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
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 asymmetric 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 non-linearities has no significant effect on the limiting amplitude and pressure waveforms.
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

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUMMARY</td>
<td>1</td>
</tr>
<tr>
<td>INTRODUCTION</td>
<td>2</td>
</tr>
<tr>
<td>SYMBOLS</td>
<td>3</td>
</tr>
<tr>
<td>NOZZLE ANALYSIS</td>
<td>5</td>
</tr>
<tr>
<td>Development of the Nozzle Wave Equation</td>
<td>6</td>
</tr>
<tr>
<td>Method of Solution</td>
<td>9</td>
</tr>
<tr>
<td>Derivation of Admittance Relations</td>
<td>12</td>
</tr>
<tr>
<td>Calculation of the Nozzle Response</td>
<td>14</td>
</tr>
<tr>
<td>COMBUSTION INSTABILITY ANALYSIS</td>
<td>17</td>
</tr>
<tr>
<td>Combustion Chamber Model</td>
<td>17</td>
</tr>
<tr>
<td>Application of Galerkin Method</td>
<td>19</td>
</tr>
<tr>
<td>Numerical Solution Procedure</td>
<td>21</td>
</tr>
<tr>
<td>RESULTS AND DISCUSSION</td>
<td>22</td>
</tr>
<tr>
<td>Admittance Coefficients</td>
<td>22</td>
</tr>
<tr>
<td>Stability Calculations</td>
<td>27</td>
</tr>
<tr>
<td>CONCLUDING REMARKS</td>
<td>31</td>
</tr>
<tr>
<td>APPENDIX A - PROGRAM NOZADM: A USER'S MANUAL</td>
<td>40</td>
</tr>
<tr>
<td>General Description</td>
<td>40</td>
</tr>
<tr>
<td>Program Structure</td>
<td>40</td>
</tr>
<tr>
<td>Input Data</td>
<td>42</td>
</tr>
</tbody>
</table>

PRECEDING PAGE BLANK NOT FILMED
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Steady-State Quantities</td>
<td>43</td>
</tr>
<tr>
<td>Coefficients</td>
<td>44</td>
</tr>
<tr>
<td>Integrals</td>
<td>44</td>
</tr>
<tr>
<td>Integration of the Differential Equations</td>
<td>46</td>
</tr>
<tr>
<td>Output</td>
<td>46</td>
</tr>
<tr>
<td>FORTRAN Listing</td>
<td>48</td>
</tr>
<tr>
<td>APPENDIX B - PROGRAM COEFFS3D: A USER'S MANUAL</td>
<td>78</td>
</tr>
<tr>
<td>FORTRAN Listing</td>
<td>81</td>
</tr>
<tr>
<td>APPENDIX C - PROGRAM LCYC3D: A USER'S MANUAL</td>
<td>108</td>
</tr>
<tr>
<td>FORTRAN Listing</td>
<td>112</td>
</tr>
<tr>
<td>REFERENCES</td>
<td>140</td>
</tr>
</tbody>
</table>
## LIST OF ILLUSTRATIONS

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Coordinate System Used for the Solution of the Oscillatory Nozzle Flow</td>
<td>8</td>
</tr>
<tr>
<td>2.</td>
<td>Nozzle Profile Used in Calculating Admittances</td>
<td>16</td>
</tr>
<tr>
<td>3.</td>
<td>Typical Mathematical Model of a Liquid Rocket Motor</td>
<td>18</td>
</tr>
<tr>
<td>4.</td>
<td>Linear Admittances for the 1T, 2T, and 1R Modes</td>
<td>23</td>
</tr>
<tr>
<td>5.</td>
<td>Nonlinear Admittances for the 2T Mode</td>
<td>24</td>
</tr>
<tr>
<td>6.</td>
<td>Nonlinear Admittances for the 1R Mode</td>
<td>25</td>
</tr>
<tr>
<td>7.</td>
<td>Relative Magnitudes of Linear and Nonlinear Admittances</td>
<td>26</td>
</tr>
<tr>
<td>8.</td>
<td>Effect of Entrance Mach Number on the Relative Magnitudes of Linear and Nonlinear Admittances</td>
<td>28</td>
</tr>
<tr>
<td>9.</td>
<td>Effect of Nozzle Half-Angle on the Relative Magnitudes of Linear and Nonlinear Admittances</td>
<td>29</td>
</tr>
<tr>
<td>10.</td>
<td>Effect of Nozzle Radii of Curvature on the Relative Magnitudes of Linear and Nonlinear Admittances for the 2T Mode</td>
<td>30</td>
</tr>
<tr>
<td>11.</td>
<td>Linear Stability Limit</td>
<td>32</td>
</tr>
<tr>
<td>12.</td>
<td>Comparison of Pressure Waveforms for a Mildly Unstable Motor</td>
<td>33</td>
</tr>
<tr>
<td>13.</td>
<td>Comparison of Pressure Waveforms for a Strongly Unstable Motor</td>
<td>34</td>
</tr>
<tr>
<td>14.</td>
<td>Comparison of Limit-Cycle Amplitudes for Different Values of $n$</td>
<td>35</td>
</tr>
<tr>
<td>15.</td>
<td>Comparison of Limit-Cycle Pressure Amplitudes for Different Values of $\varphi$</td>
<td>36</td>
</tr>
<tr>
<td>16.</td>
<td>Comparison of Limit-Cycle Amplitudes for Different Values of $M_e$</td>
<td>37</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>-------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>17.</td>
<td>Comparison of Limit-Cycle Amplitudes for Different Values of L/D.</td>
<td>38</td>
</tr>
<tr>
<td>A-1.</td>
<td>Flow Chart</td>
<td>41</td>
</tr>
</tbody>
</table>
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 who considered the case in which the oscillation of the incoming flow is one-dimensional and isothermal. Crocco 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 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 and transverse 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. 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\textsuperscript{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**

- $A_p(\phi)$: axially dependent amplitude functions in Eq. (4)
- $B_p(t)$: time dependent amplitude functions in Eq. (18)
- $R_{N}(\tilde{z}')$: nozzle boundary residual (see Eq. (10))
- $b_p$: complex axial acoustic eigenvalue
- $c$: dimensionless sonic velocity, $c^*/c_0^*$
$E_\nu(\xi')$ residual of Eq. (2)

$E_c(\xi')$ 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, $\gamma_p^*/\rho_o^* c_o^*$

$r$ 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, $u^*/c_0^*$

$\gamma_p^*$ linear admittance for the $p^{th}$ mode

$z$ dimensionless axial coordinate, $z^*/r_c^*$

$\gamma$ specific heat ratio

$\Gamma_p^*$ nonlinear admittance for the $p^{th}$ mode

$\xi_p^*$ linear admittance function

$\theta$ azimuthal coordinate

$\rho^*$ dimensionless density, $\rho^*/\rho_o^*$

$\tau$ dimensionless pressure sensitive time lag, $\frac{\tau^*}{(r_c^*/c_o^*)}$
\[ \varphi \quad \text{steady state potential function} \]
\[ \phi \quad \text{velocity potential} \]
\[ \psi \quad \text{steady state stream function} \]
\[ \omega \quad \text{dimensionless frequency} \]

Subscripts:
\[ e \quad \text{evaluated at the nozzle entrance} \]
\[ n \quad \text{radial mode number} \]
\[ r, i \quad \text{real and imaginary parts of a complex quantity, respectively} \]
\[ w \quad \text{evaluated at the nozzle wall} \]
\[ o \quad \text{stagnation quantity} \]
\[ \varphi, \psi, r, \theta, z, t \quad \text{partial differentiation with respect to } \varphi, \psi, r, \theta, z, \text{ or } t, \text{ respectively} \]

Superscripts:
\[ ( \cdot )' \quad \text{perturbation quantity} \]
\[ ( \cdot ) \quad \text{steady state quantity} \]
\[ ( \cdot )^* \quad \text{dimensional quantity, complex conjugate} \]
\[ ( \cdot )^{\text{approximate}} \quad \text{approximate solution} \]

**NOZZLE ANALYSIS**

The development of the nonlinear nozzle theory is described in detail 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,\textsuperscript{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:

\begin{equation}
\nabla^2 \phi - \frac{\delta}{\theta \phi t} = 2\nabla \phi \cdot \nabla \phi + (\gamma - 1) \frac{\phi}{\theta} \nabla^2 \phi \\
+ \frac{\gamma - 1}{2} (\nabla \phi \cdot \nabla \phi) \nabla^2 \phi + \frac{1}{2} \nabla \phi \cdot \nabla (\nabla \phi \cdot \nabla \phi)
\end{equation}

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 \((\phi, \psi, \theta)\) coordinate system used by Zinn and Crocco\textsuperscript{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:

\begin{equation}
E_{in}(\phi') = f_1(\psi) \phi' \phi' + f_2(\psi) \phi' \phi + f_3(\psi) \left[ 2(\psi \phi' \phi' + \phi' \phi') + \frac{1}{2} \phi' \phi' \right] \\
- 2 \phi' \phi' t + f_4(\psi) \phi' - \frac{1}{\theta} \phi' \theta t \\
- \left\{ 2 \phi' \phi' \phi' t + \frac{b^2}{\theta} \phi' \phi' \psi \phi' \phi' t + \frac{b^2}{\psi} \phi' \phi' \phi' \phi' t \right\}
\end{equation}
\[ + (\gamma+1)\frac{u^2}{c^2} \phi \psi \phi \psi + 2 \bar{p}u \psi \phi \phi \phi + \frac{\bar{p}u}{2\psi} \phi \phi \phi \psi \]

\[ + f_5(\phi) (\phi')^2 + f_6(\phi) (\phi')^2 + f_6(\phi) \frac{1}{4\psi} (\phi')^2 \]

\[ + (\gamma-1) \phi' \phi' \phi' - f_4(\phi) \phi' \phi' \]

\[ + (\gamma-1) \frac{p}{u} \left[ 2 \left( \psi \phi' + \phi' \right) + \frac{1}{2\psi} \phi' \theta \theta \right] \phi' \]

\[ + (\gamma-1) \bar{p}u \left[ 2 \left( \psi \phi' + \phi' \right) + \frac{1}{2\psi} \phi' \theta \theta \right] \phi' \phi' = 0 \]

where

\[ f_1(\phi) = \frac{c^2}{\phi} - \frac{u^2}{\phi} \]

\[ f_2(\phi) = \frac{1}{c^2} \frac{\partial u^2}{\partial \phi} \]

\[ f_3(\phi) = \frac{pc^2}{u} \]

\[ f_4(\phi) = - (\gamma-1) \frac{du^2}{2c^2} \frac{\partial \psi}{\partial \phi} \]

\[ f_5(\phi) = \frac{3}{2} \left[ 1 + \frac{\gamma-1}{2} \frac{u^2}{c^2} \right] \frac{\partial u^2}{\partial \phi} \]

\[ f_6(\phi) = \frac{p}{2u} \left[ 1 - (2-\gamma) \frac{u^2}{c^2} \right] \frac{\partial u^2}{\partial \phi} \]

\[ 7 \]
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, \( \tilde{\phi} \), 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:

\[
\tilde{\phi} = \sum_{p=1}^{N} \left\{ A_p(\varphi) \cos(m \theta) J_m \left[ S_{mn} \left( \frac{\psi}{\psi_w} \right) \right] e^{ik_p \omega t} \right\}
\]

where the functions \( A_p(\varphi) \) are unknown complex functions of the axial variable \( \varphi \), and \( \theta \)- and \( \psi \)-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

<table>
<thead>
<tr>
<th>p</th>
<th>m(p)</th>
<th>n(p)</th>
<th>k_p</th>
<th>Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1T</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>2T</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3R</td>
</tr>
</tbody>
</table>

In the time-dependence, \( w \) 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. For example it was found that, due to nonlinear coupling between modes, the 2T and 3R 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., \( \varphi \)) in the nozzle through the unknown functions \( A_p(\varphi) \).

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

\[
\int_T \int_S E_N(\hat{\phi}') e^{-ik_j \omega t} \cos m\theta J_m \left[ S_{mn} \left( \frac{\psi}{\psi_w} \right)^{\frac{1}{2}} \right] dS dt = 0 \quad (5)
\]

\[ j = 1, 2, \ldots, N \]

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 = 2π/ω, while the spatial integration is performed over any surface of \( \varphi = \text{constant} \) 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:

\[
H_1(\varphi) \frac{d^2A_1}{d\varphi^2} + M_1(\varphi) \frac{dA_1}{d\varphi} + N_1(\varphi)A_1(\varphi) = 0
\]

\[
H_p(\varphi) \frac{d^2A_p}{d\varphi^2} + M_p(\varphi) \frac{dA_p}{d\varphi} + N_p(\varphi)A_p(\varphi) = \sum_{p=1}^{N} \left\{ A_1, \frac{dA_1}{d\varphi}, \frac{d^2A_1}{d\varphi^2} \right\}
\]

\( p = 2, 3, \ldots N \)

where
\[ H_p(\varphi) = u^2 (c^2 \varphi^2 - \dot{\varphi}^2) \]  \hspace{1cm} (7)

\[ M_p(\varphi) = -\frac{u^2}{c^2} \left[ \frac{1}{\varphi} \frac{d\varphi^2}{d\varphi} + 2ik_\varphi \right] \]

\[ N_p(\varphi) = \left[ -\frac{S_p^2}{2\psi_w} \mu u c^2 - \frac{V_\varphi}{\varphi} \frac{1}{p} \frac{u}{c} \frac{d\varphi^2}{d\varphi} + k^2 \varphi^2 \right] \]

and \( I_p \) are inhomogeneous terms which are functions of \( \varphi \) and the amplitude of the IT mode, \( A_\parallel(\varphi) \).

