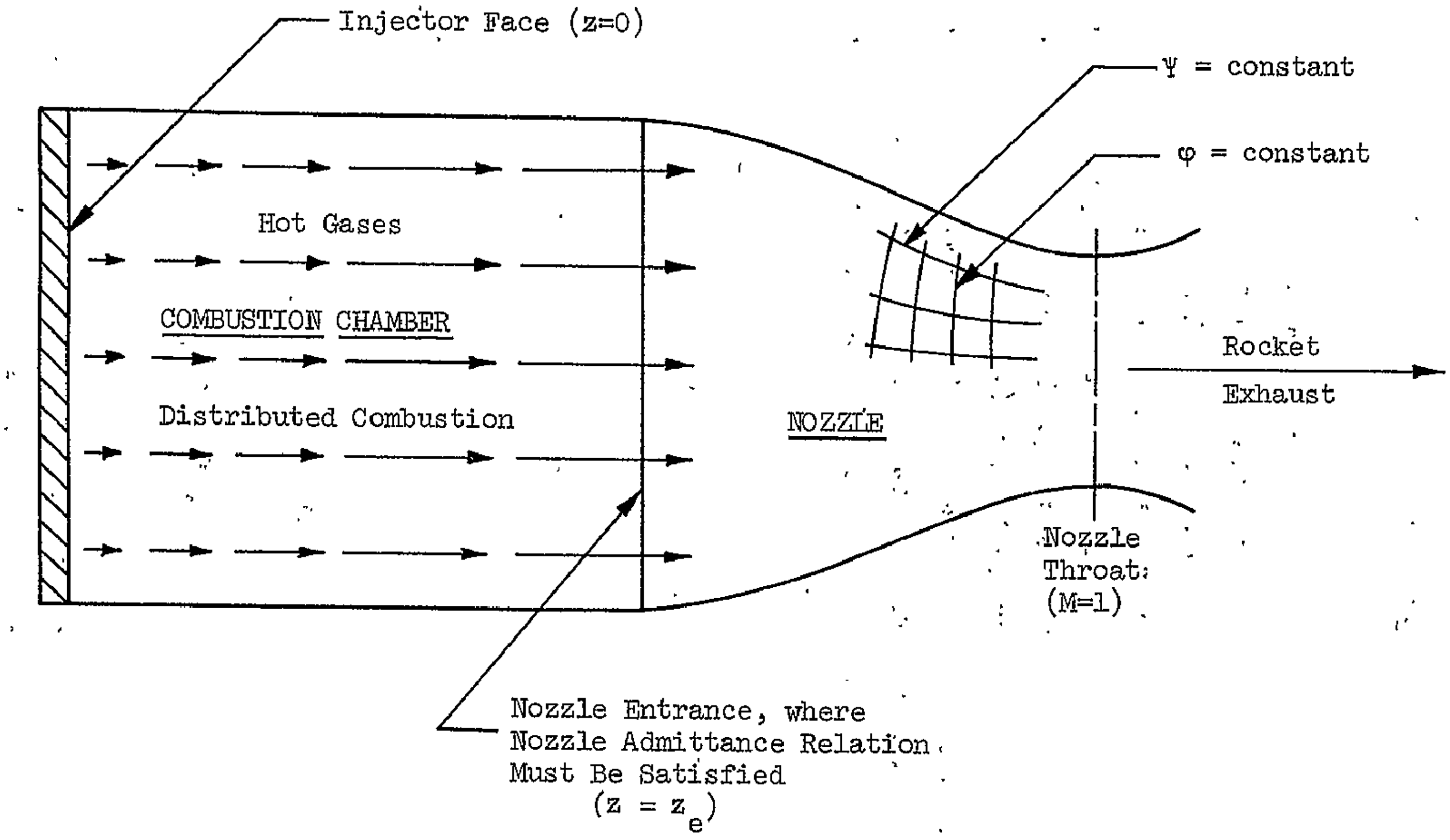


Figure 3. Typical Mathematical Model of a Liquid Rocket Motor.

Application of Galerkin Method

Assuming a series expansion of the form (see Ref. 11):

$$\tilde{\Phi}' = \sum_{p=1}^{N} \Phi_p = \sum_{p=1}^{N} B_p(t)\ \cos(m\theta)\ J_m(S_{mn}r)\ \cosh(ib_p z) \tag{18}$$

the Galerkin method was used to obtain approximate solutions to Eq. (17). In Eq. (18) the radial and azimuthal eigenfunctions are the same as those used in the nozzle analysis (see Eq. 4). Unlike the nozzle analysis where the unknown coefficients $A_p(\varphi)$ were functions of axial location in the nozzle, the unknown coefficients $B_p(t)$ in Eq. (18) are functions of time. The b_p appearing in the axial dependence are the axial acoustic eigenvalues for a chamber with a solid wall boundary condition at the injector end and a linear nozzle admittance condition at the other end.

The unknown amplitudes $B_p(t)$ were determined by substituting the assumed series expansion (i.e., Eq. (18)) into the wave equation (i.e., Eq. (17)) to form the residual $E_c(\tilde{\Phi}')$. Similarly, the series expansion was substituted into the nozzle boundary condition (i.e., Eq. (10)) to obtain the boundary residual $B_N(\tilde{\Phi}')$. The residuals $E_c(\tilde{\Phi}')$ and $B_N(\tilde{\Phi}')$ were required to satisfy the following orthogonality condition (see Ref. 11):

$$\int_0^{z_e}\int_0^{2\pi}\int_0^{1} E_c(\tilde{\Phi}')\ Z_j^*(z)\ \Theta_j(\theta)R_j(r)\ r\,dr\,d\theta\,dz \tag{19}$$

$$-\int_0^{2\pi}\int_0^{1} B_N(\tilde{\Phi}')\ Z_j^*(z_e)\ \Theta_j(\theta)R_j(r)\ r\,dr\,d\theta = 0$$

$$j = 1,2,\ \ldots\ N$$

where the Z_j^* are the complex conjugates of the axial acoustic eigenfunctions appearing in Eq. (18), and Θ_j and R_j are the azimuthal and radial eigenfunctions respectively.

Evaluating the spatial integrals in Eqs. (19) gave the following system of N complex nonlinear equations to be solved for the amplitude functions, $B_p(t)$:

$$\sum_{p=1}^{N} \Big\{ C_0(j,p)\,\frac{d^2B_p}{dt^2} + C_1(j,p)B_p(t) + \Big[C_2(j,p) - nC_3(j,p)\Big]\frac{dB_p}{dt} \qquad (20)$$

$$+\, nC_3(j,p)\,\frac{d[B_p(t-\bar{\tau})]}{dt} + C_4(j,p)e^{ik_p\omega t}\Big\}$$

$$+\sum_{p=1}^{N}\sum_{q=1}^{N}\Big\{D_1(j,p,q)B_p\,\frac{dB_q}{dt} + D_2(j,p,q)B_p\,\frac{dB_q^*}{dt}$$

$$+\, D_3(j,p,q)B_p^*\,\frac{dB_q^*}{dt} + D_4(j,p,q)B_p^*\,\frac{dB_q^*}{dt}\Big\} = 0$$

$$j = 1,2, \;\ldots\; N$$

In the above equation, the term $C_4(j,p)e^{ik_p\omega t}$ results from the presence of nozzle nonlinearities (i.e. the term involving Γ_p in Eq. (10)).

The coefficients appearing in Eq. (20) were determined by evaluating the various integrals of hyperbolic, trigonometric, and Bessel functions that arise from the spatial integrations indicated in the Galerkin orthogonality conditions. These were calculated by the computer program COEFFS3D (Appendix B).

The time-dependent behavior of an engine following the introduction of a disturbance is determined by specifying the form of the initial disturbance and then following the subsequent behavior of the individual modes by numerically integrating Eqs. (20). Once the time-dependence of the individual modes is known, the velocity potential, $\tilde{\Phi}$, is calculated from Eq. (18). The pressure perturbation at any location within the chamber is related to

Φ' by the following second-order momentum equation (see Ref. 11):

$$p' = -\gamma \left[\tilde{\Phi}'_t + \bar{u}\tilde{\Phi}'_z + \frac{1}{2}\left(\tilde{\Phi}'_r\right)^2 + \frac{1}{2r^2}\left(\tilde{\Phi}'_\theta\right)^2 + \frac{1}{2}\left(\tilde{\Phi}'_z\right)^2 - \frac{1}{2}\left(\tilde{\Phi}'\right)_t^2 \right] \qquad (21)$$

Numerical Solution Procedure

Equation (20) is a system of N ordinary differential equations which describes the behavior of the N complex time-dependent functions, $B_p(t)$. Beginning with a sinusoidal initial disturbance, a fourth order Runge-Kutta scheme was employed for the numerical integration of this system of equations. In the present calculations, a three-mode series expansion consisting of the first tangential (1T), second tangential (2T) and first radial mode (1R) was used. This is the same series expansion used in the stability calculations presented in Refs. (10) and (11). The numerical integration of Eqs. (20) is performed by the computer program, LCYC3D, which is described in Appendix C.

The oscillatory flow in the combustor and nozzle are mutually dependent on each other; that is, the combustion chamber analysis requires knowledge of the nozzle admittances, but these nozzle admittances depend on the frequency of oscillation and the pressure amplitude, which can only be determined by the combustion chamber analysis. Thus an iterative solution technique is used. In this procedure, linear nozzle admittances are first calculated for the specified nozzle geometry. Next, the combustion chamber analysis is carried out using these linear nozzle admittances ($\Gamma_p = 0$), and limit-cycle frequency and pressure amplitude of the 1T mode at the nozzle entrance are determined. This information is then used in the nozzle theory to determine the nonlinear nozzle admittances which are used in the chamber analysis to calculate new limit-cycle frequencies and pressure amplitude. If the limit-cycle amplitude obtained with the nonlinear nozzle boundary condition is significantly different from the limit-cycle amplitude obtained with the linear nozzle admittances, new values of the nonlinear admittances are calculated and the process is repeated until the change in limit-cycle amplitude is sufficiently small.

RESULTS AND DISCUSSION

Admittance Coefficients

Computations of the admittance coefficients have been performed using a three-term series expansion consisting of the first tangential, second tangential and first radial modes. An Adam-Bashforth predictor-corrector scheme was used to perform the numerical integration, while the starting values needed to apply this method were obtained using a fourth order Runge-Kutta integration scheme. Computations have been performed for several nozzles, at different frequencies and pressure amplitudes of the first tangential mode.

Figure 4 shows the frequency dependence of the linear admittance coefficients for the 1T, 2T, and 1R modes for a typical nozzle ($\theta_1 = 20^\circ$, $r_{cc} = 1.0$, $r_{ct} = 0.9234$; $M = 0.2$). Here, ω is the frequency of the 1T mode, while the frequency of the 2T and 1R modes is 2ω due to nonlinear coupling. Hence the real parts of the linear admittance coefficients for the 2T and 1R modes actually attain their peak values at a higher frequency than that for the 1T mode. The linear admittance coefficients for the 1T mode are in complete agreement with those calculated previously by Bell and Zinn[16].

The frequency dependence of the nonlinear admittance coefficient for the 2T mode is shown in Figure 5 with pressure amplitude of the 1T mode as a parameter. While the behavior of the linear admittance coefficient depends only upon the frequency of oscillations, the behavior of the nonlinear admittance coefficient is seen to depend also on the amplitude of the 1T mode. The absolute values of both Γ_r and Γ_i increase with increasing pressure amplitude of the 1T mode, which acts as a driving force. It is observed that the absolute values of Γ_r and Γ_i vary with frequency in a manner similar to the absolute values of Y_r and Y_i. The frequency dependence of the nonlinear admittance coefficient for the 1R mode is shown in Figure 6 with pressure amplitude of the 1T mode as a parameter.

Figure 7 shows the effect of pressure amplitude upon the magnitude of the ratio of nonlinear admittance coefficient to the linear admittance coefficient for the 2T and 1R modes respectively. This ratio, $|\Gamma/Y|$, increases with increasing pressure amplitude. In the limiting case of $|p_1|' = 0$, the nonlinear admittance coefficient is zero for all frequencies as expected.

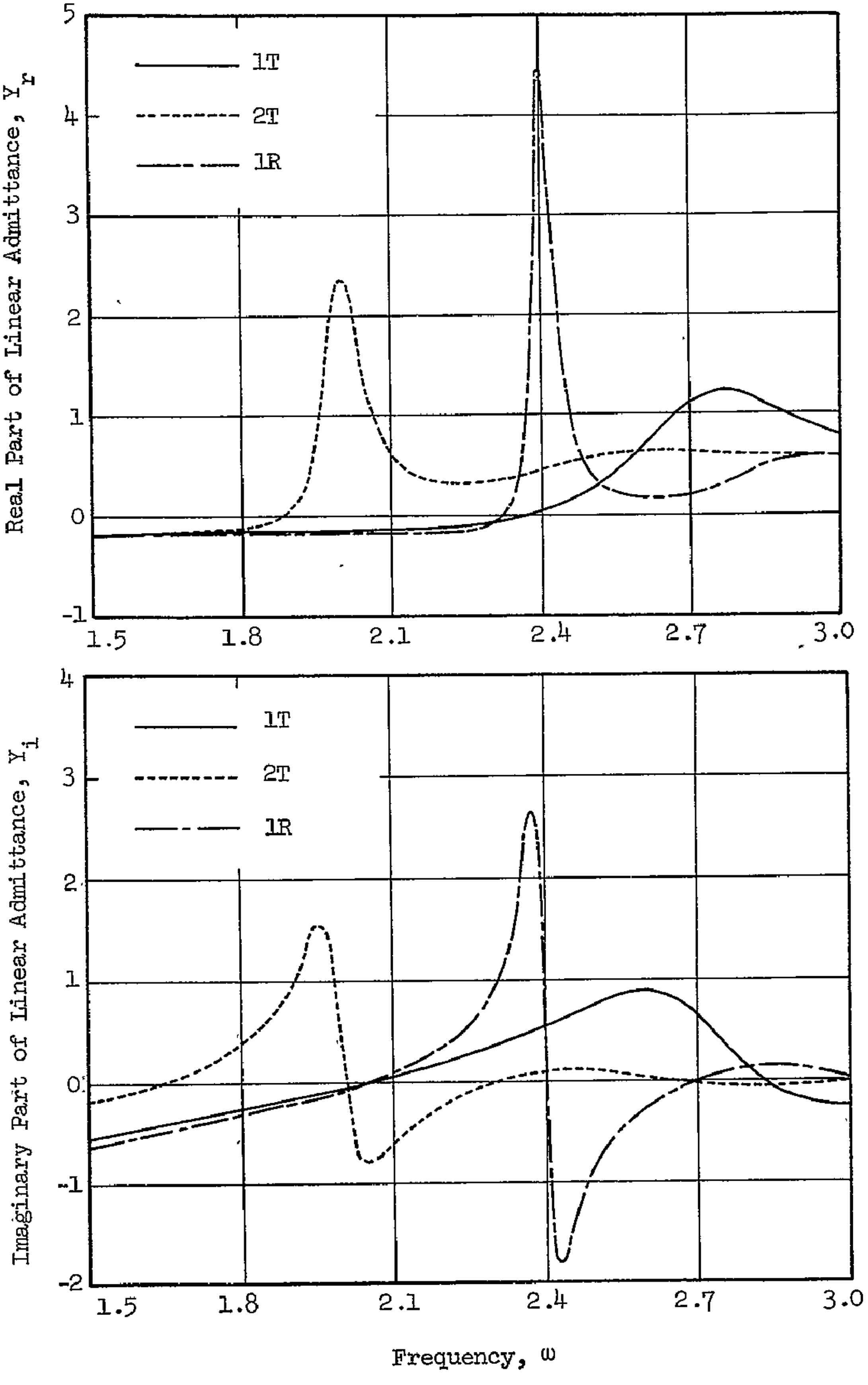


Figure 4. Linear Admittances for the 1T, 2T, and 1R Modes.

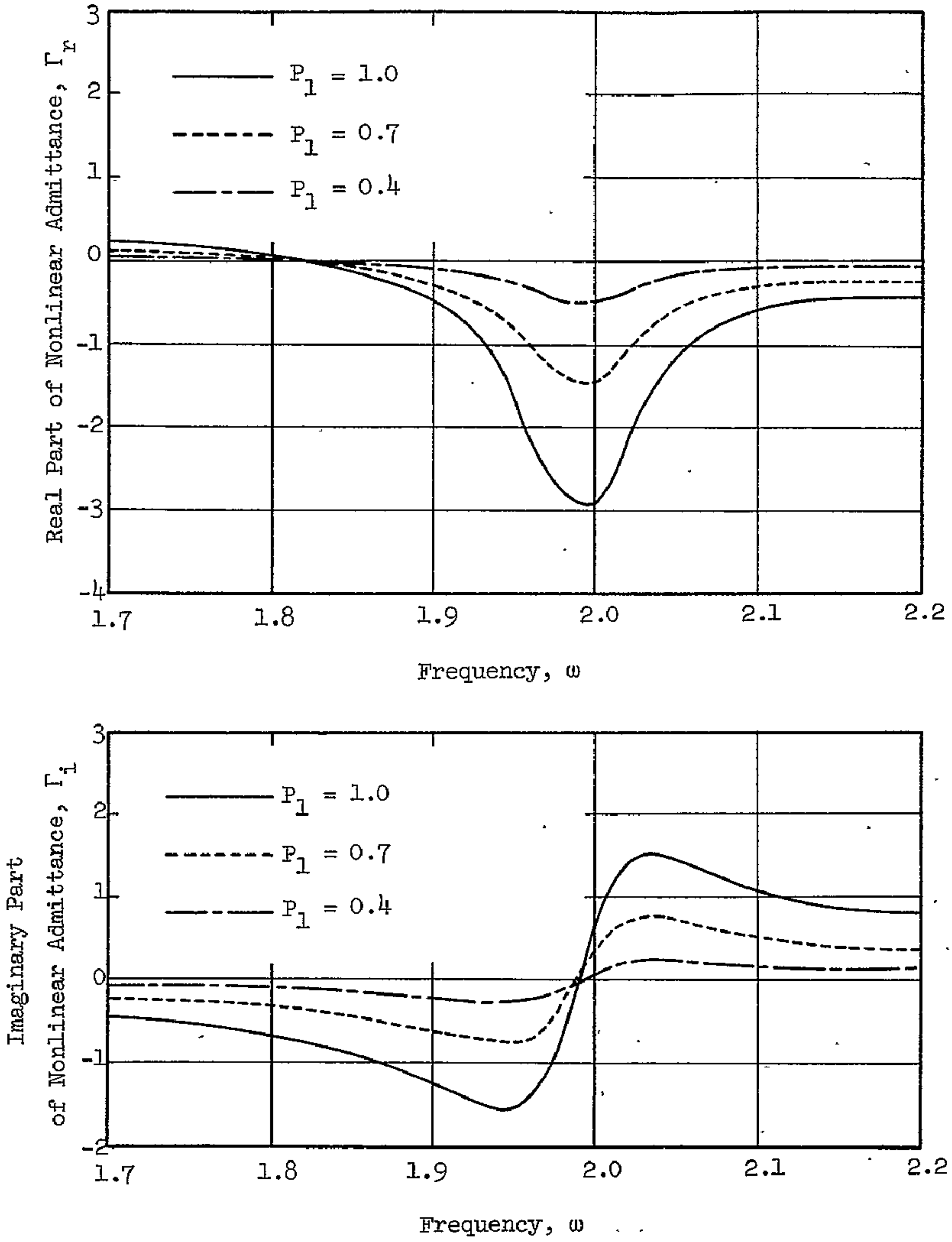


Figure 5. Nonlinear Admittances for the 2T Mode.

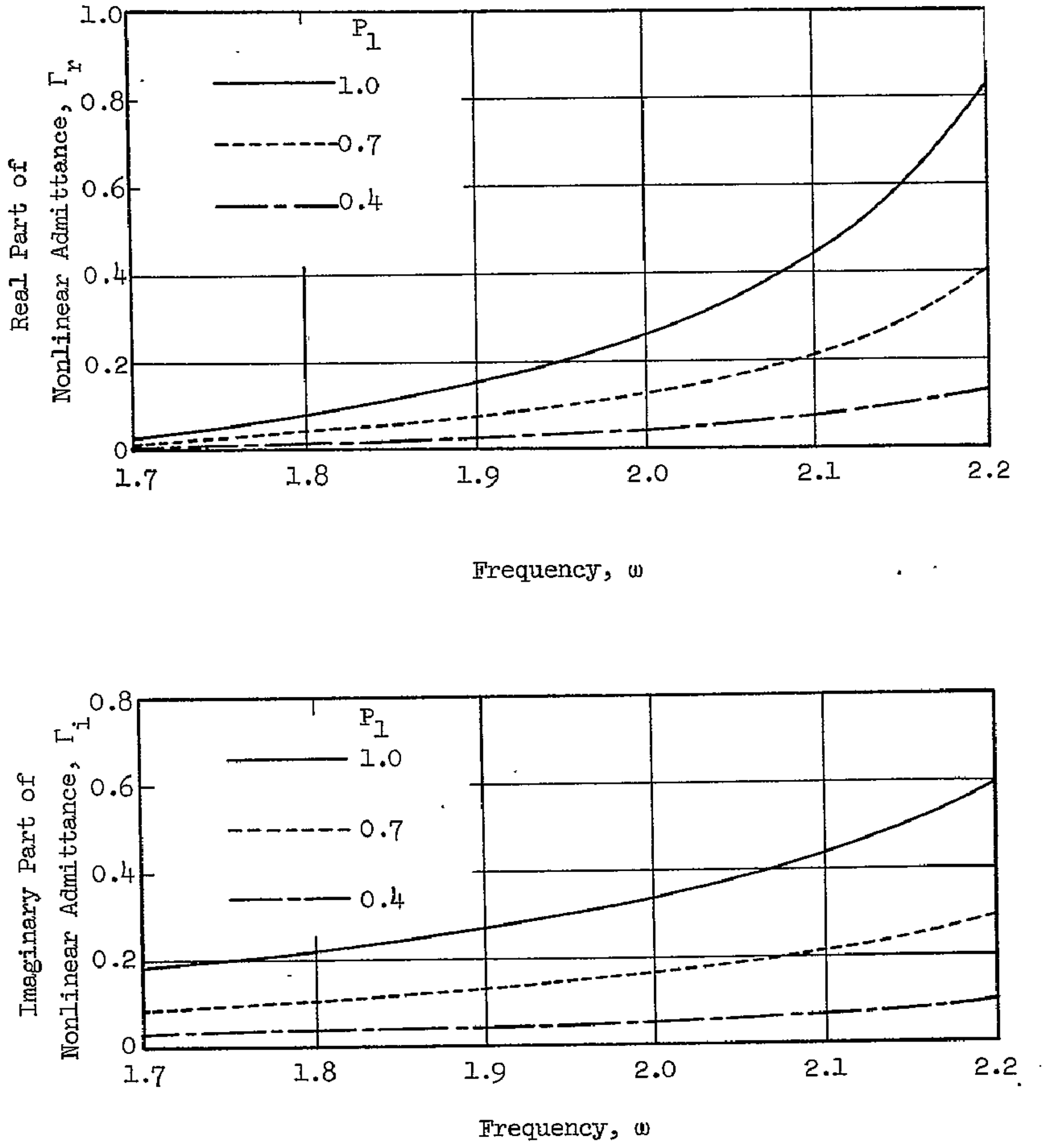


Figure 6. Nonlinear Admittances for the 1R Mode.

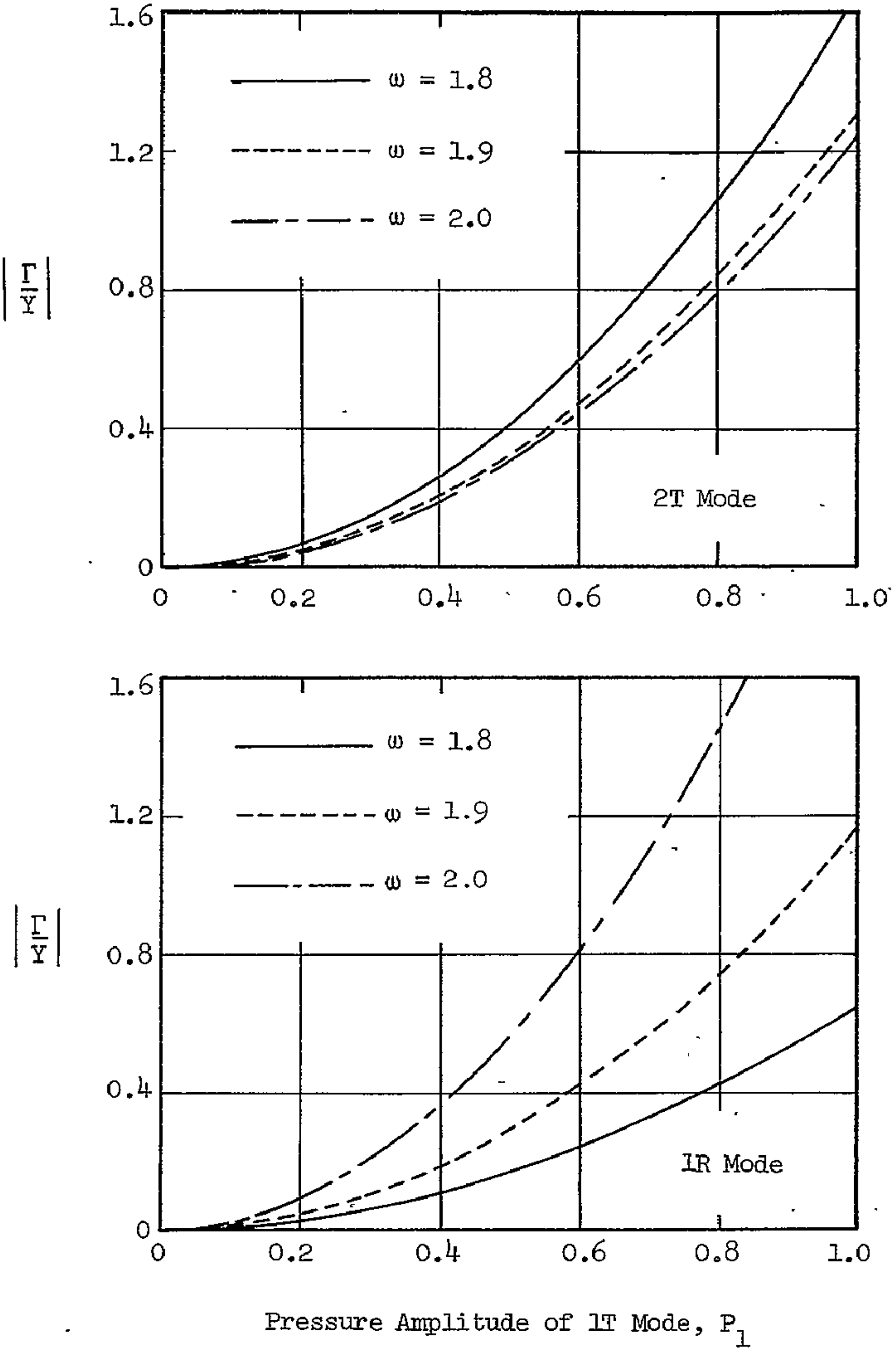


Figure 7. Relative Magnitudes of Linear and Nonlinear Admittances.

Figure 8 shows the influence of entrance Mach number M_e on the nonlinear nozzle admittance coefficients for the 2T and 1R modes respectively. Here the relative magnitudes of the linear and nonlinear admittances (i.e., $|\Gamma/Y|$) are plotted as a function of amplitude of the 1T mode. In each case there is a significant decrease in $|\Gamma/Y|$ with increasing Mach number, thus it appears that the importance of nozzle nonlinearities will be smaller at higher Mach numbers.

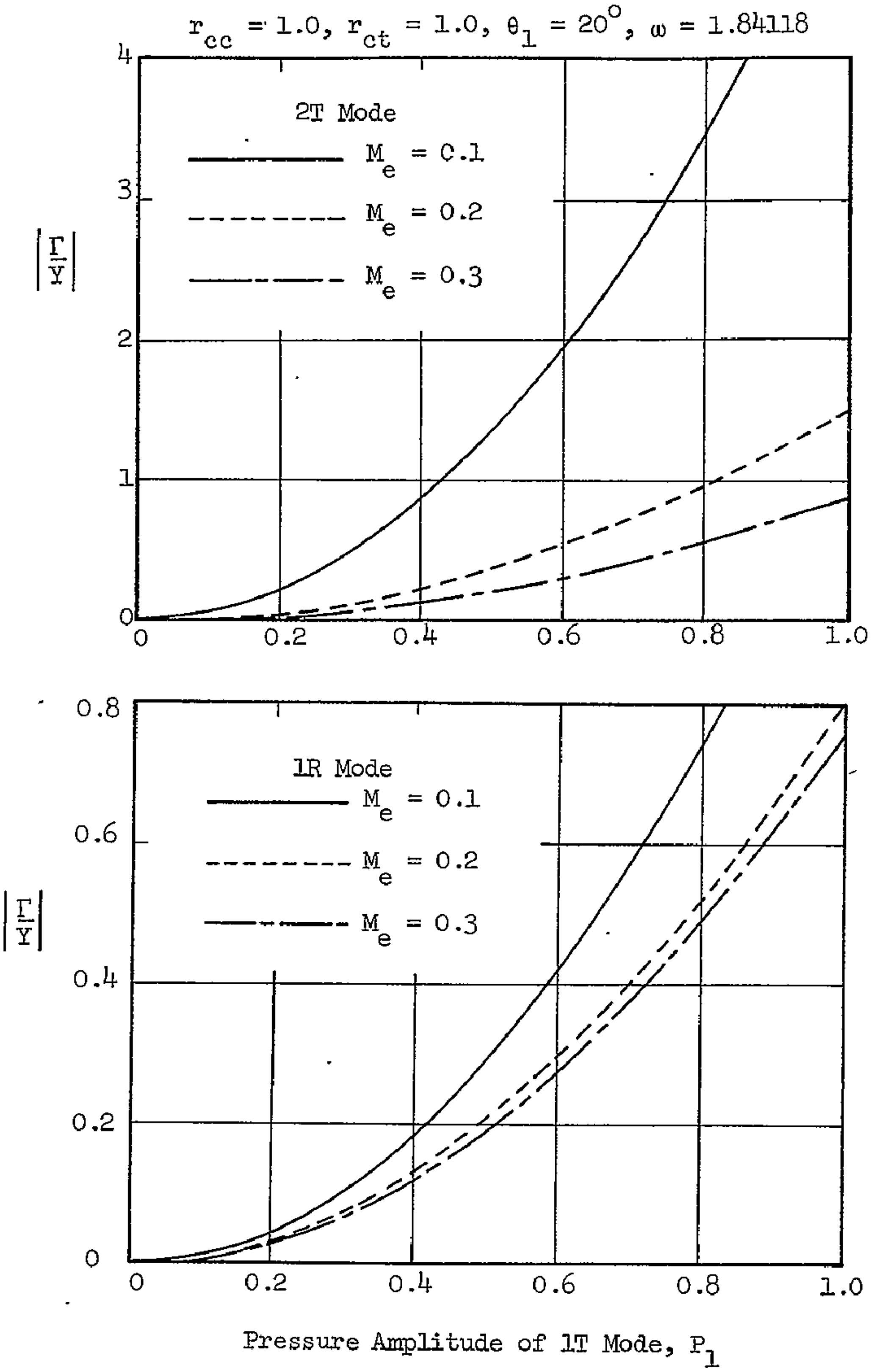
The effect of nozzle half-angle on $|\Gamma/Y|$ for the 2T and 1R modes is shown in Figure 9. It is readily seen that for θ_1 between 15 and 45 degrees there is only a slight effect of nozzle half-angle on the relative magnitudes of the linear and nonlinear admittances. However, it should be noted that both the linear and nonlinear theories are restricted to slowly convergent nozzles (i.e., small θ_1).

Figure 10 shows the effect of the nozzle radii of curvature upon the quantity $|\Gamma/Y|$ for the 2T mode. It is observed that a change in the radius of curvature of the nozzle at the throat has an insignificant effect on the relative magnitude of the linear and nonlinear admittances. On the other hand, a similar change in the radius of curvature of the nozzle at the entrance section has considerable effect on the relative magnitude of the linear and nonlinear admittances. Similar results were obtained for the 1R mode.

In summary, the results obtained in the admittance calculations indicate that the magnitude of the nonlinear admittance coefficient is comparable to that of the linear admittance coefficient, especially at large pressure amplitudes. To determine if this result has a significant effect upon combustor stability, calculations were made for typical liquid rocket combustors using the nonlinear admittances. These results were compared with similar calculations using linear admittances. The results of this investigation are discussed in the remainder of this report.

Stability Calculations

Combustion instability calculations have been made using the three mode series consisting of the 1T, 2T, and 1R modes. These calculations have been made for different values of the following parameters: (1) time lag $\bar{\tau}$, (2) interaction index n, (3) steady state Mach number at the nozzle entrance M_e, and (4) chamber length-to-diameter ratio L/D. All of the combustors that



igure 8. Effect of Entrance Mach Number on the Relative Magnitudes of Linear and Nonlinear Admittances.

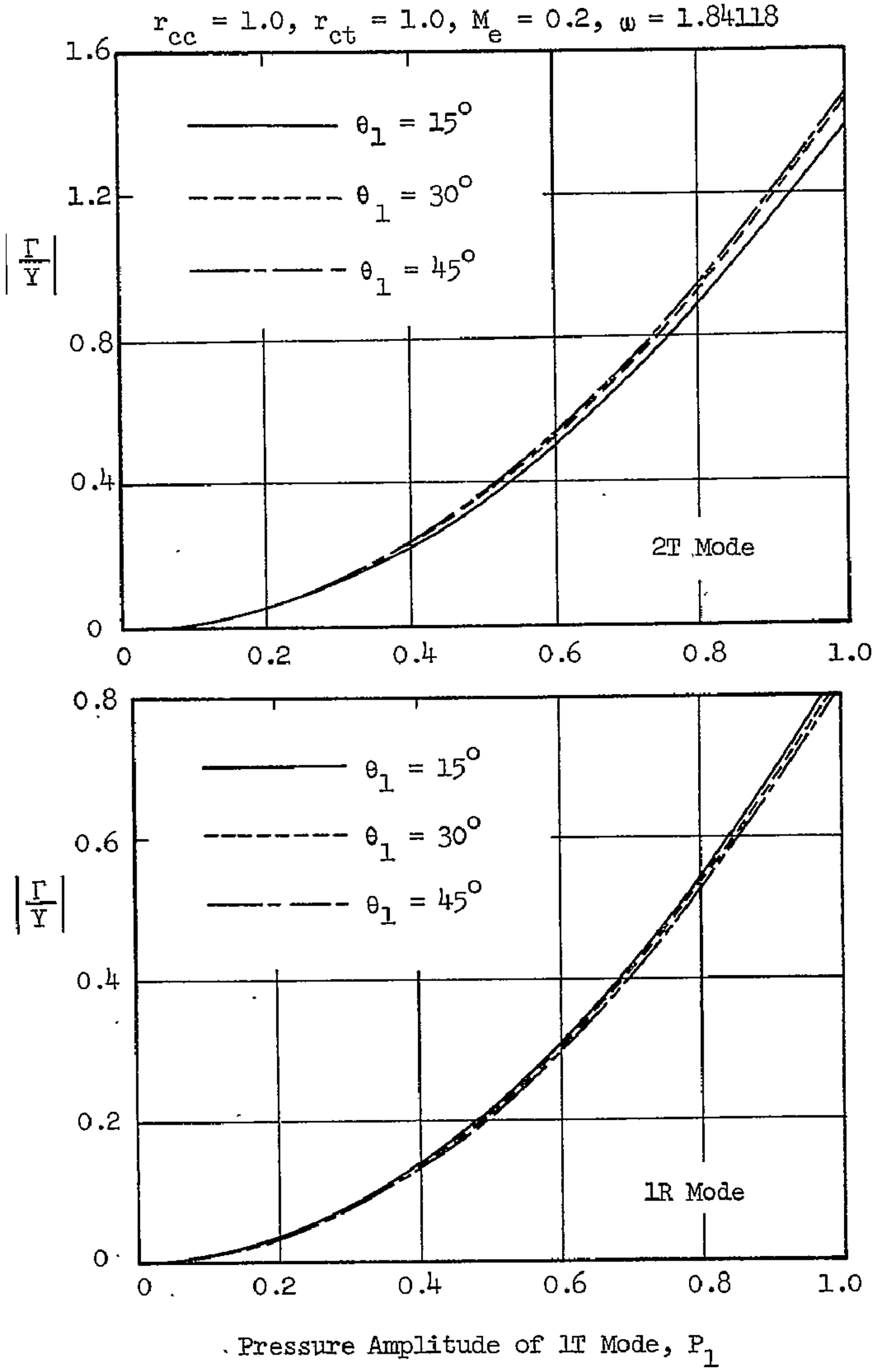


Figure 9. Effect of Nozzle Half-Angle on the Relative Magnitudes of Linear and Nonlinear Admittances.

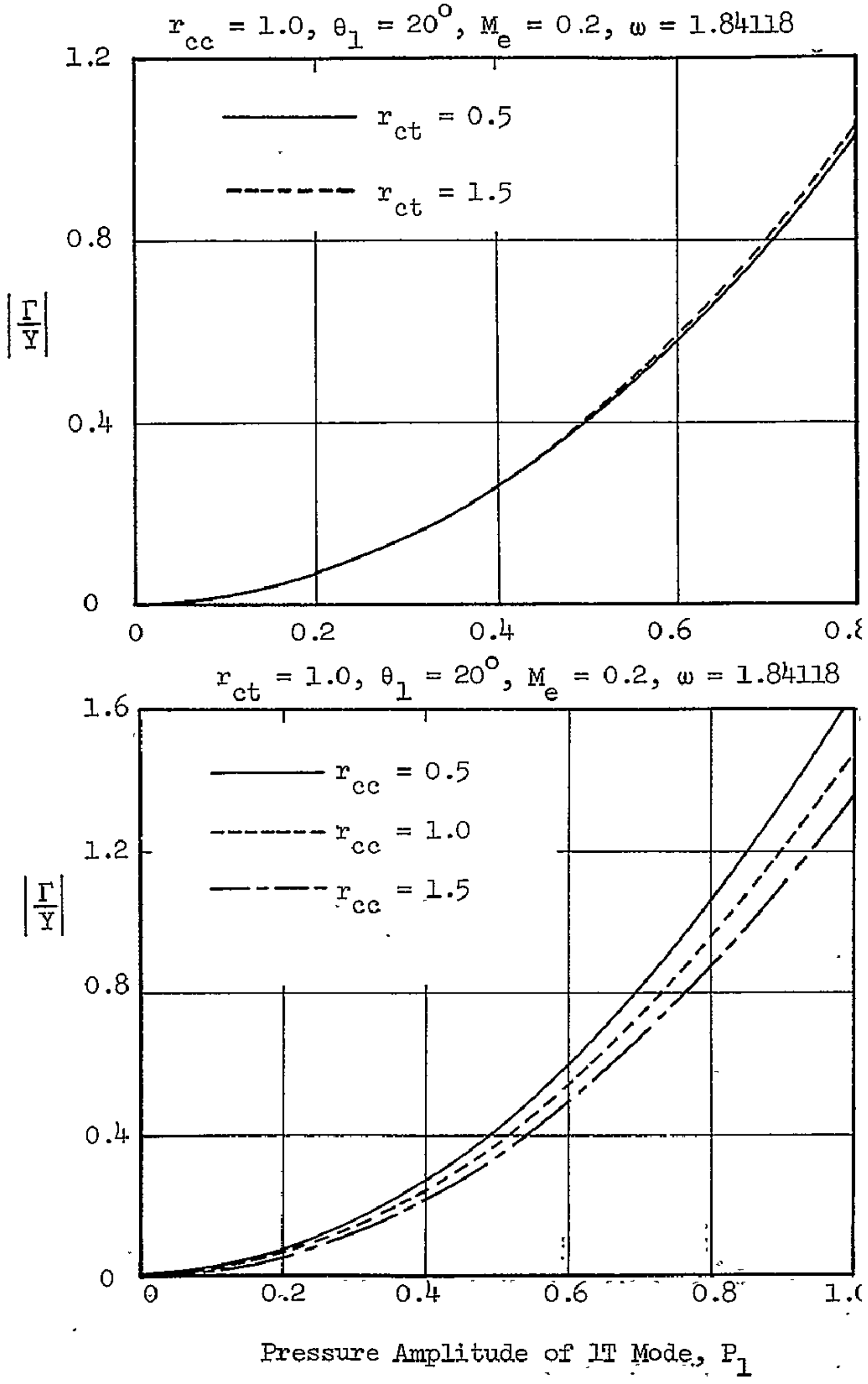


Figure 10. Effect of Nozzle Radii of Curvature on the Relative Magnitudes of Linear and Nonlinear Admittances for the 2T Mode.

have been analyzed are attached to nozzles with the following specifications: radius of curvature of nozzle at the combustion chamber, $r_{cc} = 1.0$, radius of curvature of nozzle at the throat, $r_{ct} = 1.0$; and nozzle half-angle, $\theta_1 = 20^\circ$. In each case, solutions have been obtained with both the linear and nonlinear nozzle admittances.