It can be seen that the above equations are decoupled with respect to the IT mode; that is, the solution for \( A_\parallel \) can be obtained independently of the amplitudes of the other modes. Thus to second order the nozzle nonlinearities do not affect the IT 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. (22) 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:

\[ A_p(\varphi) = K_A \varphi A_p^{(h)}(\varphi) + A_p^{(i)}(\varphi) \]  \hspace{1cm} (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 \( \bar{\varphi}' \) given by Eq. (5) is a summation of partial potentials \( \bar{\varphi}'_p \) where
\[ \ddot{\phi}_p = A_p(\phi) \cos(m\theta) J_m \left[ S_m \left( \frac{x}{y} \right) \right] e^{ix_p} \] (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:

\[ B_N(\ddot{\phi}) = \frac{\partial \ddot{\phi}}{\partial z} + \gamma_p \frac{\partial \ddot{\phi}}{\partial t} \] (10)

\[ + \bar{u}_e c_e e^{i(\cos m \theta) J_m \left[ S_m \left( \frac{x}{y} \right) \right] e^{ik \cdot \omega t}} \Gamma_p = 0 \]

where

\[ \gamma_p = \left( \frac{i\bar{u}_e}{\gamma_k \omega} \right) \frac{1}{A_p(h)} \frac{dA_p(h)}{d\varphi} \quad p = 1, 2, \ldots N \] (11)

\[ \Gamma_p = \frac{1}{c^2 A_p(h)} \left[ A_p(i) \frac{dA_p(h)}{d\varphi} - A_p(h) \frac{dA_p(i)}{d\varphi} \right] \quad p = 2, 3, \ldots N \] (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 \( \gamma_p \) and \( \Gamma_p \) are, respectively, the linear and nonlinear admittance coefficients for the \( p \)th mode. The nonlinear admittance, \( \Gamma_p \), 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:

\[ U_p - \gamma_p P_p = -\frac{u}{c} c_p \Gamma_p \]  

where \( U_p \) and \( P_p \) are the \( \varphi \)-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 \( \gamma_p \) and \( \Gamma_p \) must be determined for the nozzle under consideration. The equations governing these quantities are readily derived from Eqs. (6) using the definition of \( \Gamma_p \) (i.e., Eq. (12) to obtain:

\[ \frac{H_p}{p} \frac{d\gamma_p}{d\varphi} = -m_p \gamma_p - n_p - H_p \gamma_p \]  

\[ \frac{H_p}{p} \frac{d\Gamma_p}{d\varphi} = \left(-H_p \gamma_p + \frac{\gamma_p}{2c^2} \frac{du}{d\varphi} \frac{H_p}{p} - M_p \right) \Gamma_p = \frac{I_p}{c^2} \]  

where

\[ \gamma_p = \frac{1}{A_p} \frac{dA_p}{d\varphi} \]  

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 non-linear terms vanish for the LT mode (i.e., \( I_1 = 0, I_p = 0 \)) and it is only necessary to solve Eq. (14) to obtain \( \gamma_1 \) (and hence \( \gamma_1 \)) at the nozzle entrance. Since Eq. (14) does not depend on the higher modes, it can be solved independently for \( \gamma_1 \). Once \( \gamma_1 \) has been determined both Eqs. (14)
and (15) must be solved for the other modes. In order to do this, the amplitude \( A_1(\phi) \) must be determined since Eq. (15) depends on \( A_1(\phi) \) and its derivatives through \( I_p(\phi) \). Once \( \zeta_1(\phi) \) is known, \( A_1(\phi) \) 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 LTE mode) at the nozzle entrance. The value of \( A_1 \) thus found is introduced into Eq. (15) which is then solved for \( \Gamma_p \).

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 \( \Gamma_p \) 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 \( \phi = 0 \).

In Eqs. (14) and (15), the quantities \( H_p, M_p, N_p \) and \( \Gamma_p \) are functions of the steady-state flow variables in the nozzle and these must be computed before performing the numerical integration to obtain \( \zeta_p \) and \( \Gamma_p \). 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 \( r_0 = 1 \). 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 \( \theta_1 \), 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 \( \phi \) 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 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 \( \phi \) must satisfy the following nonlinear partial differential equation:

\[
E_c (\phi') = \frac{\phi'}{rr} + \frac{1}{r} \frac{\phi'}{rr} + \frac{1}{r} \frac{\phi'}{r} + \frac{1}{2} \frac{\phi'}{\theta \theta} + \frac{\phi'}{zz} - \frac{\phi'}{tt}
\]  \( \text{(17)} \)

\[
-2\frac{\phi'\phi'}{r} - \frac{2}{r} \frac{\phi'\phi'}{\theta} \theta \theta - 2\frac{\phi'\phi'}{z} z z
\]

\[-(\gamma-1)\frac{\phi'}{t} (\phi' + \frac{1}{r} \frac{\phi'}{r} + \frac{1}{r} \frac{\phi'}{\theta} \theta + \frac{\phi'}{zz})
\]

\[-2u\frac{\phi'}{z} = (\gamma+1)\frac{\phi'}{t} \frac{du}{dz}
\]

\[+ \gamma n \frac{du}{dz} \left[ \phi'(r, \theta, z, t) - \phi'(r, \theta, z, t - \tau) \right] = 0
\]

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):

\[ \tilde{\Psi} = \sum_{p=1}^{N} \phi_p = \sum_{p=1}^{N} B_p(t) \cos(m\phi) J_m(\frac{m\pi r}{b_p}) \cosh(\frac{m\pi z}{b_p}) \]  

(18)

developed the Galerkin method 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. 14). Unlike the nozzle analysis where the unknown coefficients \( A_p(\phi) \) 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 \( E_c(\tilde{\Psi}) \). Similarly, the series expansion was substituted into the nozzle boundary condition (i.e., Eq. (10)) to obtain the boundary residual \( B_N(\tilde{\Psi}) \). The residuals \( E_c(\tilde{\Psi}) \) and \( B_N(\tilde{\Psi}) \) were required to satisfy the following orthogonality condition (see Ref. 11):

\[ \int_{0}^{2\pi} \int_{0}^{1} \int_{0}^{1} E_c(\tilde{\Psi}) Z_j^*(z) \Theta_j(\theta) R_j(r) \ r \ r \, d\theta \, dz = 0 \]  

(19)

\[ - \int_{0}^{2\pi} \int_{0}^{1} B_N(\tilde{\Psi}) Z_j^*(z_e) \Theta_j(\theta) R_j(r) \ r \ r \, d\theta = 0 \]

\( j = 1, 2, \ldots, N \)
where the $Z_j^*$ are the complex conjugates of the axial acoustic eigenfunctions appearing in Eq. (18), and $\Theta_j$ 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, $B_p(t)$:

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

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

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

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

In the above equation, the term $C_4(j,p)e^{ik_p at}$ results from the presence of nozzle nonlinearities (i.e., the term involving $\Gamma_p$ 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, $\vec{\Phi}$, is calculated from Eq. (18).

The pressure perturbation at any location within the chamber is related to
by the following second-order momentum equation (see Ref. 11):

\[ p' = -\gamma \left[ \frac{\tilde{z}'}{t} + u\tilde{z}' + \frac{1}{2} \left( \tilde{z}' \right)^2 + \frac{1}{2r^2} \left( \tilde{z}' \right)^2 + \frac{1}{2} \left( \tilde{z}' \right)^2 - \frac{1}{2} \left( \tilde{z}' \right)^2 \right] \]  (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, \( \text{LCYC3D} \), 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 Adams-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, \( \omega \) is the frequency of the 1T mode, while the frequency of the 2T and 1R modes is \( 2\omega \) 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.

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 \( \Gamma_r \) and \( \Gamma_i \) increase with increasing pressure amplitude of the 1T mode, which acts as a driving force. It is observed that the absolute values of \( \Gamma_r \) and \( \Gamma_i \) 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 $\theta_1$ 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 $\theta_1$).

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 $\tau$, (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
Figure 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 \( n-\tau \) plane in Figure 11. Since it was desired to study the limit-cycle behavior of the motor, the values
of \( n \) and \( \tau \) considered were chosen from the unstable region of this stability
diagram.

Limit-cycle amplitudes and waveforms were calculated for \( \tau = 1.6 \)
(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 \( \tau = 1.6 \). 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
\( \tau \) 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 cal-
culating 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 $\tau$. 

$M_e = 0.2, L/D = 0.5, n - n_o = 0.1$
Figure 16. Comparison of Limit-Cycle Amplitudes for Different Values of $M_e$. 

- $n-n_0 = 0.1$
- $\tau = 1.6$
- $L/D = 0.5$
Figure 17. Comparison of Limit-Cycle Amplitudes for Different Values of L/D.
by the LT 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 LT 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 LT 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 COEFS3D (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 COEFS3D.

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 \( \zeta_p \) needs to be solved. Thus, the equations governing \( \zeta_p \) 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 (\( \zeta_1 \)) and the amplitude of the fundamental mode (\( A_1 \)) are first solved to obtain these quantities at
Figure A-1. Flow Chart.
every station in the nozzle. In the subsequent steps, the equations for $\zeta$ and $\Gamma$ 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.

<table>
<thead>
<tr>
<th>No. of Cards</th>
<th>Location</th>
<th>Type</th>
<th>Input Item</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1-10</td>
<td>F</td>
<td>CM</td>
<td>Mach number at the nozzle entrance</td>
</tr>
<tr>
<td></td>
<td>11-20</td>
<td>F</td>
<td>ANGLE</td>
<td>Nozzle half-angle</td>
</tr>
<tr>
<td></td>
<td>21-30</td>
<td>F</td>
<td>RCC</td>
<td>Radius of curvature of the nozzle at the entrance</td>
</tr>
<tr>
<td></td>
<td>31-40</td>
<td>F</td>
<td>RCT</td>
<td>Radius of curvature of the nozzle at the throat</td>
</tr>
<tr>
<td></td>
<td>41-50</td>
<td>F</td>
<td>GAM</td>
<td>Ratio of specific heats</td>
</tr>
<tr>
<td>1</td>
<td>1-5</td>
<td>I</td>
<td>NOZNL1</td>
<td>If 0: nonlinear admittances are not evaluated</td>
</tr>
<tr>
<td></td>
<td>6-10</td>
<td>I</td>
<td>NOUT</td>
<td>If 1: nonlinear admittances are evaluated</td>
</tr>
</tbody>
</table>

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
| 1-10 | F | WC | Frequency of oscillation
| 11-20 | F | PLAMPL | Pressure amplitude of the fundamental mode. Omit if only linear admittances are needed.

The nozzle parameters ANGLE, RCC and RCT correspond to $\theta_l$, $r_{cc}$ 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 COEFS3D. If, instead of providing this data to program COEFS3D 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 COEFS3D 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 KSTDY 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

$$u_p(\theta) = \cos(m_\theta \theta),$$

the integrals are as follows:

$$\text{ALPHA (1, p)} = \int_0^{2\pi} \left[ u_p(\theta) \right]^2 u_1(\theta) \, d\theta$$

$$\text{ALPHA (2, p)} = \int_0^{2\pi} \left[ u_p'(\theta) \right]^2 u_1(\theta) \, d\theta$$

$$\text{ALPHA (3, p)} = \int_0^{2\pi} \left[ u_p''(\theta) \right] u_p(\theta) u_1(\theta) \, d\theta$$

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

$$\text{ALPHA (5, p)} = \int_0^{2\pi} \left[ u_p''(\theta) \right] u_p(\theta) \, d\theta$$
The integrals involving Bessel functions are as follows:

\[
\begin{align*}
\text{BETA} (1, \, p) &= \int \left[ R_1(r) \right]^2 R_1(r) \, r \, dr \\
\text{BETA} (2, \, p) &= \int \left[ R_p(r) \right]^2 R_1(r) \frac{1}{r} \, dr \\
\text{BETA} (3, \, p) &= \int \left[ R'_p(r) \right]^2 R_1(r) \, r \, dr \\
\text{BETA} (4, \, p) &= \int \left[ R''_p(r) \right] R_p(r) R_1(r) \, r \, dr \\
\text{BETA} (5, \, p) &= \int \left[ R'_p(r) \right] R_p(r) R_1(r) \, dr \\
\text{BETA} (6, \, p) &= \int \left[ R_p(r) \right]^2 \, r \, dr \\
\text{BETA} (7, \, p) &= \int \left[ R'_p(r) \right] R_p(r) \, dr \\
\text{BETA} (8, \, p) &= \int \left[ R''_p(r) \right] R_p(r) \, r \, dr \\
\text{BETA} (9, \, p) &= \int \left[ R_p(r) \right]^2 \frac{1}{r} \, dr
\end{align*}
\]

Here \( R_p(r) = J_m \left[ S_{mn} r \right] \) where \( m \) and \( n \) are the transverse mode numbers for the \( p \)th 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 (ANGE), 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 (NPIAST). 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 COEFS3D.

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 COEFS3D. 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 COEFS3D, which is the next program to be executed.
*************** PROGRAM NOZADM **********************

THIS PROGRAM EVALUATES THE LINEAR AND NONLINEAR ADMITTANCES OF A SPECIFIED NOZZLE.

THE FOLLOWING INPUTS ARE REQUIRED:

CM IS THE MACH NUMBER AT THE NOZZLE ENTRANCE.
ANGLE IS THE SLOPE OF THE MIDDLE SECTION OF THE NOZZLE.
RCC IS THE RADIUS OF CURVATURE OF THE NOZZLE AT THE ENTRANCE.
RCT IS THE RADIUS OF CURVATURE AT THE THROAT.
GAM IS THE SPECIFIC HEATS RATIO.

NOZNL1 DETERMINES WHETHER THE NONLINEAR ADMITTANCES ARE TO BE EVALUATED:
NOZNL1 = 0 NOT EVALUATED.
NOZNL1 = 1 EVALUATED.

NOUT DETERMINES THE OUTPUT:
NOUT = 0 PRINTED OUTPUT ONLY.
NOUT = 1 PRINTED AND WRITTEN INTO A FASTRAND FILE.
NOUT = 2 PRINTED AND ADMITTANCES Punched INTO CARDS.

IEXTN DETERMINES IF THERE IS AN EXTENSION SECTION
IEXTN = 0 NO EXTENSION SECTION.
IEXTN = 1 THERE IS AN EXTENSION SECTION.

EXTNSN IS THE LENGTH OF THE EXTENSION SECTION.

WC IS THE FREQUENCY OF THE FUNDAMENTAL MODE.
PFIAMPL IS THE PRESSURE AMPLITUDE OF THE FUNDAMENTAL MODE.