A typical neutral stability curve is shown in the n-τ plane in Figure 11. Since it was desired to study the limit-cycle behavior of the motor, the values of n and $\bar{\tau}$ considered were chosen from the unstable region of this stability diagram.

Limit-cycle amplitudes and waveforms were calculated for $\bar{\tau} = 1.6$ (resonant conditions) for several values of n as shown in Figure 11. Wall pressure waveforms (antinode) are shown for a mildly unstable case (Point A, n = 0.52) and a strongly unstable case (Point B, n = 0.70) in Figures 12 and 13. Figure 14 shows limit-cycle amplitude as a function of n for $\bar{\tau} = 1.6$. In each case both linear and nonlinear nozzle admittances were used in the calculations. These results show that the nozzle nonlinearities have only a small effect on the limit-cycle amplitude and waveform even for fairly large amplitude instabilities.

Similar comparisons were made for the off-resonant values of n and $\bar{\tau}$ shown in Figure 11 (see points C, D, E, F). These results also show very little effect of nozzle nonlinearities on the limit-cycle amplitudes for off-resonant oscillations as seen in Figure 15.

Finally, comparisons of limit-cycle amplitudes are shown for various exit Mach numbers in Figure 16 and for various length-to-diameter ratios in Figure 17. Again, limit-cycle amplitudes obtained using the nonlinear nozzle boundary condition agree closely with those obtained using the linear nozzle boundary condition.

CONCLUDING REMARKS

A second-order theory and computer program have been developed for calculating three-dimensional, nonlinear nozzle admittance coefficients to be used in the analysis of nonlinear combustion instability problems. This theory is applicable to slowly convergent, supercritical nozzles under isentropic, irrotational conditions when the combustion chamber oscillations are dominated

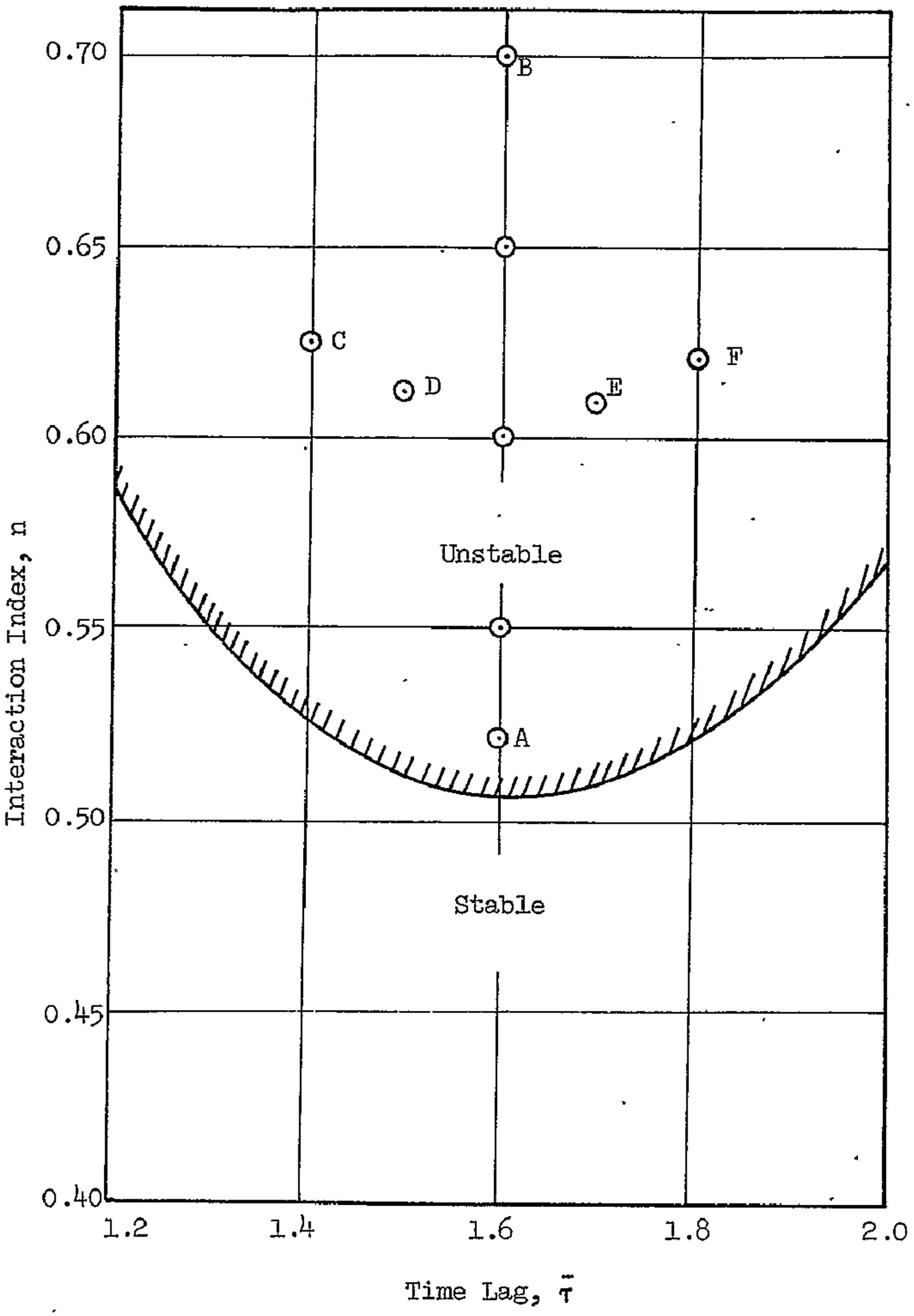


Figure 11. Linear Stability Limit.

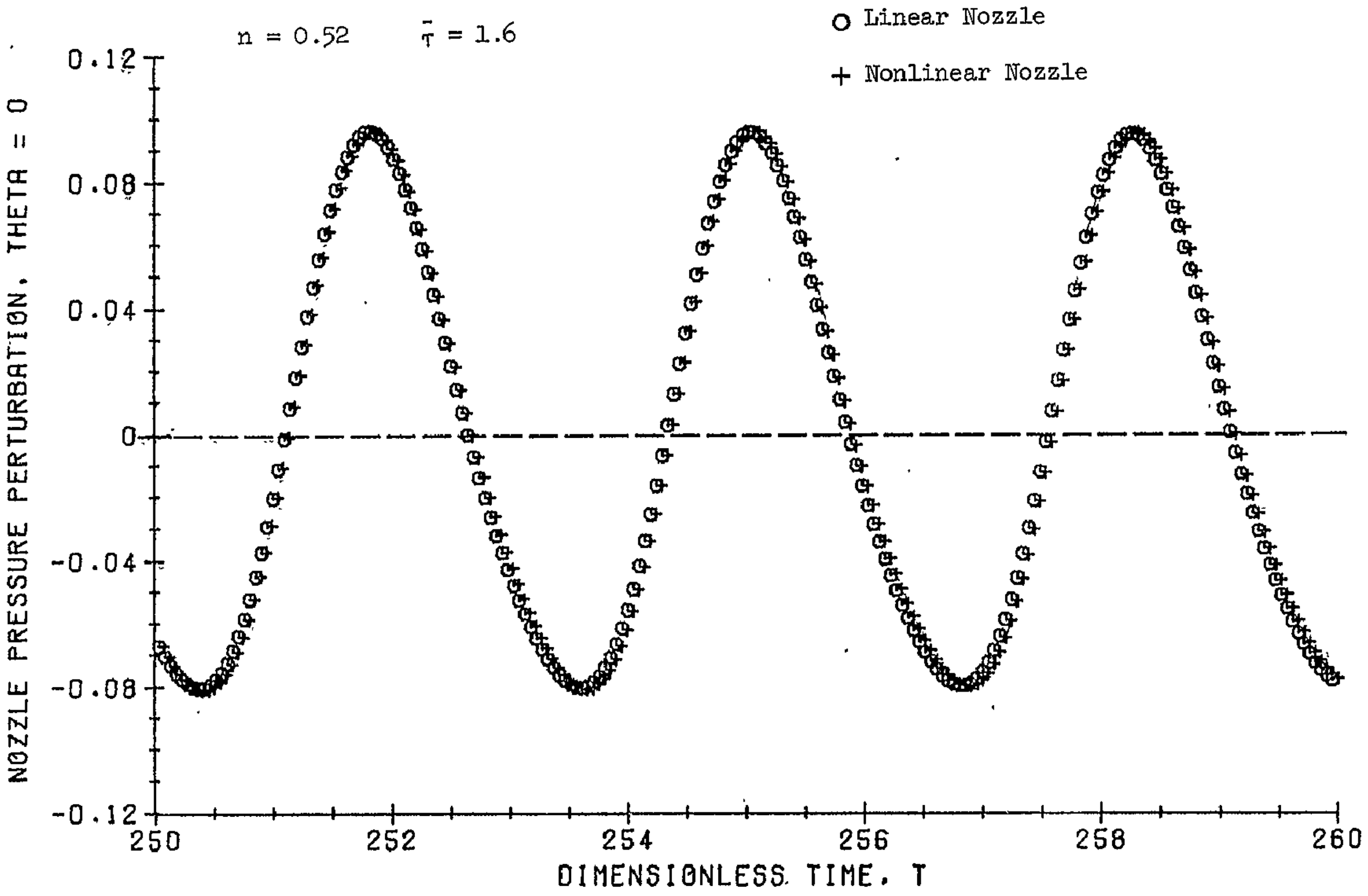


Figure 12. Comparison of Pressure Waveforms for a Mildly Unstable Motor.

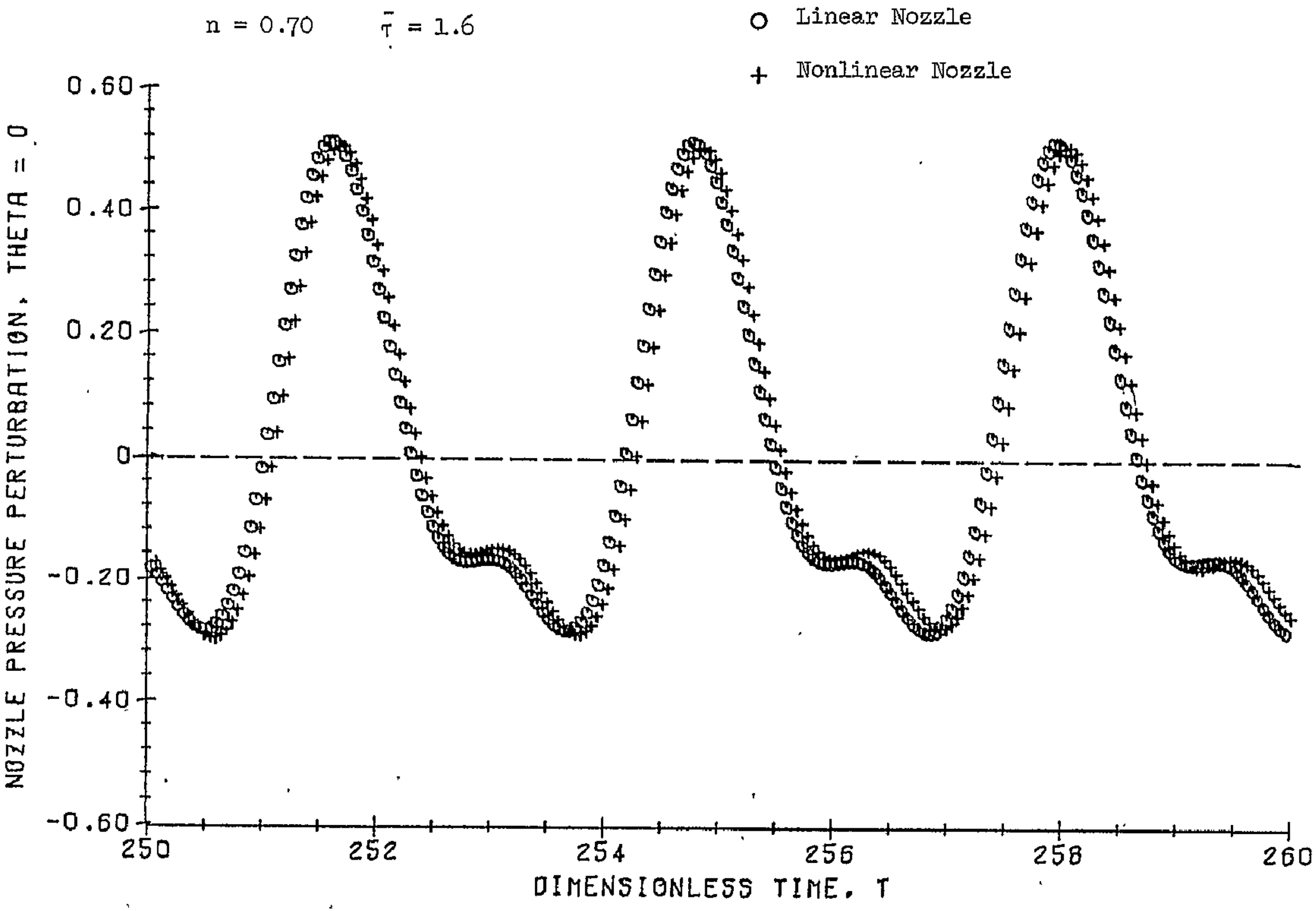


Figure 13. Comparison of Pressure Waveforms for a Strongly Unstable Motor.

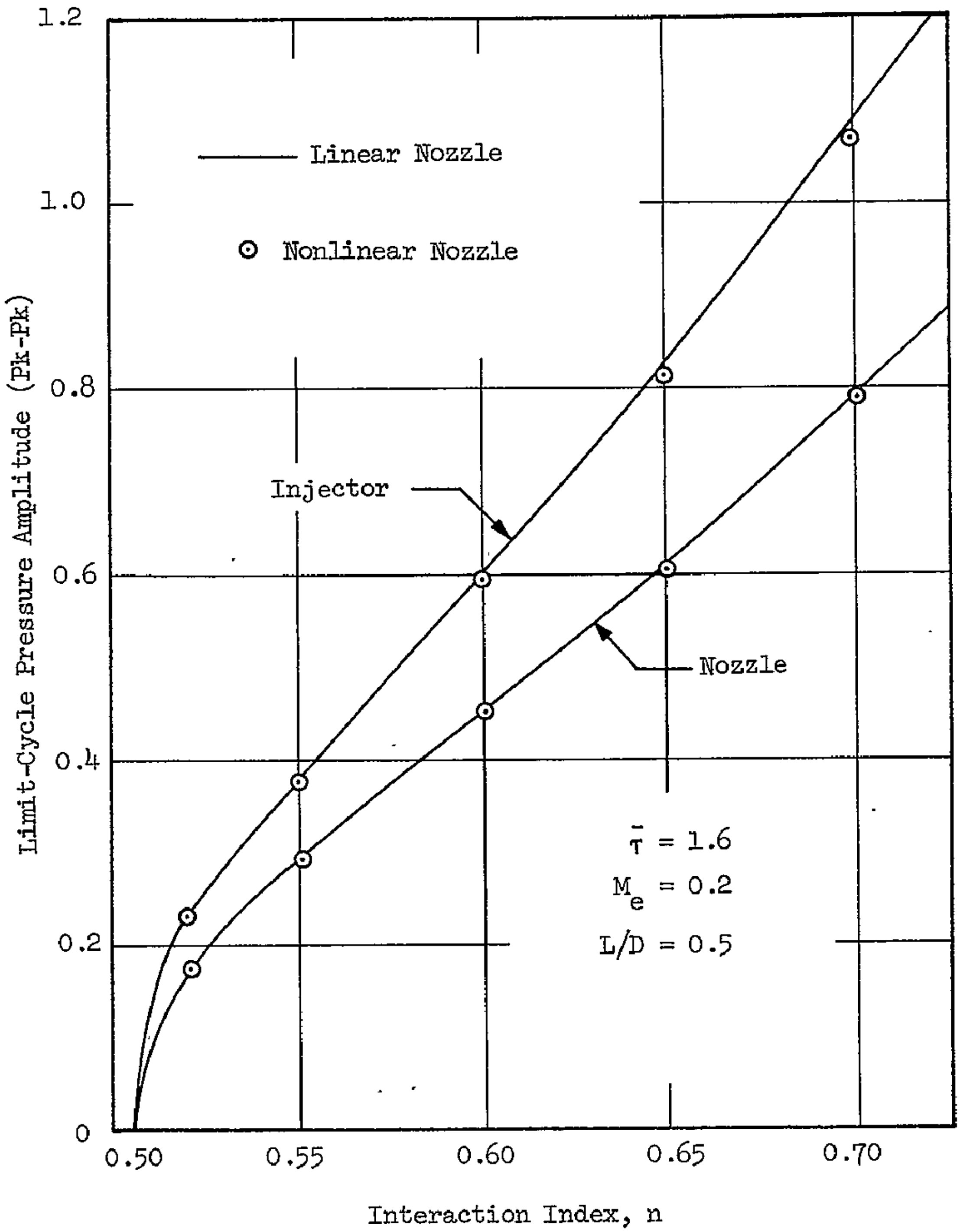


Figure 14. Comparison of Limit-Cycle Amplitudes for Different Values of n.

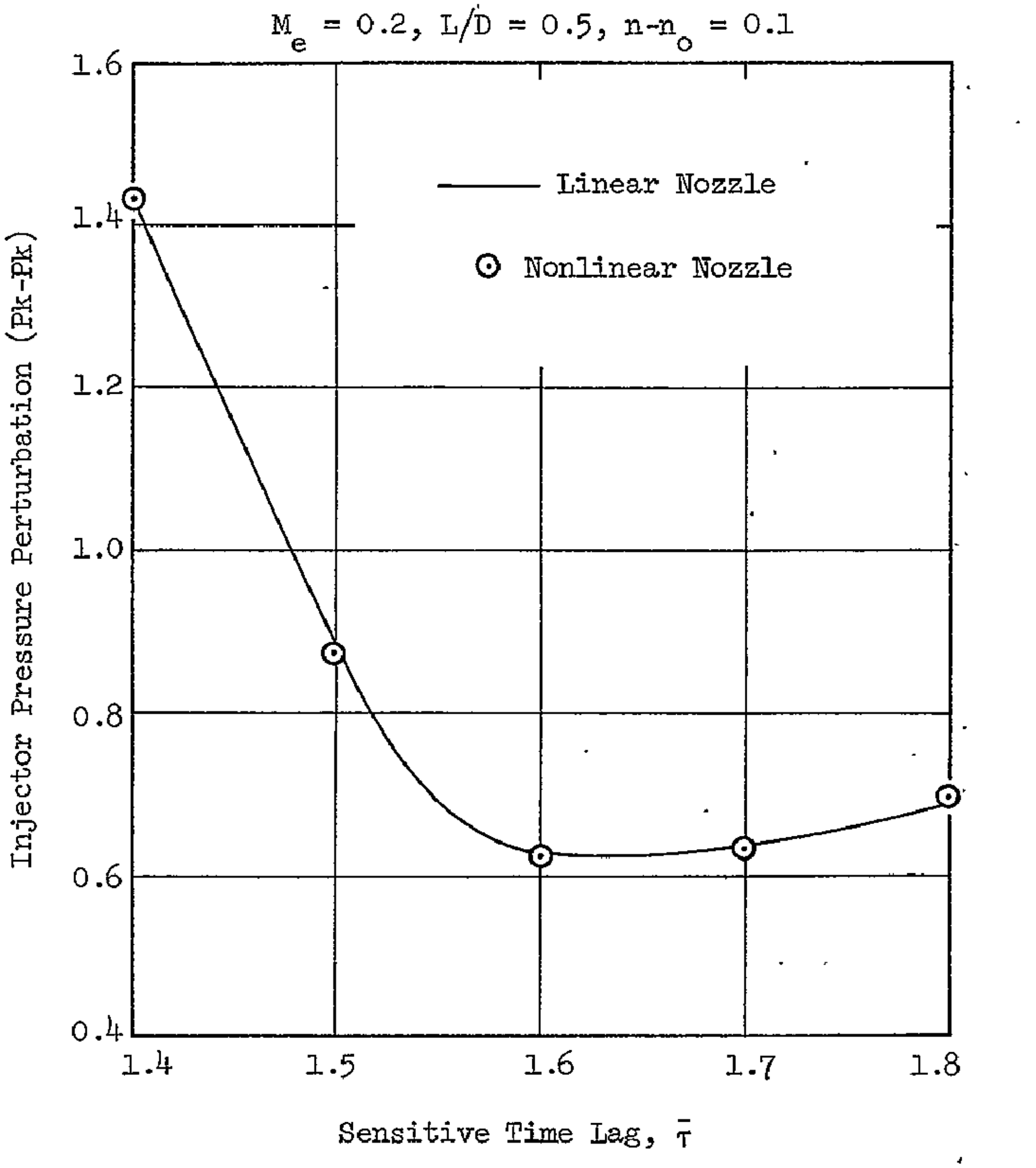


Figure 15. Comparison of Limit-Cycle Pressure Amplitudes for Different Values of $\bar{\tau}$.

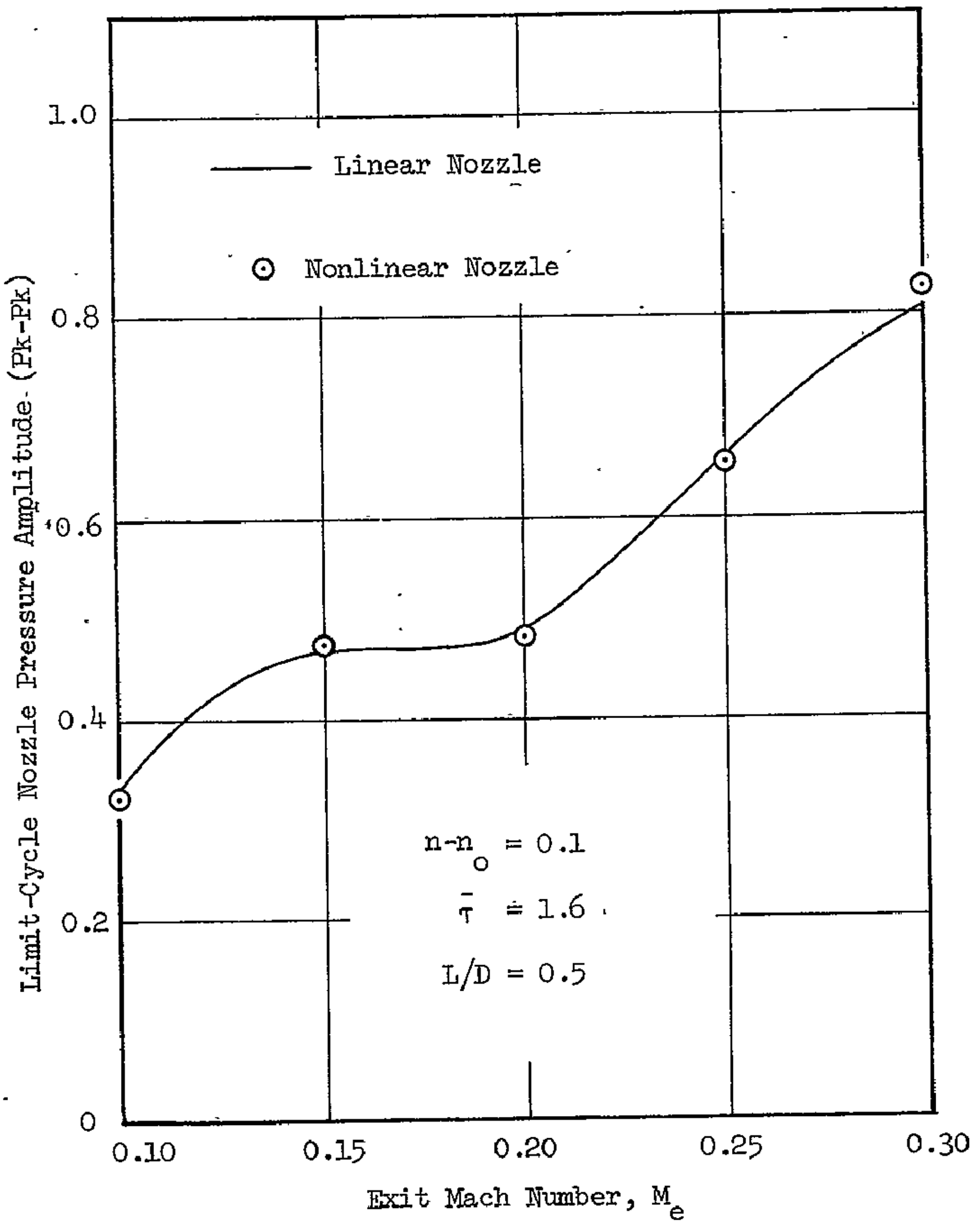


Figure 16. Comparison of Limit-Cycle Amplitudes for Different Values of M_e.

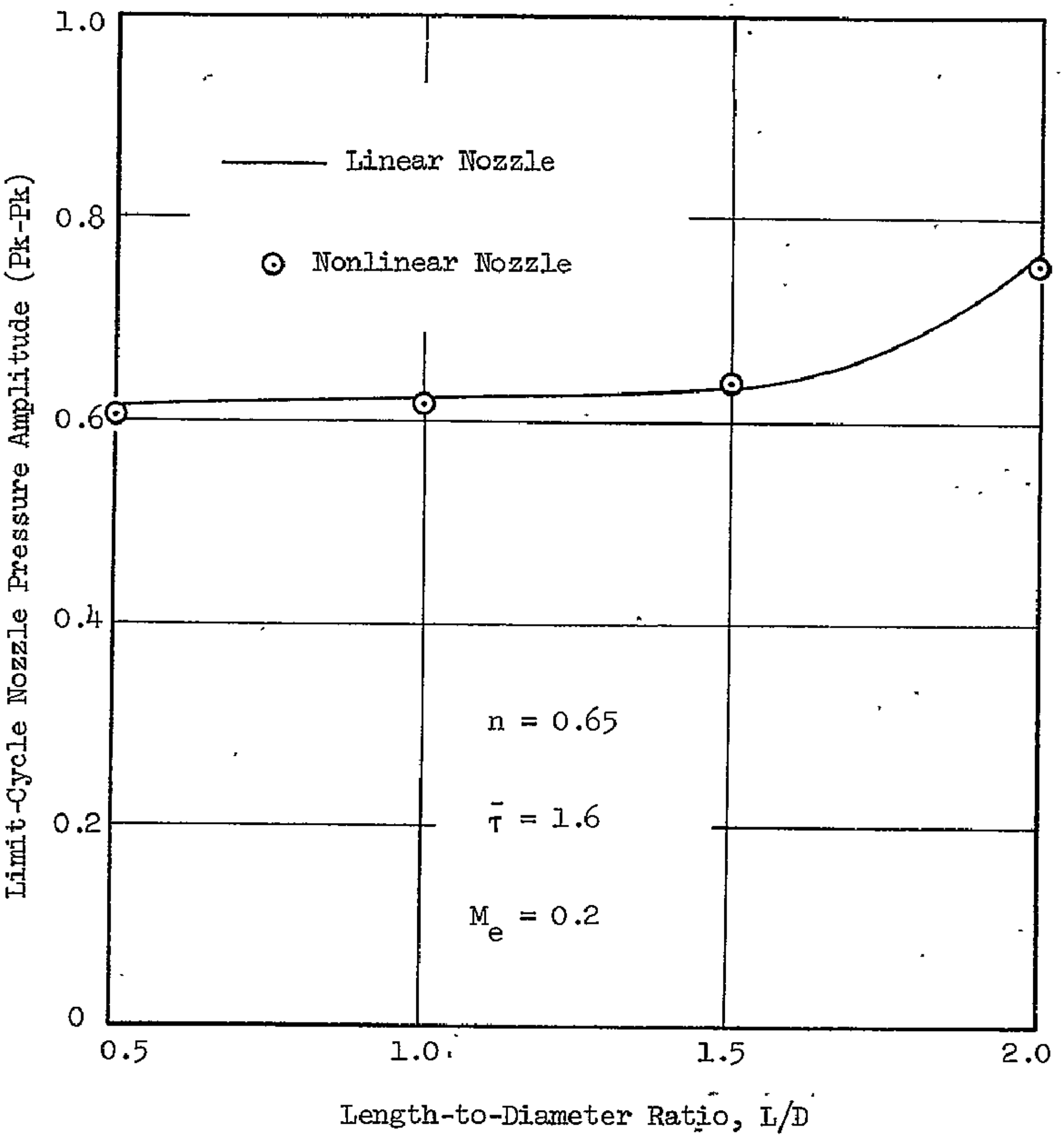


Figure 17. Comparison of Limit-Cycle Amplitudes for Different Values of L/D.

by the 1T mode. Nozzle admittances have been computed for typical nozzle geometries, and results have been shown as a function of the frequency and amplitude of the 1T mode.

The nonlinear nozzle admittances have been incorporated into the previously developed nonlinear combustion instability theory, and calculations of limit-cycle amplitudes and pressure waveforms have been made to assess the importance of the nonlinear contribution to the nozzle admittance. These results show that nozzle nonlinearities can be safely neglected in nonlinear combustion instability calculations if the following conditions are satisfied: (1) the amplitude of the oscillations are moderate, (2) the mean flow Mach number is small, and (3) the instability is dominated by the first tangential mode. Therefore, the linear nozzle boundary condition used in the previous nonlinear combustion instability analyses is adequate for most cases involving 1T mode instability.

APPENDIX A

PROGRAM NOZADM: A USER'S MANUAL

General Description

Program NOZADM calculates both the linear and the nonlinear admittance coefficients for a specified nozzle. These admittance coefficients are required as input for Program COEFFS3D (see Appendix B) which calculates the coefficients of both the linear and nonlinear terms in the combustor amplitude equation (i.e., Eq. (20)). The output of Program NOZADM is either punched onto cards or stored on disk or drum for input to Program COEFFS3D.

Program Structure

A flow chart for Program NOZADM is shown in Fig. (A-1). The program performs the following operations: (1) reads the input data, (2) calculates the steady-state flow quantities in the nozzle, (3) obtains the starting values needed to numerically integrate Eqs. (14) and (15), (4) performs the numerical integration of Eqs. (14) and (15) to obtain the desired admittance coefficients, and (5) provides the desired output.

The inputs to the program include parameters describing the nozzle, the frequency and pressure amplitude of the fundamental mode, and the various control numbers.

After reading the input, the program obtains the steady-state flow quantities at every station in the nozzle by calling the subroutine STEADY. This subroutine also calculates the number of station points (NPLAST) in the nozzle.

The evaluation of the admittance coefficients is carried out in stages. The work performed in each step depends upon whether or not the nonlinear admittances are to be evaluated. If only the linear admittances are required, only the equation for ζ_p needs to be solved. Thus, the equations govering ζ_p are solved individually for each of the modes in the series expansion. On the other hand, if the nonlinear admittances are also required the equations governing the linear admittance for the fundamental mode (ζ_1) and the amplitude of the fundamental mode (A_1) are first solved to obtain these quantities at

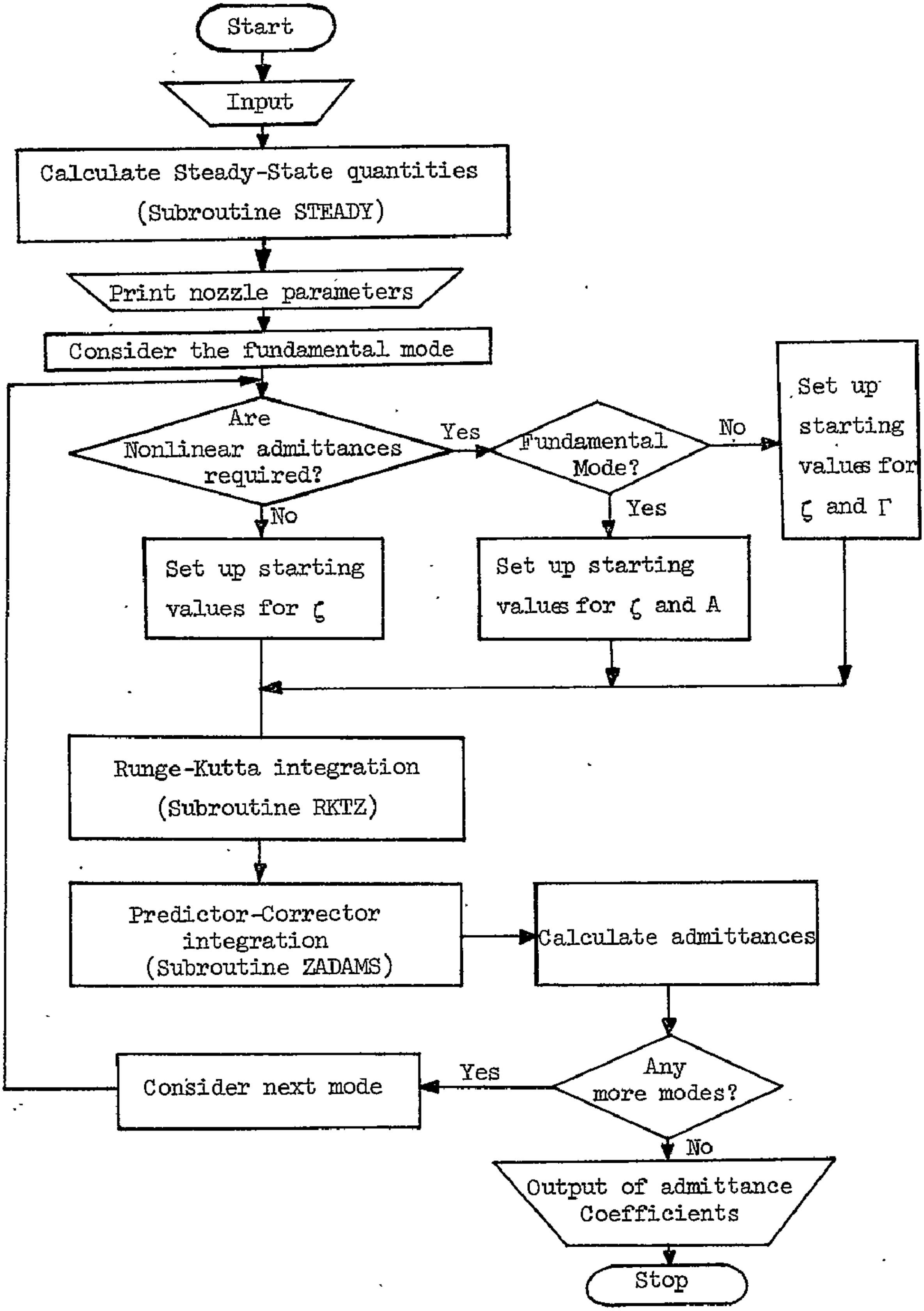

Start
Input
Calculate Steady-State quantities (Subroutine STEADY)
Print nozzle parameters
Consider the fundamental mode
Are Nonlinear admittances required?
Yes
No
Fundamental Mode?
No
Yes
Set up starting values for ζ and Γ
Set up starting values for ζ
Set up starting values for ζ and A
Runge-Kutta integration (Subroutine RKTZ)
Predictor-Corrector integration (Subroutine ZADAMS)
Calculate admittances
Any more modes?
Yes
No
Consider next mode
Output of admittance Coefficients
Stop

Figure A-1. Flow Chart.

every station in the nozzle. In the subsequent steps, the equations for ζ and Γ for each of the remaining modes are solved.

Input Data

A precise definition of the input data required to run the computer program is given below. The input is given through three data cards. In the description of the cards below, the location number refers to the columns of the card. "I" indicates integers and "F" indicates real numbers with a decimal point. For the I formats, the values are placed in fields of five locations while a field of ten locations is used with the "F" formats. In either case, the numbers must be placed in the rightmost locations of the allocated field.

No. of Cards	Location	Type	Input Item	Comments
1	1-10	F	CM	Mach number at the nozzle entrance
	11-20	F	ANGLE	Nozzle half-angle
	21-30	F	RCC	Radius of curvature of the nozzle at the entrance
	31-40	F	RCT	Radius of curvature of the nozzle at the throat
	41-50	F	GAM	Ratio of specific heats
1	1-5	I	NOZNL1	If 0: nonlinear admittances are not evaluated If 1: nonlinear admittances are evaluated
	6-10	I	NOUT	Determines output If 0: only printed output If 1: printed and stored on disk or drum (output device number 7) If 2: printed and cards punched in a format suitable for the program COEFFS3D

No of Cards	Location	Type	Input Item	Comments
	11-15	I	IEXTN	If 0: no extension section If 1: an extension section is present.
	16-25	F	EXTNSN	Length of the extension section; omit if IEXTN = 0
1	1-10	F	WC	Frequency of oscillation
	11-20	F	P1AMPL	Pressure amplitude of the fundamental mode. Omit if only linear admittances are needed.

The nozzle parameters ANGLE, RCC and RCT correspond to θ_1, r_{cc} and r_{ct} in Fig. 2. For IEXTN = 1, the integration of Eqs. (14) and (15) is continued beyond the nozzle entrance plane to a length EXTNSN within the combustion chamber. When NOUT = 1, the values of the necessary admittance coefficients are stored on disk or drum (device number 7) in a format suitable for input to program COEFFS3D. If, instead of providing this data to program COEFFS3D through data file 7, it is desirable to provide punched cards only, NOUT should be 2. Again the format is such that these cards can be fed to program COEFFS3D directly.

Steady-State Quantities

The subroutine STEADY is called to evaluate the steady-state quantities in the nozzle. This subroutine first calculates the radius of the nozzle at the throat necessary to obtain the specified Mach number at the nozzle entrance. The steady-state flow quantities at the throat are determined by the choking conditions. Starting with these values, the steady-state flow quantities at the other stations in the nozzle are calculated by numerically integrating the steady-state equations starting from the throat. The subroutine RKSTDY determines the values of the steady-state velocity near the throat using the Runge-Kutta scheme. These values are needed to start the Adam's predictor-corrector scheme for integrating the steady-state flow equation. The numerical integration is performed by the subroutine UADAMS. Starting slightly upstream of the throat, the numerical integration is continued till the nozzle entrance is reached (radius of the nozzle R = 1). The arrays U and C contain the steady-state velocity and speed of sound respectively.