COMMON
/X1/CM, ANGLE, RCC, RCT, GAM, G, RT, DF
/X2/T, R1, R2, NPLAST, NEND, IEXTN
/X3/WC, SUN, IP, MODE, NO, KF(3)
/X4/RUK(7), DUK(7), ZTHR1, GTHR1
/X5/(UK(1000), EU(1000), C(1000), RUK(1000))
/X6/AFN1, AFN2
/X7/ALPHA(5,3), BETA(9,3)
/X8/ZRBC(1000)
COMPLEX
AFN1(1000), AFN2(1000), AFN2(1000), ACHME, CONST
/C(25), CC1(25), CFH, CFM, CFN, CGF1, CFK2
/INHMI, INHMG1, ZTHR1, GTHR1, AH1, AHR1, GHR1, GHR1
/ZETA, TAU, LINAEM, ZRK, CNX(3)
DIMENSION GC(4), GP(4), Y(4), DY(4, 4), SMN(3), ISTEP(3)
/NM(3), PHASE(3), PAMPL(3)
DATA (NAME(MODE), MODE = 1-3) /SH17, 2H27, 3H17/
/(SMN(MODE), MODE = 1-3) /1.84118, 3, 05424, 3, 83171/
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  OBTAIN STEADY-STATE QUANTITIES IN THE NOZZLE.
CALL STEADY
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  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 NF = NPLAST, NEND
U(NF) = U(NPLAST)
C(NF) = C(NPLAST)
DU(NF) = DU(NPLAST)
RW(NF) = RW(NPLAST)
20  CONTINUE
25  CONTINUE
IF (NEND * GT. 1000) GO TO 550

C  CALL INTEGR
SRTR=(RT*RCT)**5
C  ACHMGR = CMPLX (F1AMFL / (WC*GAM) ** 0.)
IF (NOUT * EQ. 0) WRITE (6,1050) WC,F1AMFL
IF (NOUT * EQ. 0) WRITE (6,1055)

C  DO 500 MODE=1,3
IP=ISTEP(MODE)
SVN=SKN(MODE)
SUNR=SUN/RT

C**********************************************************************STARTING \VALUES SECTION**********************************************************************

C

P=0.
AH1 = 1.
AH1 = 0.
AH1 = CMFLX (AH1, AH1)
UP = U(1)
CP = C(1)
DUP = DUP(1)
RFP = RF(1)
CALL COEFFS (UP, DUP, CP, RFP, CC)
CFH = CC(1)
CFM = CC(2) + CC(6)
CFN = CC(3) + CC(4) + CC(5) + CC(7) + CC(8)

C**********************************************************************DERIVATIVES OF THE COEFFICIENTS AT THE THROAT**********************************************************************

C

EVALUATE DERIVATIVES OF LINEAR COEFFICIENTS:
XR = - 4./(GPL1 * SRTR)
CFH1 = CMFLX (XR, 0.)
XR = (24. + 4. * GAM) / (GPL1 * 3. * RT * RCT)
XI = - 8. * WC * KP(MODE) / (GPL1 * SRTR)
CFM1 = CMFLX (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 = CMFLX (XI, XI)

C

SET UP VALUES AT THE THROAT BY TAYLORS EXPANSION

C

STARTING VALUES FOR ZETA
ZTHR = - CFN / CFM
ZTHR1 = - (CFN1 * ZTHR + CFH1 * ZTHR + ZTHR + CFN1) / (CFH1 + CFM)
ZTH(1) = ZTHR

C

IF (MODE.NE.1) GO TO 110
AFN(1) = AH
AFN(1) = AFN(1) * ZTHR
AFN(2) = AFN(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) = AH1
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)
INHM = -CC(18) * AFN(1) * AFN2(1) - CC(12) * AFN1(1) * AFN2(1)
1 = -CC(9) + CC(15) * AFN1(1) * AFN2(1) - CGRP1 * AFN(1) * AFN(1)
2
EVALUATE DERIVATIVES OF NON-LINEAR COEFFICIENTS.
AIB1 = 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)
LO 26  J = 1*25
CC(I,J) = CMPLX (0.,0.)
XR = (2.*A1B1 * WC) / (A4B6 * GPLI * SRTR)
XI = XR
CC(9) = CMPLX (XR, XI)
XR = (4.* A1B1) / (3.*A15927 * GPLI * SRTR * A4B6)
XI = XR
CC(12) = CMPLX (XR, XI)
XR = - A1B3 / (GFLI * RT * RT * SRTR * A4B6)
XI = XR
CC(13) = CMPLX (XR, XI)
XR = - A2B2 / (GFLI * RT * RT * A4B6 * SRTR)
XI = XR
CC(14) = CMPLX (XR, XI)
XI = - XR
CC(15) = CMPLX (XR, XI)
XR = A1B3 * (9.* - 2.*GAM - GAM*GAM) / (12.* RT**3 * RCT * GPLI
1 * A4B6)
XI = - XR
CC(16) = CMPLX (XR, XI)
1 * A4B6)
XI = - XR
CC(17) = CMPLX (XR, XI)
XR = - (GMINI * WC * A1B1) / (GFLI * SRTR * A4B6)
XI = XR
CC(18) = CMPLX (XR, XI)
XR = - (GMINI * WC * A1B1) / (3.* GPLI * RT * RCT
1 * A4B6)
XI = XR
CC(19) = CMPLX (XR, XI)
XR = - (GMINI * ALPHA(1,MODE) * (BETA(4,MODE) - BETA(5,MODE))) / (GFLI * RT * RT * SRTR * A4B6)
XI = XR
CC(23) = CMPLX (XR, XI)
XR = - (GMINI * ALPHA(1,MODE) * BETA(5,MODE) * 2.*
1 / (GFLI * RT * RT * SRTR * A4B6)
XI = XR
CC(24) = CMPLX (XR, XI)
XR = - (GMINI * ALPHA(3,MODE) * BETA(2,MODE))
1 / (GFLI * RT * RT * SRTR * A4B6)
XI = XR
51


icio (25) = CMPLX (X, X)

INHMGI = - AFN2(1) * AFN2(1) * CC1(12) - AFN1(1) * AFN2(1) * 1
(CC1(18) + CC1(12) + 2 * CC1(9) + 2 * CC1(15)) - AFN2(1) * AFN1(1) * 2
AFN(1) * (CC1(18) + CGRP1) - AFN1(1) * AFN1(1) * 3
CC1(19) + CC1(23) + CC1(24)
+ CC1(25) + 2 * CGRP2) - AFN1(1) * AFN1(1) * (CC1(12) + 4
+ CC1(11) + CC1(17) + CC1(20) + CC1(21) + CC1(22))

---------------

STARTING VALUES FOR GAMMA:
GTHR = INHMG / (CF * CFM)
GTHR1 = (CF * GTHR * (GTHI + ZTHR + CFM)) / (GMIN * 5 * 4)
1
(1 + GLP1 * ZTHR) * GTHR * (GTHI + CFM) - INHMGI)
2
(CF + CFH + CF + CFM)

---------------

RUNGE-KUTTA INTEGRATION TO PROVIDE INITIAL VALUES FOR PREDICTOR-CORRECTOR INTEGRATION:
DO 30 IRK = 3, 4
CALL RKTZ (DP, F, G, GP, IRK)
P = P + DP
ZR = G(1)
ZI = G(2)
ZR(1) = CMPLX (Z, ZI)
DY(1) = GP(1)
DY(2) = GP(2)
GO TO (150, 160, 170), IP
160
AH = G(3)
AH1 = G(4)
DY(3) = GP(3)
DY(4) = GP(4)
IF (MODE = NE) GO TO 162
AFN (1) = CMPLX (G(3)*G(4))
AFN1 (1) = CMPLX (G(3)*G(4))
AF1 = G(1)*GP(3) + G(2)*GP(4) + G(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(1) = CMPLX (AF1, AI2)
162 GO TO 150
170 CONTINUE
GAMR = G(3)
GAM1 = G(4)
DY(3) = GP(3)
DY(4) = GP(4)
150 CONTINUE
30 CONTINUE
Y(1) = ZR

---------------
Y(2)=ZI
GO TO (160,190,200)* IP
190  Y(3) = AHR
Y(4) = AMI
GO TO 180
200  CONTINUE
Y(3) = GAMR
Y(4) = GAMI
180  CONTINUE
C
C  PREDICTOR-CORRECTOR INTEGRATION
CALL ZADAMS (DP,P,Y,LP,ITORZ)
C
C  CALCULATE LINEAR ADMITTANCE COEFFICIENTS.
UE = UCNEND)
CE = C(NEND)
RHOE = CE ** (1./GMIN1)
FR = WC * KF(MODE)
F = UE ** .5 / (FR*GAM)
IF (ITYOZ *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)
LINAIM = F * CMPLX(0.,1.) / TAU
40  CONTINUE
YR = REAL (LINADM)
YI = AIMAG (LINAIM)
YMAG = CABS (LINAIM)
YPHASE = ATAN2 (YI,YR) * 180. / 3.1415927
AMPL(MODE) = YMAG
PHASE(MODE) = YPHASE
C
GO TO (210,220,230)* IP
220  AHR = Y(3)
AMI = Y(4)
IF (MODE *NE. 1) GO TO 210
CONST = ACHMER / 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.
GYY = 0.0
GNOZ(1) = (0.0,0.0)
GO TO 210
CONTINUE
CALCULATE NONLINEAR ADMITTANCE COEFFICIENTS
GAMR = Y(3)
GAMI = Y(4)
GMAG = (GAMR * GAMR + GAMI * GAMI) ** .5
GPHASE = ATAN2 (GAMI,GAMR) * 180. / 3.1415927
GBYY = CMABS(CMPLX(GAMR,GAMI)/LINADM)
GNOZ(MODE) = CMPLX(GAMR,GAMI)
CONTINUE
IF (NOUT .EQ. 0) WRITE (6,1060) NAME(MODE), YR, YI,
1 YMAG, YPHASE, GAMR, GAMI, GMAG, GPHASE, GBYY
CONTINUE
CONTINUE
CONTINUE
CONTINUE
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)
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)
CONTINUE
WRITE (6,1065)
C
************** READ FORMAT SPECIFICATIONS **************
C
5005 FORMAT (6F10.0)
5010 FORMAT (3I5,F10.0)
5015 FORMAT (2F10.0)
C
************** WRITE FORMAT SPECIFICATIONS **************
C
1005 FORMAT (1H1,//////,45X,17H**************,//////)
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**************,//////)
18HNOZZLE ADMITTANCES///46X,18H**************,//////)
20X,"FREQUENCY = ",F8.6,40X,PRESSURE AMPLITUDE = ",F6.4)
1055 FORMAT (/////////5X,"MODE" 10X,2HYF,9X,2HYI,9X,"YMAG",9X,"YPHASE",
//

ORIGINAL PAGE IS OF POOR QUALITY
1  1 1X, 2HGR, 9X, 2H61, 9X, 4HGMAG, 10X, 6H6FHPASE, 13X, 3H6Y, /)
1060 FORMAT (1H0, 5X, A2, 2X, 3F12.4, F16.4, 3F12.4, 2F16.4)
1065 FORMAT (1H1)
7005 FORMAT (15, 2F10.5)
C
C  ***********************************************
C
C STOP
C END
SUBROUTINE STEADY

THIS SUBROUTINE EVALUATES STEADY-STATE QUANTITIES IN THE NOZZLE.

NOZZLE PROFILE AND FLOW PARAMETERS ARE PASSED TO THE SUBROUTINE
THROUGH THE COMMON BLOCKS X1 AND X2.

THE SUBPROGRAM PROVIDES THE OUTPUT THROUGH COMMON BLOCK X5.

U IS THE SQUARE OF THE STEADY-STATE VELOCITY;
DU IS THE DERIVATIVE OF U WITH RESPECT TO STEADY-STATE POTENTIAL;
C IS THE SQUARE OF THE SPEED OF SOUND;
Rw IS THE RADIUS OF THE NOZZLE.

THESE OUTPUT QUANTITIES ARE STORED IN THE RESPECTIVE ARRAYS AT
INTERVALS OF DP IN P (STEADY-STATE POTENTIAL).

COMMON /X1/ CM, ANGLE, RCC, RCT, GAM, O, RT, DP
COMMON /X2/ T, R1, R2, NPLAST, NEND, IEETN
COMMON /X4/ RK(1), RDU(1), ZTH1, GTH1
COMMON /X5/ UX(1000), DU(1000), C(1000), RW(1000)

T = 3*1415927*ANGLE/180
RT = (CM**2.5) * ((1+(GAM-1.)*CM**2/2.) ** ((-GAM-1.)/
((4*(GAM-1.))*((2/(GAM+1.)) ** ((-GAM-1)/(4*(GAM-1.))))
SRTR = (RT*RCT)**5.5
G = (.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.)
RUC(1) = U(1)
C(1) = U(1)
DUC(1) = 4./(GAM+1.)*SRTR
REU(1) = DUC(1)
G = U(1)
DO 30 I=2,7
CALL RKSTDY (P, G, GP)
P = P + DP/2
RUC(I) = G
RDU(I) = GP
IF (I.EQ.9) 2*(1/2) GO TO 30
IF (I.EQ.9) 2*(1/2) GO TO 30
NF = (I+1)/2
U(NF) = RUC(I)
DUC(NF) = RDU(I)
C(NF) = 1.-(GAM-1.)*U(NF)**5
RW(NF) = G**((C(NF)) ** (-1./(2.*(GAM-1.))))
(1.)*(U(NF)**5+2.5)**4.
30 CONTINUE
CALL UADAMS (P)
RETURN
END
SUBROUTINE RKSTDY(P, G, DUM)

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 HKUDIF.