Coefficients

The complex coefficients that appear in the nozzle admittance equations are evaluated in the program by calling the subroutine COEFFS. These coefficients contain certain integrals involving trigonometric and Bessel functions. The subroutine INTGRL sets up arrays for these integrals.

Integrals

The necessary trigonometric integrals are determined by the subroutine INTGRL itself. Denoting

$$\Theta_p(\theta) = \cos(m_p\theta),$$

the integrals are as follows:

$$\text{ALPHA }(1,\ p) = \int_0^{2\pi} \left[\Theta_p(\theta)\right]^2 \Theta_1(\theta)\ d\theta$$

$$\text{ALPHA }(2,\ p) = \int_0^{2\pi} \left[\Theta_p'(\theta)\right]^2 \Theta_1(\theta)\ d\theta$$

$$\text{ALPHA }(3,\ p) = \int_0^{2\pi} \Theta_p''(\theta)\ \Theta_p(\theta)\ \Theta_1(\theta)\ d\theta$$

$$\text{ALPHA }(4,\ p) = \int_0^{2\pi} \left[\Theta_p(\theta)\right]^2 d\theta$$

$$\text{ALPHA }(5,\ p) = \int^{2\pi} \Theta_p''(\theta)\ \Theta_p(\theta)\ d\theta$$

The integrals involving Bessel functions are as follows:

$$\text{BETA }(1,\ p) = \int_0^1 \left[R_1(r)\right]^2 R_1(r)\ r\ dr$$

$$\text{BETA }(2,\ p) = \int_0^1 \left[R_p(r)\right]^2 R_1(r)\ \frac{1}{r}\ dr$$

$$\text{BETA }(3,\ p) = \int_0^1 \left[R_p'(r)\right]^2 R_1(r)\ r\ dr$$

$$\text{BETA }(4,\ p) = \int_0^1 R_p''(r)\ R_p(r)\ R_1(r)\ r\ dr$$

$$\text{BETA }(5,\ p) = \int_0^1 R_p'(r)\ R_p(r)\ R_1(r)\ dr$$

$$\text{BETA }(6,\ p) = \int_0^1 \left[R_p(r)\right]^2 r\ dr$$

$$\text{BETA }(7,\ p) = \int_0^1 R_p'(r)\ R_p(r)\ dr$$

$$\text{BETA }(8,\ p) = \int_0^1 R_p''(r)\ R_p(r)\ r\ dr$$

$$\text{BETA }(9,\ p) = \int_0^1 \left[R_p(r)\right]^2 \frac{1}{r}\ dr$$

Here $R_p(r) = J_m\left[S_{mn} r\right]$ where m and n are the transverse mode numbers for the pth mode.

These integrals of Bessel functions are obtained from the functions RAD1 and RAD2. RAD2 provides the first five integrals while RAD1 provides the last four integrals. Simpson's integration scheme is used in these function subprograms to evaluate these integrals. The values of the Bessel functions of the first kind are obtained using the subroutine JBES (see Ref. 17

Integration of the Differential Equations

For the numerical integration of the differential equations, a fourth-order Adam-Bashforth predictor-corrector scheme is employed. The necessary initial values are obtained by using a fourth-order Runge-Kutta scheme near the throat. The Runge-Kutta integration is performed by subroutine RKTZ. The predictor-corrector integration is performed by subroutines TADAMS and ZADAMS. The values of the dependent variables are stored in the array Y and their derivatives are stored in the array DY. The integration is continued in steps of DP in the axial variable (steady-state velocity potential) till the combustion chamber is reached.

After the numerical integration of all the differential equations is completed, the admittance coefficients are evaluated. AMPL (J) and PHASE(J) are the amplitude and phase of the linear admittance coefficient for mode J. GNOZ(J) is the complex, nonlinear admittance coefficient for mode J.

Output

The output of the program NOZADM contains two sections.

In Section 1, the parameters of the nozzle being analyzed are printed out. The output of this section occupies only one page and is essentially a print out of the input data. The parameters, which are printed are: the Mach number at the nozzle entrance (CM), the specific heat ratio (GAM), the nozzle half-angle (ANGLE), the length of the extension section, if any (EXTNSN), the radius of curvature of the nozzle at the throat (RCT), the radius of curvature of the nozzle at the entrance (RCC), and the number of stations in the nozzle (NPLAST). Section 1 is printed for any value of the control number NOUT.

Section 2 contains the nozzle admittance coefficients. Depending on the value of the control number NOUT, Section 2 is printed, stored on disk or drum or punched onto cards. These three modes of output will now be discussed individually.

Printed output: The control number NOUT for this mode is 0. The printed output appears on one page and contains both the linear and nonlinear admittance coefficients. For each coefficient, the real and imaginary parts as well as the magnitude and phase are printed out. If nonlinear admittance coefficients are not calculated by the program (NOZNL1 = 0), zeros are entered in the spaces for the nonlinear coefficients.

This mode of output is inconvenient to use for instability analysis since it would then be necessary to manually punch all the input cards for the program COEFFS3D.

Disk or Drum Storage: The control number NOUT for this mode is 1. When disk or drum storage (like the FASTRAND System on the UNIVAC 1108) is available, this is the most convenient means of storing the output of Section 2. The necessary admittance coefficients are stored in a format suitable for input to the program COEFFS3D. The device number for this output is 7. The control statement needed to request the disk or drum storage on the computer depends on the computer facilities being used.

Punched Cards: NOUT for this mode is 2. This mode of output is the simplest way to run the instability program. The cards containing the necessary admittance coefficients are punched by the computer in a format suitable for use with program COEFFS3D, which is the next program to be executed.