C

COMMON /XI1/ CM, ANGLE, RCC, RCT, GAM, Q, RT, DP
DIMENSION A(4), FZ(4)

A(1) = 0.
A(2) = 0.5
A(3) = 0.5
A(4) = 1.
H = DP/2.
PR = P
GR = G
CALL HKUDIF(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 HKUDIF(PR, GR, DUM)
FZ(I) = DUM
30 CONTINUE
G = G + H* (FZ(1) + 2*(FZ(2)+FZ(3)) + FZ(4))/6.
CALL HKUDIF(PR, G, DUM)
RETURN
END
SUBROUTINE RKUDIF(F,S,G,F)
C
THIS SUBROUTINE EVALUATES THE DIFFERENTIAL ELEMENT IN THE
RUNGE-KUTTA INTEGRATION SCHEME FOR SOLVING THE EQUATION FOR SQUARE
OF STEADY-STATE VELOCITY.
C
P IS THE VALUE OF STEADY-STATE POTENTIAL AT THE STATION,
WHERE DIFFERENTIAL ELEMENT IS SOUGHT; INPUT.
G IS THE VALUE OF THE FUNCTION AT F; INPUT.
SP IS THE REQUIRED DIFFERENTIAL ELEMENT.
C
COMMON /X1/ CK,ANGLE,RCC,RCT,GAM,0,KT,DP
COMMON /X2/ T,R1,R2,NPLAST,NEND,IEXTN
COMMON /X3/ WC,SUN,IP,MODE,NU,KF(3)
C
10 IF (P) 15,10,15
G = 4.* ((GAM+1.) * ((RCT*RT) **.5))
GO TO 20
C = 1.-(GAM-1.)*G+.5
R = G*(((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) **.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-1.)*(R-1.)) **.5) / (1.-R.*RCC)
45 DU = -(G**-.75)*C**(2.*GAM-1) / (2.*GAM-1.)**(2.)
1
GO = DU*DR
GO TO 20
50 GP = 0.
20 RETURN
END
SUBROUTINE UADAMSCP

C THIS SUBROUTINE CARRIES OUT A MODIFIED ADAMS PREDICTOR-CORRECTOR INTEGRATION SCHEME TO SOLVE THE DIFFERENTIAL EQUATION FOR THE STEADY-STATE VELOCITY.

C P IS THE VALUE OF THE STEADY-STATE POTENTIAL AT THE STATION 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 THE STEADY-STATE QUANTITIES ARE THE OUTPUT, AND ARE PROVIDED BY MEANS OF COMMON BLOCK X5.

C

COMMON /X1/ CM,ANGLE,RCC,RCT,GAM,O,RT,H
COMMON /X2/ T,R1,R2,NPLAST,NEND,EXTN
COMMON /X5/ UC(1000),DU(1000),C(1000),RW(1000)

C NP=4 CONTINUE

PRED = U(NP) + H*(55.*DU(NP) - 59.*DU(NP-1) + 37.*DU(NP-2) + -9.*DU(NP-3))/24.0
P = P + H
NP = NP + 1
UP = PRED
CP = 1-(GAM-1.)*UP**5
R = Q*(CF**(-1./ (2.*(GAM-1.)))) * (UP**-25)**4.

C IF R = 1, THE NOZZLE ENTRANCE HAS BEEN REACHED.
IF (R=1.) 17,17,100

17 IF (R-R1) 20,25,25
20 DR = -(2.*RCT*(R-RT) - (R-RT)*(R-RT)**5) / (RT+RCT-R)
60 TO 40
25 IF (R-R2) 30,35,35
30 DR=TAN(T)
60 TO 40
35 DR = ((2.*RCC*1.-1) - (1.-R)*(1.-R)**5) / (1.-R-RCC)
40 DO = -(UP**-75) * (CP**((2.*GAM-1) / (2.*(GAM-1)))) / (Q*(1.-GAM)*UP**5))
60 DUP = DR*DO
COR = U(NP-1)+H* (9.*DUF+19.*DU(NF-1) + 5.*DU(NF-2) +DU(NP-3))/24.0
UP = (251.*COR + 19.*PRED) / 270.
CP = 1-*(GAM-1)*UP**5
R = Q*(CF**(-1./ (2.*(GAM-1.)))) * (UP**-25)**4.

C IF R = 1, THE NOZZLE ENTRANCE HAS BEEN REACHED.
IF (R=1.) 62,62,100

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 \[ \text{DR} = \tan(T) \]
    GO TO 85
80 \[ \text{DR} = \left( (2 \times \text{RCC} \times (1-R) - (1-R) \times (1-R)^{**.5}) / (1-R-HCC) \right) \]
85 \[ \text{DO} = (-((\text{UF}^{**.75}) \times (\text{CP}^{**((2 \times \text{GAM}^{}-1)) / (2 \times (\text{GAM}^{-1})))) / \right) \]
    \left( \text{C}^{*(1-(\text{GAM}+1))} \times \text{UF}^{* 5} \right) \]
    IF (NP \*6T\* 1000) GO TO 87
C
C STORE STEADY STATE QUANTITIES AT STATION NP IN RESPECTIVE ARRAYS.

DU(NP) = DR\*DO
U(NP) = UF
C(NP) = CP
RW(NP) = R
C
87 GO TO 10
100 NPLAST = NP\+1
RETURN
END
SUBROUTINE COEFFS (U, DU, C, R, CC)

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 INTEGRAL PROVIDES ALPHA & BETA, THE VALUES OF TRANSVERSE
C INTEGRALS THROUGH COMMON BLOCK X7.

COMMON /X3/ WC, SUN. IP. MODE, NU, KP(3)
COMMON/X7/ ALPHA(5, 3), BETA(9, 3)
COMPLEX CC(25)
DATA GAM/1.21
C
C = GAM - 1.*
M = MODE
A4B6 = ALPHA (4, M) * BETA (6, M)
RSQR = R * R

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)

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********** 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
CCI(10) = CMPLX (CCR, CCI)

CCR = - 5 * A2*B2 * WC / (RSQR * A4B6)
CCI = CCR
CCI(11) = CMPLX (CCR, CCI)
CCI = CCR
CCI(12) = CMPLX (CCR, CCI)
CCR = - (U * A1 * B3) / (4. * RSQR * A4B6)
CCI = CCR
CCI(13) = CMPLX (CCR, CCI)
CCR = - (U * A2 * B2) / (4. * RSQR * A4B6)
CCI = CCR
CCI(14) = CMPLX (CCR, CCI)
CCR = - 3*U * (1. + 5*GMIN1 * U*DU/C) * A1*B1 / (8.*A4B6)
CCI = CCR
CCI(15) = CMPLX (CCR, CCI)

CCR = - DU * ((1 + (2.*GAM) / U/C) * A1 * B3 / (16 * RSQR * A4B6)
CCI = - CCR
CCI(16) = CMPLX (CCR, CCI)
CCR = - DU * (1. + (2.*GAM) / U/C) * A2 * B2 / (16 * RSQR * A4B6)
CCI = - CCR
CCI(17) = CMPLX (CCR, CCI)
CCR = - (GMIN1 * WC * A1 + B1) / (4. * A4B6)
CCI = CCR
CCI(18) = CMPLX (CCR, CCI)
CCR = - (GMIN1 * WC * U* DU + A1 + B1) / (4. * C * A4B6)
CCI = CCR
CCI(19) = CMPLX (CCR, CCI)
CCR = - GMIN1 * WC * A1 + (B4 - B5) / (4. * RSQR * A4B6)
CCI = CCR
CCI(20) = CMPLX (CCR, CCI)

CCI = CCR
CCI(21) = CMPLX (CCR, CCI)
CCI = CCR
CCI(22) = CMPLX (CCR, CCI)
CCR = - GMIN1 * U*A1.* (B4 - B5) / (4. * RSQR * A4B6)
CCI = - CCR
CCI(23) = CMPLX (CCR, CCI)
CCR = - GMIN1 * U * A1 * B5 / (2.*RSQR * A4B6)
CCI = - CCR
CCI(24) = CMPLX (CCR, CCI)
CCR = - GMIN1 * U * A3 * B2 / (4.*RSQR * A4B6)
CCI = - CCR
CCI(25) = CMPLX (CCR, CCI)

CONTINUE
RETURN
END
SUBROUTINE INTOBL

This subroutine evaluates the different transverse integrals.

COMMON/X7/ ALPH(A(5,3), BETA(9,3)
S1 = 1.84118
S2 = 3.05424
S3 = 3.83171
PI = 3.1415927

***************TANGENTIAL INTEGRALS**********************

DO 20 NOPT = 1, 3
   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
   ALPHA(I, J) = PI*ALPHA(I, J)

***************RADIAL INTEGRALS***************

DO 40 MODE = 1, 3
   GO TO (110, 120, 130), MODE

100 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, M, S, S1, S1, S)
   BETA( 2, MODE) = RAD2( 2, 1, M, S, S1, S1, S)
   BETA( 3, MODE) = RAD2( 7, 1, M, S, S1, S1, S)
   BETA( 4, MODE) = RAD2( 8, 1, M, S, S1, S1, S)
   BETA( 5, MODE) = RAD2( 5, 1, M, S, 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)

THIS SUBROUTINE CALCULATES THE INTEGRAL OVER THE INTERVAL (0, 1) OF THE FOLLOWING PRODUCTS OF TWO BESSEL FUNCTIONS

\[
\begin{align*}
NOPT = 1 & \quad J_{M}(B*R) \times J_{M}(B*R) \times R \\
NOPT = 2 & \quad J_{M}(B*R) \times J_{M}(B*R)/R \\
NOPT = 3 & \quad J_{M}(B*R) \times J_{M}(B*R) \times R \\
NOPT = 4 & \quad J_{M}(B*R) \times J_{M}(B*R) \\
NOPT = 5 & \quad J_{M+1}(B*R) \times J_{M}(B*R) \times R
\end{align*}
\]

\(J_{M}\) IS THE BESSEL FUNCTION OF FIRST KIND OF ORDER M
\(J_{M+1}\) IS THE DERIVATIVE OF \(J_{M}\) WITH RESPECT TO R
\(J_{M+2}\) IS THE SECOND DERIVATIVE OF \(J_{M}\) WITH RESPECT TO R
M IS A NON-NEGATIVE INTEGER
B IS A REAL NUMBER

DIMENSION FUNCT(200)
DOUBLE PRECISION DN, DH, DSTEP, DR, ARG, BES1, BES2, BESH, 
1 BESL, PROD, FUNCT, S1, S2, S3

NN = 100
DN = NN
DH = 1.0/DN
NP1 = NN + 1

************** CALCULATION OF INTEGRANDS **************

DO 160 I = 1, NP1
DSTEP = I - 1
DR = DH * DSTEP
ARG = B * DR

CALCULATE BESSEL FUNCTIONS.
CALL JBES(M, ARG, BES2, $500)
BES1 = BES2
IF (NOPT .LT. 3) GO TO 130

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

64
GO TO 130

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)

130 PROD = BES1 * BES2

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
GO TO 160

160 CONTINUE

C *************** SIMPSONS RULE INTEGRATION **********************

NM1 = NN - 1
S1 = FUNCT(1) + FUNCT(NN1)
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
RADI = RESULT
GO TO 501
500 WRITE (6,6000)
6000 FORMAT (1H1,10HEXERROR JBES)
501 CONTINUE
RETURN
END
FUNCTION RAD2 (NOPT, L, M, N, A, B, C)

THIS SUBROUTINE CALCULATES THE INTEGRAL OVER THE INTERVAL (0, 1) OF THE FOLLOWING PRODUCTS OF THREE BESSSEL FUNCTIONS

NOPT = 1  JL(A*R) * JM(B*R) * JN(C*R) * R
NOPT = 2  JL(A*R) * JM(B*R) * JN(C*R)/R
NOPT = 3  JL(A*R) * JM(B*R) * JN(C*R)/R2
NOPT = 4  JPL(A*R) * JM(B*R) * JN(C*R) * R
NOPT = 5  JPL(A*R) * JM(B*R) * JN(C*R)
NOPT = 6  JPL(A*R) * JM(B*R) * JN(C*R)/R
NOPT = 7  JPL(A*R) * JPM(B*R) * JN(C*R) * R
NOPT = 8  JPPL(A*R) * JM(B*R) * JN(C*R) * R
NOPT = 9  JPPL(A*R) * JPM(B*R) * JN(C*R) * R

JL IS THE BESSSEL FUNCTION OF FIRST KIND OF ORDER L
JPL IS THE DERIVATIVE OF JL WITH RESPECT TO R
JPPL IS THE SECOND DERIVATIVE OF JL WITH RESPECT TO R
L, M, N ARE NON-NEGATIVE INTEGERS
A, B, C ARE REAL NUMBERS

DIMENSION FUNCT(200)
DOUBLE PRECISION DD, DH, DSTEP, DR, ARG1, ARG2, ARG3,
                BES1, BES2, BES3, BESH, BESL, PROD,
                FUNCT, BESLIM, S1, S2, S3

NN = 100
IN = NN
DH = 1.0/IN
NP1 = NN + 1

************** CALCULATION OF INTEGRANDS **************

DO 160 I = 1, NP1
    DSTEP = I - 1
    DR = DH * DSTEP
    ARG1 = A * DR
    ARG2 = B * DR
    ARG3 = C * DR

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

56
C CALCULATE FIRST DERIVATIVES OF BESSEL FUNCTIONS.

105 CALL JBE(M+1,ARG,BESL,$500)
   IF (I .EQ. 1) GO TO 107
   RM = M
   BES2 = B* (RM* BES2/ARG1 - BES1)
   GO TO 110
107 IF (M .EQ. 0) GO TO 109
   CALL JBE(M-1, ARG, BESL,$500)
   BES2 = B * (BESL - BESH)/2.0
   GO TO 110
109 CALL JBE(1, ARG, BESL,$500)
   BES2 = -BES2 * B

110 IF (NOFT .LT. 4) GO TO 130
   CALL JBE(L+1, ARG, BESL,$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 JBE(L-1, ARG, BESL,$500)
   BES1 = A * (BESL - BESH)/2.0
   GO TO 130
117 CALL JBE(1, ARG, BESL,$500)
   BES1 = -BES1 * A
   GO TO 130

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 JBE(L+2, ARG, BESL,$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 JBE(L-2, ARG, BESL,$500)
   BES1 = 0.25 * A * A * (BESL - 2.0*BES1 + BESH)

130 PROD = BES1 * BES2 * BES3

C CALCULATE WEIGHTING FUNCTIONS AND LIMITS FOR R = 0.