FORTRAN Listing

```
C
C     ***************** PROGRAM NOZADM ********************************
C
C        THIS PROGRAM EVALUATES THE LINEAR AND NONLINEAR ADMITTANCES
C     OF A SPECIFIED NOZZLE.
C
C     THE FOLLOWING INPUTS ARE REQUIRED:
C
C     CM IS THE MACH NUMBER AT THE NOZZLE ENTRANCE.
C     ANGLE IS THE SLOPE OF THE MIDDLE SECTION OF THE NOZZLE.
C     RCC IS THE RADIUS OF CURVATURE OF THE NOZZLE AT THE ENTRANCE.
C     RCT IS THE RADIUS OF CURVATURE AT THE THROAT.
C     GAM IS THE SPECIFIC HEATS RATIO.
C
C     NOZNL1 DETERMINES WHETHER THE NONLINEAR ADMITTANCES ARE TO
C     BE EVALUATED:
C        NOZNL1 = 0      NOT EVALUATED.
C        NOZNL1 = 1      EVALUATED.
C     NOUT DETERMINES THE OUTPUT:
C        NOUT = 0      PRINTED OUTPUT ONLY.
C        NOUT = 1      PRINTED AND WRITTEN INTO A FASTRAND FILE.
C        NOUT = 2      PRINTED AND ADMITTANCES PUNCHED INTO CARDS.
C     IEXTN DETERMINES IF THERE IS AN EXTENSION SECTION
C        IEXTN = 0      NO EXTENSION SECTION.
C        IEXTN = 1      THERE IS AN EXTENSION SECTION.
C     EXTNSN IS THE LENGTH OF THE EXTENSION SECTION.
C
C     WC IS THE FREQUENCY OF THE FUNDAMENTAL MODE.
C     P1AMPL IS THE PRESSURE AMPLITUDE OF THE FUNDAMENTAL MODE.
C
C
      COMMON      /X1/CM,ANGLE,RCC,RCT,GAM,Q,RT,DF
     1            /X2/T,R1,R2,NPLAST,NEND,IEXTN
     2            /X3/WC,SUN,IP,MODE,NU,KP(3)
     3            /X4/RU(7),RDU(7),ZTHR1,GTHR1
     4            /X5/U(1000),DU(1000),C(1000),RW(1000)
     5            /X6/AFN,AFN1,AFN2
     6            /X7/ALPHA(5,3), BETA(9,3)
     7            /X8/ZRK(1000)
      COMPLEX     AFN(1000),AFN1(1000),AFN2(1000),ACHMBR,CONST,
     1            CC(25),CC1(25),CFH,CFM,CFN,CGRP1,CGRP2,
     2            INHMG,INHMG1,ZTHR,ZTHR1,AH,AH1,GTHR,GTHR1,
     3            ZETA,TAU,LINADM,ZRK,GNOZ(3)
      DIMENSION   G(4),GP(4),Y(4),DY(4,4),SMN(3),ISTEP(3),
     1            NAME(3), PHASE(3), AMPL(3)
      DATA  (NAME(MODE), MODE = 1,3) /2H1T,2H2T,2H1R/
     1      (SMN(MODE), MODE = 1,3) /1.84118,3.05424,3.83171/
C
      READ (5,5005) CM,ANGLE,RCC,RCT,GAM
      READ (5,5010) NOZNL1, NOUT, IEXTN, EXTNSN
      READ (5,5015) WC, P1AMPL
      GMIN1 = GAM - 1.
      GPL1 = GAM + 1.
      DF = -0.002
C     ISTEP = 1 :      INTEGRATE FOR ZETA ONLY.
```

```
C     ISTEP = 2 :      INTEGRATE FOR ZETA & AH.
C     ISTEP = 3 :      INTEGRATE FOR ZETA & GAMMA.
      IF  (NOZNL1 .EQ. 1)  GO TO 10
      ISTEP(1) = 1
      ISTEP(2) = 1
      ISTEP(3) = 1
      GO TO 15
10    ISTEP(1) = 2
      ISTEP(2) = 3
      ISTEP(3) = 3
15    CONTINUE
      KP(1) = 1
      KP(2) = 2
      KP(3) = 2
C
C     OBTAIN STEADY-STATE QUANTITIES IN THE NOZZLE .
      CALL STEADY
C
C     PRINT OUT THE NOZZLE PARAMETERS.
      WRITE (6,1005)
      WRITE (6,1010) CM
      WRITE (6,1015) GAM
      WRITE (6,1020) ANGLE
      WRITE (6,1025) EXTNSN
      WRITE (6,1030) RCT
      WRITE (6,1035) RCC
      WRITE (6,1040) NPLAST
C
      NEND = NPLAST
      IF (IEXTN .NE. 1)  GO TO 25
C
C     DETERMINE NUMBER OF STATIONS IN THE EXTENSION REGION, AND
C     DEFINE STEADY-STATE QUANTITIES IN THAT REGION.
C
      UEXT = U(NPLAST)
      NEND = NPLAST + (EXTNSN * UEXT ** .5) / DP
      DO 20  NP = NPLAST,NEND
      U(NP) = U(NPLAST)
      C(NP) = C(NPLAST)
      DU(NP) = DU(NPLAST)
      RW(NP) = RW(NPLAST)
20    CONTINUE
25    CONTINUE
      IF (NEND .GT. 1000)  GO TO 550
C
      CALL  INTGRL
      SRTR=(RT*RCT)**.5
C
      ACHMBR = CMPLX (PIAMPL / (WC*GAM) ,0.)
      IF (NOUT .EQ. 0) WRITE (6,1050) WC,PIAMPL
      IF (NOUT .EQ. 0) WRITE (6,1055)
C
      DO 500 MODE=1,3
      IP=ISTEP(MODE)
      SVN=SMN(MODE)
      SVNR=SVN/RT
C
C*************STARTING VALUES SECTION*************************************
C
      P=0.
      AHR = 1.
      AHI = 0.
      AH = CMPLX (AHR,AHI)
      UP = U(1)
      CP = C(1)
      DUP = DU(1)
      RWP = RW(1)
      CALL  COEFFS (UP,DUP,CP,RWP,CC)
      CFH = CC(1)
      CFM = CC(2) + CC(6)
      CFN = CC(3) + CC(4) + CC(5) + CC(7) + CC(8)
C
C     *************DERIVATIVES OF THE COEFFICIENTS AT THE THROAT********
C
C     EVALUATE DERIVATIVES OF LINEAR COEFFICIENTS.
      XR = - 4./(GPL1 * SRTR)
      CFH1 = CMPLX (XR,0.)
      XR = - (24. + 4. * GAM) / (GPL1 * 3. * RT * RCT)
      XI = - 8. * WC * KP(MODE) / (GPL1 * SRTR)
      CFM1 = CMPLX (XR,XI)
      XR = - 2.*GMIN1 * (BETA (8,MODE) + BETA (7,MODE) + BETA (9,MODE)
     1         * ALPHA (5,MODE) / ALPHA (4,MODE)) / (GPL1 * RT * RT
     2          * SRTR * BETA (6,MODE))
      XI = -(12 + 2*GAM) * WC * KP(MODE) * GMIN1 / (3.*GPL1 * RT*RCT)
      CFN1 = CMPLX (XR,XI)
C
C     SET  UP  VALUES  AT  THE  THROAT  BY  TAYLORS  EXPANSION
C
C     STARTING VALUES FOR ZETA
      ZTHR = - CFN / CFM
      ZTHR1 = - (CFM1 * ZTHR + CFH1 * ZTHR * ZTHR + CFN1) / (CFH1 + CFM)
      ZRK(1) = ZTHR
C
      IF (MODE.NE.1) GO TO 110
      AFN(1) = AH
      AFN1(1) = AFN(1) * ZTHR
      AFN2(1) = AFN1(1) * ZTHR + AFN(1) * ZTHR1
110   CONTINUE
      G(1) = REAL (ZTHR)
      G(2) = AIMAG (ZTHR)
      DY (1,1) = REAL (ZTHR1)
      DY (2,1) = AIMAG (ZTHR1)
      GO TO (120,130,140), IP
130   G(3) = AHR
      G(4) = AHI
      AH1 = AH * ZTHR
      DY (3,1) = REAL (AH1)
      DY(4,1) = AIMAG (AH1)
      GO TO 120
140   CONTINUE
```

```
      CGRP1 = CC(13) + CC(14) + CC(19) + CC(23) + CC(24) + CC(25)
      CGRP2 = CC(10) + CC(11) + CC(17) + CC(20) + CC(21) + CC(22)
      INHMG = -CC(18) * AFN(1) * AFN2(1) - CC(12) * AFN1(1) * AFN2(1)
     1        -(CC(9) + CC(15)) * AFN1(1) * AFN1(1) - CGRP1 * AFN(1) *
     2        AFN1(1) - CGRP2 * AFN(1) * AFN(1)
C
C     EVALUATE DERIVATIVES OF NON-LINEAR COEFFICIENTS.
      A1B1 = ALPHA(1,MODE) * BETA(1,MODE)
      A2B2 = ALPHA(2,MODE) * BETA(2,MODE)
      A1B3 = ALPHA(1,MODE) * BETA(3,MODE)
      A4B6 = ALPHA(4,MODE) * BETA(6,MODE)
      DO 26  J = 1,25
 26   CC1(J) = CMPLX (0.,0.)
      XR = - (2.*A1B1 * WC) / (A4B6 * GPL1 * SRTR)
      XI = XR
      CC1(9) = CMPLX (XR,XI)
      XR = - (4. * A1B1) / (3.1415927 * GPL1 * SRTR * A4B6)
      XI = - XR
      CC1 (12) = CMPLX (XR,XI)
      XR = - A1B3 / (GPL1 * RT * RT * SRTR * A4B6)
      XI = - XR
      CC1 (13) = CMPLX (XR,XI)
      XR = - A2B2 / (GPL1 * RT * RT * A4B6 * SRTR)
      XI = - XR
      CC1 (14) = CMPLX (XR,XI)
      XR = - A1B1 * (3.*GPL1 * SRTR + GMIN1 * (12.+GAM)) /
     1        (2. * RT * RCT * GPL1 * GPL1 * A4B6)
      XI = - XR
      CC1 (15) = CMPLX (XR,XI)
      XR = A1B3 * (9. - 2.*GAM - GAM*GAM) / (12. * RT**3 * RCT * GPL1
     1             * A4B6)
      XI = - XR
      CC1 (16) = CMPLX (XR,XI)
      XR = A2B2 * (9. - 2.*GAM - GAM*GAM) / (12. * RT**3 * RCT * GPL1
     1             * A4B6)
      XI = - XR
      CC1 (17) = CMPLX (XR,XI)
      XR = - (GMIN1 * WC * A1B1) / (GPL1 * SRTR * A4B6)
      XI = XR
      CC1 (18) = CMPLX (XR,XI)
      XR = - (GMIN1 * (6.+GAM) * WC * A1B1) / (3. * GPL1 * RT * RCT
     1             * A4B6)
      XI = XR
      CC1 (19) = CMPLX (XR,XI)
      XR = - (GMIN1 * ALPHA (1,MODE) * (BETA (4,MODE) - BETA(5,MODE)))
     1        / (GPL1 * RT * RT * SRTR * A4B6)
      XI = - XR
      CC1 (23) = CMPLX (XR,XI)
      XR = - (GMIN1 * ALPHA (1,MODE) * BETA (5,MODE) * 2.)
     1        / (GPL1 * RT * RT * SRTR * A4B6)
      XI = - XR
      CC1 (24) = CMPLX (XR,XI)
      XR = - (GMIN1 * ALPHA (3,MODE) * BETA (2,MODE))
     1        / (GPL1 * RT * RT * SRTR * A4B6)
      XI = - XR
      CC1 (25) = CMPLX (XR,XI)
C
      INHMG1 = - AFN2(1) * AFN2(1) * CC(12) - AFN1(1) * AFN2(1) *
     1           (CC(18) + CC1(12) + 2.*CC(9) + 2.*CC(15)) - AFN2(1)
     2           * AFN(1) * (CC1(18) + CGRP1) - AFN1(1) * AFN1(1) *
     3           (CC1(9) + CC1(15) + CGRP1) - AFN1(1) * AFN(1) *
     4           (CC1(13) + CC1(14) + CC1(19) + CC1(23) + CC1(24)
     5           + CC1(25) + 2.*CGRP2) - AFN(1) * AFN(1) * (CC1(10)
     6           + CC1(11) + CC1(17) + CC1(20) + CC1(21) + CC1(22))
C
C     STARTING VALUES FOR GAMMA
      GTHR = - INHMG / (CP * CFM)
      GTHR1 = (-CP * GTHR * (CFH1 * ZTHR + CFM1) + (GMIN1 * .5 * 4. /
     1        ( GPL1 * SRTR)) * GTHR * (CFH1 + CFM) - INHMG1) /
     2        ( CP * CFH1 + CP * CFM)
C
      G(3) = REAL (GTHR)
      G(4) = AIMAG (GTHR)
      DY (3,1) = REAL (GTHR1)
      DY (4,1) = AIMAG (GTHR1)
120   CONTINUE
C
C     ************NUMERICAL COMPUTATIONS*****************************
C
C     RUNGE-KUTTA INTEGRATION TO PROVIDE INITIAL VALUES
C     FOR PREDICTOR-CORRECTOR INTEGRATION
C
      DO 30 IRK = 2,4
      CALL RKTZ(DP,P,G,GP,IRK)
      P=P+DP
      ZR=G(1)
      ZI=G(2)
      ZRK(IRK) = CMPLX (ZR,ZI)
      DY(1,IRK)=GP(1)
      DY(2,IRK)=GP(2)
      GO TO (150,160,170), IP
160   AHR = G(3)
      AHI = G(4)
      DY(3,IRK)=GP(3)
      DY(4,IRK)=GP(4)
      IF (MODE.NE.1) GO TO 162
      AFN (IRK) = CMPLX (G(3),G(4))
      AFN1 (IRK) = CMPLX (GP(3),GP(4))
      AR2 = G(1)*GP(3) - G(2)*GP(4) + GP(1)*G(3) - GP(2)*G(4)
      AI2 = G(2)*GP(3) + G(1)*GP(4) + GP(2)*G(3) + GP(1)*G(4)
      AFN2(IRK) = CMPLX (AR2,AI2)
162   GO TO 150
170   CONTINUE
      GAMR = G(3)
      GAMI = G(4)
      DY(3,IRK) = GP(3)
      DY(4,IRK) = GP(4)
150   CONTINUE
30    CONTINUE
      Y(1)=ZR
      Y(2)=ZI
      GO TO (180,190,200), IP
190   Y(3) = AHR
      Y(4) = AHI
      GO TO 180
200   CONTINUE
      Y(3) = GAMR
      Y(4) = GAMI
180   CONTINUE
C
C     PREDICTOR- CORRECTOR  INTEGRATION
      CALL  ZADAMS (DP,P,Y,DY,ITORZ)
C
C
C     CALCULATE LINEAR ADMITTANCE COEFFICIENTS.
      UE = U(NEND)
      CE = C(NEND)
      RHOE = CE ** (1./GMIN1)
      FR = WC * KP(MODE)
      F = UE ** .5 / (FR*GAM)
      IF  (ITORZ .EQ. 1) GO TO 35
      ZR=Y(1)
      ZI=Y(2)
      ZETA = CMPLX (ZR,ZI)
      LINADM = F * CMPLX(0.,1.) * ZETA
      GO TO 40
35    TR= Y(1)
      TI = Y(2)
      TAU = CMPLX (TR,TI)
      LINADM = F * CMPLX(0.,1.) / TAU
40    CONTINUE
      YR = REAL (LINADM)
      YI = AIMAG (LINADM)
      YMAG = CABS (LINADM)
      YPHASE = ATAN2 (YI,YR) * 180. / 3.1415927
      AMPL(MODE) = YMAG
      PHASE(MODE) = YPHASE
C
      GO TO (210,220,230), IP
220   AHR = Y(3)
      AHI = Y(4)
      IF  (MODE .NE. 1)  GO TO 210
      CONST = ACHMBR / AFN(NEND)
      DO 50  NP = 1,NEND
      AFN(NP) = CONST * AFN(NP)
      AFN1(NP) = CONST * AFN1(NP)
      AFN2(NP) = CONST * AFN2(NP)
50    CONTINUE
C
C     NONLINEAR ADMITTANCE COEFFICIENT IS ZERO FOR 1T MODE.
      GAMR = 0.
      GAMI = 0.
      GMAG = 0.
      GPHASE = 0.
      GBYY = 0.0
      GNOZ(1) = (0.0,0.0)
C
      GO TO 210
230   CONTINUE
C
C     CALCULATE NONLINEAR ADMITTANCE COEFFICIENTS.
      GAMR = Y(3)
      GAMI = Y(4)
      GMAG = (GAMR * GAMR + GAMI * GAMI) ** .5
      GPHASE = ATAN2 (GAMI,GAMR) * 180. / 3.1415927
      GBYY = CABS (CMPLX (GAMR,GAMI) / LINADM)
      GNOZ(MODE) = CMPLX(GAMR,GAMI)
C
210   CONTINUE
      IF (NOUT .EQ. 0) WRITE (6,1060) NAME(MODE), YR, YI,
     1    YMAG, YPHASE, GAMR, GAMI, GMAG, GPHASE, GBYY
500   CONTINUE
510   CONTINUE
520   CONTINUE
550   CONTINUE
      IF (NOUT .EQ. 0) GO TO 560
      DO 570 J = 1, 3
      IF (NOUT .EQ. 1) WRITE (7,7005) J, AMPL(J), PHASE(J)
      IF (NOUT .EQ. 2) PUNCH 7005 J, AMPL(J), PHASE(J)
570   CONTINUE
      IF (NOZNL1 .EQ. 0) GO TO 560
      DO 580 J = 1, 3
      IF (NOUT .EQ. 1) WRITE (7,7005) J, GNOZ(J)
      IF (NOUT .EQ. 2) PUNCH 7005 J, GNOZ(J)
580   CONTINUE
560   WRITE (6,1065)
C
C     ****************** READ FORMAT SPECIFICATIONS ********************
C
5005  FORMAT (6F10.0)
5010  FORMAT (3I5,F10.0)
5015  FORMAT (2F10.0)
C
C
C     ****************** WRITE FORMAT SPECIFICATIONS ********************
C
1005  FORMAT (1H1,//////////,45X,17H*****************,/,45X,
     1          17HNOZZLE PARAMETERS,/,45X,17H*****************,///////)
1010  FORMAT  (1H0,25X,"MACH NUMBER = ",F4.2)
1015  FORMAT  (1H0,25X,"GAMMA = ",F4.2)
1020  FORMAT  (1H0,25X,"NOZZLE ANGLE = ",F5.2)
1025  FORMAT  (1H0,25X,"LENGTH OF EXTENSION SECTION = ",F4.2)
1030  FORMAT  (1H0,25X,"RADIUS OF CURVATURE AT THE THROAT = ",F7.5)
1035  FORMAT  (1H0,25X,"RADIUS OF CURVATURE AT THE NOZZLE ENTRANCE = ",
     1          F7.5)
1040  FORMAT  (1H0,25X,"NUMBER OF STATIONS IN THE NOZZLE = ",I4)
1050  FORMAT  (1H1,////,46X,18H******************,/,46X,
     1          18HNOZZLE ADMITTANCES,/,46X,18H******************,//////,
     2          20X,"FREQUENCY = ",F8.6,40X,"PRESSURE AMPLITUDE = ",F6.4)
1055  FORMAT  (////////,5X,"MODE",10X,2HYR,9X,2HYI,9X,"YMAG",9X,"YPHASE",
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
     1         11X, 2HGR, 9X, 2HGI, 9X, 4HGMAG, 10X, 6HGPHASE, 13X, 3HG/Y, //)
1060  FORMAT  (1H0, 5X, A2, 2X, 3F12.4, F16.4, 3F12.4, 2F16.4)
1065  FORMAT  (1H1)
7005  FORMAT (I5, 2F10.5)
C
C     ****************************************************************
C
      STOP
      END
      SUBROUTINE STEADY
C
C     THIS SUBROUTINE EVALUATES STEADY-STATE QUANTITIES IN THE NOZZLE.
C
C     NOZZLE PROFILE AND FLOW PARAMETERS ARE PASSED TO THE SUBROUTINE
C     THROUGH THE COMMON BLOCKS X1 AND X2.
C     THE SUBPROGRAM PROVIDES THE OUTPUT THROUGH COMMON BLOCK X5.
C     U IS THE SQUARE OF THE STEADY-STATE VELOCITY;
C     DU IS THE DERIVATIVE OF U WITH RESPECT TO STEADY-STATE POTENTIAL ;
C     C IS THE SQUARE OF THE SPEED OF SOUND;
C     RW IS THE RADIUS OF THE NOZZLE.
C     THESE OUTPUT QUANTITIES ARE STORED IN THE RESPECTIVE ARRAYS AT
C     INTERVALS OF DP IN P (STEADY-STATE POTENTIAL).
C
C
      COMMON /X1/ CM,ANGLE,RCC,RCT,GAM,Q,RT,DP
      COMMON /X2/ T,R1,R2,NPLAST,NEND,IEXTN
      COMMON /X4/ RU(7),RDU(7),ZTHR1,GTHR1
      COMMON /X5/ U(1000),DU(1000),C(1000),RW(1000)
C
      T= 3.1415927*ANGLE/180.
      RT = (CM**.5) * ((1.+(GAM-1.)*CM**2/2.) ** ((-GAM-1.)/
     1     (4*(GAM-1))))*((2/(GAM+1)) ** ((-GAM-1)/(4.*(GAM-1))))
      SRTR = (RT*RCT) **.5
      Q = (.25*RT) * ((2./(GAM+1.)) ** ((GAM+1.) / (4.*(GAM-1))))
      R1 = RT+RCT*(1.-COS(T))
      R2 = 1.-RCC * (1.-COS(T))
      R=RT
      P= 0.
      RW(1) = RT
      U(1) = 2./(GAM+1.)
      RU(1) = U(1)
      C(1) = U(1)
      DU(1) = 4./((GAM+1.)*SRTR)
      RDU(1) = DU(1)
      G = U(1)
      DO 30  I=2,7
      CALL RKSTDY (P,G,GP)
      P = P + DP/2.
      RU(I) = G
      RDU(I)=GP
      IF (I .EQ. 2*(I/2))  GO TO 30
      NP = (I+1)/2
      U(NP) = RU(I)
      DU(NP) = RDU(I)
      C(NP) = 1.-(GAM-1)*U(NP)*.5
      RW(NP) = Q**((C(NP)) ** (-1./(2.*(GAM-1.))))
     1            *(U(NP)**-.25)*4.
 30   CONTINUE
      CALL  UADAMS (P)
      RETURN
      END
      SUBROUTINE RKSTDY(P,G,DUM)
C
C     THIS SUBROUTINE PERFORMS A FOURTH ORDER RUNGE-KUTTA INTEGRATION
C     TO OBTAIN STARTING VALUES OF STEADY-STATE VELOCITY FOR THE
C     PREDICTOR-CORRECTOR METHOD.
C     P IS THE CURRENT VALUE OF THE STEADY-STATE POTENTIAL: INPUT.
C     G IS THE SQUARE THE STEADY-STATE VELOCITY: INPUT AND OUTPUT.
C     AS OUTPUT, G IS THE VALUE AT THE NEXT STEP.
C     DUM IS DERIVATIVE OF THE SQUARE OF STEADY-STATE VELOCITY: OUTPUT.
C     DUM IS OBTAINED BY CALLING SUBROUTINE RKUDIF.
C
C
      COMMON /X1/ CM,ANGLE,RCC,RCT,GAM,Q,RT,DP
      DIMENSION A(4),FZ(4)
C
      A(1) = 0.
      A(2) = 0.5
      A(3) = 0.5
      A(4) = 1.
      H = DP/2.
      PR=P
      GR=G
      CALL  RKUDIF(PR,GR,DUM)
      FZ(1) = DUM
      DO  30  I=2,4
      PR = P+A(I)*H
      GR = G+A(I)*H*FZ(I-1)
      CALL  RKUDIF (PR,GR,DUM)
      FZ(I) = DUM
30    CONTINUE
      G = G + H* (FZ(1) + 2*(FZ(2)+FZ(3)) + FZ(4))/6.
      CALL  RKUDIF(PR,G,DUM)
      RETURN
      END
      SUBROUTINE  RKUDIF(P,G,GP)
C
C     THIS SUBROUTINE EVALUATES THE DIFFERENTIAL ELEMENT IN THE
C     RUNGE-KUTTA INTEGRATION SCHEME FOR SOLVING THE EQUATION FOR SQUARE
C     OF STEADY-STATE VELOCITY.
C
C     P IS THE VALUE OF STEADY-STATE POTENTIAL AT THE STATION,
C     WHERE DIFFERENTIAL ELEMENT IS SOUGHT; INPUT.
C     G IS THE VALUE OF THE FUNCTION AT P; INPUT.
C     GP IS THE REQUIRED DIFFERENTIAL ELEMENT.
C
      COMMON /X1/ CM,ANGLE,RCC,RCT,GAM,Q,RT,DP
      COMMON /X2/ T,R1,R2,NPLAST,NEND,IEXTN
      COMMON /X3/ WC,SVN,IP,MODE,NU,KP(3)
C
      IF (P) 15,10,15
10    GP = 4./ ((GAM+1.) * ((RCT*RT) **.5))
      GO TO 20
15    C = 1-(GAM-1.)*G*.5
      R = Q*((C) ** (-1./(2.*(GAM-1.)))) * (G**-.25) * 4.
      IF  (R-1.) 22,22,50
22    IF  (R-R1) 25,30,30
25    DR = -((2.*RCT*(R-RT) - (R-RT) * (R-RT))**.5) / (RT+RCT-R)
      GO TO 45
30    IF  (R-R2)  35,40,40
35    DR = -TAN(T)
      GO TO 45
40    DR = ((2.*RCC*(1-R) - (R-1)*(R-1)) **.5) / (1.-R-RCC)
45    DU = -(G**.75)*(C**((2.*GAM-1) / (2.*(GAM-1.)))) /
     1            (Q*(1.-(GAM+1.) * G*.5))
      GP = DU*DR
      GO TO 20
50    GP = 0.
20    RETURN
      END
      SUBROUTINE  UADAMS(P)
C
C     THIS SUBROUTINE CARRIES OUT A MODIFIED ADAMS PREDICTOR-CORRECTOR
C     INTEGRATION SCHEME TO SOLVE THE DIFFERENTIAL EQUATION FOR THE
C     STEADY-STATE VELOCITY.
C
C     P IS THE VALUE OF THE STEADY-STATE POTENTIAL AT THE STATION,
C     WHERE PREDICTOR-CORRECTOR INTEGRATION COMMENCES; INPUT.
C     DURING THE PROGRAM, P IS CHANGED TO THE VALUE AT CURRENT STATION.
C     H IS THE STEP-SIZE; INPUT THROUGH COMMON BLOCK X1.
C     COMMON BLOCKS X1 AND X2 PROVIDE DETAILS OF NOZZLE PROFILE.
C
C     THE STEADY-STATE QUANTITIES ARE THE OUTPUT, AND
C     ARE PROVIDED BY MEANS OF COMMON BLOCK X5.
C
C
      COMMON /X1/ CM,ANGLE,RCC,RCT,GAM,Q,RT,H
      COMMON /X2/ T,R1,R2,NPLAST,NEND,IEXTN
      COMMON /X5/ U(1000),DU(1000),C(1000),RW(1000)
C
      NP=4
10    CONTINUE
      PRED = U(NP) + H*(55.*DU(NP) - 59.*DU(NP-1) + 37.*DU(NP-2)
     1               -9.*DU(NP-3))/24.0
      P = P + H
      NP = NP + 1
      UP = PRED
      CP = 1.-(GAM-1.)*UP*.5
      R = Q*(CP**(-1./ (2.*(GAM-1.)))) * (UP**-.25)*4.
C
C     IF R = 1, THE NOZZLE ENTRANCE HAS BEEN REACHED.
      IF (R-1.)  17,17,100
C
17    IF  (R-R1) 20,25,25
20    DR = -((2.*RCT*(R-RT) - (R-RT)*(R-RT))**.5) / (RT+RCT-R)
      GO TO 40
25    IF (R-R2) 30,35,35
30    DR=-TAN(T)
      GO TO 40
35    DR = ((2.*RCC*(1.-R) - (1.-R)*(1.-R))**.5) / (1.-R-RCC)
40    DQ = -(UP**.75) * (CP**((2.*GAM-1) / (2.*(GAM-1))))/
     1         (Q*(1.-(GAM+1.) * UP * .5))
      DUP = DR*DQ
      COR = U(NP-1)+H* (9.*DUP+19.*DU(NP-1) - 5.*DU(NP-2)
     1         +DU(NP-3))/24.0
      UP = (251.*COR + 19.*PRED) / 270.
      CP = 1.-(GAM-1.)*UP*.5
      R = Q*(CP**(-1./ (2.*(GAM-1.)))) * (UP**-.25)*4.
C
C     IF R = 1, THE NOZZLE ENTRANCE HAS BEEN REACHED
      IF (R-1.)  62,62,100
C
62    IF (R-R1) 65,70,70
65    DR = -((2.*RCT*(R-RT) - (R-RT)*(R-RT))**.5) / (RT+RCT-R)
      GO TO 85
70    IF (R-R2)  75,80,80
```

```
75    DR = -TAN(T)
      GO TO 85
80    DR = ((2.*RCC*(1.-R) - (1.-R)*(1.-R))**.5) / (1.-R-RCC)
85    DQ = -(UF**.75) * (CP**((2.*GAM-1) / (2.*(GAM-1))))/
     1         (Q*(1.-(GAM+1.) * UP * .5))
      IF (NP .GT. 1000)  GO TO 87
C
C     STORE STEADY STATE QUANTITIES AT STATION NF IN RESPECTIVE ARRAYS.

      DU(NP)=DR*DQ
      U(NP) = UF
      C(NP) = CP
      RW(NP) = R
C
87    GO TO 10
100   NFLAST= NP-1
      RETURN
      END
      SUBROUTINE  COEFFS  (U,DU,C,R,CC)
C
C     THIS SUBROUTINE COMPUTES THE COEFFICIENTS.
C     U,DU,C,R ARE THE STEADY-STATE QUANTITIES AT THE AXIAL LOCATION,
C     WHERE THE COEFFICIENTS ARE REQUIRED.
C     CC ARE THE COMPLEX COEFFICIENTS.
C     SUBROUTINE INTGRL PROVIDES ALPHA & BETA, THE VALUES OF TRANSVERSE
C     INTEGRALS THROUGH COMMON BLOCK X7.
C
      COMMON /X3/ WC,SVN,IP,MODE,NU,KP(3)
      COMMON/X7/ ALPHA(5,3), BETA(9,3)
      COMPLEX CC(25)
      DATA GAM/1.2/
C
      GMIN1 = GAM - 1.
      M = MODE
      A4B6 = ALPHA (4,M) * BETA (6,M)
      RSQR = R * R
C
C********** LINEAR  COEFFICIENTS ************************************
C
      CCR = U * (C-U)
      CC(1) = CMPLX(CCR,0.0)
      CCR = - U*DU / C
      CC(2) = CMPLX(CCR,0.0)
      CCR = C * (BETA (8,M) - BETA (7,M)) / (RSQR * BETA (6,M))
      CC(3) = CMPLX(CCR,0.0)
      CCR = 2. * C * BETA (7,M) / (RSQR * BETA (6,M))
      CC(4) = CMPLX(CCR,0.0)
C
      CCR = C * ALPHA (5,M) * BETA (9,M) / (RSQR * A4B6)
      CC(5) = CMPLX(CCR,0.0)
      CCR = 0.0
      CCI = - 2. * WC * U * KP(M)
      CC(6) = CMPLX (CCR,CCI)
      CCR = 0.0
      CCI = - GMIN1 * WC * KP(M) * U * DU / (2. * C)
      CC(7) = CMPLX (CCR,CCI)
      CCR = (WC * KP(M)) **2
      CCI = 0.0
      CC(8) = CMPLX (CCR,CCI)
      IF (IP .NE. 3)  GO TO 110
C
C********** NONLINEAR  COEFFICIENTS *********************************
C
      A1 = ALPHA (1,M)
      A2 = ALPHA (2,M)
      A3 = ALPHA (3,M)
      B1 = BETA (1,M)
      B2 = BETA (2,M)
      B3 = BETA (3,M)
      B4 = BETA (4,M)
      B5 = BETA (5,M)
      CCR = - .5 * A1*B1 * WC*U / A4B6
      CCI = CCR
      CC(9) = CMPLX (CCR,CCI)
```

```
       CCR = - .5 * A1 * B3 * WC / (RSQR * A4B6)
      CCI = CCR
      CC(10) = CMPLX (CCR,CCI)
C
      CCR = - .5 * A2*B2 * WC / (RSQR * A4B6)
      CCI = CCR
      CC(11) = CMPLX (CCR,CCI)
      CCR = - ((GAM+1.) * U*U * A1*B1) / (4.*3.1415927*A4B6)
      CCI = - CCR
      CC(12) = CMPLX (CCR,CCI)
      CCR = - (U * A1 * B3) / (4. * RSQR * A4B6)
      CCI = - CCR
      CC(13) = CMPLX (CCR,CCI)
      CCR = - (U * A2 * B2) / (4. * RSQR * A4B6)
      CCI = - CCR
      CC(14) = CMPLX (CCR,CCI)
      CCR = - 3.*U * (1. + .5*GMIN1 * U*DU/C) * A1*B1 / (8.*A4B6)
      CCI = - CCR
      CC(15) = CMPLX (CCR,CCI)
C
      CCR = - DU * (1. - (2.-GAM) * U/C) * A1 * B3 / (16 * RSQR * A4B6)
      CCI = - CCR
      CC(16) = CMPLX (CCR,CCI)
      CCR = - DU * (1. - (2.-GAM) * U/C) * A2 * B2 / (16 * RSQR * A4B6)
      CCI = - CCR
      CC(17) = CMPLX (CCR,CCI)
      CCR = - (GMIN1 * WC * A1 * B1) / (4. * A4B6)
      CCI = CCR
      CC(18) = CMPLX (CCR,CCI)
      CCR = - (GMIN1 * WC * U * DU * A1 * B1) / (4. * C * A4B6)
      CCI = CCR
      CC(19) = CMPLX (CCR,CCI)
      CCR = - GMIN1 * WC * A1 * (B4 - B5) / (4. * RSQR * A4B6)
      CCI = CCR
      CC(20) = CMPLX (CCR,CCI)
C
      CCR = - GMIN1 * A1 * B5 / (2. * RSQR * A4B6)
      CCI = CCR
      CC(21) = CMPLX (CCR,CCI)
      CCR = - GMIN1 * A3 * B2 / (4. * RSQR * A4B6)
      CCI = CCR
      CC(22) = CMPLX (CCR,CCI)
      CCR = - GMIN1 * U*A1 * (B4 - B5) / (4. * RSQR * A4B6)
      CCI = - CCR
      CC(23) = CMPLX (CCR,CCI)
      CCR = - GMIN1 * U * A1 * B5 / (2.*RSQR * A4B6)
      CCI = - CCR
      CC(24) = CMPLX (CCR,CCI)
      CCR = - GMIN1 * U * A3 * B2 / (4.*RSQR * A4B6)
      CCI = - CCR
      CC(25) = CMPLX (CCR,CCI)
110   CONTINUE
      RETURN
      END
      SUBROUTINE  INTGRL
C
C      THIS SUBROUTINE EVALUATES THE DIFFERENT TRANSVERSE INTEGRALS.
C
      COMMON/X7/ ALPHA(5,3), BETA(9,3)
      S1 = 1.84118
      S2 = 3.05424
      S3 = 3.83171
      PI = 3.14159 27
C
C     **************TANGENTIAL INTEGRALS*******************************
C
      DO 20  NOPT = 1,3
20    ALPHA (NOPT,1) =0.
      ALPHA (4,1)  =  1.0
      ALPHA (5,1)  = -1.0
      ALPHA (1,2)  =  0.5
      ALPHA (2,2)  = -0.5
      ALPHA (3,2)  = -0.5
      ALPHA (4,2)  =  1.0
      ALPHA (5,2)  = -4.0
      ALPHA (1,3)  =  1.0
      ALPHA (2,3)  =  1.0
      ALPHA (3,3)  = -1.0
      ALPHA (4,3)  =  2.0
      ALPHA (5,3)  =  0.0
      DO 30  I = 1,5
      DO 30  J = 1,3
30    ALPHA(I,J)  =  PI*ALPHA(I,J)
C
C     *************RADIAL INTEGRALS*************************************
C
      DO 40  MODE = 1,3
      GO TO (110,120,130),  MODE
110   M=1
      S=S1
      GO TO 140
120   M=2
      S=S2
      GO TO 140
130   M=0
      S=S3
140   CONTINUE
      BETA (1,MODE)  =  RAD2 (1,1,1,M,S1,S1,S)
      BETA (2,MODE)  =  RAD2 (2,1,1,M,S1,S1,S)
      BETA (3,MODE)  =  RAD2 (7,1,1,M,S1,S1,S)
      BETA (4,MODE)  =  RAD2 (8,1,1,M,S1,S1,S)
      BETA (5,MODE)  =  RAD2 (5,1,1,M,S1,S1,S)
      BETA (6,MODE)  =  RAD1 (1,M,S)
      BETA (7,MODE)  =  RAD1 (4,M,S)
      BETA (8,MODE)  =  RAD1 (5,M,S)
      BETA (9,MODE)  =  RAD1 (2,M,S)
40    CONTINUE
      RETURN
      END
      FUNCTION RAD1 (NOPT,M,B)
C
C     THIS SUBROUTINE CALCULATES THE INTEGRAL OVER THE INTERVAL
C     (0,1) OF THE FOLLOWING PRODUCTS OF TWO BESSEL FUNCTIONS
C
C     NOPT = 1  JM(B*R) * JM(B*R) * R
C
C     NOPT = 2  JM(B*R) * JM(B*R)/R
C
C     NOPT = 3  JPM(B*R) * JM(B*R) * R
C
C     NOPT = 4  JPM(B*R) * JM(B*R)
C
C     NOPT = 5  JPPM(B*R) * JM(B*R) * R
C
C     JM  IS THE BESSEL FUNCTION OF FIRST KIND OF ORDER M
C     JPM IS THE DERIVATIVE OF JM WITH RESPECT TO R
C     JPPM IS THE SECOND DERIVATIVE OF JM WITH RESPECT TO R
C     M IS A NON-NEGATIVE INTEGER
C     B IS A REAL NUMBER
C
C
      DIMENSION  FUNCT(200)
      DOUBLE PRECISION  DN, DH, DSTEP, DR, ARG, BES1, BES2, BESH,
     1                  BESL, PROD, FUNCT, S1, S2, S3
C
      NN = 100
      DN = NN
      DH = 1.0/DN
      NP1 = NN + 1
C
C     ************* CALCULATION OF INTEGRANDS *************************
C
      DO 160 I = 1, NP1
      DSTEP = I - 1
      DR = DH * DSTEP
      ARG = B * DR
C
C     CALCULATE BESSEL FUNCTIONS.
      CALL JBES(M,ARG,BES2,$500)
      BES1 = BES2
      IF (NOPT .LT. 3)  GO TO 130
C
C     CALCULATE FIRST DERIVATIVES OF BESSEL FUNCTIONS.
      CALL JBES(M+1,ARG,BESH,$500)
      IF (NOPT .EQ. 5)  GO TO 120
      IF (I .EQ. 1)  GO TO 115
      RM = M
      BES1 = B * (RM*BES1/ARG - BESH)
      GO TO 130
  115 IF (M .EQ. 0)  GO TO 117
      CALL JBES(M-1,ARG,BESL,$500)
      BES1 = B * (BESL - BESH)/2.0
      GO TO 130
  117 CALL JBES(1,ARG,BES1,$500)
      BES1 = -BES1 * B
      GO TO 130
C
C     CALCULATE SECOND DERIVATIVES OF BESSEL FUNCTIONS.
  120 IF (I .EQ. 1)  GO TO 122
      RM = M
      F = RM * (RM - 1.0)/(ARG * ARG)
      BES1 = ((F - 1.0) * BES1  +  BESH/ARG) * B * B
      GO TO 130
  122 CALL JBES(M+2,ARG,BESH,$500)
      IF (M .EQ. 0)  BES1 = 0.5 * B * B * (BESH - BES1)
      IF (M .EQ. 1)  BES1 = 0.25 * B * B *(BESH - 3.0*BES1)
      IF (M .LT. 2)  GO TO 130
      CALL JBES(M-2,ARG,BESL,$500)
      BES1 = 0.25 * B * B * (BESL - 2.0*BES1 + BESH)
C
  130 PROD = BES1 * BES2
C
C     CALCULATE WEIGHTING FUNCTIONS AND LIMITS FOR R = 0.
      IF (NOPT .EQ. 2)  GO TO 140
      IF (NOPT .EQ. 4)  GO TO 150
      FUNCT(I) = PROD * DR
      GO TO 160
  140 IF (I .EQ. 1)  GO TO 145
      FUNCT(I) = PROD/DR
      GO TO 160
  145 FUNCT(I) = 0.0
      GO TO 160
  150 FUNCT(I) = PROD
C
  160 CONTINUE
C
C
C     ************** SIMPSONS RULE INTEGRATION **************************
C
      NM1 = NN - 1
      S1 = FUNCT(1) + FUNCT(NP1)
      S2 = 0.0
      S3 = 0.0
      DO 20 I = 2, NN, 2
      S2 = S2 + FUNCT(I)
   20 CONTINUE
      DO 30 I = 3, NM1, 2
      S3 = S3 + FUNCT(I)
   30 CONTINUE
      RESULT = DH * (S1 + 4.0*S2 + 2.0*S3)/3.0
      RAD1 = RESULT
      GO TO 501
  500 WRITE (6, 6000)
 6000 FORMAT (1H1,10HERROR JBES)
  501 CONTINUE
      RETURN
      END
      FUNCTION  RAD2 (NOPT,L,M,N,A,B,C)
C
C     THIS SUBROUTINE CALCULATES THE INTEGRAL OVER THE INTERVAL
C     (0,1) OF THE FOLLOWING PRODUCTS OF THREE BESSEL FUNCTIONS
C
C     NOPT = 1  JL(A*R) * JM(B*R) * JN(C*R) * R
C
C     NOPT = 2  JL(A*R) * JM(B*R) * JN(C*R)/R
C
C     NOPT = 3  JL(A*R) * JM(B*R) * JN(C*R)/(R*R)
C
C     NOPT = 4  JPL(A*R) * JM(B*R) * JN(C*R) * R
C
C     NOPT = 5  JPL(A*R) * JM(B*R) * JN(C*R)
C
C     NOPT = 6  JPL(A*R) * JM(B*R) * JN(C*R)/R
C
C     NOPT = 7  JPL(A*R) * JPM(B*R) * JN(C*R) * R
C
C     NOPT = 8  JPPL(A*R) * JM(B*R) * JN(C*R) * R
C
C     NOPT = 9  JPPL(A*R) * JPM(B*R) * JN(C*R) * R
C
C     JL  IS THE BESSEL FUNCTION OF FIRST KIND OF ORDER L
C     JPL IS THE DERIVATIVE OF JL WITH RESPECT TO R
C     JPPL IS THE SECOND DERIVATIVE OF JL WITH RESPECT TO R
C     L, M, N  ARE NON-NEGATIVE INTEGERS
C     A, B, C ARE REAL NUMBERS
C
C
      DIMENSION  FUNCT(200)
      DOUBLE PRECISION  DN, DH, DSTEP, DR, ARG1, ARG2, ARG3,
     1                  BES1, BES2, BES3, BESH, BESL, PROD,
     2                  FUNCT, BESLIM, S1, S2, S3
C
      NN = 100
      DN = NN
      DH = 1.0/DN
      NP1 = NN + 1
C
C     ************* CALCULATION OF INTEGRANDS **************************
C
      DO 160 I = 1, NP1
      DSTEP = I - 1
      DR = DH * DSTEP
      ARG1 = A * DR
      ARG2 = B * DR
      ARG3 = C * DR
C
C     CALCULATE BESSEL FUNCTIONS.
      CALL JBES(N,ARG3,BES3,$500)
      CALL JBES(L,ARG1,BES1,$500)
      CALL JBES(M,ARG2,BES2,$500)
      IF ((NOPT .EQ. 7) .OR. (NOPT .EQ. 9))  GO TO 105
      GO TO 110
C
```

```
C     CALCULATE FIRST DERIVATIVES OF BESSEL FUNCTIONS.
  105 CALL JBES(M+1,ARG2,BESH,$500)
      IF (I .EQ. 1)  GO TO 107
      RM = M
      BES2 = B * (RM*BES2/ARG2 - BESH)
      GO TO 110
  107 IF (M .EQ. 0)  GO TO 109
      CALL JBES(M-1,ARG2,BESL,$500)
      BES2 = B * (BESL - BESH)/2.0
      GO TO 110
  109 CALL JBES(1,ARG2,BES2,$500)
      BES2 = -BES2 * B
  110 IF (NOPT .LT. 4)  GO TO 130
      CALL JBES(L+1,ARG1,BESH,$500)
      IF (NOPT .GT. 7)  GO TO 120
      IF (I .EQ. 1)  GO TO 115
      RL = L
      BES1 = A * (RL*BES1/ARG1 - BESH)
      GO TO 130
  115 IF (L .EQ. 0)  GO TO 117
      CALL JBES(L-1,ARG1,BESL,$500)
      BES1 = A * (BESL - BESH)/2.0
      GO TO 130
  117 CALL JBES(1,ARG1,BES1,$500)
      BES1 = -BES1 * A
      GO TO 130
C
C     CALCULATE SECOND DERIVATIVES OF BESSEL FUNCTIONS.
  120 IF (I .EQ. 1)  GO TO 122
      RL = L
      F = RL * (RL - 1.0)/(ARG1 * ARG1)
      BES1 = ((F - 1.0) * BES1  +  BESH/ARG1) * A * A
      GO TO 130
  122 CALL JBES(L+2,ARG1,BESH,$500)
      IF (L .EQ. 0)  BES1 = 0.5 * A * A * (BESH - BES1)
      IF (L .EQ. 1)  BES1 = 0.25 * A * A *(BESH - 3.0*BES1)
      IF (L .LT. 2)  GO TO 130
      CALL JBES(L-2,ARG1,BESL,$500)
      BES1 = 0.25 * A * A * (BESL - 2.0*BES1 + BESH)
C
  130 PROD = BES1 * BES2 * BES3
C
C     CALCULATE WEIGHTING FUNCTIONS AND LIMITS FOR R = 0.
      IF ((NOPT .EQ. 2) .OR. (NOPT .EQ. 6))  GO TO 133
      IF (NOPT .EQ. 3)  GO TO 136
      IF (NOPT .EQ. 5)  GO TO 140
      FUNCT(I) = PROD * DR
      GO TO 160
  133 IF (I .EQ. 1)  GO TO 134
      FUNCT(I) = PROD/DR
      GO TO 160
  134 BESLIM = 0.0
      IF (NOPT .EQ. 6)  GO TO 135
      IF ((L.EQ.1) .AND. (M.EQ.0) .AND. (N.EQ.0))  BESLIM = A/2.0
      IF ((L.EQ.0) .AND. (M.EQ.1) .AND. (N.EQ.0))  BESLIM = B/2.0
      IF ((L.EQ.0) .AND. (M.EQ.0) .AND. (N.EQ.1))  BESLIM = C/2.0
      GO TO 155
  135 IF ((L.EQ.0) .AND. (M.EQ.0) .AND. (N.EQ.0))  BESLIM = -A*A/2.0
      IF ((L.EQ.1) .AND. (M.EQ.1) .AND. (N.EQ.0))  BESLIM = A*B/4.0
      IF ((L.EQ.1) .AND. (M.EQ.0) .AND. (N.EQ.1))  BESLIM = A*C/4.0
      IF ((L.EQ.2) .AND. (M.EQ.0) .AND. (N.EQ.0))  BESLIM = A*A/4.0
      GO TO 155
  136 IF (I .EQ. 1)  GO TO 138
      FUNCT(I) = PROD/(DR*DR)
      GO TO 160
  138 BESLIM = 0.0
      IF ((L.EQ.2) .AND. (M.EQ.0) .AND. (N.EQ.0))  BESLIM = A*A/8.0
      IF ((L.EQ.0) .AND. (M.EQ.2) .AND. (N.EQ.0))  BESLIM = B*B/8.0
      IF ((L.EQ.0) .AND. (M.EQ.0) .AND. (N.EQ.2))  BESLIM = C*C/8.0
      IF ((L.EQ.1) .AND. (M.EQ.1) .AND. (N.EQ.0))  BESLIM = A*B/4.0
      IF ((L.EQ.1) .AND. (M.EQ.0) .AND. (N.EQ.1))  BESLIM = A*C/4.0
      IF ((L.EQ.0) .AND. (M.EQ.1) .AND. (N.EQ.1))  BESLIM = B*C/4.0
      GO TO 155
  140 FUNCT(I) = PROD
      GO TO 160
  155 FUNCT(I) = BESLIM
C
  160 CONTINUE
C
C
C     ************** SIMPSONS RULE INTEGRATION *************************
C
      NM1 = NN - 1
      S1 = FUNCT(1) + FUNCT(NP1)
      S2 = 0.0
      S3 = 0.0
      DO 20 I = 2, NN, 2
      S2 = S2 + FUNCT(I)
   20 CONTINUE
      DO 30 I = 3, NM1, 2
      S3 = S3 + FUNCT(I)
   30 CONTINUE
      RESULT = DH * (S1 + 4.0*S2 + 2.0*S3)/3.0
      RAD2 = RESULT
      GO TO 501
  500 WRITE (6, 6000)
 6000 FORMAT (1H1,10HERROR JBES)
  501 CONTINUE
      RETURN
      SUBROUTINE RKTZ(H,T1,G,DUM,IRK)
C
C     THIS SUBROUTINE PERFORMS A FOURTH ORDER RUNGE-KUTTA INTEGRATION
C     TO OBTAIN THE INITIAL VALUES FOR THE PREDICTOR-CORRECTOR METHOD.
C
C     NU IS THE NUMBER OF DIFFERENTIAL EQUATIONS TO BE SOLVED.
C     IF IP = 1,INTEGRATION IS CARRIED OUT FOR ZETA ONLY (NU = 2).
C     IF IP = 2, INTEGRATION IS CARRIED OUT FOR ZETA AND AH (NU = 4).
C     IF IP = 3, INTEGRATION IS CARRIED OUT FOR ZETA AND GAMMA (NU = 4).
C     IP IS PASSED TO THIS SUBROUTINE THROUGH BLOCK COMMON X3.
C
C     H IS THE STEP-SIZE; INPUT.
C     T1 IS THE CURRENT VALUE OF STEADY STATE POTENTIAL; INPUT.
C     G ARE THE VALUES OF THE FUNCTIONS AT THE NEXT STEP; OUTPUT.
C     DUM ARE THE VALUES OF THE DERIVATIVES OF THE FUNCTIONS
C     AT THE NEXT STEP; OUTPUT.
C     DUM ARE OBTAINED BY CALLING SUBROUTINE RKDIF.
C
C
      COMMON /X3/ WC,SVN,IP,MODE,NU,KP(3)
      DIMENSION A(4),G(4),GZ(4),DUM(4),FZ(4,4)
      A(1)=0.
      A(2)=.5
      A(3)=.5
      A(4)=1.
      TZ=T1
      NU=4
      IF (IP.EQ.1) NU=2
      DO 10 J=1,NU
 10   GZ(J)=G(J)
      IK=1
      CALL RKDIF(TZ,GZ,DUM,IK,IRK)
      DO 25 J=1,NU
 25   FZ(1,J)=DUM(J)
      DO 30 IK=2,4
      TZ=T1+A(IK)*H
      DO 35 J=1,NU
 35   GZ(J)=G(J)+A(IK)*H*FZ(IK-1,J)
      CALL RKDIF(TZ,GZ,DUM,IK,IRK)
      DO 50 J=1,NU
 50   FZ(IK,J)=DUM(J)
 30   CONTINUE
      DO 55 J=1,NU
 55   G(J)=G(J)+H*(FZ(1,J)+2.*(FZ(2,J)+FZ(3,J))+FZ(4,J))/6.
      IK=4
      CALL RKDIF(TZ,G,DUM,IK,IRK)
 75   RETURN
      END
      SUBROUTINE RKDIF(P,G,GP,IK,IRK)
C
C      THIS SUBROUTINE EVALUATES THE DIFFERENTIAL ELEMENT IN THE
C     RUNGE-KUTTA INTEGRATION SCHEME.
C     P IS THE CURRENT VALUE OF STEADY-STATE POTENTIAL; INPUT.
C     G ARE THE VALUES OF THE FUNCTIONS AT P; INPUT.
C     GP ARE THE DERIVATIVES OF FUNCTIONS AT P; OUTPUT.
C
C
      COMMON /X1/ CM,ANGLE,RCC,RCT,GAM,Q,RT,DP
      COMMON /X2/ T,R1,R2,NPLAST,NEND,IEXTN
      COMMON /X3/ WC,SVN,IP,MODE,NU,KP(3)
      COMMON /X4/ RU(7),RDU(7),ZTHR1,GTHR1
      COMMON /X6/ AFN,AFN1,AFN2
      COMPLEX  AFN(1000),AFN1(1000),AFN2(1000)
      COMPLEX  CC(25),CFH,CFM,CFN,INHMG
      COMPLEX  ZETA,ZETA1,AH,AH1,CGAM,CGAM1,ZTHR1,GTHR1,AP,AP1,AP2
      DIMENSION G(4),GP(4)
C
      ZR = G(1)
      ZI = G(2)
      ZETA = CMPLX (ZR,ZI)
      GO TO (110,120,130),IP
120   AHR = G(3)
      AHI = G(4)
      AH = CMPLX (AHR,AHI)
      GO TO 110
130   CONTINUE
      GAMR = G(3)
      GAMI = G(4)
      CGAM = CMPLX (GAMR,GAMI)
110   CONTINUE
      IF (P) 15,10,15
10    GP(1) = REAL(ZTHR1)
      GP(2) = AIMAG(ZTHR1)
      GO TO (140,150,160), IP
150   AH1 = AH * ZETA
      GP(3) = REAL (AH1)
      GP(4) = AIMAG (AH1)
      GO TO 140
160   CONTINUE
      GP(3) = REAL (GTHR1)
      GP(4) = AIMAG (GTHR1)
140   CONTINUE
      GO TO 20
15    ICL = 2*IRK - 2
      IF (IK .EQ. 1)  ICL = 2*IRK - 3
      IF (JK .EQ. 4)  ICL = 2*IRK - 1
      U=RU(ICL)
      DU=RDU(ICL)
      C=1.+(GAM-1.)*U*.5
      R=Q*((C)**(-1/(2*(GAM-1.))))*(U**-.25)*4.
      CALL  COEFFS (U,DU,C,R,CC)
      CFH = CC(1)
      CFM = CC(2) + CC(6)
      CFN = CC(3) + CC(4) + CC(5) + CC(7) + CC(8)
      ZETA1 = ( -CFM * ZETA - CFN) / CFH - ZETA * ZETA
      GP(1) = REAL (ZETA1)
      GP(2) = AIMAG (ZETA1)
      GO TO (170,180,190), IP
180   AH1 = AH * ZETA
      GP(3) = REAL (AH1)
      GP(4) = AIMAG (AH1)
      GO TO 170
190   CONTINUE
      GO TO (30,40,40,50), IK
30    AP = AFN (IRK-1)
      AP1 = AFN1 (IRK-1)
      AP2 = AFN2 (IRK-1)
      GO TO 60
40    AP = .5 * (AFN (IRK-1) + AFN (IRK))
      AP1 = .5 * (AFN1 (IRK-1) + AFN1 (IRK))
      AP2 = .5 * (AFN2 (IRK-1) + AFN2 (IRK))
      GO TO 60
50    AP = AFN (IRK)
      AP1 = AFN1 (IRK)
      AP2 = AFN2 (IRK)
60    CONTINUE
      INHMG = - CC(18) * AP * AP2 - CC(12) * AP1 * AP2 - (CC(9)
     1        + CC(15)) * AP1 * AP1 - (CC(13) + CC(14) + CC(19)
     2        + CC(23) + CC(24) + CC(25)) * AP1 * AP - (CC(10) + CC(11)
     3        + CC(17) + CC(20) + CC(21) + CC(22)) * AP * AP
      CGAM1 = ( - ZETA + .5* (GAM-1.) * DU/C - CFM/CFH) * CGAM
     1          - INHMG / (C * CFH)
      GP(3) = REAL (CGAM1)
      GP(4) = AIMAG (CGAM1)
170   CONTINUE
20    RETURN
      END
      SUBROUTINE  ZADAMS (H,X,Y,DY,ITORZ)
C
C     THIS SUBROUTINE CARRIES OUT A MODIFIED ADAMS PREDICTOR-CORRECTOR
C     INTEGRATION SCHEME TO SOLVE THE VARIOUS DIFFERENTIAL EQUATIONS AS
C     DESCRIBED BELOW
C     IF IP = 1, INTEGRATION IS CARRIED OUT FOR ZETA ONLY;
C     IF IP = 2, INTEGRATION IS CARRIED OUT FOR ZETA AND AH;
C     IF IP = 3, INTEGRATION IS CARRIED OUT FOR ZETA AND GAMMA.
C     IP IS PASSED TO THE SUBROUTINE THROUGH COMMON BLOCK X3.
C
C     H IS THE STEP-SIZE; INPUT.
C     X IS THE VALUE OF STEADY-STATE POTENTIAL AT THE STATION ,
C     WHERE THE PREDICTOR-CORRECTOR INTEGRATION STARTS; INPUT.
C     DURING THE PROGRAM, X IS CHANGED TO VALUE AT CURRENT STATION.
C     Y ARE THE VALUES AT X , OF THE FUNCTIONS, WHOSE EQUATIONS ARE
C     BEING SOLVED; INPUT AND OUTPUT.
C     DY ARE THE DERIVATIVES OF Y; INPUT AND OUTPUT.
C
C     ITORZ PASSES TO MAIN PROGRAM THE INFORMATION AS TO WHICH VARIABLE
C     (TAU OR ZETA) HAS BEEN INTEGRATED.
C     ITORZ = 1  :  INTEGRATION OF EQUATION FOR TAU.
C     ITORZ = 2  :  INTEGRATION OF EQUATION FOR ZETA.
C
C
      COMMON /X1/ CM,ANGLE,RCC,RCT,GAM,Q,RT
      COMMON /X2/ T,R1,R2,NFLAST,NEND,IEXTN
      COMMON /X3/ WC,SVN,IP,MODE,NU,KP(3)
      COMMON /X5/ U(1000),DU(1000),C(1000),RW(1000)
      COMMON /X6/ AFN,AFN1,AFN2
      COMMON /X8/ ZETA,TAU,CCEXT
      COMPLEX  ZETA(1000),TAU(1000),CCEXT(25)
      COMPLEX  AFN(1000),AFN1(1000),AFN2(1000)
      COMPLEX  CC(25),CFH,CFM,CFN,INHMG,ZETA1,AH,AH1,AH2,AP,AP1,AP2,
     1         CGAM,CGAM1
      DIMENSION  Y(4),DY(4,4),DP(4),PRED(4),COR(4)
C
      NP=4
      ITORZ = 2
      IF (IEXTN .NE. 1)  GO TO 10
C
C     DEFINE STEADY STATE QUANTITIES IN THE EXTENSION REGION.
C
      UEXT = U(NEND)
      CEXT = C(NEND)
      REXT = RW(NEND)
      DUEXT = DU(NEND)
      CALL COEFFS (UEXT,DUEXT,CEXT,REXT,CCEXT)
C
C     NU IS THE NUMBER OF EQUATIONS TO BE SOLVED.
C
10    CONTINUE
      DO 15 J=1,NU
      PRED(J)=Y(J)+H*(55.*DY(J,4)-59.*DY(J,3)+37.*DY(J,2)
     1      -9.*DY(J,1))/24.
15    CONTINUE
      X=X+H
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      NP=NP+1
      ZR=PRED(1)
      ZI=PRED(2)
      ZETA(NP) = CMPLX (ZR,ZI)
      GO TO (110,120,130), IP
120   AHR = PRED(3)
      AHI = PRED(4)
      AH = CMPLX (AHR,AHI)
      GO TO 110
130   CONTINUE
      CGAM = CMPLX (PRED(3),PRED(4))
110   CONTINUE
      IF (NP .LE. NPLAST)  GO TO 20
      DO 25  I = 1,25
25    CC(I) = CCEXT(I)
      GO TO 30
20    CONTINUE
      UP=U(NP)
      DUP=DU(NP)
      CP=C(NP)
      R=RW(NP)
      CALL  COEFFS (UP,DUP,CP,R,CC)
30    CONTINUE
      CFH = CC(1)
      CFM = CC(2) + CC(6)
      CFN = CC(3) + CC(4) + CC(5) + CC(7) + CC(8)
      ZETA1 = ( - CFM * ZETA(NP) - CFN) / CFH - ZETA(NP) **2
      DP(1) = REAL (ZETA1)
      DP(2) = AIMAG (ZETA1)
      GO TO (140,150,160), IP
150   AH1 = AH * ZETA(NP)
      DP(3) = REAL (AH1)
      DP(4) = AIMAG (AH1)
      GO TO 140
160   CONTINUE
C
C     AP,AP1 AND AP2 ARE THE VALUES OF THE AMPLITUDE FUNCTION AND
C     THEIR DERIVATIVES AT THE CURRENT STATION.
      AP = AFN(NP)
      AP1 = AFN1(NP)
      AP2 = AFN2(NP)
C
      INHMG = - CC(18) * AP * AP2 - CC(12) * AP1 * AP2 - (CC(9)
     1        + CC(15)) * AP1 * AP1 - (CC(13) + CC(14) + CC(19)
     2        + CC(23) + CC(24) + CC(25)) * AP1 * AP - (CC(10) + CC(11)
     3        + CC(17) + CC(20) + CC(21) + CC(22)) * AP * AP
      CGAM1 = (- ZETA(NP) + .5* (GAM-1.) * DUP/CP - CFM/CFH) * CGAM
     1         - INHMG / (CP * CFH)
      DP(3) = REAL (CGAM1)
      DP(4) = AIMAG (CGAM1)
140   CONTINUE
      DO 45 J=1,NU
      COR(J)= Y(J) + H*(DY(J,2)-5.*DY(J,3)+19.*DY(J,4)
     1              +9.*DP(J))/24.0
45    Y(J)= (251.*COR(J)+19.*PRED(J))/270.
```

```
      DO 55 I=1,NU
      DO 55 J=1,3
55    DY(I,J) = DY(I,J+1)
      ZR=Y(1)
      ZI=Y(2)
      ZETA(NP) = CMPLX (ZR,ZI)
      ZETA1 = ( - CFM * ZETA(NP) - CFN) / CFH - ZETA(NP) **2
      DY (1,4) = REAL (ZETA1)
      DY (2,4) = AIMAG (ZETA1)
      GO TO (170,180,190), IP
180   AH = CMPLX (Y(3),Y(4))
      AH1 = AH * ZETA(NP)
      DY(3,4) = REAL (AH1)
      DY(4,4) = AIMAG (AH1)
      IF (MODE.NE.1) GO TO 182
      AH2 = AH1 * ZETA(NP) + AH * ZETA1
      AFN(NP) = AH
      AFN1(NP) = AH1
      AFN2(NP) = AH2
182   GO TO 170
190   CONTINUE
      CGAM = CMPLX (Y(3),Y(4))
      CGAM1 = (- ZETA(NP) + .5* (GAM-1.) * DUP/CP - CFM/CFH) * CGAM
     1          - INHMG / (CP * CFH)
      DY(3,4) = REAL (CGAM1)
      DY (4,4) = AIMAG (CGAM1)
170   CONTINUE
      IF (NP .EQ. NEND)  GO TO 100
C
C     DECIDE  WHICH  EQUATION  IS  TO  BE  INTEGRATED :  TAU  OR  ZETA
C
      IF (CABS (ZETA(NP)) .LT. 10)  GO TO 10
      ITORZ = 1
C
C     CALCULATE  VALUE OF TAU AND ITS DERIVATIVE AT  LAST FOUR STATIONS.
      DO 410 I = 1,4
410   TAU (NP-4+I) = 1./ZETA(NP-4+I)
      Y(1) = REAL (TAU(NP))
      Y(2) = AIMAG (TAU(NP))
      DO 420  I = 1,4
      TSQR =  REAL (TAU(NP-4+I) * TAU(NP-4+I))
      TSQI = AIMAG (TAU(NP-4+I) * TAU(NP-4+I))
      ZPR = DY(1,I)
      ZPI = DY(2,I)
      DY(1,I) = - TSQR*ZPR + TSQI*ZPI
      DY(2,I) = - TSQR*ZPI - TSQI*ZPR
420   CONTINUE
C
      CALL  TADAMS (H,NP,X,Y,DY,IQ,ITORZ)
      GO TO (10,100), IQ
100   RETURN
      END
      SUBROUTINE  TADAMS (H,NP,X,Y,DY,IQ,ITORZ)
C
C     THIS SUBROUTINE CARRIES OUT A MODIFIED ADAMS PREDICTOR-CORRECTOR
C     INTEGRATION SCHEME TO SOLVE THE VARIOUS DIFFERENTIAL EQUATIONS AS
C     DESCRIBED BELOW
C     IF IP = 1, INTEGRATION IS CARRIED OUT FOR TAU ONLY;
C     IF IP = 2, INTEGRATION IS CARRIED OUT FOR TAU  AND AH;
C     IF IP = 3, INTEGRATION IS CARRIED OUT FOR TAU  AND GAMMA.
C     IP IS PASSED TO THE SUBROUTINE THROUGH COMMON BLOCK X3.
C
C     H IS THE STEP-SIZE; INPUT.
C     X IS THE VALUE OF STEADY-STATE POTENTIAL AT THE STATION ,
C     WHERE THE PREDICTOR-CORRECTOR INTEGRATION STARTS; INPUT.
C     DURING THE PROGRAM, X IS CHANGED TO THE VALUE AT CURRENT STATION.
C     Y ARE THE VALUES AT X , OF THE FUNCTIONS, WHOSE EQUATIONS ARE
C     BEING SOLVED; INPUT AND OUTPUT.
C     DY ARE THE DERIVATIVES OF Y; INPUT AND OUTPUT.
C     IQ INDICATES WHETHER INTEGRATION IS COMPLETE; OUTPUT.
C     IQ = 1 :   INTEGRATION IS TO BE CONTINUED BY SUBROUTINE ZADAMS.
C     IQ = 2 :   INTEGRATION IS COMPLETE.
C     ITORZ INDICATES WHICH EQUATION SHOULD BE INTEGRATED :
C     ITORZ = 1 :   INTEGRATION OF EQUATION FOR ZETA.
C     ITORZ = 2 :   INTEGRATION OF EQUATION FOR TAU.
C
C
      COMMON /X1/ CM,ANGLE,RCC,RCT,GAM,G,RT
      COMMON /X2/ T,R1,R2,NFLAST,NEND,IEXTN
      COMMON /X3/ WC,SVN,IP,MODE,NU,KP(3)
      COMMON /X5/ U(1000),DU(1000),C(1000),RW(1000)
      COMMON /X6/ AFN,AFN1,AFN2
      COMMON /X8/ ZETA,TAU,CCEXT
      COMPLEX  AFN(1000),AFN1(1000),AFN2(1000)
      COMPLEX  CC(25),CFH,CFM,CFN,INHMG,AH,AH1,AP,AP1,AP2,CGAM,CGAM1
      COMPLEX  ZETA(1000),TAU(1000),TAU1,CCEXT(25)
      DIMENSION  Y(4),DY(4,4),DP(4),PRED(4),COR(4)
C
 10   CONTINUE
C     NU IS THE NUMBER OF EQUATIONS TO BE SOLVED.
      DO 15  J = 1,NU
      PRED(J)=Y(J)+H*(55.*DY(J,4)-59.*DY(J,3)+37.*DY(J,2)
     1          -9.*DY(J,1))/24.
 15   CONTINUE
      X = X+H
      NP = NP + 1
      TR = PRED (1)
      TI = PRED (2)
      TAU (NP) = CMPLX (TR,TI)
      ZETA (NP) = 1./ TAU(NP)
      GO TO (110,120,130), IP
 120  AHR = PRED(3)
      AHI = PRED (4)
      AH = CMPLX (AHR,AHI)
      GO TO 110
 130  CONTINUE
      CGAM = CMPLX (PRED(3),PRED(4))
```

```
110   CONTINUE
      IF (NP .LE. NPLAST)  GO TO 20
C
C     OBTAIN COEFFICIENTS IN THE EXTENSION SECTION.
      DO 25  I = 1,25
25    CC(I) = CCEXT(I)
C
      GO TO 30
20    CONTINUE
      DUP = DU(NP)
      UP = U(NP)
      CP = C(NP)
      R = RW (NP)
      CALL  COEFFS (UP,DUP,CP,R,CC)
30    CONTINUE
      CFH = CC(1)
      CFM = CC(2) + CC(6)
      CFN = CC(3) + CC(4) + CC(5) + CC(7) + CC(8)
      TAU1 = 1. + (CFM + CFN * TAU(NP)) * TAU(NP) / CFH
      DP(1) = REAL (TAU1)
      DP(2) = AIMAG (TAU1)
      GO TO (140,150,160), IP
150   AH1 = AH / TAU(NP)
      DP(3) = REAL (AH1)
      DP(4) = AIMAG (AH1)
      GO TO 140
160   CONTINUE
C
C     AP,AP1 AND AP2 ARE THE VALUES OF THE AMPLITUDE FUNCTION AND
C     THEIR DERIVATIVES AT THE CURRENT STATION.
      AP = APN(NP)
      AP1 = APN1(NP)
      AP2 = APN2(NP)
C
      INHMG = - CC(18) * AP * AP2 - CC(12) * AP1 * AP2 - (CC(9)
     1        + CC(15)) * AP1 * AP1 - (CC(13) + CC(14) + CC(19)
     2        + CC(23) + CC(24) + CC(25)) * AP1 * AP - (CC(10) + CC(11)
     3        + CC(17) + CC(20) + CC(21) + CC(22)) * AP * AP
      CGAM1 = ( - ZETA(NP) + .5 * (GAM - 1.) * DUP/CP - CFM/CFH) * CGAM
     1          - INHMG / (CP * CFH)
      DP(3) = REAL (CGAM1)
      DP(4) = AIMAG (CGAM1)
140   CONTINUE
      DO 45 J=1,NU
      COR(J)= Y(J) + H*(DY(J,2)-5.*DY(J,3)+19.*DY(J,4)
     1            +9.*DP(J))/24.0
45    Y(J)= (251.*COR(J)+19.*PRED(J))/270.
      DO 55 I=1,NU
      DO 55 J=1,3
55    DY(I,J) = DY(I,J+1)
      TR = Y(1)
      TI = Y(2)
      T2 = TR*TR + TI*TI
      TAU (NP) = CMPLX (TR,TI)
      ZETA (NP) = 1./ TAU(NP)
      TAU1 = 1. + (CFM + CFN * TAU(NP)) * TAU(NP) / CFH
      DY (1,4) = REAL (TAU1)
      DY (2,4) = AIMAG (TAU1)
      GO TO (170,180,190), IP
180   AHR = Y(3)
      AHI = Y(4)
      AH = CMPLX (AHR,AHI)
      AH1 = AH / TAU(NP)
      DY (3,4) = REAL (AH1)
      DY (4,4) = AIMAG (AH1)
      IF (MODE .NE. 1)  GO TO 182
      AFN(NP) = AH
      AFN1(NP) = AH1
      AFN2 (NP) = ( TAU(NP) * AFN1(NP) - TAU1 * AFN(NP) ) /
     1             ( TAU(NP)*TAU(NP) )
182   GO TO 170
190   CONTINUE
      CGAM = CMPLX (Y(3),Y(4))
      CGAM1 = ( - ZETA(NP) + .5 * (GAM - 1.) * DUP/CP - CFM/CFH) * CGAM
     1           - INHMG / (CP * CFH)
      DY (3,4) = REAL (CGAM1)
      DY (4,4) = AIMAG (CGAM1)
170   CONTINUE
      IF (NP .EQ. NEND)  GO TO 100
C
C     DECIDE  WHICH  EQUATION  IS  TO  BE  INTEGRATED :  TAU  OR  ZETA
C
      IF  (CABS(TAU(NP))  .LT.  10)  GO TO 10
      ITORZ = 2
      Y(1) = REAL ( ZETA(NP) )
      Y(2) = AIMAG ( ZETA(NP) )
C
C     CALCULATE DERIVATIVES OF ZETA AT THE LAST FOUR POINTS.
      DO 420 I = 1,4
      ZSQR =  REAL ( ZETA(NP-4+I) * ZETA(NP-4+I) )
      ZSQI = AIMAG ( ZETA(NP-4+I) * ZETA(NP-4+I) )
      TPR = DY(1,I)
      TPI = DY(2,I)
      DY(1,I) = - ZSQR*TPR + ZSQI*TPI
      DY(2,I) = - ZSQR*TPI - ZSQI*TPR
420   CONTINUE
C
      IQ = 1
      RETURN
100   IQ = 2
105   RETURN
      END
```

APPENDIX B

PROGRAM COEFFS3D: A USER'S MANUA

Program COEFFS3D calculates the coefficients of both the linear and nonlinear terms that appear in Eq. (20). These coefficients are required as input for Program LCYC3D (see Appendix C) which numerically integrates this system of equations. Program COEFFS3D is a slightly modified version of the program described in detail in Appendix C of Ref. 11. The modification lies in the evaluation of one more coefficient, $C_4(j, p)$ defined by

$$C_4(j,p) = \bar{u}_e \bar{c}_e^2 \Gamma_p Z_j^*(z_e) \int_0^{2\pi} \Theta_p \Theta_j d\theta \int_0^1 R_p R_j r dr.$$

This coefficient represents the effect of nozzle nonlinearities. Except for this additional coefficient, the two programs are very similar in the structure of their numerical calculations and their output. Hence in this user's manual, only the listing of the entire program together with a precise description of the necessary input is given. For details of the program, one is referred to Appendix C of Ref. 11.

In the following description of the input, the location number refers to columns of the card. Three formats are used for input: "A" indicates alphanumeric characters, "I" indicates integers and "F" indicates real numbers with a decimal point. For the "I" and "F" formats the values are placed in fields of five and ten locations respectively (right justified).

No. of Cards	Location	Type	Input Item	Comments
1	1-72	A	Title	Title of the case
1	1-10	F	GAMMA	Ratio of specific heats
	11-20	F	UE	Steady-state Mach number at nozzle entrance
	21-30	F	RLD	Length-to-diameter ratio
	31-40	F	ZCOMB	Length of the combustion zone

Location	Type	Input Item	Comments
41-45	I	NDROPS	If 0: droplet momentum source neglected If 1: droplet momentum source included
46-50	I	NOZZLE	If 0: quasi-steady nozzle If 1: conventional nozzle
1-5	I	NJMAX	Number of series terms (complex)
6-10	I	NONLIN	If 0: linear terms only If 1: both linear and nonlinear terms
11-15	I	NEGL	If 0: all non-zero coefficients calculated If 1: small coefficients neglected
16-20	I	NOUT	If 0: printed output only If 1: printed and written into file If 2: written into file only If 3: card output only
21-25	I	NOZNL1	If 0: nozzle nonlinearities neglected If 1: nozzle nonlinearities included
26-30	I	NZDATA	If 0: nozzle admittance values input through cards If 1: nozzle admittance values input through file If NZDATA is 1, NOUT in program NOZADM should be 1

The next card is necessary only if NEGL = 1.

No. of Cards	Location	Type	Input Item	Comments
1	1-10	F	SM1	Linear coefficients with absolute value less than SM1 neglected
	11-20	F	SM2	Nonlinear coefficients with absolute value less than SM2 neglected

The next NJMAX cards are necessary only if NOZZLE = 1 and NZDATA = 0.

No. of Cards	Location	Type	Input Item	Comments
NJMAX	1-5	I	J	Integer which identifies the series term
	6-15	F	AMPL(J)	Amplitude of the linear nozzle admittance
	16-25	F	PHASE(J)	Phase of the linear nozzle admittance

The next NJMAX cards are necessary only if NZDATA = 0 and NOZNL1 = 1.

No. of Cards	Location	Type	Input Item	Comments
NJMAX	1-5	I	J	Integer which identifies the series term
	6-15	F	GNOZ(J)	Real part of the nonlinear nozzle admittance
	16-25	F	GNOZ(J)	Imaginary part of the nonlinear nozzle admittance
NJMAX	1-5	I	J	Integer which identifies series term
	6-10	I	L(J)	Axial mode number, ℓ
	11-15	I	M(J)	Tangential mode number, m
	16-20	I	N(J)	Radial mode number, n
	21-25	I	NS(J)	NS(J) = 1: $\Theta_j = \sin(m\theta)$ NS(J) = 2: $\Theta_j = \cos(m\theta)$
	26-30	A	NAME(J)	Four character name

FORTRAN Listing

```
C
C
C     ********************** PROGRAM COEFFS3D ************************
C
C          THIS PROGRAM COMPUTES THE COEFFICIENTS WHICH APPEAR
C     IN THE DIFFERENTIAL EQUATIONS WHICH GOVERN THE MODE-AMPLITUDE
C     FUNCTIONS.   THESE COEFFICIENTS ARE PUNCHED ONTO CARDS FOR
C     INPUT INTO PROGRAM LIMCYC.
C
C     THE FOLLOWING INPUTS ARE REQUIRED:
C     THE TITLE OF THE CASE.
C     GAMMA IS THE SPECIFIC HEAT RATIO.
C     UE IS THE STEADY STATE MACH NUMBER AT THE NOZZLE ENTRANCE.
C     RLD IS THE LENGTH-TO-DIAMETER RATIO.
C     ZCOMB IS THE LENGTH OF THE REGION OF UNIFORMLY DISTRIBUTED
C     COMBUSTION,EXPRESSED AS A FRACTION OF THE CHAMBER LENGTH.
C     NDROPS DETERMINES THE PRESENCE OF DROPLET MOMENTUM SOURCES:
C        NDROPS = 0  DROPLET MOMENTUM SOURCE NEGLECTED.
C        NDROPS = 1  DROPLET MOMENTUM SOURCE INCLUDED.
C     NOZZLE SPECIFIES THE TYPE OF NOZZLE USED:
C        NOZZLE = 0      QUASI-STEADY
C        NOZZLE = 1      CONVENTIONAL NOZZLE.
C     FOR CONVENTIONAL NOZZLE
C     AMPL IS THE NOZZLE AMPLITUDE RATIO.
C     PHASE IS THE NOZZLE PHASE SHIFT.
C     NOZNL1 DETERMINES THE PRESENCE OF NOZZLE NONLINEARITIES
C        NOZNL1 = 0      NOZZLE NONLINEARITIES NEGLECTED.
C        NOZNL1 = 1      NOZZLE NONLINEARITIES INCLUDED.
C     NZDATA DETERMINES HOW THE NOZZLE DATA IS SUPPLIED
C        NZDATA = 0      FROM CARDS.
C        NZDATA = 1      FROM A FASTRAND FILE.
C     NJMAX IS THE NUMBER OF MODE-AMPLITUDE FUNCTIONS IN THE ASSUMED
C     SERIES SOLUTION.  NJMAX MUST NOT EXCEED MX.
C     THE COEFFICIENTS COMPUTED ARE DETERMINED BY NONLIN AS FOLLOWS
C        NONLIN = 0   LINEAR COEFFICIENTS ONLY
C        NONLIN = 1   BOTH LINEAR AND NONLINEAR COEFFICIENTS
C     COEFFICIENTS TO BE NEGLECTED ARE DETERMINED BY NEGL
C     AS FOLLOWS:
C        NEGL = 0  TERMS SMALLER THAN 0.00001 ARE NEGLECTED.
C        NEGL = 1  LINEAR TERMS SMALLER THAN SM1 AND NONLINEAR
C                  TERMS SMALLER THAN SM2 ARE NEGLECTED.
C     THE OUTPUT IS DETERMINED BY NOUT AS FOLLOWS
C        NOUT = 0  PRINTED OUTPUT ONLY
C        NOUT = 1  PRINTED AND WRITTEN ON FASTRAND FILE.
C        NOUT = 2  FASTRAND FILE ONLY.
C        NOUT = 3  CARD OUTPUT ONLY.
C     EACH MODE-AMPLITUDE IS ASSIGNED AN INTEGER J.
C     THE MODE IS SPECIFIED BY THE INDICES L(J), M(J), AND N(J).
C     L(J) IS THE AXIAL MODE NUMBER AND MUST NOT EXCEED 5.
C     M(J) IS THE AZIMUTHAL MODE NUMBER AND MUST NOT EXCEED 8.
C     N(J) IS THE RADIAL MODE NUMBER AND MUST NOT EXCEED 5.
C     THE INTEGER NS(J) IS ASSIGNED AS FOLLOWS:
C        NS = 1   A-FUNCTION    SIN(M*THETA) * COSH(I*B*Z)
C        NS = 2   B-FUNCTION    COS(M*THETA) * COSH(I*B*Z)
C     NAME(J) IS A FOUR-CHARACTER NAME.
```

```
C
C     ******************************************************************
C
      PARAMETER    MX=5, MX2=10, MX4=20
      DIMENSION    L(MX), N(MX), NAME(MX), S(MX), SJ(MX), TITLE(80),
     1             RJROOT(10,5), RJVAL(10,5), C1(MX2,MX2), C(4,MX2,MX2),
     2             D(MX2,MX2,MX2), AMPL(MX), PHASE(MX), AZI(2),
     3             BES1(9,9,9), BES2(9,9,9), BES3(9,9,9),
     4             V(2), JC(MX2), TS(4,MX2), TSQ(MX2), KMAX(5)
      COMPLEX      CRSLT, CI, ZEJ, ZEP1, ZEP2, CZE, CAZ, CRAD,
     1             G1, DCOEF, CGAM, CAX, B(MX), BC(MX), YNOZ(MX),
     2             CNORM(MX), CSSQ(MX), TANINT(2), RADINT(3),
     3             AXINT(4,3), CC(5,MX,MX), CD1(MX,MX,MX),
     4             CD2(MX,MX,MX), AX(4), T1, T2, D1, D2, D3, D4,
     5             CD3(MX,MX,MX), CD4(MX,MX,MX), GNOZ(MX)
      COMMON       B    /BLK2/  M(MX), NS(MX)
C
C     DATA INPUT.
C
      PI = 3.1415927
      SM1 = 0.00001
      SM2 = 0.00001
      SM3 = 0.00001
      CI = (0.0,1.0)
C
C     INPUT ROOTS AND VALUES OF BESSEL FUNCTIONS.
      DATA ((RJROOT(I,J), J = 1,5), I = 1,9)/
     1   3.83171,  7.01559, 10.17347, 13.32369, 16.47063,
     2   1.84118,  5.33144,  8.53632, 11.70600, 14.86359,
     3   3.05424,  6.70613,  9.96947, 13.17037, 16.34752,
     4   4.20119,  8.01524, 11.34592, 14.58585, 17.78875,
     5   5.31755,  9.28240, 12.68191, 15.96411, 19.19603,
     6   6.41562, 10.51986, 13.98719, 17.31284, 20.57551,
     7   7.50127, 11.73494, 15.26818, 18.63744, 21.93172,
     8   8.57784, 12.93239, 16.52937, 19.94185, 23.26805,
     9   9.64742, 14.11552, 17.77401, 21.22906, 24.58720/
      DATA ((RJVAL(I,J), J = 1,5), I = 1,9)/
     1  -0.40276,  0.30012, -0.24970,  0.21836, -0.19647,
     2   0.58187, -0.34613,  0.27330, -0.23330,  0.20701,
     3   0.48650, -0.31353,  0.25474, -0.22088,  0.19794,
     4   0.43439, -0.29116,  0.24074, -0.21097,  0.19042,
     5   0.39965, -0.27438,  0.22959, -0.20276,  0.18403,
     6   0.37409, -0.26109,  0.22039, -0.19580,  0.17849,
     7   0.35414, -0.25017,  0.21261, -0.18978,  0.17363,
     8   0.33793, -0.24096,  0.20588, -0.18449,  0.16929,
     9   0.32438, -0.23303,  0.19998, -0.17979,  0.16539/
C
C     INPUT PARAMETERS.
    4 READ (5,5000, END = 600)  (TITLE(I), I = 1, 72)
      READ (5,5001)  GAMMA, UE, RLD, ZCOMB, NDROPS, NOZZLE
      IF (GAMMA) 600, 600, 8
    8 READ (5,5004) NJMAX, NONLIN, NEGL, NOUT, NOZNL1, NZDATA
      IF (NEGL .EQ. 1) READ (5,5005) SM1, SM2
      IF (NOZZLE .EQ. 1)  GO TO 5
C     COMPUTE ADMITTANCE FOR QUASI-STEADY NOZZLE.
```

```
      Y = (GAMMA - 1.0) * UE/(2.0 * GAMMA)
      DO 3 J = 1, NJMAX
      AMPL(J) = Y
      PHASE(J) = 0.0
    3 CONTINUE
      GO TO 7
    5 CONTINUE
      IF (NZDATA .EQ. 0) NZDATA = 5
      IF (NZDATA .EQ. 1) NZDATA = 7
      DO 6 I = 1, NJMAX
      READ (NZDATA,5003) J, AMPL(J), PHASE(J)
    6 CONTINUE
      IF (NOZNL1 .NE. 1) GO TO 7
      DO 710  I = 1,NJMAX
      READ (NZDATA,5003)  J, GNOZ(J)
  710 CONTINUE
    7 DO 10 I = 1, NJMAX
      READ (5,5002) J, L(J), M(J), N(J), NS(J), NAME(J)
   10 CONTINUE
C
      DO 12 J = 1, NJMAX
      THETA = PHASE(J) * PI/180.0
      YR = AMPL(J) * COS(THETA)
      YI = AMPL(J) * SIN(THETA)
      YNOZ(J) = CMPLX(YR,YI)
   12 CONTINUE
C
      ZE = 2.0 * RLD
      CZE = CMPLX(ZE,0.0)
      CGAM = CMPLX(GAMMA,0.0)
      CAX = CGAM
      IF (NDROPS .EQ. 1) CAX = CGAM + (1.0,0.0)
C
C     ******************************************************************
C
C     ASSIGN ARRAYS FOR ROOTS OF BESSEL FUNCTIONS.
      DO 20 J = 1, NJMAX
      IF ((M(J) .EQ. 0) .AND. (N(J) .EQ. 0))  GO TO 15
      MM = M(J) + 1
      NN = N(J)
      S(J) = RJROOT(MM,NN)
      SJ(J) = RJVAL(MM,NN)
      GO TO 25
   15 S(J) = 0.0
      SJ(J) = 1.0
   25 SSQ = S(J) * S(J)
      CSSQ(J)  = CMPLX(SSQ,0.0)
   20 CONTINUE
C
C     ******************************************************************
C
C     CALCULATE AXIAL ACOUSTIC EIGENVALUES.
C
C     FIND MAXIMUM VALUES OF L(J), M(J), AND N(J).
      KN = 0
      LMAX = 0
      MMAX = 0
      NMAX = 0
      DO 30 J = 1, NJMAX
      IF (L(J) .GT. LMAX)  LMAX = L(J)
      IF (M(J) .GT. MMAX)  MMAX = M(J)
      IF (N(J) .GT. NMAX)  NMAX = N(J)
      IF (N(J) .NE. N(1))  KN = 1
   30 CONTINUE
      LMAX = LMAX + 1
      MMAX = MMAX + 1
C
C     COMPUTE EIGENVALUES.
      DO 40 J = 1, NJMAX
      LL = L(J)
      SMN = S(J)
      YAMPL = AMPL(J)
      YPHASE = PHASE(J)
      CALL EIGVAL(LL,SMN,GAMMA,ZE,YAMPL,YPHASE,CRSLT)
      B(J) = CRSLT
      BC(J) = CONJG(CRSLT)
   40 CONTINUE
C
C     ******************************************************************
C
C     CALCULATE LINEAR COEFFICIENTS.
C
C     CALCULATE THE NUMBER OF LINEAR COEFFICIENTS.
C
      NCOEFF = 4
      IF (NOZNL1 .EQ. 1) NCOEFF = 5
      NCFM1 = NCOEFF-1
C
      DO 100 NJ = 1, NJMAX
      DO 100 NP = 1, NJMAX
C
C     ZERO COEFFICIENT ARRAYS.
      DO 105 KC = 1, NCOEFF
      CC(KC,NJ,NP) = (0.0,0.0)
  105 CONTINUE
C
C     ORTHOGONALITY PROPERTY OF TANGENTIAL EIGENFUNCTIONS.
      IF ( NS(NP) .NE. NS(NJ) )  GO TO 100
      IF (M(NP) .NE. M(NJ))  GO TO 100
      IF (M(NJ) .EQ. 0)  GO TO 112
      AZ = PI
      GO TO 120
  112 IF ( NS(NJ) .EQ. 1)  GO TO 100
      AZ = 2.0 * PI
C
C     ORTHOGONALITY PROPERTY OF RADIAL EIGENFUNCTIONS.
  120 IF (N(NP) .NE. N(NJ))  GO TO 100
      IF (S(NP))  125, 122, 125
  125 SQM = M(NJ) * M(NJ)
      SSQ = S(NP) * S(NP)
      SJSQ = SJ(NJ) * SJ(NJ)
      RAD = (SSQ - SQM) * SJSQ/(2.0 * SSQ)
      GO TO 127
  122 RAD = 0.5
C
C     CALCULATE AXIAL INTEGRALS.
  127 DO 130 NOPT = 1, 4
      CALL AXIAL1 (NOPT, NP, NJ, UE, ZE, ZCOMB, CRSLT)
      AX(NOPT) = CRSLT
  130 CONTINUE
C
C     EVALUATE FUNCTIONS AT NOZZLE END.
      ZEJ = CCOSH(CI*BC(NJ)*CZE)
      ZEP1 = CCOSH(CI*B(NP)*CZE)
      ZEP2 = CI * B(NP) * CSINH(CI*B(NP)*CZE)
C
      CAZ = CMPLX(AZ,0.0)
      CRAD = CMPLX(RAD,0.0)
C
C     COEFFICIENT OF THE SECOND DERIVATIVE OF A(P).
      CC(1,NJ,NP) = AX(1) * CAZ * CRAD
C
C     COEFFICIENT OF A(P).
      CC(2,NJ,NP) = (CSSQ(NP)*AX(1) - AX(2) + ZEP2*ZEJ) * CAZ * CRAD
C
C     COEFFICIENT OF THE FIRST DERIVATIVE OF A(P).
      CC(3,NJ,NP) = (CAX*AX(3) + (2.0,0.0)*AX(4)
     1              + CGAM*YNOZ(NP)*ZEP1*ZEJ) * CAZ * CRAD
C
C     COEFFICIENT OF THE RETARDED DERIVATIVE OF A(P).
      CC(4,NJ,NP) = CGAM * AX(3) * CAZ * CRAD
C
      IF (NOZNL1 .NE. 1) GO TO 100
C
C     COEFFICIENT DUE TO NOZZLE NONLINEARITIES.
      CESQ = 1- (GAMMA-1) * UE/2.
      CC(5,NJ,NP) = UE * CESQ * GNOZ(NP) * ZEJ * CAZ * CRAD
C
  100 CONTINUE
C
C     NORMALIZE LINEAR COEFFICIENTS.
      DO 140 NJ = 1, NJMAX
      CNORM(NJ) = CC(1,NJ,NJ)
      DO 140 NP = 1, NJMAX
      DO 140 KC = 1, NCOEFF
      CC(KC,NJ,NP) = CC(KC,NJ,NP)/CNORM(NJ)
  140 CONTINUE
C
C     ******************************************************************
C
C     COMPUTE NONLINEAR COEFFICIENTS.
C
      IF (NONLIN .EQ. 0)  GO TO 402
      G1 = (CGAM - (1.0,0.0)) * (0.5,0.0)
```

```
C     COMPUTATIONS OF BESSEL INTEGRALS WHEN ALL SERIES TERMS HAVE THE
C     SAME RADIAL MODE NUMBER N(J).
      IF (KN .EQ. 1)  GO TO 170
      DO 150 MP = 1, MMAX
      DO 150 MQ = 1, MMAX
      DO 150 MJ = 1, MMAX
      BES1(MP,MQ,MJ) = 0.0
      BES2(MP,MQ,MJ) = 0.0
      BES3(MP,MQ,MJ) = 0.0
      L1 = MP - 1
      L2 = MQ - 1
      L3 = MJ - 1
      LM = L1 + L2
      LN = L1 + L3
      MN = L2 + L3
      IF ((L3.EQ.LM) .OR. (L2.EQ.LN) .OR. (L1.EQ.MN))  GO TO 160
      GO TO 150
  160 IF (NMAX .EQ. 0)  GO TO 165
      A1 = RJROOT(MP,NMAX)
      A2 = RJROOT(MQ,NMAX)
      A3 = RJROOT(MJ,NMAX)
      GO TO 167
  165 A1 = 0.0
      A2 = 0.0
      A3 = 0.0
  167 CALL RADIAL(1,L1,L2,L3,A1,A2,A3,RESULT)
      BES1(MP,MQ,MJ) = RESULT
      CALL RADIAL(2,L1,L2,L3,A1,A2,A3,RESULT)
      BES2(MP,MQ,MJ) = RESULT
      CALL RADIAL(3,L1,L2,L3,A1,A2,A3,RESULT)
      BES3(MP,MQ,MJ) = RESULT
  150 CONTINUE
C
  170 DO 200 NJ = 1, NJMAX
      DO 200 NP = 1, NJMAX
      DO 200 NQ = 1, NJMAX
C
      CD1(NJ,NP,NQ) = (0.0,0.0)
      CD2(NJ,NP,NQ) = (0.0,0.0)
      CD3(NJ,NP,NQ) = (0.0,0.0)
      CD4(NJ,NP,NQ) = (0.0,0.0)
C
      DO 210 J = 1, 2
      CALL AZIMTL(J,NP,NQ,NJ,RESULT)
      AZI(J) = RESULT
      TANINT(J) = CMPLX(RESULT,0.0)
  210 CONTINUE
C
      IF (AZI(1)) 220, 225, 220
  225 IF (AZI(2)) 220, 200, 220
C
  220 IF (KN .EQ. 0)  GO TO 222
      L1 = M(NP)
      L2 = M(NQ)
      L3 = M(NJ)
      A1 = S(NP)
      A2 = S(NQ)
      A3 = S(NJ)
      GO TO 244
C
  222 MP = M(NP) + 1
      MQ = M(NQ) + 1
      MJ = M(NJ) + 1
      RADINT(1) = CMPLX(BES1(MP,MQ,MJ),0.0)
      RADINT(2) = CMPLX(BES2(MP,MQ,MJ),0.0)
      RADINT(3) = CMPLX(BES3(MP,MQ,MJ),0.0)
C
  244 DO 240 J = 1, 3
      IF (KN .EQ. 0)  GO TO 242
      CALL RADIAL (J,L1,L2,L3,A1,A2,A3,RESULT)
      RADINT(J) = CMPLX(RESULT,0.0)
  242 DO 240 NC = 1,4
      CALL AXIAL2 (J,NC,NP,NQ,NJ,ZE,CRSLT)
      AXINT(NC,J) = CRSLT
  240 CONTINUE
C
C
      DO 250 J = 1,4
      T1 = G1 * CSSQ(NP) * AXINT(J,1)
      T2 = G1 * AXINT(J,3)
      D1 = AXINT(J,1) * TANINT(1) * RADINT(3)
      D2 = AXINT(J,1) * TANINT(2) * RADINT(2)
      D3 = AXINT(J,2) * TANINT(1) * RADINT(1)
      D4 = (T2 - T1) * TANINT(1) * RADINT(1)
      DCOEF = (0.5,0.0) * (D1 + D2 + D3 + D4)/CNORM(NJ)
      IF (J .EQ. 1)  CD1(NJ,NP,NQ) = (1.0,-1.0) * DCOEF
      IF (J .EQ. 2)  CD2(NJ,NP,NQ) = (1.0,1.0) * DCOEF
      IF (J .EQ. 3)  CD3(NJ,NP,NQ) = (1.0,1.0) * DCOEF
      IF (J .EQ. 4)  CD4(NJ,NP,NQ) = (1.0,-1.0) * DCOEF
  250 CONTINUE
  200 CONTINUE
C
C     ****************************************************************
C
C     CALCULATE COEFFICIENTS FOR EQUIVALENT REAL SYSTEM.
C
  402 DO 350 NJ = 1, NJMAX
      NEWJ = (2 * NJ) - 1
      NEWJ1 = NEWJ + 1
      DO 350 NP = 1, NJMAX
      NEWP = (2 * NP) - 1
      NEWP1 = NEWP + 1
C
C     COEFFICIENTS OF LINEAR TERMS.
      CCR = REAL(CC(1,NJ,NP))
      CCI = AIMAG(CC(1,NJ,NP))
      C1(NEWJ,NEWP) = CCR
      C1(NEWJ,NEWP1) = -CCI
      C1(NEWJ1,NEWP) = CCI
      C1(NEWJ1,NEWP1) = CCR
      CCR = REAL(CC(KC+1,NJ,NP))
      CCI = AIMAG(CC(KC+1,NJ,NP))
      C(KC,NEWJ,NEWP) = CCR
      C(KC,NEWJ,NEWP1) = -CCI
      C(KC,NEWJ1,NEWP) = CCI
      C(KC,NEWJ1,NEWP1) = CCR
  360 CONTINUE
C
C     COEFFICIENTS OF NONLINEAR TERMS.
      IF (NONLIN .EQ. 0)  GO TO 350
      DO 370 NQ = 1, NJMAX
      NEWQ = (2 * NQ) - 1
      NEWQ1 = NEWQ + 1
      CD1R = REAL(CD1(NJ,NP,NQ))
      CD1I = AIMAG(CD1(NJ,NP,NQ))
      CD2R = REAL(CD2(NJ,NP,NQ))
      CD2I = AIMAG(CD2(NJ,NP,NQ))
      CD3R = REAL(CD3(NJ,NP,NQ))
      CD3I = AIMAG(CD3(NJ,NP,NQ))
      CD4R = REAL(CD4(NJ,NP,NQ))
      CD4I = AIMAG(CD4(NJ,NP,NQ))
      D(NEWJ,NEWP,NEWQ) = CD1R + CD2R + CD3R + CD4R
      D(NEWJ,NEWP,NEWQ1) = -CD1I + CD2I - CD3I + CD4I
      D(NEWJ,NEWP1,NEWQ) = -CD1I - CD2I + CD3I + CD4I
      D(NEWJ,NEWP1,NEWQ1) = -CD1R + CD2R + CD3R - CD4R
      D(NEWJ1,NEWP,NEWQ) = CD1I + CD2I + CD3I + CD4I
      D(NEWJ1,NEWP,NEWQ1) = CD1R - CD2R + CD3R - CD4R
      D(NEWJ1,NEWP1,NEWQ) = CD1R + CD2R - CD3R - CD4R
      D(NEWJ1,NEWP1,NEWQ1) = -CD1I + CD2I + CD3I - CD4I
  370 CONTINUE
  350 CONTINUE
C
C     *****************************************************************
C
C     COMPUTE COEFFICIENTS FOR THE EQUATIONS WHICH ARE DECOUPLED
C     IN THE SECOND DERIVATIVES.
C
      DO 405 KC = 1, NCOEFF
      KMAX(KC) = 0
  405 CONTINUE
C
C     CALCULATE INVERSE OF THE MATRIX C1(I,J).
      JMAX = NJMAX
      NJMAX = 2 * NJMAX
C
      V(1) = 1
      CALL GJR(C1,MX2,MX2,NJMAX,0,$500,JC,V)
C
C     USE INVERSE TO CALCULATE DECOUPLED COEFFICIENTS.
C
      DO 410 NP = 1, NJMAX
C
C     LINEAR COEFFICIENTS.
      DO 420 NJ = 1, NJMAX
      DO 420 KC = 1, NCFM1
      TS(KC,NJ) = 0.0
      DO 420 K = 1, NJMAX
      TS(KC,NJ) = TS(KC,NJ) + C1(NJ,K) * C(KC,K,NP)
  420 CONTINUE
      DO 430 NJ = 1, NJMAX
      DO 425 KC = 1, 3
      C(KC,NJ,NP) = TS(KC,NJ)
      ABSVAL = ABS(C(KC,NJ,NP))
      IF (ABSVAL .GE. SM1) KMAX(KC) = KMAX(KC) + 1
  425 CONTINUE
      IF (NOZNL1 .NE. 1) GO TO 430
      C(4,NJ,NP) = TS(4,NJ)
      ABSVAL = ABS(C(4,NJ,NP))
      IF (ABSVAL .GE. SM3) KMAX(4) = KMAX(4) + 1
  430 CONTINUE
C
C     NONLINEAR COEFFICIENTS.
      IF (NONLIN .EQ. 0)  GO TO 410
      DO 415 NQ = 1, NJMAX
      DO 440 NJ = 1, NJMAX
      TSQ(NJ) = 0.0
      DO 440 K = 1, NJMAX
      TSQ(NJ) = TSQ(NJ) + C1(NJ,K) * D(K,NP,NQ)
  440 CONTINUE
      DO 445 NJ = 1, NJMAX
      D(NJ,NP,NQ) = TSQ(NJ)
      ABSVAL = ABS(D(NJ,NP,NQ))
      IF (ABSVAL .GE. SM2) KMAX(NCOEFF) = KMAX(NCOEFF) + 1
  445 CONTINUE
  415 CONTINUE
C
  410 CONTINUE
C
C     ****************************************************************
C     OUTPUT.
C
      IF (NOUT .GE. 2)  GO TO 455
C
C     PRINTED OUTPUT
      WRITE (6,6001)  (TITLE(I), I = 1, 72)
      WRITE (6,6002) GAMMA,UE,RLD,ZCOMB
      IF (NDROPS .EQ. 0) WRITE (6,6020)
      IF (NDROPS .EQ. 1) WRITE (6,6021)
      IF (NOZZLE .EQ. 0)  WRITE (6,6012)
      IF (NOZNL1 .EQ. 1)  GO TO 760
      WRITE (6,6022)
      WRITE (6,6004)
      DO 310 J = 1, JMAX
      WRITE (6,6003)  NAME(J), J, L(J), M(J), N(J), NS(J),
     1                S(J), SJ(J), B(J), YNOZ(J)
  310 CONTINUE
      GO TO 765
  760 CONTINUE
      WRITE (6,6023)
      WRITE (6,6025)
      DO 770 J = 1, JMAX
      WRITE (6,6026)  NAME(J), J, L(J), M(J), N(J), NS(J),
     1                S(J), SJ(J), B(J), YNOZ(J), GNOZ(J)
  770 CONTINUE
  765 CONTINUE
      IF (NONLIN .EQ. 0)  WRITE (6,6013)
C
C     OUTPUT OF LINEAR COEFFICIENTS.
      DO 320 KC = 1, NCFM1
      IF (KC .EQ. 1)  WRITE (6,6005)
      IF (KC .EQ. 2)  WRITE (6,6006)
      IF (KC .EQ. 3)  WRITE (6,6007)
      IF (KC .EQ. 4) WRITE (6,6024)
      WRITE (6,6008)  (J, J = 1, NJMAX)
      WRITE (6,6014)
      DO 320 NJ = 1, NJMAX
      WRITE (6,6009)  NJ, (C(KC,NJ,NP), NP = 1, NJMAX)
  320 CONTINUE
C
C     OUTPUT OF NONLINEAR COEFFICIENTS.
      IF (NONLIN .EQ. 0)  GO 10 452
      DO 400 NJ = 1, NJMAX
      WRITE (6,6010)  NJ
      WRITE (6,6011)  (J, J = 1, NJMAX)
      WRITE (6,6015)
      DO 400 NP = 1, NJMAX
      WRITE (6,6009) NP, (D(NJ,NP,NQ),  NQ = 1, NJMAX)
  400 CONTINUE
  452 IF (NOUT .EQ. 0)  GO TO 4
C
  455 IF (NOUT .EQ. 3) GO TO 460
C
C     WRITE COEFFICIENTS ON FASTRAND FILE.
C
      WRITE (9,7001) GAMMA, UE, ZE, ZCOMB, NDROPS, NJMAX, NOZNL1
C
      DO 450 J = 1, JMAX
      WRITE (9,7002) J, L(J), M(J), N(J), NS(J), S(J), SJ(J),
     1            NAME(J)
  450 CONTINUE
C
      DO 457 J = 1, JMAX
      WRITE (9,7006) J, YNOZ(J), B(J)
  457 CONTINUE
      IF (NOZNL1 .NE. 1) GO TO 720
      DO 730 J = 1, JMAX
      WRITE (9,7007) J, GNOZ(J)
  730 CONTINUE
  720 CONTINUE
C
      DO 460 KC = 1, 3
      WRITE (9,7003) KMAX(KC)
      DO 460 NJ = 1, NJMAX
      DO 460 NP = 1, NJMAX
```