133 IF (NOPT .EQ. 6) GO TO 136
   IF (NOPT .EQ. 5) GO TO 140
   FUNCT(I) = PROD * DR
   GO TO 160
134 BESLIM = 0.0

135 IF (L .EQ. 1) GO TO 134
   FUNCT(I) = PROD/DR
   GO TO 160
136 IF (NOPT .EQ. 6) BESLIM = A/2.0
137 IF (L .EQ. 1) BESLIM = B/2.0
138 IF (L .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.2) AND (M.EQ.0) AND (N.EQ.1)) BESLIM = A*C/4.0
   GO TO 155
136 IF (1.EQ.1) GO TO 135
   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.3) 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
*************** SIMPSONS RULE INTEGRATION **********************
C
C
NM1 = NN - 1
S1 = FUNCT(I) + FUNCT(NF1)
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
   RAB2 = RESULT
   GO TO 501
500 WRITE (6, 6000)
6000 FORMAT (1H1, 10ERROR JBES)
501 CONTINUE
RETURN
SUBROUTINE RKTZ(H, TI, 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 TI 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 = TI
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, IK)
DO 25 J = 1, NU
25 FZ(1, J) = DUM(J)
DO 30 IK = 2, 4
TZ = TI + 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, IK)
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, IK)
75 RETURN
END
SUBROUTINE RKDIF(F, G, GP, IK, IFK)
C
C THIS SUBROUTINE EVALUATES THE DIFFERENTIAL ELEMENT IN THE
C RUNGE-KUTTA INTEGRATION SCHEME.
C F IS THE CURRENT VALUE OF STEADY-STATE POTENTIAL; INPUT.
C G ARE THE VALUES OF THE FUNCTIONS AT F; INPUT.
C GP ARE THE DERIVATIVES OF FUNCTIONS AT F; OUTPUT.

COMMON /XI/ CM*R,ECT, GAM, Q, RT, DP
COMMON /X2/ TR, R, NPLAST, NEND, IEXTN
COMMON /X3/ WC, SVN, IP, MODE, NUP, KP(3)
COMMON /X4/ HU(7), RDU(7), ZTHRI, GTHRI
COMMON /X6/ AFN, AFN1, AFN2
COMPLEX AFN(1000), AFN1(1000), AFN2(1000)
COMPLEX CC(25), CFH, CFM, CFN, INHM
COMPLEX ZETA, ZIA1, AH, AH1, CGAM, CGAM1, ZTHRI, GTHRI, AP, AP1, AF2
DIMENSION G(4), GP(4)
C
ZR = G(1)
ZI = G(2)
ZETA = CMPLX (ZR, ZI)
GO TO (110, 120, 130), IF
120 AHR = G(3)
AH1 = G(4)
AH = CMPLX (AHR, AH1)
GO TO 110
130 CONTINUE
GAMR = G(3)
GAMI = G(4)
CGAM = CMPLX (GAMR, GAMI)
110 CONTINUE
IF (F) 15, 10, 15
10 GP(1) = REAL(ZTHRI)
GP(2) = AIMAG(ZTHRI)
GO TO (140, 150, 160), IF
150 AH1 = AH * ZETA
GP(3) = REAL (AH1)
GP(4) = AIMAG (AH1)
GO TO 140
160 CONTINUE
GP(3) = REAL (GTHRI)
GP(4) = AIMAG (GTHRI)
140 CONTINUE
GO TO 20
15 ICL = 2*IRK - 2
IF (IK .EQ. 1) ICL = 2*IRK - 3
IF (IK .EQ. 4) ICL = 2*IRK - 1
U = RU(ICL)
DU = RDU(ICL)
C = 1 - (GAM-1) * U * S
R = Q * (C) ** (-1/(2*(GAM-1))) * (U**-4) * 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)
\[
\begin{align*}
ZETA1 &= (\text{-}CFM \cdot ZETA - CFN) / CFH \cdot ZETA \cdot ZETA \\
GP(1) &= \text{REAL}(ZETA1) \\
GP(2) &= \text{AIMAG}(ZETA1) \\
\text{GO TO} (170, 180, 190), \text{ IP} \\
180 &= \text{AH1} = \text{AH} \cdot ZETA \\
GP(3) &= \text{REAL}(AH1) \\
GP(4) &= \text{AIMAG}(AH1) \\
\text{GO TO} 170 \\
190 &= \text{CONTINUE} \\
\text{GO TO} (30, 40, 40, 50), \text{ IK} \\
30 &= \text{AP} = \text{AFN}(IRK-1) \\
\text{AP1} &= \text{AFN1}(IRK-1) \\
\text{AP2} &= \text{AFN2}(IRK-1) \\
\text{GO TO} 60 \\
40 &= \text{AP} = .5 \cdot (\text{AFN}(IRK-1) + \text{AFN}(IRK)) \\
\text{AP1} &= .5 \cdot (\text{AFN1}(IRK-1) + \text{AFN1}(IRK)) \\
\text{AP2} &= .5 \cdot (\text{AFN2}(IRK-1) + \text{AFN2}(IRK)) \\
\text{GO TO} 60 \\
50 &= \text{AP} = \text{AFN}(IRK) \\
\text{AP1} &= \text{AFN1}(IRK) \\
\text{AP2} &= \text{AFN2}(IRK) \\
60 &= \text{CONTINUE} \\
\text{INHMG} &= \text{-}CC(18) \cdot \text{AP} \cdot \text{AP2} - \text{CC}(12) \cdot \text{AP1} \cdot \text{AP2} - (\text{CC}(9) \\
&+ \text{CC}(15)) \cdot \text{AP1} \cdot \text{AP1} - (\text{CC}(13) + \text{CC}(14) + \text{CC}(19) \\
2 &= \text{CC}(23) + \text{CC}(24) + \text{CC}(25)) \cdot \text{AP1} \cdot \text{AP} - (\text{CC}(10) + \text{CC}(11) \\
3 &= \text{CC}(17) + \text{CC}(20) + \text{CC}(21) + \text{CC}(22)) \cdot \text{AP} \cdot \text{AP} \\
\text{CGAM1} &= (\text{-}ZETA + .5 \cdot \text{GAM-1}) \cdot \text{DU/C} - \text{CFM/CFH} \cdot \text{CGAM} \\
1 &= \text{INHMG} / (\text{C} \cdot \text{CFH}) \\
GP(3) &= \text{REAL}(CGAM1) \\
GP(4) &= \text{AIMAG}(CGAM1) \\
170 &= \text{CONTINUE} \\
20 &= \text{RETURN} \\
\text{END}
\end{align*}
\]
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 A;
C IF IP = 3, INTEGRATION IS CARRIED OUT FOR ZETA AND GAMMA.
C IP IS PASSED TO THE SUBROUTINE THROUGH COMMON BLOCK X3.
C
H IS THE STEP-SIZE; INPUT.
X IS THE VALUE OF STEADY-STATE POTENTIAL AT THE STATION;
WHERE THE PREDICTOR-CORRECTOR INTEGRATION STARTS; INPUT.
DURING THE PROGRAM, X IS CHANGED TO VALUE AT CURRENT STATION.
Y ARE THE VALUES AT X OF THE FUNCTIONS, WHOSE EQUATIONS ARE
BEING SOLVED; INPUT AND OUTPUT.
LY ARE THE DERIVATIVES OF Y; INPUT AND OUTPUT.
ITORZ PASSES TO MAIN PROGRAM THE INFORMATION AS TO WHICH VARIABLE
(TAU OR ZETA) HAS BEEN INTEGRATED.
ITORZ = 1 : INTEGRATION OF EQUATION FOR TAU.
ITORZ = 2 : INTEGRATION OF EQUATION FOR ZETA.

COMMON /X1/ CM, ANGLE, RCC, RCT, GAM, G, RT
COMMON /X2/ T, R1, R2, NPLAST, TEND, IEXT
COMMON /X3/ VQ, SVN, IP, MODE, NU, KFC(3)
COMMON /X5/ U(1000), DU(1000), C(1000), RW(1000)
COMMON /X6/ AFN, AFN1, AFN2
COMMON /X8/ ZETA, TAU, CEXT
COMPLEX ZETA, TAU, CC, C1000, RW1000)
COMPLEX AFN, AFN1, AFN2, AF1, AF2, AFN, CFH, CFM, CN, CIHMG, ZETA, AH1, AH2, AF1, AF2,
DIMENSION Y(4), DY(4, 4), DF(4), PRED(4), COR(4)

NP=4
ITORZ = 2
IF (IEXTN .NE. 1) GO TO 10

DEFINE STEADY STATE QUANTITIES IN THE EXTENSION REGION.
UEXT = U(NEND)
CEXT = C(NEND)
REXT = RW(NEND)
DUEXT = DUCNEND)
CALL COEFFS (UEXT, DUEXT, CEXT, REXT, CCEXT)
NU IS THE NUMBER OF EQUATIONS TO BE SOLVED.
10 CONTINUE
DO 15 J=1, NU
FRED(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(3)
ZI=PRED(4)
ZETA(NP) = CMPLX (ZR*ZI)
GO TO (110, 120, 130), IP
120 AHR = PRED(3)
AH1 = PRED(4)
AH = CMPLX (AHR, AH1)
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=CT(NP)
P=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
AP, AP1 AND AP2 ARE THE VALUES OF THE AMPLITUDE FUNCTION AND
THEIR DERIVATIVES AT THE CURRENT STATION.
AP = AFNX(NP)
AP1 = AFNX1(NP)
AP2 = AFNX2(NP)
C
INHMG = - CC(18) * AP * AP2 - CC(19) * AP1 * AP2 - (CC(9)
1 + CC(15)) * AP1 * AP1 - (CC(13) + CC(14) + CC(19)
2 + CC(23)) * CC(24) + CC(25)) * AP1 * AP2 - (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)

ZK=Y(1)
ZL=Y(2)
ZETA(NF) = CMPLX (ZK,ZL)
ZETA1 = (- CFM * ZETA(NF) - CFN) / CFH - ZETA(NF)**2

DY (1,4) = REAL (ZETA1)
DY (2,4) = AIMAG (ZETA1)
GO TO (170,180,190), IF

180

AH = CMPLX (Y(3),Y(4))

AH1 = AH * ZETA(NF)

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(NF) = AH1

AFN2(NF) = AH2

182

GO TO 170

190

CONTINUE

CGAM = CMPLX (Y(3),Y(4))

CGAM1 = (- ZETA(NF) + .5* (GAM-1.) * DUF/CF - CFM/CFH) * CGAM

1 = INHKG / (CF * CFH)

DY(3,4) = REAL (CGAM1)

DY (4,4) = AIMAG (CGAM1)

170

CONTINUE

IF (NP.EQ. NEND) GO TO 10

10 CONTINUE

C

IF (CABS (ZETA(NF)) *LT. 10) GO TO 10

ITORZ = 1

C

C

C

C

CALCULATE VALUE OF TAU AND ITS DERIVATIVE AT LAST FOUR STATIONS.

DO 410 I=1,4

410

TAU (NF+4+I) = 1./ZETA(NF+4+I)

Y(1) = REAL (TAU(NP))

Y(2) = AIMAG (TAU(NP))

DO 420 I=1,4

TSOR = REAL (TAU(NF-4+I) * TAU(NF-4+I))

TSOI = AIMAG (TAU(NF-4+I) * TAU(NF-4+I))

ZFR = DY(1,1)

ZPI = DY(2,1)

DY(1,1) = - TSCR*ZFR + TSOI*ZPI

DY(2,1) = - TSCR*ZPI - TSOI*ZFR

420

CONTINUE

C

CALL TADAMS (H,NF,X,Y,DY,ITORZ)

GO TO (10,100), 10

100

RETURN

END
SUBROUTINE TAIJAMS (H,NF,X,Y,DY,IC,ITORZ)

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 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 I TORZ INDICATES WHICH EQUATION SHOULD BE INTEGRATED:
C ITORZ = 1: INTEGRATION OF EQUATION FOR ZETA.
C ITORZ = 2: INTEGRATION OF EQUATION FOR TAU.

COMMON /X1/ CM, ANG, RCG, RCI, GAM, G, RT
COMMON /X2/ TR1, h2, FLAST, NEN, IXTN
COMMON /X3/ WC, SUN, IP, MDE, NU, KF(3)
COMMON /X5/ UC(1000), DU(1000), C(1000), RV(1000)
COMMON /X6/ AFN, AFN1, AFN2
COMMON /X8/ ZETA, TAU, CCEXT
COMPLEX AFN(1000), AFN1(1000), AFN2(1000)
COMPLEX CC(25), CFH, CFM, CFN, INKM, AH, AH1, AF, AF1, AF2, CGAM, CGAM1
COMPLEX ZETA(1000), TAU(1000), TAU1, CCEXT(25)
DIMENSION Y(4), DY(4, 4), DP(4), PRED(4), COR(4)

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
C X IS THE VALUE OF STEADY-STATE POTENTIAL AT THE STATION
C WHERE THE PREDICTOR-CORRECTOR INTEGRATION STARTS: INPUT.
C TR = FRED(I)
C TI = FRED(2)
C TAU (NP) = CMPLX (TR, TI)
C ZETA (NP) = CMPLX (TR1)
C GO TO (110, 120, 130) IP
120 AH = PRED(3)
   AH = CMPLX (AH, AH1)
   GO TO 110
130 CONTINUE
C CGAM = CMPLX (PRED(3), PRED(4))
CONTINUE
IF (NP == NPLAST) GO TO 20

CONTAIN COEFFICIENTS IN THE EXTENSION SECTION.
DO 25 I = 1, 25
25 CC(I) = CCEXT(I)

GO TO 30

CONTINUE
DUF = DU(NP)
UP = UK(NP)
CF = CNF(NP)
R = RW(NP)
CALL COEFFS (UP, DUF, CF, R, CC)

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(NF)) * TAU(NP) / CFH
IF(1) = REAL (TAU1)
DP(2) = AIMAG (TAU1)
GO TO (140, 160) IF

AH1 = AH / TAU(NP)
DP(3) = REAL (AH1)
DP(4) = AIMAG (AH1)
GO TO 140

CONTINUE

AP, API AND AFP ARE THE VALUES OF THE AMPLITUDE FUNCTION AND
THEIR DERIVATIVES AT THE CURRENT STATION.
AP = AFN1(NP)
API = AFN11(NP)
AP2 = AFN2(NP)

INHMG = - CC(18) * AP * AP2 - CC(12) * AH1 * AP2 - (CC(9)
1 + CC(15)) * 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 = (- ZFACNF) + 5 * (GAM - 1.) * DUF/CF - CFM/CFH) * CGAM
1 - INHMG / (CF * CFH)

DP(3) = REAL (CGAM1)
DP(4) = AIMAG (CGAM1)

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
LO 55 J = 1, 3
55 DY(I, J) = DY(I, J+1)
TR = Y(1)
TI = Y(2)
T2 = TR*TR + TI*TI
TAU (NF) = CMFLX (TR, TI)
ZETA (NP) = 1./ TAU(NF)
TAU1 = 1 + (CFM * CFN * TAU(NP)) * TAU(NF) / CFH
DY (1,4) = REAL (TAU1)
DY (2,4) = AIMAG (TAU1)
GO TO (170,180,190), IF
180 AHR = Y(3)
AH1 = Y(4)
AH = CMFLX (AHR,AH1)
AH1 = AH / TAU(NF)
DY (3,4) = REAL (AH1)
DY (4,4) = AIMAG (AH1)
IF (MODE *NE* 1) GO TO 182
AFN(NF) = AH
AFN1(NF) = AH
AFN2 (NP) = ( TAU(NP) * AFN1(NF) - TAU1 * AFN(NF) ) / TAU(NF)
182 GO TO 170
190 CONTINUE
CGAM = CMFLX (Y(3),Y(4))
CGAM1 = ( - ZETA(NP) + .5 * (GAM - 1) ) * LUP/CF = CFM/CFH) * CGAM
1 = INHKG / (CP * CFH)
DY (3,4) = REAL (CGAM1)
DY (4,4) = AIMAG (CGAM1)
170 CONTINUE
IF (NP *EQ* NENF) GO TO 100
C
C DECIDE WHICH EQUATION IS TO BE INTEGRATED: TAU OR ZETA
C
IF (CABS(TAU(NF)) *LT* .10) GO TO 10
ITORZ = 2
Y(1) = REAL ( ZETA(NF) )
Y(2) = AIMAG ( ZETA(NF) )
C
C CALCULATE DERIVATIVES OF ZETA AT THE LAST FOUR POINTS.
DO 420 I = 1,4
ZSQR = REAL ( ZETA(NF-4+1) * ZETA(NF-4+1) )
ZSQI = AIMAG ( ZETA(NF-4+1) * ZETA(NF-4+1) )
TFR = DY(1,I)
TPI = DY(2,I)
DY(1,I) = - ZSQR*TFR + ZSQI*TPI
DY(2,I) = - ZSQR*TPI - ZSQI*TFR
420 CONTINUE
C
I0 = 1
RETURN
100 I0 = 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, $\mathcal{C}_4(j, p)$ defined by

$$
\mathcal{C}_4(j, p) = \frac{u_e}{c_e} \frac{e^2}{p} \int_{z_e}^{2\pi} \Theta \int_{j}^{1} R_p R_j R_z \, d\Theta.
$$

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).

<table>
<thead>
<tr>
<th>No. of Cards</th>
<th>Location</th>
<th>Type</th>
<th>Input Item</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1-72</td>
<td>A</td>
<td>Title</td>
<td>Title of the case</td>
</tr>
<tr>
<td>1</td>
<td>1-10</td>
<td>F</td>
<td>GAMMA</td>
<td>Ratio of specific heats</td>
</tr>
<tr>
<td></td>
<td>11-20</td>
<td>F</td>
<td>UE</td>
<td>Steady-state Mach number at nozzle entrance</td>
</tr>
<tr>
<td></td>
<td>21-30</td>
<td>F</td>
<td>RID</td>
<td>Length-to-diameter ratio</td>
</tr>
<tr>
<td></td>
<td>31-40</td>
<td>F</td>
<td>ZCOMB</td>
<td>Length of the combustion zone</td>
</tr>
<tr>
<td>Location</td>
<td>Type</td>
<td>Input Item</td>
<td>Comments</td>
<td></td>
</tr>
<tr>
<td>----------</td>
<td>------</td>
<td>------------</td>
<td>----------</td>
<td></td>
</tr>
<tr>
<td>41-45</td>
<td>I</td>
<td>NDROPS</td>
<td>If 0: droplet momentum source neglected. If 1: droplet momentum source included.</td>
<td></td>
</tr>
<tr>
<td>46-50</td>
<td>I</td>
<td>NOZZLE</td>
<td>If 0: quasi-steady nozzle. If 1: conventional nozzle.</td>
<td></td>
</tr>
<tr>
<td>1-5</td>
<td>I</td>
<td>NJMAX</td>
<td>Number of series terms (complex).</td>
<td></td>
</tr>
<tr>
<td>6-10</td>
<td>I</td>
<td>NONLIN</td>
<td>If 0: linear terms only. If 1: both linear and nonlinear terms.</td>
<td></td>
</tr>
<tr>
<td>11-15</td>
<td>I</td>
<td>NEGL</td>
<td>If 0: all non-zero coefficients calculated. If 1: small coefficients neglected.</td>
<td></td>
</tr>
<tr>
<td>16-20</td>
<td>I</td>
<td>NOUT</td>
<td>If 0: printed output only. If 1: printed and written into file. If 2: written into file only. If 3: card output only.</td>
<td></td>
</tr>
<tr>
<td>21-25</td>
<td>I</td>
<td>NOZNIL</td>
<td>If 0: nozzle nonlinearities neglected. If 1: nozzle nonlinearities included.</td>
<td></td>
</tr>
<tr>
<td>26-30</td>
<td>I</td>
<td>NZDATA</td>
<td>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.</td>
<td></td>
</tr>
</tbody>
</table>

The next card is necessary only if NEGL = 1.
<table>
<thead>
<tr>
<th>No. of Cards</th>
<th>Location</th>
<th>Type</th>
<th>Input Item</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1-10</td>
<td>F</td>
<td>SM1</td>
<td>Linear coefficients with absolute value less than SM1 neglected</td>
</tr>
<tr>
<td></td>
<td>11-20</td>
<td>F</td>
<td>SM2</td>
<td>Nonlinear coefficients with absolute value less than SM2 neglected</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>NJMAX</th>
<th>Location</th>
<th>Type</th>
<th>Input Item</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-5</td>
<td>I</td>
<td>J</td>
<td></td>
<td>Integer which identifies the series term</td>
</tr>
<tr>
<td>6-15</td>
<td>F</td>
<td>AMPL(J)</td>
<td></td>
<td>Amplitude of the linear nozzle admittance</td>
</tr>
<tr>
<td>16-25</td>
<td>F</td>
<td>PHASE(J)</td>
<td></td>
<td>Phase of the linear nozzle admittance</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>NJMAX</th>
<th>Location</th>
<th>Type</th>
<th>Input Item</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-5</td>
<td>I</td>
<td>J</td>
<td></td>
<td>Integer which identifies the series term</td>
</tr>
<tr>
<td>6-15</td>
<td>F</td>
<td>GNOZ(J)</td>
<td></td>
<td>Real part of the nonlinear nozzle admittance</td>
</tr>
<tr>
<td>16-25</td>
<td>F</td>
<td>GNOZ(J)</td>
<td></td>
<td>Imaginary part of the nonlinear nozzle admittance</td>
</tr>
<tr>
<td>1-5</td>
<td>I</td>
<td>J</td>
<td></td>
<td>Integer which identifies series term</td>
</tr>
<tr>
<td>6-10</td>
<td>I</td>
<td>L(J)</td>
<td></td>
<td>Axial mode number, ( \ell )</td>
</tr>
<tr>
<td>11-15</td>
<td>I</td>
<td>M(J)</td>
<td></td>
<td>Tangential mode number, ( m )</td>
</tr>
<tr>
<td>16-20</td>
<td>I</td>
<td>N(J)</td>
<td></td>
<td>Radial mode number, ( n )</td>
</tr>
<tr>
<td>21-25</td>
<td>I</td>
<td>NS(J)</td>
<td></td>
<td>( \text{NS}(J) = 1: \ \Theta_j = \sin (m\theta) ) ( \text{NS}(J) = 2: \ \Theta_j = \cos (m\theta) )</td>
</tr>
<tr>
<td>26-30</td>
<td>A</td>
<td>NAME(J)</td>
<td></td>
<td>Four character name</td>
</tr>
</tbody>
</table>

80
FORTRAN Listing

*************** PROGRAM COEFFS3D ***************

THIS PROGRAM COMPUTES THE COEFFICIENTS WHICH APPEAR
IN THE DIFFERENTIAL EQUATIONS WHICH GOVERN THE MODE-AMPLITUDE
FUNCTIONS. THESE COEFFICIENTS ARE PUNCHED ONTO CARDS FOR
INPUT INTO PROGRAM LIMCYC.

THE FOLLOWING INPUTS ARE REQUIRED:
THE TITLE OF THE CASE,
GAMMA IS THE SPECIFIC HEAT RATIO,
UE IS THE STEADY STATE MACH NUMBER AT THE NOZZLE ENTRANCE,
RLD IS THE LENGTH-TO-DIAMETER RATIO,
ZCMB IS THE LENGTH OF THE REGION OF UNIFORMLY DISTRIBUTED
COMBUSTION, EXPRESSED AS A FRACTION OF THE CHAMBER LENGTH,
NDROPS DETERMINES THE PRESENCE OF DROplet MOMENTUM SOURCES:
  NDROPS = 0 DROplet MOMENTUM SOURCE NEGLECTED,
  NDROPS = 1 DROplet MOMENTUM SOURCE INCLUDED,
NOZZLE SPECIFIES THE TYPE OF NOZZLE USED:
  NOZZLE = 0 QUASI-STEADY
  NOZZLE = 1 CONVENTIONAL NOZZLE,
AWFL IS THE NOZZLE AMPLITUDE RATIO,
PHASE IS THE NOZZLE PHASE SHIFT,
NOZNL1 DETERMINES THE PRESENCE OF NOZZLE NONLINEARITIES:
  NOZNL1 = 0 NOZZLE NONLINEARITIES NEGLECTED,
  NOZNL1 = 1 NOZZLE NONLINEARITIES INCLUDED,
NZDATA DETERMINES HOW THE NOZZLE DATA IS SUPPLIED:
  NZDATA = 0 FROM CARDS,
  NZDATA = 1 FROM A FASTRAND FILE,
NMAX IS THE NUMBER OF MODE-AMPLITUDE FUNCTIONS IN THE ASSUMED
SERIES SOLUTION. NMAX MUST NOT EXCEED MX.
THE COEFFICIENTS COMPUTED ARE DETERMINED BY NONLIN AS FOLLOWS:
  NONLIN = 0 LINEAR COEFFICIENTS ONLY
  NONLIN = 1 BOTH LINEAR AND NONLINEAR COEFFICIENTS
COEFFICIENTS TO BE NEGLECTED ARE DETERMINED BY NEGL
AS FOLLOWS:
  NEGL = 0 TERMS SMALLER THAN 0.00001 ARE NEGLECTED
  NEGL = 1 LINEAR TERMS SMALLER THAN SM1 AND NONLINEAR
  TERMS SMALLER THAN SM2 ARE NEGLECTED
THE OUTPUT IS DETERMINED BY NOUT AS FOLLOWS:
  NOUT = 0 PRINTED OUTPUT ONLY
  NOUT = 1 PRINTED AND WRITTEN ON FASTRAND FILE
  NOUT = 2 FASTRAND FILE ONLY
  NOUT = 3 CARD OUTPUT ONLY
EACH MODE-AMPLITUDE IS ASSIGNED AN INTEGER J.
THE MODE IS SPECIFIED BY THE INDICES L(J), M(J), AND N(J).
L(J) IS THE AXIAL MODE NUMBER AND MUST NOT EXCEED S.
M(J) IS THE AZIMUTHAL MODE NUMBER AND MUST NOT EXCEED 8.
N(J) IS THE RADIAL MODE NUMBER AND MUST NOT EXCEED 5.
The integer NS(J) is assigned as follows:
  NS = 1 A-Function \sin(M*Theta) \cosh(I*B*Z)
  NS = 2 B-Function \cos(M*Theta) \cosh(I*B*Z)
NAME(J) IS A FOUR-CHARACTER NAME.
PARAMETER MX=5, MX2=10, MX4=20

DIMENSION L(KX), N(MX), NAME(MX), S(MX), SJ(MX), TITLE(80),
RJROOT(10,5), RJVAL(10,5), CI(MX2,MX2), C(4,MX2,MX2),
D(MX2,MX2,MX2), AMPL(MX), PHASE(MX), AZI(2),
BES1(9,9,9), BES2(9,9,9), BES3(9,9,9),
V(2), JC(MX2), TS4(MX2), TSQ(MX2), KMAX(5)

COMPLEX CRSLT, CI, ZE1, ZEP1, ZEP2, CZE, CAZ, CRAD,
G(MX,MX2), C(t f X2, MX2) p

DATA INPUT.
PI = 3.1415927
SM1 = 0.00001
SM2 = 0.00001
SM3 = 0.00001
CI = (0.0, 1.0)

INPUT ROOTS AND VALUES OF BESSEL FUNCTIONS:
DATA ((FJROOT(I,J)), J = 1,5, I = 1,9)/
2 1.84118 - 5.33144 - 8.53632 - 11.70600 - 14.86359
4 3.05424 - 6.17347 - 9.32369 - 12.47063 - 15.6309
9 7.50127 - 11.24118 - 14.40613 - 17.57037 - 20.7325

DATA ((FJVAL(I,J)), J = 1,5, I = 1,9)/
1 -0.80276 - 0.30012 - 0.24970 - 0.21836 - 0.19647
2 0.85118 - 0.34613 - 0.27330 - 0.23300 - 0.20701
3 0.48650 - 0.31553 - 0.25474 - 0.20688 - 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 - 0.19560 - 0.17849
7 0.35414 - 0.25017 - 0.21631 - 0.18976 - 0.17363
8 0.33793 - 0.24096 - 0.20586 - 0.18449 - 0.16929
9 0.32438 - 0.23303 - 0.19998 - 0.17979 - 0.16539

INPUT PARAMETERS:
4 READ (5,5000) END = 600) (TITLE(I), I = 1, 72)
READ (5,5001) GAMMA, UE, RL, ZCOMB, NLROS, NOZZLE
IF (.NOT. GAMMA) 600, 600, 8
8 READ (5,5004) NUMAX, NONLIN, NFGL, NOUT, NOZNL1, NZDATA
IF (.NOT. NFGL = .EQ. 1) READ (5,5005) SM1, SM2
IF (.NOT. NOZZLE = .EQ. 1) GO TO 5

COMPUTE ADMITTANCE FOR QUASI-STEADY NOZZLE.