C.2

```
      ABSVAL = ABS(C(KC,NJ,NP))
      IF (ABSVAL .GE. SM1) WRITE (9,7004) NJ,NP, C(KC,NJ,NP)
  460 CONTINUE
C
      IF (NOZNL1 .NE. 1) GO TO 464
      WRITE (9,7003) KMAX(4)
      DO 462 NJ = 1, NJMAX
      DO 462 NP = 1, NJMAX
      ABSVAL = ABS(C(4,NJ,NP))
      IF (ABSVAL .GE. SM3) WRITE (9,7004) NJ, NP, C(4,NJ,NP)
  462 CONTINUE
  464 CONTINUE
C
      WRITE (9,7003) KMAX(NCOEFF)
      IF (NONLIN .EQ. 0)  GO TO 4
      DO 470 NJ = 1, NJMAX
      DO 470 NP = 1, NJMAX
      DO 470 NQ = 1, NJMAX
      ABSVAL = ABS(D(NJ,NP,NQ))
      IF (ABSVAL .GE. SM2) WRITE (9,7005)NJ, NP, NQ, D(NJ,NP,NQ)
  470 CONTINUE
      GO TO 4
C
C     PUNCHED CARD OUTPUT.
C
  480 PUNCH 7001 GAMMA, UE, ZE, ZCOMB, NDROPS, NJMAX, NOZNL1
C
      DO 482 J = 1, JMAX
      PUNCH 7002 J, L(J), M(J), N(J), NS(J), S(J), SJ(J),
     1           NAME(J)
  482 CONTINUE
C
      DO 484 J = 1,JMAX
      PUNCH 7006 J, YNOZ(J), B(J)
  484 CONTINUE
      IF (NOZNL1 .NE. 1) GO TO 740
      DO 750 J = 1, JMAX
      PUNCH 7007 J,GNOZ(J)
  750 CONTINUE
  740 CONTINUE
C
      DO 486 KC = 1, 3
      PUNCH 7003 KMAX(KC)
      DO 486 NJ = 1, NJMAX
      DO 486 NP = 1, NJMAX
      ABSVAL = ABS(C(KC,NJ,NP))
      IF (ABSVAL .GE. SM1) PUNCH 7004 NJ, NP, C(KC,NJ,NP)
  486 CONTINUE
C
      IF (NOZNL1 .NE. 1) GO TO 490
      PUNCH 7003 KMAX(4)
      DO 492 NJ = 1, NJMAX
      DO 492 NP = 1, NJMAX
      ABSVAL = ABS(C(4,NJ,NP))
      IF (ABSVAL .GE. SM3) PUNCH 7004 NJ, NP, C(4,NJ,NP)
  492 CONTINUE
  490 CONTINUE
C
      PUNCH 7003  KMAX (NCOEFF)
      IF (NONLIN .EQ. 0) GO TO 4
      DO 488 NJ = 1, NJMAX
      DO 488 NP = 1, NJMAX
      DO 488 NQ = 1, NJMAX
      ABSVAL = ABS(D(NJ,NP,NQ))
      IF (ABSVAL .GE. SM2) PUNCH 7005 NJ, NP, NQ ,  D(NJ,NP,NQ)
  488 CONTINUE
      GO TO 4
C
C     ERROR EXIT
  500 IF (JC(1))  510, 510, 520
  510 JC(1) = ABS(JC(1))
      WRITE (6,6017) JC(1)
      GO TO 4
  520 WRITE (6,6018) JC(1)
      GO TO 4
  600 CONTINUE
      WRITE (6,6027)
C
C     ******************************************************************
C
C     FORMAT SPECIFICATIONS.
 5000 FORMAT (72A1)
 5001 FORMAT (4F10.0,2I5)
 5002 FORMAT (5I5,1X,A4)
 5003 FORMAT (I5,2F10.0)
 5004 FORMAT (6I5)
 5005 FORMAT (2F10.0)
 6001 FORMAT (1H1,1X,72A1//)
 6002 FORMAT (2X,8HGAMMA = ,F5.2,5X,5HUE = ,F5.2,5X,6HL/D = ,F8.5,
    1        5X,8HZCOMB = ,F5.2/)
 6003 FORMAT (2X,A4,5I5,4F10.5,2F11.5/)
 6004 FORMAT (2X////2X,29HNAME     J     L     M     N    NS,7X,3HSMN,3X,
    1        7HJM(SMN),7X,3HEPS,7X,3HETA,8X,2HYR,8X,2HYI//)
 6005 FORMAT (1H1,45H  DECOUPLED COEFFICIENT OF B(P):      C(1,J,P)///)
 6006 FORMAT (1H1,44H  DECOUPLED COEFFICIENT OF THE DERIVATIVE OF,
    1        6H B(P):,5X,8HC(2,J,P)///)
 6007 FORMAT (1H1,39H  DECOUPLED COEFFICIENT OF THE RETARDED,
    1        20H DERIVATIVE OF B(P):,5X,8HC(3,J,P)///)
 6008 FORMAT (7X,1HP,I8,9I12)
 6009 FORMAT (2X//2X,I3,3X,10F12.6)
 6010 FORMAT (1H1,42H  DECOUPLED COEFFICIENT OF B(P) * DB(Q)/DT,
    1        19H IN EQUATION FOR B(,I2,1H)///)
 6011 FORMAT (7X,1HQ,I8,9I12)
 6012 FORMAT (2X,19HQUASI-STEADY NOZZLE/)
 6013 FORMAT (2X//2X,24HLINEAR COEFFICIENTS ONLY)
 6014 FORMAT (4X,1HJ)
 6015 FORMAT (4X,1HP)
 6017 FORMAT (1H1,31H OVERFLOW DETECTED, LAST ROW = ,I5)
 6018 FORMAT (1H1,34H SINGULARITY DETECTED, LAST ROW = ,I5)
 6020 FORMAT (2X,"DROPLET MOMENTUM SOURCE NEGLECTED"/)
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
6021 FORMAT (2X,"DROPLET MOMENTUM SOURCE INCLUDED"/)
6022 FORMAT (2X,"NOZZLE NONLINEARITIES NEGLECTED"/)
6023 FORMAT (2X,"NOZZLE NONLINEARITIES INCLUDED"/)
6024 FORMAT (1H1,"  DECOUPLED COEFFICIENT DUE TO NOZZLE",
   1          " NONLINEARITIES:",5X,8HC(4,J,P)////)
6025 FORMAT (2X////2X,29HNAME     J     L     M     N    NS,7X,3HSMN,3X,
   1          7HJM(SMN),7X,3HEPS,7X,3HETA,8X,2HYR,8X,2HYI,
   2          8X,2HGR,8X,2HGI//)
6026 FORMAT (2X,A4,5I5,4F10.5,4F11.5/)
6027 FORMAT (1H1)
7001 FORMAT (4F10.5,3I5)
7002 FORMAT (5I5,2F10.5,1X,A4)
7003 FORMAT (I5)
7004 FORMAT (2I5,F15.6)
7005 FORMAT (3I5,F15.6)
7006 FORMAT (I5,4F10.5)
7007 FORMAT (I5,2F10.5)
     END
```

```
      SUBROUTINE EIGVAL(L,SMN,GAMMA,ZE,YAMPL,YPHASE,RESULT)
C
      COMPLEX   RESULT
      COMMON  /BLK1/ GSQ, ABSQ, ALBET, SMNSQ
C
C     ******************************************************************
C
C     THIS SUBROUTINE COMPUTES THE COMPLEX AXIAL ACOUSTIC EIGENVALUES
C     FOR A CYLINDRICAL CHAMBER WITH A NOZZLE AND STORES THEM IN
C     RESULT.
C     THE EIGENVALUES ARE COMPUTED BY MEANS OF NEWTONS METHOD.
C
C     THE INPUT PARAMETERS ARE AS FOLLOWS
C     L IS THE AXIAL MODE NUMBER.
C     SMN IS THE DIMENSIONLESS ACOUSTIC FREQUENCY.
C     GAMMA IS THE SPECIFIC HEAT RATIO.
C     ZE IS THE LENGTH-TO-RADIUS RATIO.
C     YAMPL IS THE NOZZLE AMPLITUDE FACTOR.
C     YPHASE IS THE NOZZLE PHASE SHIFT IN DEGREES.
C
C     ******************************************************************
C
      PI = 3.1415927
      ERR = 0.0000001
C
      IF (YAMPL)  5,60,5
C     CALCULATE CONSTANTS.
    5 PHASE = YPHASE * PI/180.0
      ALPHA = YAMPL * COS(PHASE)
      BETA  = YAMPL * SIN(PHASE)
      GSQ = GAMMA * GAMMA
      ABSQ = (ALPHA * ALPHA) - (BETA * BETA)
      ALBET = ALPHA * BETA
      SMNSQ = SMN * SMN
C
C     ASSIGN INITIAL GUESS FOR EIGENVALUE.
      IF (L .EQ. 0)  GO TO 45
      RL = L
      PHI = PI/2.0 + PHASE
      XM = RL * PI/ZE
      A = YAMPL/ZE
      XO = XM + A*COS(PHI)
      YO = A*SIN(PHI)
      GO TO 47
   45 CONTINUE
      YPHI = YPHASE
      IF (YPHASE .GT. 180) YPHI = YPHASE - 180.
      IF (YPHASE .LT. 0) YPHI = YPHASE + 180.
      IF (YAMPL .LT. 0.1) GO TO 110
      IF (YAMPL .LT. 0.4) GO TO 120
      IF (YAMPL .LT. 0.8) GO TO 150
      IF (YAMPL .LT. 1.2) GO TO 160
      XO = 1.0 * YAMPL
      GO TO 170
  160 XO = 1.25 * YAMPL
  170 IF (YPHI .LE. 30.) TANPSI = -0.4
      IF (YPHI.GT.30. .AND. YPHI.LE.60.) TANPSI = -0.2
      IF (YPHI.GT.60. .AND. YPHI.LE.120.) TANPSI = 0.0
      IF (YPHI.GT.120. .AND. YPHI.LE.150.) TANPSI = 0.2
      IF (YPHI .GT. 150.) TANPSI = 0.4
      GO TO 140
  150 XO = 2.0 * YAMPL
      IF (YPHI .LE. 30.) TANPSI = -0.6
      IF (YPHI.GT.30. .AND. YPHI.LE.60.) TANPSI = -0.3
      IF (YPHI.GT.60. .AND. YPHI.LE.120.) TANPSI = 0.0
      IF (YPHI.GT.120. .AND. YPHI.LE.150.) TANPSI = 0.3
      IF (YPHI .GT. 150.) TANPSI = 0.6
      GO TO 140
  110 XO = 5. * YAMPL
      GO TO 130
  120 XO = 3. * YAMPL
  130 CONTINUE
      IF (YPHI .LE. 30.) TANPSI = -0.75
      IF (YPHI.GT.30. .AND. YPHI.LE.60.) TANPSI = -0.4
      IF (YPHI.GT.60. .AND. YPHI.LE.120.) TANPSI = 0.0
      IF (YPHI.GT.120. .AND. YPHI.LE.150.) TANPSI = 0.4
      IF (YPHI .GT. 150.) TANPSI = 0.75
  140 CONTINUE
      YO = XO * TANPSI
C
C     ITERATION USING NEWTONS METHOD FOR A SYSTEM OF TWO EQUATIONS
C     IN TWO UNKNOWNS.
   47 L1 = 0
      X = XO
      Y = YO
   40 CALL FCNS(X,Y,ZE,F,G,FX,FY,GX,GY)
      IF (L1 .EQ. 40)  GO TO 50
      RJFG = (FX * GY) - (GX * FY)
      IF (RJFG) 20, 30, 20
   20 DELTAX = (-F * GY  +  G * FY)/RJFG
      DELTAY = (-G * FX  +  F * GX)/RJFG
      L1 = L1 + 1
      X = X + DELTAX
      Y = Y + DELTAY
C
C     TEST FOR CONVERGENCE.
      IF (ABS(DELTAX) .GE. ERR .OR. ABS(DELTAY) .GE. ERR)  GO TO 40
      GO TO 10
C
C     WARNING MESSAGES
   30 WRITE (6,6005)
      GO TO 10
   50 WRITE (6,6006)
      GO TO 10
C
C     CASE OF HARD WALL (YAMPL = 0).
   60 RL = L
```

```
      X = RL * PI/ZE
      Y = 0.0
   10 RESULT = CMPLX(X,Y)
C
C     FORMAT SPECIFICATIONS.
 6005 FORMAT (2X//2X,16HJACOBIAN IS ZERO//)
 6006 FORMAT (2X//2X,35HFAILED TO CONVERGE IN 40 ITERATIONS//)
      RETURN
      END
      SUBROUTINE FCNS(X,Y,ZE,F,G,FX,FY,GX,GY)
C
C     THIS SUBROUTINE COMPUTES THE FUNCTIONS F(X,Y) AND G(X,Y)
C     AND THEIR PARTIAL DERIVATIVES WITH RESPECT TO X AND Y.
C
      COMMON  /BLK1/ GSQ, ABSQ, ALBET, SMNSQ
C
C     COMPUTE THE TRIGONOMETRIC FUNCTIONS, THE HYPERBOLIC FUNCTIONS
C     AND THEIR SQUARES.
C
      I = 1
      ARGX = ZE * X
      ARGY = ZE * Y
   10 SX = SIN(ARGX)
      CX = COS(ARGX)
      SHY = SINH(ARGY)
      CHY = COSH(ARGY)
      IF (I .EQ. 2)  GO TO 20
      SXSQ = SX * SX
      CXSQ = CX * CX
      SHYSQ = SHY * SHY
      CHYSQ = CHY * CHY
      ARGX = 2.0 * ARGX
      ARGY = 2.0 * ARGY
      I = 2
      GO TO 10
C
C     COMPUTE TRANSCENDENTAL FUNCTIONS AND THEIR DERIVATIVES
C
   20 FF = (SXSQ * CHYSQ) - (CXSQ * SHYSQ)
      GG = (CXSQ * CHYSQ) - (SXSQ * SHYSQ)
      HH = 0.25 * SX * SHY
      FFX = ZE * SX * CHY
      GGY = ZE * CX * SHY
      FFY = -GGY
      GGX = -FFX
      HHX = 0.5 * GGY
      HHY = 0.5 * FFX
C
C     COMPUTE FACTORS
      XYSQ = (X * X)' - (Y * Y)
      XY = X * Y
      SMNXY = SMNSQ + XYSQ
      F1 = (ABSQ * SMNXY) - (4.0 * ALBET * XY)
      F2 = (ALBET * SMNXY) + (ABSQ * XY)
      G1 = (ABSQ * SMNXY) + (4.0 * ALBET * XY)
      FX1 = (2.0 * X * ABSQ) - (4.0 * ALBET * Y)
      FX2 = (2.0 * X * ALBET) + (ABSQ * Y)
      FY1 = (-2.0 * Y * ABSQ) - (4.0 * ALBET * X)
      FY2 = (-2.0 * Y * ALBET) + (ABSQ * X)
      GX1 = (2.0 * X * ABSQ) + (4.0 * ALBET * Y)
      GY1 = (-2.0 * Y * ABSQ) + (4.0 * ALBET * X)
C
C     COMPUTE F(X,Y) AND G(X;Y)
C
      F = (XYSQ * FF) - (4.0 * XY * HH)
     1    + GSQ * ((F1 * GG) + (4.0 * F2 * HH))
      G = (XYSQ * HH) + (XY * FF)
     1    + GSQ * ((F2 * GG) - (G1 * HH))
C
C     COMPUTE THE PARTIAL DERIVATIVES OF F AND G
C
      FX = (2.0 * X * FF) + (XYSQ * FFX)
     1    -4.0 * ((Y * HH) + (XY * HHX))
     2    + GSQ * ((FX1 * GG) + (F1 * GGX)
     3    + (4.0 * FX2 * HH) + (4.0 * F2 * HHX))
      FY = (-2.0 * Y * FF) + (XYSQ * FFY)
     1    -4.0 * ((X * HH) + (XY * HHY))
     2    + GSQ * ((FY1 * GG) + (F1 * GGY)
     3    + (4.0 * FY2 * HH) + (4.0 * F2 * HHY))
      GX = (2.0 * X * HH) + (XYSQ * HHX)
     1    + (Y * FF) + (XY * FFX)
     2    + GSQ * ((FX2 * GG) + (F2 * GGX)
     3    -(GX1 * HH) - (G1 * HHX))
      GY = (-2.0 * Y * HH) + (XYSQ * HHY)
     1    + (X * FF) + (XY * FFY)
     2    + GSQ * ((FY2 * GG) + (F2 * GGY)
     3    -(GY1 * HH) - (G1 * HHY))
      RETURN
      END
      SUBROUTINE AXIAL1 (NOPT,NP,NJ,UE,ZE,ZCOMB,RESULT)
C
C
C     THIS SUBROUTINE CALCULATES THE INTEGRAL OVER THE INTERVAL
C     (O,ZE) OF THE FOLLOWING FUNCTIONS ACCORDING TO THE VALUE
C     OF NOPT
C
C     NOPT = 1       Z(NP) * ZC(NJ)
C     NOPT = 2       ZPP(NP) * ZC(NJ)
C     NOPT = 3       UP * Z(NP) * ZC(NJ)
C     NOPT = 4       U * ZP(NP) * ZC(NJ)
C
C     IN THE ABOVE EQUATIONS:
C     Z(NP) IS THE AXIAL ACOUSTIC EIGENFUNCTION OF INDEX NP.
C     Z(NJ) IS THE AXIAL ACOUSTIC EIGENFUNCTION OF INDEX NJ.
C     ZC IS THE COMPLEX CONJUGATE OF THE AXIAL EIGENFUNCTION.
C     ZP AND ZPP ARE THE FIRST AND SECOND DERIVATIVES OF THE
C     AXIAL EIGENFUNCTIONS RESPECTIVELY.
C     U IS THE STEADY STATE VELOCITY DISTRIBUTION AND UP IS ITS
C     AXIAL DERIVATIVE.
C     THE VELOCITY DISTRIBUTION IS COMPUTED BY THE SUBROUTINE UBAR.
C
C
      PARAMETER MX = 5
      REAL     MAG
      COMPLEX  CI, CZE, BP, BJ, T1, T2, CH, F1, F2, F3, CZ, ARG,
     1         S1, S2, S3, RESULT, FUNCT(500),B(MX)
      COMMON   B
C
      CI = (0.0,1.0)
      CZE = CMPLX(ZE,0.0)
      BP = B(NP)
      BJ = CONJG(B(NJ))
C
      IF (NOPT .GT. 2)  GO TO 50
C     CALCULATE INTEGRALS BY MEANS OF ANALYTICAL EXPRESSIONS FOR
C     NOPT = 1 AND NOPT = 2.
      ARG = (BP + BJ) * CI
      MAG = CABS(ARG)
      IF (MAG) 20, 25, 20
   20 T1 = CSINH(ARG*CZE)/ARG
      GO TO 30
   25 T1 = CZE
   30 ARG = (BP - BJ) * CI
      MAG = CABS(ARG)
      IF (MAG) 35, 40, 35
   35 T2 = CSINH(ARG*CZE)/ARG
      GO TO 45
   40 T2 = CZE
   45 RESULT = (T1 + T2) * (0.5,0.0)
      IF (NOPT .EQ. 2)  RESULT = -B(NP) * B(NP) * RESULT
      GO TO 100
C
```

```
C     NUMERICAL EVALUATION OF INTEGRALS FOR NOPT = 3 AND NOPT = 4.
C
C     COMPUTE STEP SIZE FOR SIMPSON INTEGRATION.
   50 N = 50
      RN = N
      RESULT = (0.0,0.0)
      IC = ZCOMB
      IC = 2 - IC
C
      DO 90 J = 1, IC
      IF (J .EQ. 1)  H = ZCOMB * ZE/RN
      IF (J .EQ. 2)  H = (1.0 - ZCOMB) * ZE/RN
      IF (J .EQ. 1)  Z0 = 0.0
      IF (J .EQ. 2)  Z0 = ZCOMB * ZE
      NP1 = N + 1
      CH = CMPLX(H,0.0)
C
C     COMPUTE INTEGRANDS.
      DO 60 I = 1, NP1
      STEP = I - 1
      Z = (STEP * H) + Z0
      IF ((I.EQ.1) .AND. (J.EQ.2)) Z = Z + H/100.0
      IF (NOPT .EQ. 3)  CALL UBAR(2,UE,ZE,ZCOMB,Z,F)
      IF (NOPT .EQ. 4)  CALL UBAR(1,UE,ZE,ZCOMB,Z,F)
      F1 = CMPLX(F,0.0)
      CZ = CMPLX(Z,0.0)
      ARG = CI * BP
      IF (NOPT .EQ. 3)  F2 = CCOSH(ARG*CZ)
      IF (NOPT .EQ. 4)  F2 = ARG * CSINH(ARG*CZ)
      ARG = CI * BJ
      F3 = CCOSH(ARG*CZ)
      FUNCT(I) = F1 * F2 * F3
   60 CONTINUE
C
C     PERFORM SIMPSON INTEGRATION.
      NM1 = N - 1
      S1 = FUNCT(1) + FUNCT(NP1)
      S2 = (0.0,0.0)
      S3 = (0.0,0.0)
      DO 70 I = 2, N, 2
      S2 = S2 + FUNCT(I)
   70 CONTINUE
      DO 80 I = 3, NM1, 2
      S3 = S3 + FUNCT(I)
   80 CONTINUE
      RESULT = RESULT +
     1         CH * (S1 + (4.0,0.0)*S2 + (2.0,0.0)*S3)/(3.0,0.0)
   90 CONTINUE
C
  100 CONTINUE
      RETURN
      END
      SUBROUTINE AXIAL2(NOPT,NCONJ,NP,NQ,NJ,ZE,RESULT)
C
C
C     THIS SUBROUTINE CALCULATES THE INTEGRAL OVER THE INTERVAL
C     (0,ZE) OF THE FOLLOWING FUNCTIONS ACCORDING TO THE VALUES
C     OF NOPT AND NCONJ
C
C     FOR NCONJ = 1 AND
C     NOPT = 1      Z(NP) * Z(NQ) * ZC(NJ)
C     NOPT = 2      ZP(NP) * ZP(NQ) * ZC(NJ)
C     NOPT = 3      ZPP(NP) * Z(NQ) * ZC(NJ)
C
C     FOR NCONJ = 2 AND
C     NOPT = 1      Z(NP) * ZC(NQ) * ZC(NJ)
C     NOPT = 2      ZP(NP) * ZPC(NQ) * ZC(NJ)
C     NOPT = 3      ZPP(NP) * ZC(NQ) * ZC(NJ)
C
C     FOR NCONJ = 3 AND
C     NOPT = 1      ZC(NP) * Z(NQ) * ZC(NJ)
C     NOPT = 2      ZPC(NP) * ZP(NQ) * ZC(NJ)
C     NOPT = 3      ZPPC(NP) * Z(NQ) * ZC(NJ)
C
C     FOR NCONJ = 4 AND
C     NOPT = 1      ZC(NP) * ZC(NQ) * ZC(NJ)
C     NOPT = 2      ZPC(NP) * ZPC(NQ) * ZC(NJ)
C     NOPT = 3      ZPPC(NP) * ZC(NQ) * ZC(NJ)
C
C     IN THE ABOVE EQUATIONS:
C     Z(NP), Z(NQ), AND Z(NJ) ARE THE AXIAL ACOUSTIC EIGENFUNCTIONS
C     AND NP, NQ, AND NJ ARE THEIR INDICES.
C     ZP IS THE FIRST DERIVATIVE OF THE AXIAL EIGENFUNCTIONS.
C     ZPP IS THE SECOND DERIVATIVE OF THE AXIAL EIGENFUNCTIONS.
C     ZC AND ZPC ARE COMPLEX CONJUGATES OF Z AND ZP RESPECTIVELY.
C
C
      PARAMETER MX = 5
      REAL      MAG
      COMPLEX   CI, CF, CZE, BP, BQ, BJ, SUM, RESULT,
     1          ARG(4), FUNCT(4), B(MX)
      COMMON    B
C
C     CALCULATE INTEGRALS BY MEANS OF ANALYTICAL EXPRESSIONS.
      CI = (0.0,1.0)
      CF = (0.25,0.0)
      CZE = CMPLX(ZE,0.0)
      BP = B(NP)
      BQ = B(NQ)
      BJ = CONJG(B(NJ))
      IF ((NCONJ.EQ.2) .OR. (NCONJ.EQ.4))  BQ = CONJG(BQ)
      IF (NCONJ .GT. 2)  BP = CONJG(BP)
      ARG(1) = (BP + BQ + BJ) * CI
      ARG(2) = (BP + BQ - BJ) * CI
      ARG(3) = (BP - BQ + BJ) * CI
      ARG(4) = (BP - BQ - BJ) * CI
      DO 10 J = 1,4
      MAG = CABS(ARG(J))
      IF (MAG) 12, 15, 12
   12 FUNCT(J) = CSINH(ARG(J)*CZE)/ARG(
      GO TO 10
   15 FUNCT(J) = CZE
   10 CONTINUE
      IF (NOPT .EQ. 2)  GO TO 30
      SUM = FUNCT(1) + FUNCT(2) + FUNCT(3) + FUNCT(4)
      RESULT = CF * SUM
      IF (NOPT .EQ. 3)  RESULT = -BP * BP * RESULT
      GO TO 50
   30 SUM = FUNCT(1) + FUNCT(2) - FUNCT(3) - FUNCT(4)
      RESULT = -CF * BP * BQ * SUM
   50 CONTINUE
      RETURN
      END
      SUBROUTINE AZIMTL(NOPT,NP,NQ,NJ,RESULT)
C
      PARAMETER MX = 5
      DIMENSION    NFCN(3), SG(2)
      COMMON  /BLK2/    M(MX), NS(MX)
C
C     *****************************************************************
C
C     THIS SUBROUTINE CALCULATES THE INTEGRAL OVER THE INTERVAL
C     (0, 2*PI)  OF THE FOLLOWING FUNCTIONS ACCORDING TO THE VALUE
C     OF NOPT
C
C     NOPT = 1      TH(NP) * TH(NQ) * TH(NJ)
C
C     NOPT = 2      THP(NP) * THP(NQ) * TH(NJ)
C
C     IN THE ABOVE EQUATIONS:
C     TH(NP), TH(NQ), AND TH(NJ) ARE THE TANGENTIAL EIGENFUNCTIONS
C     AND NP, NQ, AND NJ ARE THEIR INDICES.
C     THP IS THE DERIVATIVE OF THE TANGENTIAL EIGENFUNCTIONS.
C
C     IF NS = 1  TH = SIN(M*THETA)
C     IF NS = 2  TH = COS(M*THETA)
C
C     *****************************************************************
C
      RESULT = 0.0
      FACTOR = 1.0
      PI = 3.1415927
C
C     DISTINGUISH BETWEEN SINES AND COSINES.
      DO 10 K1 = 1, 3
      NFCN(K1) = 1
   10 CONTINUE
      IF  (NS(NJ).EQ.2) NFCN(3) = 2
      IF (NOPT .EQ. 2)  GO TO 20
      IF  (NS(NP).EQ.2) NFCN(1) = 2
      IF  (NS(NQ).EQ.2) NFCN(2) = 2
      GO TO 30
   20 IF  (NS(NP).EQ.1) NFCN(1) = 2
      IF  (NS(NQ).EQ.1) NFCN(2) = 2
      DO 40 K1 = 1,2
      SG(K1) = 1.0
      IF (NFCN(K1) .EQ. 1)  SG(K1) = -1.0
   40 CONTINUE
      FACTOR = SG(1) * SG(2) * M(NP) * M(NQ)
C
   30 NSUM = 0
      DO 50 K1 = 1, 3
      NSUM = NSUM + NFCN(K1)
   50 CONTINUE
C
```

```
      IF ((NSUM .EQ. 3) .OR. (NSUM .EQ. 5))  GO TO 60
      IF (NSUM .EQ. 4)  GO TO 70
      IF (NSUM .EQ. 6)  GO TO 80
C
   70 KOPT = 2
      IF (NFCN(1) .EQ. 2)  GO TO 72
      GO TO 74
   72 LL = M(NP)
      MM = M(NQ)
      NN = M(NJ)
      GO TO 90
   74 IF (NFCN(2) .EQ. 2)  GO TO 76
      GO TO 78
   76 LL = M(NQ)
      MM = M(NP)
      NN = M(NJ)
      GO TO 90
   78 LL = M(NJ)
      MM = M(NP)
      NN = M(NQ)
      GO TO 90
C
   80 KOPT = 1
      LL = M(NP)
      MM = M(NQ)
      NN = M(NJ)
C
C     COMPUTE VALUES OF THE INTEGRALS.
C
   90 IF ((LL.NE.0) .AND. (MM.NE.0) .AND. (NN.NE.0))  GO TO 101
      GO TO 103
  101 LM = LL + MM
      LN = LL + NN
      MN = MM + NN
      IF ((NN.EQ.LM) .OR. (MM.EQ.LN))   RESULT = PI/2.0
      IF (LL .EQ. MN)  GO TO 102
      GO TO 104
  102 IF (KOPT .EQ. 1)  RESULT = PI/2.0
      IF (KOPT .EQ. 2)  RESULT = -PI/2.0
      GO TO 104
  103 IF ((LL.EQ.0) .AND. (MM.EQ.0) .AND. (NN.EQ.0))  GO TO 105
      IF ((KOPT.EQ.1) .AND. (NN.EQ.0) .AND. (LL.EQ.MM)) RESULT = PI
      IF ((KOPT.EQ.1) .AND. (MM.EQ.0) .AND. (LL.EQ.NN)) RESULT = PI
      IF ((LL .EQ. 0) .AND. (MM .EQ. NN))  RESULT = PI
      GO TO 104
  105 IF (KOPT .EQ. 1)  RESULT = 2.0 * PI
  104 CONTINUE
      RESULT = FACTOR * RESULT
   60 CONTINUE
      RETURN
      END
      SUBROUTINE RADIAL(NOPT,L,M,N,A,B,C,RESULT)
C
C     THIS SUBROUTINE CALCULATES THE INTEGRAL OVER THE INTERVAL
C     (0,1) OF THE FOLLOWING PRODUCTS OF THREE BESSEL FUNCTIONS:
C
C     NOPT = 1  JL(A*R) * JM(B*R) * JN(C*R) * R
C
C     NOPT = 2  JL(A*R) * JM(B*R) * JN(C*R)/R
C
C     NOPT = 3  JPL(A*R) * JPM(B*R) * JN(C*R) * R
C
C     JL  IS THE BESSEL FUNCTION OF FIRST KIND OF ORDER L
C     JPL IS THE DERIVATIVE OF JL WITH RESPECT TO R
C     L, M, N  ARE NON-NEGATIVE INTEGERS
C     A, B, C  ARE REAL NUMBERS
C
      DIMENSION  FUNCT(200)
      DOUBLE PRECISION  DN, DH, DSTEP, DR, ARG1, ARG2, ARG3,
     1                  BES1, BES2, BES3, BESH, BESL, PROD,
     2                  FUNCT, BESLIM, S1, S2, S3
C
      NN = 100
      DN = NN
      DH = 1.0/DN
      NP1 = NN + 1
C
      DO 10 I = 1, NP1
      DSTEP = I - 1
      DR = DH * DSTEP
      ARG1 = A * DR
      ARG2 = B * DR
      ARG3 = C * DR
C
      CALL JBES(N,ARG3,BES3,$500)
      IF (NOPT .EQ. 3)  GO TO 101
      CALL JBES(L,ARG1,BES1,$500)
      CALL JBES(M,ARG2,BES2,$500)
      GO TO 102
  101 IF (L .EQ. 0)  GO TO 103
      CALL JBES(L+1,ARG1,BESH,$500)
      CALL JBES(L-1,ARG1,BESL,$500)
      BES1 = A * (BESL - BESH)/2.0
      GO TO 104
  103 CALL JBES(1,ARG1,BES1,$500)
      BES1 = -BES1 * A
  104 IF (M .EO. 0)  GO TO 105
      CALL JBES(M+1,ARG2,BESH,$500)
      CALL JBES(M-1,ARG2,BESL,$500)
      BES2 = B * (BESL - BESH)/2.0
      GO TO 102
  105 CALL JBES(1,ARG2,BES2,$500)
      BES2 = -BES2 * B
  102 PROD = BES1 * BES2 * BES3
C
      IF (NOPT .EQ. 2)  GO TO 110
      FUNCT(I) = PROD * DR
      GO TO 10
  110 IF (I .EQ. 1)  GO TO 111
      FUNCT(I) = PROD/DR
      GO TO 10
  111 BESLIM = 0.0
      IF ((L.EQ.1) .AND. (M.EQ.0) .AND. (N.EQ.0))  BESLIM = A/2.0
      IF ((L.EQ.0) .AND. (M.EQ.1) .AND. (N.EQ.0))  BESLIM = B/2.0
      IF ((L.EQ.0) .AND. (M.EQ.0) .AND. (N.EQ.1))  BESLIM = C/2.0
      FUNCT(I) = BESLIM
   10 CONTINUE
C
      NM1 = NN - 1
      S1 = FUNCT(1) + FUNCT(NP1)
      S2 = 0.0
      S3 = 0.0
      DO 20 I = 2, NN, 2
      S2 = S2 + FUNCT(I)
   20 CONTINUE
      DO 30 I = 3, NM1, 2
      S3 = S3 + FUNCT(I)
   30 CONTINUE
      RESULT = DH * (S1 + 4.0*S2 + 2.0*S3)/3.0
      GO TO 501
  500 WRITE (6, 6000)
 6000 FORMAT (1H1,10HERROR JBES)
  501 CONTINUE
      RETURN
      END
      SUBROUTINE UBAR(NOPT,UE,ZE,ZCOMB,Z,RESULT)
C
C     THIS SUBROUTINE CALCULATES THE STEADY STATE VELOCITY
C     DISTRIBUTION FOR UNIFORMLY DISTRIBUTED COMBUSTION COMPLETED AT
C     Z = ZCOMB * ZE WHERE:
C     UE IS THE EXIT MACH NUMBER.
C     ZE IS THE DIMENSIONLESS LENGTH.
C     Z IS THE AXIAL COORDINATE.
C
C     IF NOPT = 1 THE DISTRIBUTION IS CALCULATED.
C     IF NOPT = 2 THE DERIVATIVE IS CALCULATED.
C     IF NOPT = 3 THE SECOND DERIVATIVE IS CALCULATED.
C
C
      ECZ = ZCOMB * ZE
      GO TO (10,20,30), NOPT
   10 IF (Z .LE. ECZ) RESULT = UE * Z/ECZ
      IF (Z .GT. ECZ) RESULT = UE
      GO TO 40
   20 IF (Z .LE. ECZ) RESULT = UE/ECZ
      IF (Z .GT. ECZ) RESULT = 0.0
      GO TO 40
   30 RESULT = 0.0
   40 CONTINUE
      RETURN
      END
```

APPENDIX C

PROGRAM LCYC3D: A USER'S MANUAL

Program LCYC3D calculates the nonlinear stability characteristics of the combustion chamber described in Fig. 3 by numerically integrating the system of differential equations given by Eq. (20). Except for the term $C_4(j,p)\ e^{ik_p\omega t}$, this equation is the same as Eq. (12) of Ref. 11, whose solution is carried out by the program LCYC3D described in detail in Appendix D of Ref. 11. The present computer program is very similar to Program LCYC3D of Ref. 11 in its general structure, input and output. Hence in this user's manual, only the complete listing of the present program, along with a precise description of the necessary input, is given; for details about the program (including input) one is referred to Appendix D of Ref. 11.

No. of Cards	Location	Type	Input Item	Comments
1	1-5	I	NOUTCF	If 0: coefficients are not printed out If 1: only the linear coefficients are printed out If 2: all the coefficients are printed out
	6-10	I	NOZNL2	If 0: nozzle nonlinearities not included If 1: nozzle nonlinearities included
1	1-72	A	TITLE	Title used to label the plots
1	1-10	F	EN	Interaction index, n
	11-20	F	TAU	Time lag, $\bar{\tau}$
	21-30	F	H	Time increment for numerical integration
	31-40	F	TSTART	Time at which output of solution begins

No. of Cards	Location	Type	Input Item	Comments
	41-50	F	TQUIT	Time at which output of solution ends
1	1-5	I	NTEST	If 0: compute transient behavior If 1: compute limit-cycle behavior
	6-10	I	JMODE	Identifies the amplitude function used to test for limit-cycles
	11-15	I	NLOC	Determines location for wall pressure maxima and minima If 1: $z = 0, \quad \theta = 0^\circ$ If 2: $z = 0, \quad \theta = 45^\circ$ If 3: $z = 0, \quad \theta = 90^\circ$
	16-20	I	NTERMS	Number of amplitude functions given initial values
	21-25	I	NPZ	Determines how secondary instability zones are handled If 0: all instability zones included If 1: secondary zones eliminated
	26-30	I	NOUT	Determines output If 0: printed output only If $1 \leq$ NOUT ≤ 6: both printed and plotted output; NOUT being the number of the last plot produced
	31-35	I	ICTYPE	If 1: amplitudes selected to satisfy the nozzle boundary condition If 2: amplitudes selected to eliminate the extraneous solution

The next three cards are necessary only if 1 ≤ NOUT ≤ 6.

No. of Cards	Location	Type	Input Item	Comments
1	1-10	F	YHI(1)	Maximum ordinate for pressure plots
	11-20	F	YHI(5)	Maximum ordinate for velocity plots
	21-30	F	YLAB(1)	Interval for ordinate labeling of pressure plots
	31-40	F	YLAB(5)	Interval for ordinate labeling of velocity plots
1	1-5	I	ITICY(1)	Number of ordinate tic marks for pressure plots
	6-10	I	ITICY(5)	Number of ordinate tic marks for velocity plots
	11-15	I	NFIRST	Gives the number of the first plot produced
	16-20	I	NOMIT	If 0: time-history plot produced If 1: time-history plot omitted
1	1-5	I	MDPLOT(1)	If 0: plot of the first mode amplitude not produced If 1: plot of the first mode amplitude is produced
	6-10	I	MDPLOT(2)	If 0: plot of the second mode amplitude not produced If 1: plot of the second mode amplitude is produced
	11-15	I	MDPLOT(3)	If 0: plot of the third mode amplitude not produced If 1: plot of the third mode amplitude is produced

No. of Cards	Location	Type	Input Item	Comments
	16-20	I	MDPLOT(4)	If 0: plot of the pressure amplitude of the first mode not produced If 1: plot of the pressure amplitude of the first mode is produced

The next card is necessary only if plot of any mode-amplitude is desired.

1	1-10	F	YHIMD	Maximum ordinate for mode-amplitude plots
	11-20	F	YIABMD	Interval for ordinate labeling of mode-amplitude plots
	21-25	I	ITICMD	Number of ordinate tic marks for mode-amplitude plots
NTERMS	1-5	I	J	Identifies complex amplitude function
	6-15	F	AST	Amplitude of sin(ωt) terms in initial conditions
	16-25	F	ACT	Amplitude of cos(ωt) terms in initial conditions

The next card is necessary only if ICTYPE = 2.

1	1-10	F	DAMP	Damping factor in initial condition, obtained from linear stability analysis (Appendix E of Ref. 11)
	11-20	F	FREQ	Corresponding frequency