82
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

DO 12 J = 1, NJMAX
   THETA = PHASE(J) * PI/180.0
   YR = AMPL(J) * COS(THETA)
   YI = AMPL(J) * SING(THETA)
   YNOZ(J) = CMPLX(YR,YI)
12 CONTINUE

ZE = 2.0 * ELD
CZE = CMPLX(ZE,0.0)
CGAM = CMPLX(GAMMA,0.0)
CAX = CGAM
IF (NDROPS *EQ* 1) CAX = CGAM + (1.0,0.0)

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
20 CONTINUE
15 S(J) = 0.0
   SJ(J) = 1.0
25 SSQ = S(J) * S(J)
   CSSQ(J) = CMPLX(SSQ,0.0)
20 CONTINUE

CALCULATE AXIAL ACOUSTIC EIGENVALUES.
FIND MAXIMUM VALUES OF L(J), M(J), AND N(J).
KN = 0
LMAX = 0
MMAX = 0
NFAK = 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)
YAMFL = AMFL(J)
YHASE = PHASE(J)
CALL EIGVAL(LL, SMN, GAMMA, ZE, YAMFL, YHASE, 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 (NZNL1 *EQ* 1) NCOEFF = 5
NCFM1 = NCOEFF - 1

C
DO 100 NJ = 1, NJMAX
DO 100 NF = 1, NJMAX

C
C ZERO COEFFICIENT ARRAYS.
DO 105 KC = 1, NCOEFF
C(KC,NJ,NF) = (0.0,0.0,0.0,0.0)
105 CONTINUE

C
OIHOGONALITY 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
120

C
OIHOGONALITY PROPERTY OF RADIAL EIGENFUNCTIONS.
120 IF (M(NP) *NE* M(NJ)) GO TO 100
IF (S(NP)) 125, 122, 125
125 SGM = M(NJ) * M(NJ)
SS0 = S(NP) * S(NP)
SS = SJ(NJ) * SJ(NJ)
RAD = (SSQ - SG) * SJSG/(2.0 * SSQ)
GO TO 127
122 RAD = 0.5

C CALCULATE AXIAL INTEGRALS.
127 DO 130 NOPT = 1, 4
CALL AXIAL1 (NOPT, NF, NJ, UE, ZE, ZCOME, CRSLT)
AX(NOPT) = CRSLT
130 CONTINUE

C EVALUATE FUNCTIONS AT NOZZLE END.
ZEJ = CCOSH(CI*BC(NJ)*CZE)
ZEP1 = CCOSH(CI*B(NF)*CZE)
ZEP2 = CI * B(NF) * CSINH(CI*B(NF)*CZE)

C CAZ = CMPLX(AZ,0.0)
CRAD = CMPLX(RAD,0.0)

C COEFFICIENT OF THE SECOND DERIVATIVE OF A(P).
CC(1,NJ,NF) = AX(1) * CAZ * CHAD

C COEFFICIENT OF A(P).
CC(2,NJ,NF) = (CSSQ(NF)*AX(1) - AX(2) + ZEP2*ZEJ) * CAZ * CRAD

C COEFFICIENT OF THE FIRST DERIVATIVE OF A(P).
CC(3,NJ,NF) = (CAZ*AX(3) + (2*0.0*0)*AX(4)
+ CGAM*YN0Z(NF)*ZEP1*ZEJ) * CAZ * CRAD

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.
CES0 = 1 - (GAMMA-1) * UE/2
CC(5,NJ,NP) = UE * CESO * G0Z(NP) * ZEJ * CAZ * CRAD

100 CONTINUE

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 COMPUTE NONLINEAR COEFFICIENTS.
C IF (NONLIN .EQ. 0) GO TO 402
61 = (CGAM - (1.0,0.0)) * (0.5,0.0)

85
COMPUTATIONS OF BESSEL INTEGRALS WHEN ALL SERIES TERMS HAVE THE
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 = RJQ00T(MP, NMAX)
A2 = RJQ00T(MQ, NMAX)
A3 = RJQ00T(NJ, 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

DO 200 NJ = 1, NJMAX
DO 200 NF = 1, NJMAX
DO 200 NQ = 1, NJMAX
CD1(NJ, NF, NQ) = (0.0, 0.0)
CD2(NJ, NF, NQ) = (0.0, 0.0)
CD3(NJ, NF, NQ) = (0.0, 0.0)
CD4(NJ, NF, NQ) = (0.0, 0.0)

DO 210 J = 1, 2
CALL AZIMTL(J, NF, NQ, NJ, RESULT)
AZ1(J) = RESULT
TANINT(J) = CMPLX(RESULT, 0.0)
210 CONTINUE

IF (AZ1(1)) 220, 225, 220
225 IF (AZ1(2)) 220, 200, 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
MG = M(NQ) + 1
MJ = M(NJ) + 1
RADINT(1) = CMPLX(BES1(MP, MG, MJ), 0.0)
RADINT(2) = CMPLX(BES2(MP, MG, MJ), 0.0)
RADINT(3) = CMPLX(BES3(MP, MG, 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, CPSLT)
AXINT(NC, J) = CPSLT

240 CONTINUE

C

DO 250 J = 1, 4
T1 = G1 * CSSG(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 = (0.0, 0.0) * (T1 + T2 + D3 + D4) / NORM(NJ)
IF (J + EQ. 1) CD1(NJ, NP, NJ) = (1.0, -1.0) * DCOEF
IF (J + EQ. 2) CD2(NJ, NP, NJ) = (1.0, 1.0) * DCOEF
IF (J + EQ. 3) CD3(NJ, NP, NJ) = (1.0, 1.0) * DCOEF
IF (J + EQ. 4) CD4(NJ, NP, NJ) = (1.0, -1.0) * DCOEF

250 CONTINUE

200 CONTINUE

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

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

87
COEFFICIENTS OF NONLINEAR TERMS.
IF (NONLIN NE 0) GC TO 350
END 370 NO = 1, NJMAX
NEWQ = (2 * NO) - 1
NEWQ = NEWQ + 1
CD1R = REAL(CD1(NJ, NF, NG))
CL1I = AIMAG(CD1(NJ, NF, NG))
CD2R = REAL(CD2(NJ, NF, NG))
CD2I = AIMAG(CD2(NJ, NF, NG))
CD3R = REAL(CD3(NJ, NF, NG))
CD3I = AIMAG(CD3(NJ, NF, NG))
CD4R = REAL(CD4(NJ, NF, NG))
CD4I = AIMAG(CD4(NJ, NF, NG))
D(NEWJ, NEWF, NEWQ) = CD1R + CD2R + CD3R + CD4R
D(NEWJ, NEWF, NEWQ) = -CL1I - CD3I - CD4I
D(NEWJ, NEWF, NEWQ) = -CD1I - CD2I + CD3I + CD4I
D(NEWJ, NEWF, NEWQ) = -CL1R + CD2R + CD3R - CD4R
D(NEWJ, NEWF, NEWQ) = CD1I + CD2I + CD3I + CD4I
D(NEWJ, NEWF, NEWQ) = CD1R + CD2R + CD3R - CD4R
D(NEWJ, NEWF, NEWQ) = CD1R + CD2R - CD3R - CD4R
D(NEWJ, NEWF, NEWQ) = -CD1I + CD2I + CD3I - CD4I

CONTINUE

350 CONTINUE

**************************************************************************************************

COMPUTE COEFFICIENTS FOR THE EQUATIONS WHICH ARE DECOUPLED
IN THE SECOND DERIVATIVES.

DO 405 KC = 1, NCOEFF
NMAX(KC) = 0
405 CONTINUE

CALCULATE INVERSE OF THE MATRIX C1(I, J).
JMAX = NJMAX
NMAX = 2 * NJMAX
V(1) = 1
CALL GJR(C1, MX2, MX2, NJMAX, 0, 500, JC, V)

USE INVERSE TO CALCULATE DECOUPLED COEFFICIENTS.
DO 410 NF = 1, NJMAX
LINEAR COEFFICIENTS.
DO 420 NJ = 1, NJMAX
DO 420 KC = 1, NCFM1
TS(KC,NJ) = 0.0
DO 425 K = 1, NJMAX
TS(KC,NJ) = TS(KC,NJ) + CI(NJ,K) * C(KC,K,NF)
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 NONLINEAR COEFFICIENTS
IF (NONLIN .EQ. 0) GO TO 410
DO 415 NJ = 1, NJMAX
DO 440 NJ = 1, NJMAX
TS(Q(NJ)) = 0.0
DO 440 H = 1, NJMAX
CI(H) = I(KNP,NJ)
TSO(NJ) = TSO(NJ) + CI(NJ,K) * D(K,NP,NJ)
440 CONTINUE
DO 445 NJ = 1, NJMAX
DN(J,NP,NJ) = TSQ(NJ)
ABSVAL = ABS(D(NJ,NP,NJ))
IF (ABSVAL *GE. SM2) KMAX(NCOEFF) = KMAX(NCOEFF) + 1
445 CONTINUE
415 CONTINUE

410 CONTINUE

C OUTPUT
C IF (NOUT .GE. 2) GO TO 455
C PRINTED OUTPUT
WRITE (6,6001) (TITLE(I), I = 1, 72)
WRITE (6,6002) GAMMA, UE, ZL, ZCOMB
IF (NDROPS .EQ. 0) WRITE (6,6020)
IF (NDROPS .EQ. 1) WRITE (6,6021)
IF (NOZZLE .EQ. 0) WRITE (6,6012)
IF (NOZNLI .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), S(J), SJ(J), E(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),
SK(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, NCM1
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 TO 1100
ND = 1, NJMAX
WRITE (6,6010) ND, NJMAX
WRITE (6,6011)
DO 400 NF = 1, NJMAX
WRITE (6,6009) NF, (D(NJ,KP,NG), NG = 1, NJMAX)
400 CONTINUE
C
C WRITE COEFFICIENTS ON FASTRAN FILE.
C
WRITE (9,7001) GAMMA, UE, ZE, ZCOMB, NDOPS, NJMAX, NOZNL1
C
DO 450 J = 1, JMAX
WRITE (9,7002) J, L(J), M(J), N(J), NS(J), S(J), SJ(J),
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
460 CONTINUE
C
C
ABSVAL = ABS(C(KC,NJ,NP))
        IF (ABSVAL .GE. SM1) WRITE (9,7004) NJ,NP, C(KC,NJ,NP)

460 CONTINUE

C

IF (NOZN1 .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 740
WRITE (9,7005) NJ, NP, NO, D(NJ,NP,NO)

470 CONTINUE

GO TO 4
C

C

PUNCH CARD OUTPUT.

480 PUNCH 7001 GAMMA, UE, ZE, ZCOMB, NDROPS, NJMAX, NOZN1

C

DO 482 J = 1, JMAX
PUNCH 7002 J, L(J), M(J), N(J), NS(J), S(J), S(J), 1
NAME(J)

482 CONTINUE

C

DO 484 J = 1, JMAX
PUNCH 7006 J, YNOZ(J), B(J)

484 CONTINUE

IF (NOZN1 .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 (NOZN1 .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)
CONTINUE

CONTINUE

C

PUNCH 7003 KMAX (NCOEFF)

IF (NOMLIN * EQ. 0) GO TO 4

DO 488 NJ = 1, NJMAX

DO 488 NP = 1, NJMAX

DO 488 NO = 1, NJMAX

ABSVAL = ABS(D(NJ, NP, NO))

IF (ABSVAL * GE. Sb2) PUNCH 7005 NJ, NP, NO, D(NJ, NP, NO)

CONTINUE

GO TO 4

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

FORMAT SPECIFICATIONS

5000 FORMAT (72A1)

5001 FORMAT (4F10.0, 215)

5002 FORMAT (SI5, 1X, A4)

5003 FORMAT (15, 2F10.0)

5004 FORMAT (615)

5005 FORMAT (2F10.0)

6001 FORMAT (1HI, 1X, 72AI//)

6002 FORMAT (2X, 8HEGAM, A = F5.2, 5X, 5HEU = F5.2, 5X, 6HL/D = F8.5,

1 5X, 6HZCOME = F6.2/)}

6003 FORMAT (2X, 4A4, 515, 4F10.5, 2F11.5/)}

6004 FORMAT (2X, 4//, EX, 6E9NAME, J, L, M, N, NS, 7X, 5HSMN, 3X,

1 7HMJ (SN), 7K, 3HEFS, 7X, 3HEFA8X, 2IFY, 8X, 2HYI//)

6005 FORMAT (1HI, 4SH DECOUPLED COEFFICIENT OF B(P): C(1, J, P)///)

6006 FORMAT (1HI, 4SH DECOUPLED COEFFICIENT OF THE DERIVATIVE OF,

1 6H B(P)t, 5X, 8HC(2, J, P)///)

6007 FORMAT (1HI, 39H DECOUPLED COEFFICIENT OF THE RETARDED,

1 20H DERIVATIVE OF B(P)t, 5X, 8HC(3, J, P)///)