FORTRAN Listing

```
C      *************** PROGRAM LCYC3D ********************************
C
C         THIS PROGRAM CALCULATES THE NONLINEAR BEHAVIOR OF
C      TRANSVERSE, AXIAL, OR COMBINED LONGITUDINAL-TRANSVERSE
C      INSTABILITIES IN A CYLINDRICAL COMBUSTION CHAMBER WITH
C      UNIFORM PROPELLANT INJECTION, DISTRIBUTED COMBUSTION
C      PROCESS, AND A CONVENTIONAL NOZZLE.  THE COMBUSTION PROCESS
C      IS DESCRIBED BY CROCCO"S TIME-LAG MODEL.  BOTH TRANSIENT
C      AND LIMIT-CYCLE SOLUTIONS ARE CALCULATED.
C
C      THE FOLLOWING INPUTS ARE REQUIRED
C
C      (1)  THE CONTROL NUMBERS, NOUTCF AND NOZNL2.
C      (2)  THE COEFFICIENTS FROM PROGRAM COEFFS3D.
C      (3)  THE DATA DECK.
C
C      NOUTCF DETERMINES PRINTOUT OF COEFFICIENTS.
C         IF NOUTCF = 0  COEFFICIENTS ARE NOT PRINTED OUT.
C         IF NOUTCF = 1  LINEAR COEFFICIENTS ONLY ARE PRINTED OUT.
C         IF NOUTCF = 2  ALL COEFFICIENTS ARE PRINTED OUT.
C      NOZNL2 DETERMINES IF THE NOZZLE NONLINEARITIES ARE TO BE INCLUDED.
C         IF NOZNL2 = 0  NOZZLE NONLINEARITIES NOT INCLUDED.
C         IF NOZNL2 = 1  NOZZLE NONLINEARITIES INCLUDED.
C
C      THE DATA DECK CONTAINS THE FOLLOWING INFORMATION:
C
C      TITLE OF THE RUN.
C
C      EN IS THE INTERACTION INDEX.
C      TAU IS THE TIME LAG.
C      H IS THE INTEGRATION STEP SIZE.
C      TSTART IS THE TIME AT WHICH OUTPUT STARTS.
C      TQUIT IS THE TIME AT WHICH COMPUTATIONS ARE TERMINATED.
C
C      NTEST IS TASK CONTROL NUMBER:
C         IF NTEST = 0   COMPUTE TRANSIENT BEHAVIOR.
C         IF NTEST = 1   COMPUTE THE LIMIT-CYCLE BEHAVIOR.
C      JMODE IS THE MODE-AMPLITUDE USED TO TEST FOR LIMIT-CYCLES.
C      NLOC DETERMINES THE LOCATION OF THE WALL PRESSURE MAXIMA
C      AND MINIMA:
C         IF NLOC = 1    LOCATION IS Z = 0, THETA = 0 DEGREES.
C         IF NLOC = 2    LOCATION IS Z = 0, THETA = 45 DEGREES.
C         IF NLOC = 3    LOCATION IS Z = 0, THETA = 90 DEGREES.
C      NTERMS IS THE NUMBER OF TERMS GIVEN INITIAL VALUES.
C      NPZ DETERMINES HOW SECONDARY STABILITY ZONES (PHANTOM
C      ZONES) ARE HANDLED.
C         IF NPZ = 0 PHANTOM ZONES ARE RETAINED.
C         IF NPZ = 1 PHANTOM ZONES ARE ELIMINATED.
C      NOUT IS THE OUTPUT CONTROL NUMBER.
C         IF NOUT = 0    PRINTED OUTPUT ONLY.
C         IF NOUT > 0    BOTH PRINTED AND PLOTTED OUTPUT, NOUT
C                        DETERMINES THE NUMBER OF THE LAST PLOT
C                        PRODUCED.
C      ICTYPE IS THE INITIAL CONDITION CONTROL NUMBER:
C         IF ICTYPE = 1     AMPLITUDES SELECTED TO SATISFY
```

```
C                       THE NOZZLE BOUNDARY CONDITION.
C        IF ICTYPE = 2    AMPLITUDES SELECTED TO ELIMINATE THE
C                       EXTRANEOUS SOLUTION.
C
C     DATA FOR SETTING UP PLOTS :
C
C     YHI(1) IS THE MAXIMUM ORDINATE FOR PRESSURE PLOTS.
C     YHI(5) IS THE MAXIMUM ORDINATE FOR VELOCITY PLOTS.
C     NOTE: THE ORDINATE SCALES FOR PRESSURE AND VELOCITY PLOTS
C           ARE SYMMETRIC ABOUT ZERO.
C     YLAB IS THE INTERVAL FOR ORDINATE LABELING FOR ABOVE PLOTS.
C     ITICY IS THE NUMBER OF ORDINATE TIC MARKS FOR ABOVE PLOTS.
C     NOTE: ITICY SHOULD BE NEGATIVE FOR PRESSURE AND VELOCITY PLOTS
C           TO OBTAIN CENTERLINE.
C     NFIRST IS THE NUMBER OF THE FIRST PLOT PRODUCED.
C     NOMIT DETERMINES WHETHER AMPLITUDE PLOT IS PRODUCED
C        IF NOMIT = 0 AMPLITUDE PLOT IS PRODUCED.
C        IF NOMIT = 1 AMPLITUDE PLOT IS OMITTED.
C
C     MDPLOT DETERMINES IF THE PLOT OF THE MODE-AMPLITUDE IS REQUIRED.
C        IF MDPLOT = 0  PLOT NOT REQUIRED.
C        IF MDPLOT = 1  PLOT REQUIRED.
C
C     YHIMD IS THE MAXIMUM ORDINATE FOR AMPLITUDE PLOTS.
C     YLABMD IS THE INTERVAL FOR ORDINATE LABELING OF AMPLITUDE PLOTS.
C     ITICMD IS THE NUMBER OF ORDINATE TIC MARKS.
C     NOTE: ITICMD SHOULD BE NEGATIVE TO OBTAIN THE CENTERLINE.
C
C     INITIAL AMPLITUDES OF F-FUNCTIONS (REMAINING CARDS)
C
C     AS(J) IS THE AMPLITUDE OF THE SINE TERM.
C     AC(J) IS THE AMPLITUDE OF THE COSINE TERM.
C
C     DAMP AND FREQ ARE THE DAMPING COEFFICIENT AND THE FREQUENCY FROM
C     THE LINEAR STABILITY PROGRAM.
C
C
C
      PARAMETER   MX=5, MX2=10, MX4=20, MX2SQ=100
      COMPLEX     YNOZ(MX), B(MX), C1, C2, C3, CPHIT(MX), CSUM, A
      COMPLEX     GNOZ(MX), CAXI, CI
      DIMENSION   L(MX), N(MX), S(MX), NAME(MX), AS(MX2), AC(MX2),
     1            U(250,MX4), Y(MX4), FZ(4,MX4), YP(MX4), UZ(MX4),
     2            CP(4,MX2,MX2), FR01(MX2), IMP1(MX2), UMAX(500),
     3            Z(6), ANGLE(6), THETA(6), CFT(6,MX2), YI(MX2),
     4            CFTH(6,MX2), CFZ(6,MX2), PRESS(6), AXVEL(3), YR(MX2),
     5            TPLOT(500), YPLOT(6,500), DUMMYT(500), DUMMYY(500),
     6            IBUF(3000), ITT(4), ITY1(7), ITY2(7), ITY3(7),
     7            ITY4(7), ITY5(6), TAUCUT(MX2), ITY6(8), UAVG(100),
     8            ITP(3),  TITLE(12), PRS(500), TI(500), PMAX(500),
     9            TIMAX(500), YLO(6), YHI(6), YLAB(6), ITICY(6),
     1            KFREQ(MX), WKP(MX), AA(4), UPLOT(MX,500), PRIT(500),
     2           MDPLOT(4), MTITL1(4), MTITL2(4), MTITL3(4),
     3            MTITL(4), PRTITL(5)
C
```

```
      COMMON          RV(MX2,4), C(4,MX2,MX2), D(MX2,MX2SQ),
     1                KPMAX(4,MX2), IC(4,MX2,MX2), KPGMAX(MX2),
     2                IDP(MX2,MX2SQ), IDO(MX2,MX2SQ)
      COMMON          /BLK2/    M(MX), NS(MX), SJ(MX), B
      COMMON          /BLK3/    NJMAX, NLMAX, GAMMA, COEF(3,MX2)
      COMMON          /NLTERM/  NOZNL2, EXTRA(MX2,4)
C
      DATA      ITT/"DIMENSIONLESS TIME, T"/,
     1          ITY1/"INJECTOR PRESSURE PERTURBATION, THETA = 0"/,
     2          ITY2/"INJECTOR PRESSURE PERTURBATION, THETA = 45"/,
     3          ITY3/"INJECTOR PRESSURE PERTURBATION, THETA = 90"/,
     4          ITY4/"NOZZLE PRESSURE PERTURBATION, THETA = 0"/,
     5          ITY5/"NOZZLE AXIAL VELOCITY, THETA = 0"/,
     6          ITY6/"NOZZLE B.C. (RE(-GAMMA*Y*PHIT)) AT THETA = 0"/,
     7          ITP/"PRESSURE PEAKS"/
     8          MTITL1/"AMPLITUDE OF 1T MODE"/
     9          MTITL2/"AMPLITUDE OF 2T MODE"/
     1          MTITL3/"AMPLITUDE OF 1R MODE"/
     2          PRTITL/"PRESSURE AMPLITUDE OF 1T MODE"/
C
      LAST = 250
      ERR = 0.001
      TDEL = 10.0
      NPT = 0
      AA(1) = 0.0
      AA(2) = 0.5
      AA(3) = 0.5
      AA(4) = 1.0
      PI = 3.1415927
      READ (5,5003) NOUTCF, NOZNL2
C
C     ************* COEFFICIENT INPUT SECTION *************************
C
C     THIS VERSION OF LCYC3D READS THE COEFFICIENT DATA FROM
C     A FASTRAND FILE GENERATED BY PROGRAM COEFFS3D.  TO READ
C     THIS DATA FROM CARDS, USE READ (5,XXXX) INSTEAD OF
C     READ (9,XXXX) IN THIS SECTION.
C
C     INPUT OF MOTOR PARAMETERS AND NUMBER OF TERMS.
      READ (9,5001)  GAMMA, UE, ZE, ZCOMB, NDROPS, NJMAX, NOZNL1
      WRITE (6,6001)  GAMMA, UE, ZE, ZCOMB, NJMAX
      IF (NDROPS .EQ. 0)  WRITE (6,6030)
      IF (NDROPS .EQ. 1)  WRITE (6,6031)
      IF (NOZNL2 .EQ. 0)  WRITE (6,6032)
      IF (NOZNL2 .EQ. 1)  WRITE (6,6033)
      NU = 2 * NJMAX
      JMX = NJMAX/2
      BLD = 0.5 * ZE
C
      WRITE (6,6002)
C
C     INPUT OF DESCRIPTION OF SERIES EXPANSION.
      DO 10 K = 1, JMX
      READ (9,5002) NJ, L(NJ), M(NJ), N(NJ), NS(NJ), S(NJ), SJ(NJ),
     1               NAME(NJ)
```

```
      WRITE (6,6003)  NAME(NJ), NJ, L(NJ), M(NJ), N(NJ), NS(NJ),
     1                S(NJ), SJ(NJ)
   10 CONTINUE
C
      WRITE (6,6010)
      DO 15 K = 1, JMX
      READ (9,5010) J, YNOZ(J), B(J)
      WRITE (6,6015) J, YNOZ(J), B(J)
      NJ = (2 * J) - 1
      YR(NJ) = REAL(YNOZ(J))
      YI(NJ) = AIMAG(YNOZ(J))
      YR(NJ+1) = YR(NJ)
      YI(NJ+1) = YI(NJ)
   15 CONTINUE
      IF (NOZNL1 .NE. 1) GO TO 815
      WRITE (6,6034)
      DO 820 K = 1, JMX
      READ (9,5011) J, GNOZ(J)
      WRITE (6,6035) J, GNOZ(J)
  820 CONTINUE
  815 CONTINUE
C
C     CALCULATE THE NUMBER OF TYPES OF LINEAR COEFFICIENTS.
      NCOEFF = 4
      IF (NOZNL1 .EQ. 1) NCOEFF = 5
      NCFM1 = NCOEFF -1
C
C     ZERO LINEAR COEFFICIENT ARRAYS.
      DO 20 KC = 1, NCFM1
      DO 20 NJ = 1, MX2
      DO 20 NP = 1, MX2
      C(KC,NJ,NP) = 0.0
      CP(KC,NJ,NP) = 0.0
   20 CONTINUE
C
C     ZERO NONLINEAR COEFFICIENT ARRAY.
      DO 30 NJ = 1, MX2
      DO 30 NPQ = 1, MX2SQ
      D(NJ,NPQ) = 0.0
   30 CONTINUE
C
C     INPUT OF LINEAR COEFFICIENTS.
      DO 40 KC = 1, NCFM1
      READ (9,5003) KMAX
      IF (NOUTCF .GT. 0) WRITE (6,6004) KC, KMAX
      IF (KMAX .EQ. 0)  GO TO 40
      DO 45 K = 1, KMAX
      READ (9,5004)  NJ, NP, CP(KC,NJ,NP)
      IF (NOUTCF .GT. 0)  WRITE (6,6005) KC, NJ, NP, CP(KC,NJ,NP)
   45 CONTINUE
   40 CONTINUE
C
C
C     INPUT OF NONLINEAR COEFFICIENTS.
      READ (9,5003) NLMAX
      IF (NOUTCF .EQ. 2)  WRITE (6,6006) NLMAX
      IF (NLMAX .EQ. 0)  GO TO 50
      DO 52 NJ = 1, MX2
      KPQMAX(NJ) = 0
   52 CONTINUE
      DO 55 K = 1, NLMAX
      READ (9,5005)  NJ, NP, NQ, DT
      IF (NOUTCF .EQ. 2)  WRITE (6,6007)  NJ, NP, NQ, DT
      KPQMAX(NJ) = KPQMAX(NJ) + 1
      KPQ = KPQMAX(NJ)
      IDP(NJ,KPQ) = NP
      IDQ(NJ,KPQ) = NQ
      D(NJ,KPQ) = DT
   55 CONTINUE
   50 CONTINUE
C
C     ************* PRESSURE COEFFICIENT SECTION *********************
C
C     CALCULATE SPATIAL COORDINATES FOR PRESSURE COMPUTATION.
      DO 51 NPRES = 1, 3
      Z(NPRES) = 0.0
      RTHETA = NPRES - 1
      ANGLE(NPRES) = RTHETA * 45.0
      THETA(NPRES) = RTHETA * PI/4.0
      Z(NPRES + 3) = ZE
      ANGLE(NPRES + 3) = ANGLE(NPRES)
      THETA(NPRES + 3) = THETA(NPRES)
   51 CONTINUE
C
C     CALCULATE COEFFICIENTS FOR PRESSURE TIME HISTORIES.
      DO 53 NPRES = 1, 6
      DO 53 J = 1, JMX
      NP = (2 * J) - 1
      Z1 = Z(NPRES)
      ANG = THETA(NPRES)
      CALL PHICFS(J,Z1,ANG,C1,C2,C3)
      IF (NPRES .EQ. 4)  CPHIT(J) = C1
      CFT(NPRES,NP) = REAL(C1)
      CFT(NPRES,NP+1) = -AIMAG(C1)
      CFTH(NPRES,NP) = REAL(C2)
      CFTH(NPRES,NP+1) = -AIMAG(C2)
      CFZ(NPRES,NP) = REAL(C3)
      CFZ(NPRES,NP+1) = -AIMAG(C3)
   53 CONTINUE
C
      CI = (0.0,1.0)
      CAXI = GAMMA * CCOSH(CI * B(1) * ZE)
      CAXIR = REAL(CAXI)
      CAXII = AIMAG(CAXI)
C
C     OUTPUT OF COEFFICIENTS FOR PRESSURE TIME HISTORIES.
      WRITE (6,6020)
      DO 56 NPRES = 1, 6
      WRITE (6,6014)
      DO 56 J = 1, NJMAX
      WRITE (6,6021)            J, Z(NPRES), ANGLE(NPRES),
     1                 CFT(NPRES,J), CFTH(NPRES,J), CFZ(NPRES,J)
   56 CONTINUE
C
C     ************* DATA INPUT SECTION *********************************
C
      READ (5,5000) TITLE
C
C     ZERO INITIAL VALUE AND FREQUENCY ARRAYS.
    5 DO 57 K = 1, NJMAX
      AS(K) = 0.0
      AC(K) = 0.0
      FRQ1(K) = 0.0
   57 CONTINUE
C
C
C     READ COMBUSTION AND CONTROL PARAMETERS.
      READ (5,5006, END = 300) EN, TAU, H, TSTART, TQUIT
C
C     READ CONTROL NUMBERS.
      READ (5,5008)  NTEST, JMODE, NLOC, NTERMS, NFZ, NOUT, ICTYPE
      JMODE = (2 * JMODE) - 1
      JFMODE = JMODE + NJMAX
      IF (NOZNL2 .NE. 1) GO TO 825
      FREQ = S(1)
      KFREQ(1) = 1
      KFREQ(2) = 2
      KFREQ(3) = 2
      DO 830 K = 1, JMX
      WKF(J) = FREQ * KFREQ(J)
  830 CONTINUE
  825 CONTINUE
C
      IF (NOUT .GT. 0)  NFT = 1
      IF (NOUT .EQ. 0)  GO TO 9
C     READ DATA FOR SETTING UP PLOTS.
      READ (5,5009) YHI(1), YHI(5), YLAB(1), YLAB(5)
      READ (5,5008) ITICY(1), ITICY(5), NFIRST, NOMIT
      READ (5,5014) MDPLOT
      MDPLTL = 0
      DO 320 K = 1, JMX
      MDPLTL = MDPLTL + MDPLOT(K)
  320 CONTINUE
      IF (MDPLTL .EQ. 0) GO TO 9
      READ (5,5015) YHIMD, YLABMD, ITICMD
      YLOMD = - YHIMD
C
C     ************ INITIAL AMPLITUDES SECTION ***************************
C
    9 DO 58 K = 1, NTERMS
C
C     INPUT INITIAL AMPLITUDES FOR F-FUNCTIONS.
      READ (5,5007)  J, AST, ACT
      NJ = (2 * J) - 1
      AS(NJ) = AST
      AC(NJ) = ACT
C
C     CALCULATE FREQUENCY AND DAMPING.
      IF (ICTYPE .EQ. 2) GO TO 584
      RL = L(J)
      AX = RL * PI/ZE
      AXSQ = AX * AX
      SSQ = S(J) * S(J)
      FRQ1(NJ) = SQRT(SSQ + AXSQ)
      DMP1(NJ) = 0.0
      GO TO 586
  584 LONG = L(J)
      SMN = S(J)
      READ (5,5099) DAMP,FREQ
      DMP1(NJ) = DAMP
      FRQ1(NJ) = FREQ
  586 CONTINUE
      FRQ1(NJ+1) = FRQ1(NJ)
      DMP1(NJ+1) = DMP1(NJ)
C
      IF (ICTYPE .EQ. 2) GO TO 582
C     CALCULATE INITIAL AMPLITUDES FOR G-FUNCTIONS.
C
      IF (FRQ1(NJ)) 58, 58, 581
  581 GYRU = GAMMA*YR(NJ)*UE
      GYIF = GAMMA*YI(NJ)*FRQ1(NJ)
      GYRF = GAMMA*YR(NJ)*FRQ1(NJ)
      GYIU = GAMMA*YI(NJ)*UE
C
      NPRES = 4
      IF (NS(J) .EQ. 1)   NPRES = 6
C
      A1 = (1.0 + GYRU)*CFZ(NPRES,NJ+1)
     1     - GYIF*CFT(NPRES,NJ+1)
      A2 = GYRF*CFT(NPRES,NJ+1) + GYIU*CFZ(NPRES,NJ+1)
      A3 = -(1.0 + GYRU)*CFZ(NPRES,NJ) + GYIF*CFT(NPRES,NJ)
      A4 = GYRF*CFT(NPRES,NJ) + GYIU*CFZ(NPRES,NJ)
C
      DET = A1*A1 + A2*A2
      IF (DET .LT. 0.0000001) GO TO 583
      R1 = A3*AC(NJ) - A4*AS(NJ)
      R2 = -A4*AC(NJ) - A3*AS(NJ)
C
      AC(NJ+1) = (R1*A1 + R2*A2)/DET
      AS(NJ+1) = -(R2*A1 - R1*A2)/DET
      GO TO 58
  583 AC(NJ+1) = -AS(NJ)
      AS(NJ+1) = AC(NJ)
      GO TO 58
C
  582 ARG = FRQ1(NJ) * TAU
      FSIN = SIN(ARG)
      FCOS = 1. - COS(ARG)
      FSQ = FRQ1(NJ) * FRQ1(NJ)
      DSQ = DMP1(NJ) * DMP1(NJ)
      A1 = DSQ - FSQ + DMP1(NJ) * (CP(2,NJ,NJ)
     1        - EN * CP(3,NJ,NJ) * FCOS)
     2        + EN * CP(3,NJ,NJ) * FRQ1(NJ) * FSIN
     3        + CP(1,NJ,NJ)
      A2 = (2.0 * DMP1(NJ)   +   CP(2,NJ,NJ)
     1        - EN * CP(3,NJ,NJ) * FCOS) * FRQ1(NJ)
     2        - EN * CP(3,NJ,NJ) * DMP1(NJ) * FSIN
      A3 = CP(2,NJ,NJ+1) * DMP1(NJ) + CP(1,NJ,NJ+1)
      A4 = CP(2,NJ,NJ+1) * FRQ1(NJ)
      DEN = A3*A3 + A4*A4
      IF (DEN .LT. 0.0000001) GO TO 585
      R1 = A1*A3 +A2*A4
      R2 = A1*A4 - A2*A3
      AC(NJ+1) = (-R1*AC(NJ) + R2*AS(NJ))/DEN
      AS(NJ+1) = -(R2*AC(NJ) + R1*AS(NJ))/DEN
      GO TO 58
  585 AC(NJ+1) = -AS(NJ)
      AS(NJ+1) = AC(NJ)
C
   58 CONTINUE
C
C
C     OUTPUT OF INITIAL AMPLITUDES.
      WRITE (6,6016)
      DO 590 J = 1, NJMAX
      IF (AS(J))  591, 592, 591
  592 IF (AC(J))  591, 590, 591
  591 WRITE (6,6017) J, DMP1(J), FRQ1(J), AC(J), AS(J)
  590 CONTINUE
      IF (NTEST .EQ. 0)  WRITE (6,6025)
      IF (NTEST .EQ. 1)  WRITE (6,6026)
      IF (NPZ .EQ. 1) WRITE (6,6028)
      IF (NOUT .GE. 1)  WRITE (6,6027)
C
C     ************* LINEAR COEFFICIENTS SECTION ***********************
C
      DO 59 KC = 1, NCFM1
      DO 59 NJ = 1, MX2
      KPMAX(KC,NJ) = 0
   59 CONTINUE
C
      IF (NPZ .EQ. 0) GO TO 605
      DO 602 J = 1, JMX
      NJ = (2 * J) - 1
      RL = L(J)
      AX = RL * PI/ZE
      AXSQ = AX * AX
      SSQ = S(J) * S(J)
      OMEGA = SQRT(SSQ + AXSQ)
      TAUCUT(NJ) = 2.0 * PI/OMEGA
      TAUCUT(NJ+1) = TAUCUT(NJ)
  602 CONTINUE
C
      DO 604 NJ = 1, NJMAX
      DO 604 NP = 1, NJMAX
      IF (TAU .GT. TAUCUT(NP))  CP(3,NJ,NP) = 0.0
  604 CONTINUE
C
C     COMPUTE LINEAR COEFFICIENTS FOR GIVEN VALUES OF EN AND TAU.
  605 DO 60 NJ = 1, NJMAX
      DO 60 NP = 1, NJMAX
      CT = CP(1,NJ,NP)
      IF (CT) 61, 62, 61
   61 KPMAX(1,NJ) = KPMAX(1,NJ) + 1
      KP = KPMAX(1,NJ)
      IC(1,NJ,KP) = NP
      C(1,NJ,KP) = CT
   62 CT = CP(2,NJ,NP) - EN*CP(3,NJ,NP)
      IF (CT) 63, 64, 63
   63 KPMAX(2,NJ) = KPMAX(2,NJ) + 1
      KP = KPMAX(2,NJ)
      IC(2,NJ,KP) = NP
      C(2,NJ,KP) = CT
   64 CT = EN * CP(3,NJ,NP)
      IF (CT) 65, 66, 65
   65 KPMAX(3,NJ) = KPMAX(3,NJ) + 1
      KP = KPMAX(3,NJ)
      IC(3,NJ,KP) = NP
      C(3,NJ,KP) = CT
   66 IF (NOZNL2 .NE. 1) GO TO 60
      CT = CP(4,NJ,NP)
      IF (CT) 67,60,67
   67 KPMAX(4,NJ) = KPMAX(4,NJ) + 1
      KP = KPMAX(4,NJ)
      IC(4,NJ,KP) = NP
      C(4,NJ,KP) = CT
   60 CONTINUE
C
C     ************* STEP-SIZE COMPUTATION ******************************
C
      NDIV = 1.0 + TAU/H
      RN = NDIV
      H = TAU/RN
      H6 = H/6.0
C
C     ************* INITIAL VALUES SECTION ******************************
C
      WRITE (6,6008)  EN, TAU, GAMMA, UE, RLD
      WRITE (6,6009)
      WRITE (6,6022) (ANGLE(J), J = 1,6), (ANGLE(J), J = 1,3)
      WRITE (6,6012)
      NP1 = NDIV + 1
      DO 70 I = 1, NP1
      NSTEP = I - NP1
      RSTEP = NSTEP
      TIME = RSTEP * H
      TI(I) = TIME
      DO 75 J = 1, NJMAX
      JP = J + NJMAX
      IF (AC(J)) 751, 753, 751
  753 IF (AS(J)) 751, 752, 751
  752 U(I,J) = 0.0
      U(I,JP) = 0.0
      GO TO 75
  751 ARG = FRQ1(J) * TIME
      FSIN = SIN(ARG)
      FCOS = COS(ARG)
      FEXP = EXP(DMP1(J)*TIME)
      U(I,J) = (AS(J)*FSIN + AC(J)*FCOS) * FEXP
      U(I,JP) = ((AS(J) * FCOS) - (AC(J) * FSIN)) * FRQ1(J) * FEXP
     1          + DMP1(J) * U(I,J)
   75 CONTINUE
C     CALCULATE INITIAL VALUES OF PRESSURE AND VELOCITY.
      DO 704 NPRES = 1, 6
      DO 702 J = 1, NJMAX
      COEF(1,J) = CFT(NPRES,J)
      COEF(2,J) = CFTH(NPRES,J)
      COEF(3,J) = CFZ(NPRES,J)
  702 CONTINUE
      DO 703 J = 1, NU
      Y(J) = U(I,J)
  703 CONTINUE
      UBAR = 0.0
      IF (NPRES .GT. 3) UBAR = UE
      UMS = 0.0
      IF ((NDROPS.EQ.1) .AND. (NPRES.LT.4)) UMS = UE/(ZE*ZCOMB)
      CALL PRSVEL(UBAR,UMS,Y,P,VTH,VZ)
      PRESS(NPRES) = P
      IF (NPRES .GT. 3)  AXVEL(NPRES - 3) = VZ
  704 CONTINUE
      PRS(I) = PRESS(NLOC)
C
C     CALCULATE INITIAL VALUES OF NOZZLE B.C.
      CSUM = (0.0,0.0)
      DO 710 J = 1, JMX
      JP = NJMAX + (2 * J) - 1
      FT = Y(JP)
      GT = Y(JP+1)
      A = CMPLX(FT,GT)
      CSUM = CSUM + YNOZ(J) * CPHIT(J) * A
  710 CONTINUE
      SUM = REAL(CSUM)
      YPHI = -GAMMA * SUM
      WRITE (6,6011)  NSTEP, TIME, (PRESS(J), J = 1,6),
     1                (AXVEL(J), J = 1,3), YPHI
   70 CONTINUE
C
      WRITE (6,6008)  EN, TAU, GAMMA, UE, RLD
      WRITE (6,6022) (ANGLE(J), J = 1,6), (ANGLE(J), J = 1,3)
C
C     ************* INITIALIZE CONTROL NUMBERS *************************
C
      LINE = 8
      K = 0
      MAXNO = 0
      MAXP = 0
      IF (NOUT .EQ. 0)  GO TO 100
      JPLOT = 0
      TMIN = TSTART
      TMAX = TSTART + TDEL
      YLO(1) = -YHI(1)
      DO 90 J = 2,4
      YHI(J) = YHI(1)
      YLO(J) = YLO(1)
      YLAB(J) = YLAB(1)
      ITICY(J) = ITICY(1)
   90 CONTINUE
      YLO(5) = -YHI(5)
      YHI(6) = YHI(5)
      YLO(6) = YLO(5)
      YLAB(6) = YLAB(5)
      ITICY(6) = ITICY(5)
C
C     ************* NUMERICAL CALCULATIONS SECTION *****************
C
  100 I = NP1
C
C     RUNGE-KUTTA INTEGRATION SCHEME.
  105 NSTEP = (I - NP1 + (LAST - NP1) * K)
      RSTEP = NSTEP
      TIME = RSTEP * H
      TI(I) = TIME
      DO 110 J = 1, NJMAX
      JP = J + NJMAX
      RV(J,1) = U(I-NDIV,JP)
      RV(J,4) = U(I-NDIV+1,JP)
      RV(J,2) = 0.375*RV(J,1) + 0.75*RV(J,4) - 0.125*U(I-NDIV+2,JP)
      RV(J,3) = RV(J,2)
  110 CONTINUE
      IF (NOZNL2 .NE. 1)  GO TO 835
      DO 840 II = 1,4
      TZ = TIME + AA(II)*H
      DO 840 J = 1,JMX
      JODD = 2*J - 1
      JEVEN = 2*J
      EXTRA(JODD,II) = COS(WKP(J)*TZ)
      EXTRA(JEVEN,II) = SIN(WKP(J)*TZ)
  840 CONTINUE
  835 CONTINUE
      DO 120 J = 1, NU
      Y(J) = U(I,J)
  120 CONTINUE
      CALL RHS(NU,1,Y,YP)
      DO 130 J = 1, NU
      FZ(1,J) = YP(J)
  130 CONTINUE
      DO 140 II = 2,4
      DO 144 J = 1, NU
      UZ(J) = Y(J) + AA(II) * H * FZ(II-1,J)
  144 CONTINUE
```

```
      CALL RHS(NU,II,UZ,YP)
      DO 148 J = 1, NU
      FZ(II,J) = YP(J)
  148 CONTINUE
  140 CONTINUE
      DO 150 J = 1, NU
      U(I+1,J) = Y(J) + (FZ(1,J)+2.0*(FZ(2,J)+FZ(3,J)) + FZ(4,J)) * H6
  150 CONTINUE
C
C     CALCULATE PRESSURE TIME HISTORIES.
      DO 154 NPRES = 1, 6
      DO 152 J = 1, NJMAX
      COEF(1,J) = CFT(NPRES,J)
      COEF(2,J) = CFTH(NPRES,J)
      COEF(3,J) = CFZ(NPRES,J)
  152 CONTINUE
      UBAR = 0.0
      IF (NPRES .GT. 3) UBAR = UE
      UMS = 0.0
      IF ((NDROPS.EQ.1) .AND. (NPRES.LT.4)) UMS = UE/(ZE*ZCOMB)
      CALL PRSVEL(UBAR,UMS,Y,P,VTH,VZ)
      PRESS(NPRES) = P
      IF (NPRES .GT. 3)  AXVEL(NPRES - 3) = VZ
  154 CONTINUE
      PRS(I) = PRESS(NLOC)
C
C     CALCULATE VALUES OF NOZZLE B.C.
      CSUM = (0.0,0.0)
      DO 650 J = 1, JMX
      JP = NJMAX + (2 * J) - 1
      FT = Y(JP)
      GT = Y(JP+1)
      A = CMPLX(FT,GT)
      CSUM = CSUM + YNOZ(J) * CPHIT(J) * A
  650 CONTINUE
      SUM = REAL(CSUM)
      YPHI = -GAMMA * SUM
C
C
C     DETERMINE MAXIMA AND MINIMA OF PRINCIPAL MODE-AMPLITUDE
C     FUNCTION FOR USE IN DETERMINING LIMIT-CYCLE BEHAVIOR.
      IF (U(I,JPMODE) * U(I+1,JPMODE)) 170, 170, 160
  170 PDEN = U(I,JPMODE) - U(I+1,JPMODE)
      IF (PDEN) 171, 160, 171
  171 PP = U(I,JPMODE)/PDEN
      PA = (PP - 1.0) * PP * 0.5
      PB = 1.0 - (PP * PP)
      PC = (PP + 1.0) * PP * 0.5
      MAXNO = MAXNO + 1
      UMAX(MAXNO) = PA*U(I-1,JMODE) + PB*U(I,JMODE) + PC*U(I+1,JMODE)
      IF (MAXNO .GE. 500)  GO TO 250
  160 CONTINUE
C
C     DETERMINE MAXIMUM AND MINIMUM PRESSURE AT LOCATION SPECIFIED
C     BY NLOC.
```

```
      DPL = PRS(I) - PRS(I-1)
      DPS = PRS(I-1) - PRS(I-2)
      IF (DPL*DPS) 173, 173, 175
  173 PNUM = PRS(I-2) - PRS(I)
      PDEN = 2.0 * (PRS(I-2) + PRS(I) - 2.0*PRS(I-1))
      IF (PDEN) 174, 175, 174
  174 PP = PNUM/PDEN
      PA = (PP - 1.0) * PP * 0.5
      PB = 1.0 - (PP * PP)
      PC = (PP + 1.0) * PP * 0.5
      MAXP = MAXP + 1
      PMAX(MAXP) = PA*PRS(I-2) + PB*PRS(I-1) + PC*PRS(I)
      TIMAX(MAXP) = TI(I-1) + PP*H
      IF (MAXP .GE. 500)  GO TO 250
  175 CONTINUE
C
      IF (NTEST .EQ. 1)  GO TO 155
      IF (TIME .LT. TSTART)  GO TO 155
      IF ((NOUT .EQ. 0) .OR. (NOUT .GT. 6))  GO TO 156
C
C     ************* TIME HISTORY PLOTTING SECTION *********************
C
      IF (TMAX .GT. TQUIT)  GO TO 156
      IF ((TIME .GT. TMAX) .OR. (JPLOT .GE. 500))  GO TO 1000
C
      JPLOT = JPLOT + 1
C
C     FILL TIME ARRAY FOR PLOTTING.
      TPLOT(JPLOT) = TIME
C
C     FILL INJECTOR PRESSURE ARRAYS FOR PLOTTING (THETA = 0, 45, 90)
      DO 1001 J = 1,3
      YPLOT(J,JPLOT) = PRESS(J)
 1001 CONTINUE
C
C     FILL NOZZLE PRESSURE ARRAY FOR PLOTTING (THETA = 0)
      YPLOT(4,JPLOT) = PRESS(4)
C
C     FILL NOZZLE AXIAL VELOCITY ARRAY FOR PLOTTING (THETA = 0)
      YPLOT(5,JPLOT) = AXVEL(1)
C
C     FILL NOZZLE B.C. ARRAY FOR PLOTTING (THETA = 0).
      YPLOT(6,JPLOT) = YPHI
C
      IF (MDPLTL .EQ. 0) GO TO 156
C
C     FILL MODE AMPLITUDE ARRAYS FOR PLOTTING.
      DO 322 J = 1, JMX
      IF (MDPLOT(J) .EQ. 0) GO TO 322
      J12 = 2*J - 1
      UPLOT(J,JPLOT) = U(I,J12)
  322 CONTINUE
C
      J1T1 = NJMAX + 1
      J1T2 = NJMAX + 2
      PRIT(JPLOT) = CAXIR*U(I,JIT1) - CAXII*U(I,JIT2)
C
      GO TO 156
C
 1000 NUM = JPLOT
C
C     PLOT TIME HISTORIES.
C
      DO 1020 NPLOT = NFIRST, NOUT
C
      JPLOT = 0
C
C     ASSIGN PLOTTING PARAMETERS.
      YMIN = YLO(NPLOT)
      YMAX = YHI(NPLOT)
      NTICY = ITICY(NPLOT)
      DELY = YLAB(NPLOT)
C
C     ELIMINATE POINTS THAT ARE OUT OF THE ORDINATE RANGE.
      DO 1010 J = 1, NUM
      IF ((YPLOT(NPLOT,J) .LT. YMIN) .OR. (YPLOT(NPLOT,J) .GT. YMAX))
     1   GO TO 1010
      JPLOT = JPLOT + 1
      DUMMYT(JPLOT) = TPLOT(J)
      DUMMYY(JPLOT) = YPLOT(NPLOT,J)
 1010 CONTINUE
C
      IF (JPLOT .EQ. 0)  GO TO 1020
      GO TO (1011,1012,1013,1014,1015,1016), NPLOT
C
C     PLOT INJECTOR PRESSURE AT THETA = 0 DEGREES.
 1011 CALL GRAPHS(IBUF,3000,4,JPLOT,51,NTICY,TMAX,YMAX,TMIN,YMIN,
     1               ITT,ITY1,21,41,DUMMYT,DUMMYY,2.0,DELY,TITLE)
      GO TO 1020
C
C     PLOT INJECTOR PRESSURE AT THETA = 45 DEGREES.
 1012 IF (M(JMODE) .EQ. 0)  GO TO 1020
      CALL GRAPHS(IBUF,3000,4,JPLOT,51,NTICY,TMAX,YMAX,TMIN,YMIN,
     1               ITT,ITY2,21,42,DUMMYT,DUMMYY,2.0,DELY,TITLE)
      GO TO 1020
C
C     PLOT INJECTOR PRESSURE AT THETA = 90 DEGREES.
 1013 IF (M(JMODE) .EQ. 0)  GO TO 1020
      CALL GRAPHS(IBUF,3000,4,JPLOT,51,NTICY,TMAX,YMAX,TMIN,YMIN,
     1               ITT,ITY3,21,42,DUMMYT,DUMMYY,2.0,DELY,TITLE)
      GO TO 1020
C
C     PLOT NOZZLE PRESSURE AT THETA = 0 DEGREES.
 1014 CALL GRAPHS(IBUF,3000,4,JPLOT,51,NTICY,TMAX,YMAX,TMIN,YMIN,
     1               ITT,ITY4,21,39,DUMMYT,DUMMYY,2.0,DELY,TITLE)
      GO TO 1020
C
C     PLOT NOZZLE AXIAL VELOCITY AT THETA = 0 DEGREES.
 1015 CALL GRAPHS(IBUF,3000,4,JPLOT,51,NTICY,TMAX,YMAX,TMIN,YMIN,
     1               ITT,ITY5,21,32,DUMMYT,DUMMYY,2.0,DELY,TITLE)
```

```
      GO TO 1020
C
C     PLOT NOZZLE B.C. AT THETA = 0 DEGREES.
 1016 CALL GRAPHS(IBUF,3000,4,JPLOT,51,NTICY,TMAX,YMAX,TMIN,YMIN,
     1              ITT,ITY6,21,44,DUMMYT,DUMMYY,2.0,DELY,TITLE)
C
 1020 CONTINUE
C
      IF (MDPLTL .EQ. 0) GO TO 330
      DO 324 NPLOT = 1, JMX
      IF (MDPLOT(NPLOT) .EQ. 0) GO TO 324
      JPLOT = 0
      DO 328 J123 = 1, 4
      IF (NPLOT .EQ. 1) MTITL(J123) = MTITL1(J123)
      IF (NPLOT .EQ. 2) MTITL(J123) = MTITL2(J123)
      IF (NPLOT .EQ. 3) MTITL(J123) = MTITL3(J123)
  328 CONTINUE
C
      DO 326 J = 1, NUM
      IF ((UPLOT(NPLOT,J) .LT. YLOMD) .OR. (UPLOT(NPLOT,J)
     1    .GT. YHIMD)) GO TO 326
      JPLOT = JPLOT + 1
      DUMMYT(JPLOT) = TPLOT(J)
      DUMMYY(JPLOT) = UPLOT(NPLOT,J)
  326 CONTINUE
      IF (JPLOT .EQ. 0) GO TO 324
C
C     PLOT AMPLITUDES OF DIFFERENT MODES.
      CALL GRAPHS(IBUF,3000,4,JPLOT,51,ITICMD,TMAX,YHIMD,TMIN,
     1            YLOMD,ITT,MTITL,21,20,DUMMYT,DUMMYY,2.0,YLABML,TITLE)
  324 CONTINUE
C
      IF (MDPLOT(4) .EQ. 0) GO TO 330
      JPLOT = 0
      DO 332 J = 1, NUM
      IF ((PR1T(J) .LT. YLOMD) .OR. (PR1T(J) .GT. YHIMD)) GO TO 332
      JPLOT = JPLOT + 1
      DUMMYT(JPLOT) = TPLOT(J)
      DUMMYY(JPLOT) = PR1T(J)
  332 CONTINUE
      IF (JPLOT .EQ. 0) GO TO 330
C
C     PLOT PRESSURE AMPLITUDE OF 1T MODE.
      CALL GRAPHS(IBUF,3000,4,JPLOT,51,ITICMD,TMAX,YHIMD,TMIN,
     1     YLOMD,ITT,PRTITL,21,29,DUMMYT,DUMMYY,2.0,YLABMD,TITLE)
  330 CONTINUE
C
C     REASSIGN PLOTTING PARAMETERS FOR NEXT SET OF PLOTS.
      JPLOT = 0
      TMIN = TMAX
      TMAX = TMAX + TDEL
C
C     ************* TIME HISTORY PRINTED OUTPUT SECTION ***************
C
  156 WRITE (6,6011)  NSTEP, TIME, (PRESS(J), J = 1,6),
     1                 (AXVEL(J), J = 1,3), YPHI
      LINE = LINE + 1
  157 IF (TIME .GT. TQUIT)  GO TO 250
      IF (LINE .LT. 52)  GO TO 155
      WRITE (6,6013)
      WRITE (6,6022) (ANGLE(J), J = 1,6), (ANGLE(J), J = 1,3)
      LINE = 4
C
  155 I = I + 1
      IF (I .LT. LAST)  GO TO 105
C
C     ************* LIMIT-CYCLE SECTION ********************************
C
C     TEST FOR LIMIT CYCLE.
      K = K + 1
      IF ((NTEST .EQ. 0) .OR. (MAXNO .LT. 80))  GO TO 190
      UTOT = 0.0
      DO 180 J = 0, 3
      JMAX = MAXNO - J
      UTOT = UTOT + ABS(UMAX(JMAX))
  180 CONTINUE
      UAVG(K) = UTOT/4.0
      IF (K .EQ. 1)  GO TO 190
      CHANGE = UAVG(K) - UAVG(K-1)
      ABSCHG = ABS(CHANGE/UAVG(K))
      IF (ABSCHG .GT. ERR)  GO TO 190
      TM = TIME/2.0
      ITM = TM
      ITM = 2*ITM + 2
      TM = ITM
      TSTART = TM + TSTART
      TQUIT = TM + TQUIT
      TMIN = TSTART
      TMAX = TSTART + TDEL
      NTEST = 0
C
C     RE-ASSIGN ARRAYS.
  190 DO 200 I = 1, NP1
      ILAST = LAST - NP1 + I
      PRS(I) = PRS(ILAST)
      TI(I) = TI(ILAST)
      DO 200 J = 1, NU
      U(I,J) = U(ILAST,J)
  200 CONTINUE
      GO TO 100
C
C
C     ************* PRESSURE MAXIMA AND MINIMA PRINTOUT ****************
C
  250 WRITE (6,6023) Z(NLOC), ANGLE(NLOC), MAXP
      LINE = 4
      DO 255 JST = 1, MAXP, 8
      JSTART = JST
      JSTOP = JST + 7
      IF (JSTOP .GT. MAXP) JSTOP = MAXP
      WRITE (6,6024) (PMAX(J), J = JSTART, JSTOP)
      WRITE (6,6024) (TIMAX(J), J = JSTART, JSTOP)
      WRITE (6,6014)
      LINE = LINE + 3
      IF (LINE .LT. 52)  GO TO 255
      LINE = 0
      WRITE (6,6013)
  255 CONTINUE
      IF ((NOUT .EQ. 0) .OR. (NOMIT .EQ. 1))  GO TO 5
C
C     ************* PRESSURE MAXIMA PLOTTING SECTION *******************
C
C     DETERMINE LARGEST VALUE OF PMAX.
      AMPMAX = 0.0
      DO 260 J = 1, MAXP
      IF (PMAX(J) .LT. AMPMAX)  GO TO 260
      AMPMAX = PMAX(J)
  260 CONTINUE
C
C     RANGE OF PLOT AND COORDINATE LABELING.
      ITM = AMPMAX + 1.0
      AMPMAX = ITM
      ITM = 1.0 + TIMAX(MAXP)/50.0
      TMAX = ITM * 50
      DELX = TMAX/10.0
      DELY = AMPMAX/10.0
C
C     ELIMINATE NEGATIVE VALUES.
      JPLOT = 0
      DO 262 J = 1, MAXP
      IF (PMAX(J)) 262, 264, 264
  264 JPLOT = JPLOT + 1
      DUMMYT(JPLOT) = TIMAX(J)
      DUMMYY(JPLOT) = PMAX(J)
  262 CONTINUE
C
C     PLOT VALUES.
      CALL GRAPHS(IBUF,3000,4,JPLOT,101,101,TMAX,AMPMAX,0.0,0.0,
     1            ITT,ITF,21,14,DUMMYT,DUMMYY,DELX,DELY,TITLE)
C
      GO TO 5
C
C     TURN OFF PLOTTING ROUTINE.
  300 IF (NPT .EQ. 1)  CALL SHPARG
C
C     ************* READ FORMAT SPECIFICATIONS *************************
C
 5000 FORMAT (12A6)
 5001 FORMAT (4F10.0,3I5)
 5002 FORMAT (5I5,2F10.5,1X,A4)
 5003 FORMAT (2I5)
 5004 FORMAT (2I5,F15.6)
 5005 FORMAT (3I5,F15.6)
 5006 FORMAT (5F10.0)
 5007 FORMAT (I5,2F10.0)
 5008 FORMAT (7I5)
 5009 FORMAT (7F10.0)
 5010 FORMAT (I5,4F10.5)
 5011 FORMAT (I5,2F10.5)
 5012 FORMAT (F10.0)
 5014 FORMAT (4I5)
 5015 FORMAT (2F10.0,I5)
 5099 FORMAT (2F10.0)
C
C     ************* WHITE FORMAT SPECIFICATIONS **********************
C
 6001 FORMAT (1H1,9H GAMMA = ,F5.3,5X,5HUE = ,F5.3,
     1         5X,5HZE = ,F8.5,5X,8HZCOMB = ,F5.2,
     2         5X,8HNJMAX = ,I2//)
 6002 FORMAT (2X,29HNAME     J    L    M    N   NS,7X,3HSMN,3X,
     1         7HJM(SMN)/)
 6003 FORMAT (2X,A4,5I5,2F10.5)
 6004 FORMAT (1H0,26H NUMBER OF COEFFICIENTS C(,I1,10H,NJ,NP) IS,I5/)
 6005 FORMAT (2X,2HC(,I1,1H,,I2,1H,,I2,4H) = ,F10.5)
 6006 FORMAT (1H0,38H NUMBER OF COEFFICIENTS D(NJ,NP,NQ) IS,I5/)
 6007 FORMAT (2X,2HD(,I2,1H,,I2,1H,,I2,4H) = ,F10.5)
 6008 FORMAT(1H1,45H COMBUSTION PARAMETERS:  INTERACTION INDEX = ,F7.5,
     1         12X,11HTIME-LAG = ,F7.5/2X,17HMOTOR PARAMETERS:,19X,
     2         8HGAMMA = ,F7.5,23H     EXIT MACH NUMBER = ,F7.5,
     3         22H     LENGTH/DIAMETER = ,F7.5//)
 6009 FORMAT (2X,18HINITIAL CONDITIONS//)
 6010 FORMAT (1H0,5X,1HJ,8X,2HYR,8X,2HYI,7X,3HEPS,7X,3HETA//)
 6011 FORMAT (2X,I5,F12.5,10F10.5)
 6012 FORMAT (1H0)
 6013 FORMAT (1H1)
 6014 FORMAT (1H )
 6015 FORMAT (2X,I5,4F10.5)
 6016 FORMAT (1H1,36H INITIAL CONDITIONS ARE OF THE FORM://
     1         2X,49HU(I,J) =  AC(J)*COS(FREQ*T) + AS(J)*SIN(FREQ*T)),
     2         14H * EXP(DAMP*T)///6X,1HJ,8X,7HDAMPING,
     3         6X,9HFREQUENCY,10X,5HAC(J),10X,5HAS(J)//)
 6017 FORMAT (2X,I5,4F15.8/)
 6020 FORMAT (1H1,46H COEFFICIENTS FOR COMPUTATION OF WALL PRESSURE,
     1         10H WAVEFORMS///43X,27HCOEFFICIENTS IN SERIES FOR://
     2         22X,5HTHETA,10X,4HTIME,10X,5HTHETA,10X,5HAXIAL/
     3                       6X,1HJ,9X,1HZ,3X,9H(DEGREES),5X,10HDERIVATIVE,
     4         5X,10HDERIVATIVE,5X,10HDERIVATIVE//)
 6021 FORMAT (2X,I5,F10.3,F12.1,3F15.7)
 6022 FORMAT (26X,17HINJECTOR PRESSURE,14X,15HNOZZLE PRESSURE,
     1         12X,21HNOZZLE AXIAL VELOCITY/3X,4HSTEP,8X,4HTIME,
     2         F5.0,5H DEG.,F5.0,5H DEG.,F5.0,5H DEG.,
     3         F5.0,5H DEG.,F5.0,5H DEG.,F5.0,5H DEG.,
     4         F5.0,5H DEG.,F5.0,5H DEG.,F5.0,5H DEG.,6X,4HYPHI//)
 6023 FORMAT (1H1,38H PRESSURE MAXIMA AND MINIMA AT:    Z = ,F5.2,
     1         11H   THETA = ,F4.1/19H VALUES COMPUTED:  ,I3//)
 6024 FORMAT (1H ,7X,8F13.6)
 6025 FORMAT (2X//2X,37HTHE TRANSIENT BEHAVIOR IS CALCULATED.)
 6026 FORMAT (2X//2X,39HTHE LIMIT-CYCLE BEHAVIOR IS CALCULATED.)
 6027 FORMAT (2X//2X,33HTHIS RUN PRODUCES PLOTTED OUTPUT.)
 6028 FORMAT (2X//2X,"THE PHANTOM ZONES ARE ELIMINATED.")
```

```
6030 FORMAT (2X,"DROPLET MOMENTUM SOURCE IS NEGLECTED"/)
6031 FORMAT (2X,"DROPLET MOMENTUM SOURCE IS INCLUDED"/)
6032 FORMAT (2X,"NOZZLE NONLINEARITIES NEGLECTED"/)
6033 FORMAT (2X,"NOZZLE NONLINEARITIES INCLUDED"/)
6034 FORMAT (1H0,8X,1HJ,10X,2HGR,10X,2HGI//)
6035 FORMAT (5X,I5,2F12.5)
     END
```

```
      SUBROUTINE PHICFS(NP,Z,THETA,CT,CTH,CZ)
C
C     THIS SUBROUTINE COMPUTES THE COEFFICIENTS NEEDED TO
C     CALCULATE THE WALL PRESSURE PERTURBATION.
C
C     NP IS THE INDEX OF THE COMPLEX SERIES TERM.
C     Z IS THE AXIAL LOCATION.
C     THETA IS THE AZIMUTHAL LOCATION.
C     CT IS THE COEFFICIENT IN THE SERIES FOR THE TIME DERIVATIVE O
C     THE VELOCITY POTENTIAL.
C     CTH IS THE COEFFICIENT IN THE SERIES FOR THE THETA DERIVATIVE
C     OF THE VELOCITY POTENTIAL.
C     CZ IS THE COEFFICIENT IN THE SERIES FOR THE AXIAL DERIVATIVE
C     OF THE VELOCITY POTENTIAL.
C
      PARAMETER  MX = 5
      COMPLEX    CI, CZ, CAXI, CAXIZ, CRAD, CAZI, CAZITH,
     1           B(MX), CT, CTH, CZ
      COMMON       /BLK2/  M(MX), NS(MX), SJ(MX), B
C
      CI = (0.0,1.0)
      CZ = CMPLX(Z,0.0)
      CAXI = CCOSH(CI * B(NP) * CZ)
      CAXIZ = CI * B(NP) * CSINH(CI * B(NP) * CZ)
      CRAD = CMPLX(SJ(NP),0.0)
      EM = M(NP)
      ARG = EM * THETA
      FSIN = SIN(ARG)
      FCOS = COS(ARG)
      AZI = FCOS
      IF (NS(NP) .EQ. 1)   AZI = FSIN
      AZITH = EM * FCOS
      IF (NS(NP) .EQ. 2)   AZITH = -EM * FSIN
      CAZI = CMPLX(AZI,0.0)
      CAZITH = CMPLX(AZITH,0.0)
C
      CT = CAZI * CAXI * CRAD
      CTH = CAZITH * CAXI * CRAD
      CZ = CAZI * CAXIZ * CRAD
C
      RETURN
      END
      SUBROUTINE PRSVEL(UBAR,UMS,Y,P,VTH,VZ)
C
C     THIS SUBROUTINE COMPUTES THE WALL PRESSURE AND VELOCITY.
C
C     UBAR IS THE LOCAL AXIAL STEADY STATE MACH NUMBER.
C     UMS IS THE DERIVATIVE OF THE MACH NUMBER FOR THE CASE
C      WHEN DROPLET MOMENTUM SOURCES ARE INCLUDED.
C     Y IS THE ARRAY CONTAINING VALUES OF THE MODE-AMPLITUDE
C     FUNCTIONS AND THEIR DERIVATIVES.
C     P IS THE VALUE OF THE WALL PRESSURE PERTURBATION.
C     VTH IS THE TANGENTIAL COMPONENT OF VELOCITY AT THE WALL.
C     VZ IS THE AXIAL COMPONENT OF VELOCITY AT THE WALL.
C
      PARAMETER     MX2=10, MX4=20
      DIMENSION     Y(MX4), SUM(4), SUMSQ(3)
      COMMON        /BLK3/    NJMAX, NLMAX, GAMMA, COEF(3,MX2)
C
      DO 10 I = 1, 4
      SUM(I) = 0.0
   10 CONTINUE
C
      DO 20 I = 1, 4
      DO 20 J = 1, NJMAX
      JY = J
      IF (I .EQ. 1)  JY = J + NJMAX
      II = I
      IF (I .EQ. 4) II = 1
      SUM(I) = SUM(I) + Y(JY) * COEF(II,J)
   20 CONTINUE
C
      PLIN = SUM(1) + UBAR*SUM(3) + UMS*SUM(4)
      PNL = 0.0
      IF (NLMAX .EQ. 0)  GO TO 40
      DO 30 I = 1, 3
      SUMSQ(I) = SUM(I) * SUM(I)
   30 CONTINUE
      PNL = 0.5 * (SUMSQ(2) + SUMSQ(3) - SUMSQ(1))
C
   40 P = -GAMMA * (PLIN + PNL)
      VTH = SUM(2)
      VZ = SUM(3)
C
      RETURN
      END
      SUBROUTINE RHS(NU,II,U,UP)
C
      PARAMETER      MX=5, MX2=10, MX4=20, MX2SQ=100
      DIMENSION      U(NU), UP(NU)
      COMMON         RV(MX2,4), C(4,MX2,MX2), D(MX2,MX2SQ),
     1               KPMAX(4,MX2), IC(4,MX2,MX2), KPQMAX(MX2),
     2               IDP(MX2,MX2SQ), ILQ(MX2,MX2SQ)
      COMMON         /BLK3/    NJMAX, NLMAX, GAMMA, COEF(3,MX2
      COMMON         /NLTERM/ NOZNL2, EXTRA(MX2,4)
C
      DO 10 NJ = 1, NJMAX
      NJP = NJ + NJMAX
      UP(NJ) = U(NJP)
      SL1 = 0.0
      SL2 = 0.0
      SL3 = 0.0
      SL4 = 0.0
      SNL = 0.0
      MAX = KPMAX(1,NJ)
      IF (MAX .EQ. 0) GO TO 25
      DO 20 KP = 1, MAX
      NP = IC(1,NJ,KP)
      SL1 = SL1 + (C(1,NJ,KP) * U(NP))
   20 CONTINUE
   25 MAX = KPMAX(2,NJ)
      IF (MAX .EQ. 0) GO TO 35
      DO 30 KP = 1, MAX
      NPP = IC(2,NJ,KP) + NJMAX
      SL2 = SL2 + (C(2,NJ,KP) * U(NPP))
   30 CONTINUE
   35 MAX = KPMAX(3,NJ)
      IF (MAX .EQ. 0) GO TO 45
      DO 40 KP = 1, MAX
      NP = IC(3,NJ,KP)
      SL3 = SL3 + (C(3,NJ,KP) * RV(NP,II))
   40 CONTINUE
   45 IF (NOZNL2 .NE. 1)  GO TO 65
      MAX = KPMAX(4,NJ)
      IF (MAX .EQ. 0)  GO TO 65
      DO 60 KP = 1, MAX
      NP = IC(4,NJ,KP)
      SL4 = SL4 + (C(4,NJ,KP) * EXTRA(NP,II))
   60 CONTINUE
   65 IF (NLMAX .EQ. 0)  GO TO 55
      MAX = KPQMAX(NJ)
      IF (MAX .EQ. 0) GO TO 55
      DO 50 KPQ = 1, MAX
      NP = IDP(NJ,KPQ)
      NQP = IDQ(NJ,KPQ) + NJMAX
      SNL = SNL + (D(NJ,KPQ) * U(NP) * U(NQP))
   50 CONTINUE
   55 UP(NJP) = -(SL1 + SL2 + SL3 + SL4 + SNL)
   10 CONTINUE
      RETURN
      END
      COMPILER (FLD=ABS)
      SUBROUTINE GRAPHS(IBUF,NLOC,LDEV,NTOT,NTICX,NTICY,
     1  XMAX,YMAX,XMIN,YMIN,ITITLX,ITITLY,LTITLX,LTITLY,XARRAY,
     2  YARRAY,DELX,DELY,TITLE)
C-----------------------------------------------------------------------
C
C IDENTIFIER               MEANING                                  TYPE
C
C IBUF:    ADDRESS OF BUFFER AREA FOR PLOT OUTPUT          INTEGER
C NLOC:    NUMBER OF LOCATIONS IN BUFFER AREA (>=2000)     INTEGER
C LDEV:    LOGICAL DEVICE NUMBER FOR PLOT                  INTEGER
C NTOT:    NUMBER OF POINTS TO BE PLOTTED                  INTEGER
C NTICX:   NUMBER OF TIC MARKS ON ABSCISSA (>=2)           INTEGER
C NTICY:   NUMBER OF TIC MARKS ON ORDINATE (>=2)           INTEGER
C XMAX:    UPPER LIMIT OF ABSCISSA DOMAIN                  REAL
C YMAX:    UPPER LIMIT OF ORDINATE RANGE                   REAL
C XMIN:    LOWER LIMIT OF ABSCISSA DOMAIN                  REAL
C YMIN:    LOWER LIMIT OF ORDINATE RANGE                   REAL
C ITITLX:  ABSCISSA LABEL                                  FIELDATA ARRAY
C ITITLY:  ORDINATE LABEL                                  FIELDATA ARRAY

C LTITLX:  NUMBER OF CHARACTERS IN ITITLX                  INTEGER
C LTITLY:  NUMBER OF CHARACTERS IN ITITLY                  INTEGER
C XARRAY:  ABSCISSA POINTS IN TERMS OF XMIN-XMAX COORD'S   REAL ARRAY
C YARRAY:  ORDINATE POINTS IN TERMS OF YMIN-YMAX COORD'S   REAL ARRAY
C DELX:    INTERVALS OF ABSCISSA TIC MARK LABELING
C          IN TERMS OF XMIN-XMAX COORDINATES               REAL
C DELY:    INTERVALS OF ORDINATE TIC MARK LABELING
C          IN TERMS OF YMIN-YMAX COORDINATES               REAL
C TITLE:   LABEL FOR THE WHOLE RUN                         FIELDATA ARRAY
C
C-----------------------------------------------------------------------
      DIMENSION IBUF(NLCC),XARRAY(NTOT),YARRAY(NTOT),ITITLX(1),
     1  ITITLY(1),YLIT(100)
      DIMENSION TITLE(1)
C-----------------------------------------------------------------------
C
C     FIXED BASIC PARAMETERS
C
C-----------------------------------------------------------------------
      LOGICAL ZERO
      DEFINEZERO=NDEC.LT.0.AND.ABS(FPN).LT..5
     1  .OR.NDEC.GT.0.AND.ABS(FPN).LT.5.*10.**(-NDEC-1)
      DEFINE DNDEC=NDEC-FLD(0,36,ZERO)*NDEC-FLD(0,36,ZERO)
      DEFINE IFIX(FARG)=INT(FARG+.5)
      DATA J/1/
      DATA HEIGHT/.105/
      DATA INTEG/1/
      DATA ABSCIS/8./
      DATA ORDINA/6./
      DATA ICODE/-1/
      DATA TOPMAR/1./
      DATA BOTMAR/1.5/
      REAL LEFMAR
      DATA LEFMAR/1.9/
      DATA RYTMAR/1.1/
      DATA FACT/1./
      DATA MAXIS/1/
      DATA MLINE/1/
      DATA HTLAB/.105/
C -------------------------------------------------------------
C
C     19 INITIAL COMPUTATION OF DERIVED PARAMETERS
C        AND INITIAL PLOTS CALL
C     20 SKIPS PRELIMINARIES FOR 2ND AND SUBSEQUENT CALLS
C
C -------------------------------------------------------------
      GO TO (19,20),J
19    YDIT(1) = 3./19.
      TICKLE = HEIGHT/2.
      ROTFAC =  - 3./14. * HEIGHT - 4./7. * HEIGHT
      STARTL = 6 * HEIGHT + ROTFAC + TICKLE
      SEPLAB = STARTL + 1.5 * HEIGHT
      SYMBLH = 0.070
      REAL LABSEP
      LABSEP = 4. * HEIGHT
      ASTART = 2. * HEIGHT
      DO 1 I = 2,100
1     YDIT(I) = YDIT(I - 1) + (2 * MOD(I,2) + 1)/19.
      YDIT(100) = YDIT(100) + .5
      CALL PLOTS(IBUF,NLOC,LDEV)
      CALL FACTOR(1.)
      J = 2
      CALL SYMBOL (HEIGHT,36 * HEIGHT + 5.5,HEIGHT,TITLE,270.,72)
      CALL PLOT(1., - .5, - 3)
3     DO 2 I = 1,100
2     CALL PLOT(0.,YDIT(I),3 - MOD(I,2))
      DO 33 I = 1,100
33    YDIT(I) = YDIT(I) - ABSCIS - RYTMAR
C ----------------------------------------------------------------------
C
C     RESET ORIGIN
C
C -----------------------------------------
      XPAGE = BOTMAR + ORDINA
      GO TO 2019
20    XPAGE = BOTMAR + ORDINA + TOPMAR
2019  CALL WHERE(RXPAGE,RYPAGE,FACT)
      YPAGE = RYPAGE - LEFMAR
      CALL PLOT(XPAGE,YPAGE, - 3)
      CALL FACTOR(FACT)
```

```
C -----------------------------------------------------------
C
C      DRAW AXES AND LABELING MAXIS TIMES
C
C -----------------------------------------------------------
       DO 100 I = 1,MAXIS
100    CALL MYAXIS
C -----------------------------------------------------------
C
C      DRAW POINTS, OPTIONAL CENTERLINE,AND PAGE SCISSORL
C      MLINE TIMES
C
C -----------------------------------------------------------
       DO 200 I = 1,MLINE
200    CALL MYLINE
       RETURN
C -----------------------------------------------------------
C
C      ENTRY POINT SHPARG
C      TERMINATE PLOTTING SEQUENCE
C
C -----------------------------------------------------------
       ENTRY SHPARG
       CALL WHERE(RXPAGE,RYPAGE,I)
       CALL PLOT(RXPAGE,RYPAGE,999)
       RETURN
C -----------------------------------------------------------
C
C      SUBROUTINE MYAXIS (INTERNAL)
C
C -----------------------------------------------------------
       SUBROUTINE MYAXIS
       STARTL = 6 * HEIGHT + ROTFAC + TICKLE
       IMAX = IFIX((YMAX - YMIN)/DELY)
       TICSEP = ORDINA/(ABS(NTICY) - 1)
       CALL DENDEC(YMAX,DELY,NDEC)
       K = 1
       N = (ABS(NTICY)/IMAX) - 1 + MOD(ABS(NTICY),2)
       DO 9 I = 0,IMAX
       GO TO (11,12),K
11     IF(2 * I.LT.IMAX)GO TO 12
       CALL AXLAB(0.,ITITLY,LTITLY,HTLAB)
       K = 2
12     FPN = YMAX - I * DELY
       IF(ZERO)FPN = 0.
       TMID = 1.
       XPAGE =  - I * ORDINA/IMAX - .5 * HEIGHT
       IF(FPN)113,122,118
113    IF(NDEC - 2)115,114,112
114    YPAGE = STARTL@5CHAR
```

```
      GO TO 112
115   IF(NDEC - 1)117,116,112
116   YPAGE = STARTL - HEIGHT@4CHAR
      GO TO 112
117   IF(ABS(FPN) - 100.)119,116,116
119   IF(ABS(FPN) - 10.)120,121,121
120   YPAGE = STARTL - 3 * HEIGHT@2CHAR
      GO TO 112
121   YPAGE = STARTL - 2 * HEIGHT@3CHAR
      GO TO 112
122   YPAGE = STARTL - 4 * HEIGHT@1CHAR
      GO TO 112
118   IF(NDEC - 2)123,116,112
123   IF(NDEC - 1)125,124,112
124   IF(FPN - 10.)121,116,116
125   IF(FPN - 10.)122,120,126
126   IF(FPN - 100.)120,121,127
127   IF(FPN - 1000.)121,116,128
128   IF(FPN - 10000.)116,114,114
112   NNDEC = DNDEC
      CALL NUMBER(XPAGE,YPAGE,HEIGHT,FPN,270.,NNDEC)
      XPAGE =  - I * (ORDINA/IMAX)
      DO 10 JJ = 1,N
      YPAGE = TICKLE * TMID
      CALL PLOT(XPAGE,YPAGE,3)
      YPAGE = YPAGE * ( - 1 + I/IMAX * .5)
      CALL PLOT(XPAGE,YPAGE,2)
      IF(I/IMAX)110,110,9
110   YPAGE = 0
      CALL PLOT(XPAGE,YPAGE,3)
      XPAGE = XPAGE - TICSEP
      CALL PLOT(XPAGE, YPAGE,2)
      TMID = .5
10    CONTINUE
9     CONTINUE
      K = 1
      IMAX = IFIX((XMAX - XMIN)/DELX)
      TICSEP = ABSCIS/(NTICX - 1)
      XPAGE =  - ASTART - ORDINA
      CALL DENDEC(XMAX,DELX,NDEC)
      DO 28 I = 0,IMAX
      STARTL =  - I * ABSCIS/IMAX
      GO TO (24,25),K
24    IF(2 * I.LT.IMAX)GO TO 25
      CALL AXLAB(270.,ITITLX,LTITLX,HTLAB)
      K = 2
      XPAGE =  - ASTART - ORDINA
25    FPN = XMIN + I * DELX
      IF(ZERO)FPN = 0.
      IF(FPN)813,822,818
813   IF(NDEC - 2)815,814,23
814   YPAGE = STARTL + 16./7. * HEIGHT@5CHAR
      GO TO 23
815   IF(NDEC - 1)817,816,23
```

```
816     YPAGE = STARTL + 25./14. * HEIGHT@4CHAR
        GO TO 23
817     IF(ABS(FPN) - 100.)819,816,816
819     IF(ABS(FPN) - 10.)820,821,821
820     YPAGE = STARTL + 11./14. * HEIGHT@2CHAR
        GO TO 23
821     YPAGE = STARTL + 9./7. * HEIGHT@3CHAR
        GO TO 23
822     YPAGE = STARTL + 2./7. * HEIGHT@1CHAR
        GO TO 23
818     IF(NDEC - 2)823,816,23
823     IF(NDEC - 1)825,824,23
824     IF(FPN - 10.)821,816,816
825     IF(FPN - 10.)822,820,826
826     IF(FPN - 100.)820,821,827
827     IF(FPN - 1000.)821,816,828
828     IF(FPN - 10000.)816,814,814
23      NNDEC = DNDEC.
28      CALL NUMBER(XPAGE,YPAGE,HEIGHT,FPN,270.,NNDEC)
        N = (NTICX/IMAX) - 1 + MOD(NTICX,2)
        DO 26 I = IMAX,0, - 1
        TMID = 1.
        YPAGE =  - I * ABSCIS/IMAX.
        DO 27 JJ = 1,N
        XPAGE =  - ORDINA - TICKLE * TMID
        CALL PLOT(XPAGE,YPAGE,3)
        XPAGE = XPAGE + (TICKLE + FLD(0,36,I.NE.0) * TICKLE) * TMID
        CALL PLOT(XPAGE,YPAGE,2)
        IF(I)111,26,111
111     XPAGE =  - ORDINA
        CALL PLOT(XPAGE,YPAGE,3)
        YPAGE = YPAGE + TICSEP
        CALL PLOT(XPAGE,YPAGE,2)
        TMID = .5
27      CONTINUE
26      CONTINUE
        RETURN
C -------------------------------------------------------
C
C       SUBROUTINE MYLINE (INTERNAL)
C
C -------------------------------------------------------
        SUBROUTINE MYLINE
        ITOP = IFIX((ABSCIS + RYTMAR + .5)/11. * 99.)
        IBOT = IFIX(RYTMAR/11. * 99.)
        DO 17 I = 1,NTOT
        XPAGE = (YARRAY(I) - YMAX)/(YMAX - YMIN) * ORDINA
        YPAGE = (XMIN - XARRAY(I))/(XMAX - XMIN) * ABSCIS
17      CALL SYMBOL(XPAGE,YPAGE,SYMBLH,INTEQ,270.,ICODE)
        IF(NTICY.GE.0)GO TO 22
        XPAGE =  - ORDINA/2.
        YPAGE =  - ABSCIS
        CALL PLOT(XPAGE,YPAGE,3)
        DO 18 I = IBOT,ITOP
```

```
18    CALL PLOT(XPAGE,YDIT(I),3 - MOD(I,2))
22    XPAGE = TOPMAR
      YPAGE =  - ABSCIS - RYTMAR - .5
      CALL PLOT(XPAGE,YPAGE,3)
      DO 21 I = 1,100
21    CALL PLOT(XPAGE,YDIT(I),3 - MOD(I,2))
      RETURN
C ---------------------------------------
C
C     SUBROUTINE AXLAB (INTERNAL)
C
C ------------------------------------------------
      SUBROUTINE AXLAB(ANGLE,IBCD,NCHARX,HEIGHT)
      DIMENSION IBCD(7)
      LOGICAL S
      INTEGER QSQ/' S'/
      K = 2
      NCHAR = NCHARX
      S = .FALSE.
      IF(ABS(ANGLE).GT..1)GO TO 30
      XPAGE =  - ORDINA/2. - NCHAR * HEIGHT/2
      YPAGE = SEPLAB
      GO TO 31
30    XPAGE =  - ORDINA - LABSEP
      YPAGE =  - ABSCIS/2. + NCHAR * HEIGHT/2
31    LSTART = 6 * MOD(NCHAR,6) - 12
      IF(LSTART.EQ. - 12)LSTART = 24
      LOOK = NCHAR/6 + 1.1
      IF(LSTART.EQ. - 6)GO TO 13
      IF(FLD(0,12,',S').EQ.FLD(LSTART,12,IBCD(LOOK)))GO TO 15
      GO TO 14
13    IF(FLD(0,6,',').NE.FLD(30,6,IBCD(LOOK - 1)))GO TO 14
      IF(FLD(0,6,'S').NE.FLD(0,6,IBCD(LOOK)))GO TO 14
15    NCHAR = NCHAR - 1
      S = .TRUE.
14    CALL SYMBOL(XPAGE,YPAGE,HEIGHT,IBCD,ANGLE,NCHAR)
      IF(S)CALL SYMBOL(999.,999.,2 * HEIGHT/3,QSQ,ANGLE,2)
      RETURN
C ---------------------------------------
C
C     SUBROUTINE DENDEC (INTERNAL)
C
C ------------------------------------------------
      SUBROUTINE DENDEC(QMAX,DELQ,NDEC)
      IF(INT(ABS(QMAX)).GE.10)GO TO 5
      IF(AMOD(ABS(QMAX - DELQ),.1).GE..01)GO TO 7
      NDEC = 1
      RETURN
5     NDEC =  - 1
      RETURN
7     NDEC = 2
      RETURN
      END
```

REFERENCES

1. Tsien, H. S., "The Transfer Functions of Rocket Nozzles," American Rocket Society Journal, Vol. 22, 1952, pp 139-143.

2. Crocco, L. and Cheng, S. I., Theory of Combustion Instability in Liquid Propellant Rocket Motors, Appendix B. AGARD Monograph No. 8, Butterworths, London, 1956.

3. Crocco, L. and Sirignano, W. A., "Behavior of Supercritical Nozzles Under Three Dimensional Oscillatory Conditions," Princeton University Department of Aerospace and Mechanical Sciences, Report No. 790, April 1967.

4. Zinn, B. T., "A Theoretical Study of Nonlinear Transverse Combustion Instability in Liquid Propellant Rocket Motors," Princeton University Department of Aerospace and Mechanical Sciences, Report No. 732, May 1966.

5. Zinn, B. T. and Crocco, L., "Periodic Finite-Amplitude Oscillations in Slowly Converging Nozzles," Astronautica Acta, Vol. 13, 1968, pp. 481-488.

6. Zinn, B. T. and Crocco, L., "The Nozzle Boundary Condition in the Nonlinear Rocket Instability Problem," Astronautica Acta, Vol. 13, 1968, pp. 489-496.

7. Lores, M. E. and Zinn, B. T., "The Prediction of Nonlinear Longitudinal Combustion Instability in Liquid Propellant Rockets," NASA CR-120904, April 1972.

8. Lores, M. E. and Zinn, B. T., "Nonlinear Longitudinal Combustion Instability in Rocket Motors," presented at the AIAA 11th Aerospace Sciences Meeting, January 1973.

9. Zinn, B. T. and Powell, E. A., "Nonlinear Combustion Instability in Liquid Propellant Rocket Engines," Proceedings of the 13th Symposium (International) on Combustion, The Combustion Institute, pp. 491-503.

10. Powell, E. A. and Zinn, B. T., "The Prediction of the Nonlinear Behavior of Unstable Liquid Rockets," NASA CR-72902, July 1971.

11. Powell, E. A. and Zinn, B. T., "The Prediction of Nonlinear Three-Dimensional Combustion Instability in Liquid Rockets with Conventional Nozzles," NASA CR-121279, October 1973.

12. Padmanabhan, M. S., Powell, E. A., and Zinn, B. T., "Application of the Galerkin Method in the Prediction of Nonlinear Nozzle Admittances," Proceedings of the 11th JANNAF Combustion Meeting, CPIA Publication 261, Vol. II, December 1974, pp. 141-163.

13. Padmanabhan, M. S., "The Effect of Nozzle Nonlinearities Upon the Nonlinear Stability of Liquid Rocket Motors," Ph.D. Thesis, Georgia Institute of Technology (December 1975).

14. Finlayson, B. A. and Scriven, L. E., "The Method of Weighted Residuals -- A Review," Applied Mechanics Reviews, Vol. 19, No. 9, September 1966, pp. 735-744.

15. Ames, W. F., Nonlinear Partial Differential Equations in Engineering, Academic Press, New York, 1965, pp. 243-262.

16. Bell, W. A. and Zinn, B. T., "The Prediction of Three-Dimensional Liquid-Propellant Rocket Nozzle Admittances," NASA CR-121129, February 1973.

17. McCracken, D. D., FORTRAN with Engineering Applications, John Wiley and Sons, Inc., New York, 1967, pp. 146-153.

REPORT DISTRIBUTION LIST

NASA-Lewis Research Center
Attn: Dr. R. J. Priem/MS 500-204
21000 Brookpark Road
Cleveland, OH 44135

NASA-Lewis Research Center
Attn: Library/MS 60-3
21000 Brookpark Road
Cleveland, OH 44135

NASA-Lewis Research Center
Attn: Report Control Office/MS 5-5
21000 Brookpark Road
Cleveland, OH 44135

NASA-Lewis Research Center
Attn: E. A. Bourke/MS 500-205
21000 Brookpark Road
Cleveland, OH 44135

NASA Headquarters
Attn: RPS/Robert A. Wasel
600 Independence Ave., SW, Room 526
Washington, DC 20546

NASA-Lewis Research Center
Attn: Procurement Section
Mail Stop 500-313
21000 Brookpark Road
Cleveland, OH 44135

NASA-Lyndon B. Johnson Space Center
Attn: EP/Joseph G. Thibodaux
Houston, TX 77058

NASA-George C. Marshall Space Flight
Center
Attn: S&E-ASTN-PP/ R. J. Richmond
Huntsville, AL 35812

NASA Scientific & Technical Information
Facility - Acquisitions Branch
P. O. Box 33
College Park, MD 20740

Aerojet Liquid Rocket Company
Attn: David A. Fairchild
Bldg. 20001/Sec. 9732
P. O. Box 13222
Sacramento, CA 95813

Aerojet General Corporation
Propulsion Division
Attn: R. Stiff
P. O. Box 15847
Sacramento, CA 95803

Aerospace Corporation
Attn: O. W. Dykema
P. O. Box 92957
Los Angeles, CA 90045

Aerospace Corporation
Attn: Library-Documents
2400 E. El Segundo Boulevard
Los Angeles, CA 90045

Air Force Rocket Propulsion Lab. (RPM)
Attn: Library
Edwards, CA 93523

Air Force Office of Scientific Research
Chief Propulsion Division
Attn: Dr. J. F. Masi (NAE)
1400 Wilson Boulevard
Arlington, VA 22209

Air Force Rocket Propulsion Laboratory
Attn: Daweel George
Edwards, CA 93523

AFAPL
Research & Technology Div.
AF Systems Command
U. S. Air Force
Attn: Library/APRP
Wright Patterson AFB, OH 45433

Air Force Rocket Propulsion Laboratory
Attn: Richard R. Weiss
Edwards, CA 93523

AFAPL
Attn: Frank D. Stull (RJT)
Wright Patterson AFB, OH 45433

Army Ballistics Research Labs
Attn: Austin W. Barrows
Code AMXBR-1B
Aberdeen Proving Grounds, MD 21005

Army Ballistic Research Labs
Attn: Ingo W. May
Code AMXBR-1B
Aberdeen Proving Grounds, MD 21005

Army Material Command
Missile Systems Div.
Attn: Stephen R. Matos
Code AMCRD-MT
5001 Eisenhower Ave.
Alexandria, VA 22304

Air Force Systems Command
Arnold Engineering Development Center
Attn: Dr. H. K. Doetsch
Tullahoma, TN 37389

Aeronutronic Div. of Philco Ford Corp.
Technical Information Dept.
Ford Road
Newport Beach, CA 92663

Battelle Memorial Institute
Attn: Report Library, Room 6A
505 King Ave
Columbus, OH 43201

Bell Aerosystems, Inc.
Attn: Library
Box 1
Buffalo, NY 14205

Bell Aerospace Company
Attn: T. F. Ferger
Post Office Box 1
Mail Zone, J-81
Buffalo, NY 14205

Bureau of Naval Weapons
Department of the Navy
Attn: Library
Washington, DC

Brooklyn Polytechnic Institute
Long Island Graduate Center
Attn: V. D. Agosta
Route 110
Farmingdale, NY 11735

California Institute of Technology
Jet Propulsion Laboratory
Attn: Fred E. C. Culick
4800 Oak Grove Drive
Pasadena, CA 91103

California Institute of Technology
Jet Propulsion Laboratory
Attn: Jack H. Rupe
4800 Oak Grove Drive
Pasadena, CA 91103

California State Univ. Sacramento
School of Engineering
Attn: Frederick H. Reardon
6000 J. Street
Sacramento, CA 95819

Chemical Propulsion Information Agency
Johns Hopkins University/APL
Attn: T. W. Christian
8621 Georgia Avenue
Silver Spring, MD 20910

Colorado State University
Attn: Charles E. Mitchell
Fort Collins, CO 80521

Frankford Arsenal
Attn: Martin Visnov
NDP-R, Bldg. 64-2
Bridge & Tacony Streets
Philadelphia, PA 19137

General Electric Company
Flight Propulsion Laboratory Dept.
Attn: D. Suichu
Cincinnati, OH 45215

Georgia Institute of Technology
Georgia Tech. Res. Inst.
Attn: Warren C. Strahle
Atlanta, GA 30332

Georgia Institute of Technology
Georgia Tech. Res. Inst.
Attn: Ben T. Zinn
Atlanta, GA 30322

Melvin Gerstein
P. O. Box 452
Altadena, CA 91001

Marquardt Corporation
16555 Saticory Street
Box 2013 - South Annex
Van Nuys, CA 91409

Massachusetts Institute of Technology
Department of Mechanical Engineering
Attn: T. Y. Toong
77 Massachusetts Avenue
Cambridge, MA 02139

McDonald Douglas Corporation
McDonnel Douglas Astronautics Co.
Attn: William T. Webber
5301 Bolsa Avenue
Huntington Beach, CA 92647

D. E. Mock
Advanced Research Projects Agency
Washington, DC 20525

Lockheed Aircraft Corp
Lockheed Propulsion Co., Div.
Attn: Norman S. Cohen
P. O. Box 111
Redlands, CA 92363

Naval Postgraduate School
Department of Aeronautics
Attn: David W. Netzer
Monterey, CA 93940

Naval Underwater Systems Center
Energy Conversion Dept.
Attn: Robert S. Lazar, Code 5B331
Newport, RI 02840

Naval Weapons Center
Attn: R. L. Derr, Code 608
China Lake, CA 93555

Naval Weapons Center
Attn: Charles J. Thelen, Code 4505
China Lake, CA 93555

Naval Postgraduate School
Department of Aeronautics
Attn: Allen F. Fuhs
Monterey, CA 93940

Ohio State University
Department of Aeronautical and
Astronautical Engineering
Attn: R. Edse
Columbus, OH 43210

Pennsylvania State University
Mechanical Engineering Department
Attn: G. M. Faeth
207 Mechanical Engineering Bldg.
University Park, PA 16802

Princeton University
Forrestal Campus Library
Attn: Irvin Glassman
P. O. Box 710
Princeton, NJ 08450

Princeton University
Forrestal Campus Library
Attn: David T. Harrje
P. O. Box 710
Princeton, N.J. 08540

Princeton University
Forrestal Campus Library
Attn: Martin Summerfield
P. O. Box 710
Princeton, N.J. 08540

Propulsion Sciences, Inc.
Attn: Vito Agosta
P. O. Box 814
Melville, NY 11746

Purdue University
Jet Propulsion Laboratory
Project Squid
Attn: Robert Goulard
West Lafayette, IN 47907

Purdue University Res. Foundation
School of Mechanical Engineering
Attn: John R. Osborn
Thermal Sci. Propulsion Center
West Lafayette, IN 47906

Purdue University Res. Foundation
School of Mechanical Engineering
Attn: Bruce A. Reese
Thermal Sci. Propulsion Center
West Lafayette, IN 47906

Research and Development Associates
Atten: Raymond B. Edelman
P. O. Box 3580
525 Wilshire Blvd.
Santa Monica, CA 90402

Rockwell International Corporation
Rocketdyne Division
Attn: L. P. Combs, D/991-350
Zone 11
6633 Canoga Avenue
Canoga Park, CA 91304

Rockwell International Corporation
Rocketdyne Division
Attn: James A. Nestlerode
Dept. 596-124, AC46
6633 Canoga Ave
Canoga Park, CA 91304

Rockwell International Corp.
Rocketdyne Division
Attn: Carl L. Oberg
Dept. 589-197-SS11
6633 Canoga Ave
Canoga Park, CA 91304

Rockwell International Corp.
Rocketdyne Division
Attn: Library Dept. 596-306
6633 Canoga Avenue
Canoga Park, CA 91304

Stanford Research Institute
333 Ravenswood Avenue
Menlo Park, CA 94025

Susquehanna Corp.
Atlantic Research Division
Attn: Library
Shirley Highway and Edsall Road
Alexandria, VA 22314

TISIA
Defense Documentation Center
Cameron Station, Bldg. 5
5010 Duke Street
Alexandria, VA 22314

Tennessee Technological University
Dept. of Mech. Engrg.
Attn: Kenneth R. Purdy
P. O. Box 5014
Cookeville, TN 38501

Textron, Inc.
Bell Aerospace Div.
Research Department
Attn: John H. Morgenthaler, C-84
P. O. Box One
Buffalo, NY 14240

TRW, Inc.
TRW Systems Gp.
Attn: A. C. Ellings
One Space Park
Redondo Beach, CA 90278

TRW Systems
Attn: G. W. Elverun
One Space Park
Redondo Beach, CA 90278

TRW Systems Group
STL Tech. Lib. Doc. Acquisitions
One Space Park
Redondo Beach, CA 90278

Tulane University
Attn: J. C. O'Hara
6823 St. Charles Ave
New Orleans, LA 70118

Ultrasystems, Inc.
Attn: Thomas J. Tyson
500 Newport Center Dr.
Newport Beach, CA

United Aircraft Corp.
Pratt & Whitney Division
Florida Research & Development Center
Attn: Library
West Palm Beach, FL 33402

United Aircraft Corporation
Pratt & Whitney Aircraft Division
Attn: Thomas C. Mayes
P. O. Box 2691
West Palm Beach, FL 33402

United Aircraft Corporation
Attn: R. H. Woodward Waesche
400 Main Street
East Hartford, CT 06108

United Aircraft Corporation
United Technology Center
Attn: Library
P. O. Box 358
Sunnyvale, CA 94088

University of California
Aerospace Engineering Dept.
Attn: F. A. Williams
Post Office Box 109
LaJolla, CA 92037

University of California, Berkeley
Dept. of Mechanical Engineering
Attn: A. K. Oppenheim
Berkeley, CA 94720

University of Illinois
Aeronautics/Astronautics Engr. Dept.
Attn: R. A. Strehlow
Transportation Bldg. Room 101
Urbana, Ill 61801

University of Michigan
Attn: James A, Nicholls
P. O. Box 622
Ann Arbor, MI 48107

University of Utah
Dept. of Chemical Engineering
Atten: Alva D. Baer
Park Bldg, Room 307
Salt Lake City, UT 84112

University of Wisconsin
Mechanical Engineering Dept.
Attn: P. S. Myers
1513 University Avenue
Madison, WI 53706

United States Naval Research Laboratory
Director (Code 6180)
Attn: Library
Washington, DC 20390

Virginia Polytechnic Institute
State University
Attn: J. A. Schetz
Blacksburg, VA 24061

Office of Assistant Director
(Chemical Technician)
Office of the Director of Defense
Research & Engineering
Washington, DC 20301