6008 FORMAT (7X, 1HF, 18, 9112)

6009 FORMAT (2X, /2X, I3, 3X, 10F12.6)

6010 FORMAT (1HI, 42H DECOUPLED COEFFICIENT OF E(F) * DC0/D1,

1 19H IN EQUATION FOR B(12, 1H)///)

6011 FORMAT (7X, 1HF, 18, 9112)

6012 FORMAT (2X, 19HQUASI-STEELY NOZZLE///)

6013 FORMAT (2X, 8X, 24HLINEAR COEFFICIENTS ONLY)

6014 FORMAT (4X, 1HJ)

6015 FORMAT (4X, 1HF)

6017 FORMAT (1HI, 31H CEFFLOW DETECTED LAST ROW = 15)

6018 FORMAT (1HI, 34H SINGULARITY DETECTED LAST ROW = 15)

6020 FORMAT (2X, "DEFOIL MOMENTUM SOURCE NEGLECTED")
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(4D6, F/))
6025 FORMAT (2X///, 2X, 29WEEN X L M N NS X 3H XM, 3X-
1   7HYM(SMN), 7X, 3H EPS, 7X, 3HETA, 8X, 2HYN, 6X, 2HYI,
2   6X, 2HGR, 6X, 2HGI/)
6026 FORMAT (2X, A4, 5I5, 4F10-5, 4F11-5/)
6027 FORMAT (IH1)
7001 FORMAT (4F10-5, 3I5)
7002 FORMAT (5I5, 2F10-5, 1X, A4)
7003 FORMAT (15)
7004 FORMAT (15, F15-6)
7005 FORMAT (3I5, F15-6)
7006 FORMAT (15, 4F10-5)
7007 FORMAT (15, 2F10-5)
END
SUBROUTINE EI GVAL(L, SMN, GAMMA, ZE, YAMIL, YPHASE, RESULT)

COMPLEX RESULT
COMMON /ELK1/ GS6, ABS0, ALBET, SMNSG

************************************************************************

THIS SUBROUTINE COMPUTES THE COMPLEX AXIAL ACOUSTIC EIGENVALUES
FOR A CYLINDRICAL CHAMBER WITH A NOZZLE AND STORES THEM IN
RESULT.
THE EIGENVALUES ARE COMPUTED BY MEANS OF NEWTONS METHOD.

THE INPUT PARAMETERS ARE AS FOLLOWS
L IS THE AXIAL MODE NUMBER.
SMN IS THE DIMENSIONLESS ACOUSTIC FREQUENCY.
GAMMA IS THE SPECIFIC HEAT RATIO.
ZE IS THE LENGTH-TO-RADIUS RATIO.
YAMIL IS THE NOZZLE AMPLITUDE FACTOR.
YPHASE IS THE NOZZLE PHASE SHIFT IN DEGREES.

************************************************************************

PI = 3.1415927
ERR = 0.00000001

IF (YAMIL) 5, 60, 5
CALCULATE CONSTANTS.

5 PHASE = YPHASE * PI/180.0
ALPHA = YAMIL * COS(PHASE)
BETA = YAMIL * SIN(PHASE)
GS6 = GAMMA * GAMMA
ABS0 = (ALPHA * ALPHA) - (BETA * BETA)
ALBET = ALPHA * BETA
SMNSG = SMN * SMN

ASSIGN INITIAL GUESS FOR EIGENVALUE.
IF (L .EQ. 0) GO TO 45
RL = L
PHI = PI/2.0 + PHASE
XM = RL * PI/ZE
A = YAMIL/ZE
X0 = 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 (YAMIL .LT. 0.1) GO TO 110
IF (YAMIL .LT. 0.4) GO TO 120
IF (YAMIL .LT. 0.8) GO TO 150
IF (YAMIL .LT. 1.2) GO TO 160
X0 = 1.0 * YAMIL
GO TO 170

160 X0 = 1.25 * YAMIL
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 * YAML
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 = 5.0 * YAML
GO TO 130
120 XO = 3.0 * YAML
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
Y0 = XO * TANPSI
C
C ITERATION USING NEWTON'S METHOD FOR A SYSTEM OF TWO EQUATIONS
C IN TWO UNKNOWNS.
C
47 L1 = 0
X = XO
Y = Y0
40 CALL FCNS(X,Y,Z,E,F,G,FX,IFY,GX,6Y)
IF (L1 *LE* 40) GO TO 50
RJFG = (FX * FY) - (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) *LE* ERR OR* ABS(DELTAY) *LE* ERR) GO TO 40
GO TO 10
C
C WARNING MESSAGES
30 WRITE (6,6005)
GO TO 10
50 WRITE (6,6006)
-60 TO 10
C
C CASE OF HARD WALL (YAML = 0).
60 NL = L
\[ X = \pi \times \frac{1}{z} \]
\[ Y = 0.0 \]
10 RESULT = CMPLX(X, Y)

C

C FORMAT SPECIFICATIONS.
6005 FORMAT (2X//2X,16HJACOBIAN IS ZERO//)
6006 FORMAT (2X//2X,3SHFAILED 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
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
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
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)
\[ \begin{align*}
G_{X1} &= \text{compute } F(X) \quad \text{and } G(X) \\
G_{Y1} &= \text{compute the partial derivatives of } F \quad \text{and } G
\end{align*} \]

\[\begin{align*}
F' &= (X Y S Q \ast \text{FF}) - (4 \cdot X Y \ast \text{HH}) \\
1 + G S Q \ast ((F1 \ast \text{GG}) + (4 \cdot F2 \ast \text{HH})) \\
G &= (X Y S Q \ast \text{HH}) + (X Y \ast \text{FF}) \\
1 + G S Q \ast ((F2 \ast \text{GG}) - (G1 \ast \text{HH}))
\end{align*}\]
SUBROUTINE AXIAL1 (NOPT, NP, NJ, UEZ, ZE, ZC, RESULT)

C

C THIS SUBROUTINE CALCULATES THE INTEGRAL OVER THE INTERVAL
C (0, ZE) OF THE FOLLOWING FUNCTIONS ACCORDING TO THE VALUE
C OF NOPT
C
C NOPT = 1  Z(NP) * ZC(NJ)
C NOPT = 2  ZFP(NP) * ZC(NJ)
C NOPT = 3  UF * Z(NP) * ZC(NJ)
C NOPT = 4  U * ZP(NP) * ZC(NJ)

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 UF 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,
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
C ARG = (BP + BJ) * CI
C MAG = CABS(ARG)
C 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)
C 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)
C IF (NOPT LT 2) RESULT = -B(NP) * B(NP) * RESULT
GO TO 100
NUMERICAL EVALUATION OF INTEGRALS FOR NOPT = '3' AND NOPT = '4'.

COMPUTE STEP SIZE FOR SIMPSON INTEGRATION.

50 N = 50
RN = N
RESULT = (0.0, 0.0)
IC = ZCOMB
IC = 2 - IC

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) ZO = 0.0
IF (J .EQ. 2) ZO = ZCOMB * ZE
NP1 = N + 1
CH = CMPLX(H, 0.0)

COMPUTE INTEGRANDS.

DO 60 I = 1, NP1
STEP = I - 1,
Z = (STEP * H) + ZO
IF (I .EQ. 1) .AND. (J .EQ. 2) Z = Z + H/100.0
IF (NOPT .EQ. 3) CALL UBARC2SUE.ZESZCOMB.ZSF)
IF (NOPT .EQ. 4) CALL UBARC2SUE.ZESZCOMB.ZSF)
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

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

100 CONTINUE
RETURN
END
SUBROUTINE AXIAL2(NOFT, NCONJ, NP, NO, NJ, ZE, RESULT)

THIS SUBROUTINE CALCULATES THE INTEGRAL OVER THE INTERVAL (0, ZE) OF THE FOLLOWING FUNCTIONS ACCORDING TO THE VALUES OF NOFT AND NCONJ

FOR NCONJ = 1 AND
NOFT = 1  Z(NF) * Z(NO) * ZC(NJ)
NOFT = 2  ZP(NP) * ZP(NO) * ZC(NJ)
NOFT = 3  ZPP(NP) * Z(NO) * ZC(NJ)

FOR NCONJ = 2 AND
NOFT = 1  Z(NF) * ZC(NO) * ZC(NJ)
NOFT = 2  ZP(NP) * ZFC(NO) * ZC(NJ)
NOFT = 3  ZPP(NP) * ZC(NO) * ZC(NJ)

FOR NCONJ = 3 AND
NOFT = 1  ZC(NP) * ZC(NO) * ZC(NJ)
NOFT = 2  ZFC(NF) * ZFC(NO) * ZC(NJ)
NOFT = 3  ZPPC(NF) * ZC(NO) * ZC(NJ)

FOR NCONJ = 4 AND
NOFT = 1  ZC(NP) * ZC(NO) * ZC(NJ)
NOFT = 2  ZFC(NF) * ZFC(NO) * ZC(NJ)
NOFT = 3  ZPPC(NF) * ZC(NO) * ZC(NJ)

IN THE ABOVE EQUATIONS:
Z(NP), Z(NO), AND Z(NJ) ARE THE AXIAL ACOUSTIC EIGENFUNCTIONS
AND NP, NO, AND NJ ARE THEIR INDICES.
ZP IS THE FIRST DERIVATIVE OF THE AXIAL EIGENFUNCTIONS.
ZPP IS THE SECOND DERIVATIVE OF THE AXIAL EIGENFUNCTIONS.
ZC AND ZPC ARE COMPLEX CONJUGATES OF Z AND ZP RESPECTIVELY.

PARAMETER MX = 5
REAL MAG
COMPLEX CI, CF, CZE, BF, BG, BJ, SUM, RESULT
1
COMMON B

CALCULATE INTEGRALS BY MEANS OF ANALYTICAL EXPRESSIONS.
CI = (0.0, 1.0)
CF = (0.25, 0.0)
CZE = CMPLX(ZE, 0.0)
BF = B(NP)
BG = B(NO)
BJ = CONJG(B(NJ))
IF (NCONJ .EQ. 2 .OR. (NCONJ .EQ. 4)) BG = CONJG(BF)
IF (NCONJ .GT. 2) BF = CONJG(BF)
ARG(1) = (BF + BG + 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 = CAEBS(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 AZIMUL(NOPT, NP, NQ, NJ, RESULT)

PARAMETER MX = 5
DIMENSION NFCN(3), SG(2)
COMMON /BLK2/ M(MX), NS(MX)

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

THIS SUBROUTINE CALCULATES THE INTEGRAL OVER THE INTERVAL
(0, 2*PI) OF THE FOLLOWING FUNCTIONS ACCORDING TO THE VALUE
OF NOPT

NOPT = 1  TH(NP) * TH(NQ) * TH(NJ)
NOPT = 2  THP(NP) * THP(NQ) * TH(NJ)

IN THE ABOVE EQUATIONS:
TH(NP), TH(NQ), AND TH(NJ) ARE THE TANGENTIAL EIGENFUNCTIONS
AND NP, NQ, AND NJ ARE THEIR INDICES.
THP IS THE DERIVATIVE OF THE TANGENTIAL EIGENFUNCTIONS.

IF NS = 1  TH = SIN(M*THETA)
IF NS = 2  TH = COS(M*THETA)

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

RESULT = 0.0
FACTOR = 1.0
PI = 3.1415927

DISTINGUISH BETWEEN SINES AND COSINES.

DO 10 K1 = 1, 3
   NFCN(K1) = 1
10 CONTINUE
IF (NS(NJ) .GE. 2) NFCN(3) = 2
IF (NOPT .EQ. 2) GO TO 20
IF (NS(NP) .GE. 2) NFCN(1) = 2
IF (NS(NQ) .EQ. 2) NFCN(2) = 2
GO TO 30
20 IF (NS(NP) .EQ. 1) NFCN(1) = 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)

30 NSUM = 0
   DO 50 K1 = 1, 3
      NSUM = NSUM + NFCN(K1)
50 CONTINUE
IF (CNSU .EQ. 3) OR (NSUM .EQ. 5) GO TO 60
IF (NSUM .EQ. 4) GO TO 70
IF (NSUM .EQ. 6) GO TO 60

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

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)

THIS SUBROUTINE CALCULATES THE INTEGRAL OVER THE INTERVAL (0,1) OF THE FOLLOWING PRODUCTS OF THREE BESSEL FUNCTIONS:

NOPT = 1  \(J_L(A*R) \times J_M(B*R) \times J_N(C*R) \times R\)

NOPT = 2  \(J_L(A*R) \times J_M(B*R) \times J_N(C*R)/R\)

NOPT = 3  \(J_P(L)(A*R) \times J_P(M)(B*R) \times J_N(C*R) \times R\)

\(J_L\) IS THE BESSEL FUNCTION OF FIRST KIND OF ORDER \(L\)
\(J_P\) IS THE DERIVATIVE OF \(J_L\) WITH RESPECT TO \(R\)
\(L, M, N\) ARE NON-NEGATIVE INTEGERS
\(A, B, C\) ARE REAL NUMBERS

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

NN = 100
DN = NN
DH = 1.0/DN
NP1 = NN + 1

DO 10 I = 1, NP1
DSTEP = I - 1
DR = DH * DSTEP
ARG1 = A * DR
ARG2 = B * DR
ARG3 = C * DR

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, BES1, $500)
CALL JBES(L-1, ARG1, BES1, $500)
BES1 = A \times (BESL - BESH)/2.0
GO TO 104

103 CALL JBES(1, ARG1, BES1, $500)
BES1 = -BES1 \times A

104 IF (M .EQ. 0) GO TO 105
CALL JBES(M+1, ARG2, BES2, $500)
CALL JBES(N-1, ARG2, BESL, $500)
BES2 = B \times (BESL - BESH)/2.0
GO TO 102

105
CALL JBES(1, ARG2, BES2, S500)
BES2 = -BES2 * B
102 PROD = BES1 * BES2 * BES3

C
   IF (.NOT. OPT * 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, IOERROR 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
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
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 $C_k(j,p) e^{-\gamma t}$, 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.

<table>
<thead>
<tr>
<th>No. of Cards</th>
<th>Location</th>
<th>Type</th>
<th>Input Item</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1-5</td>
<td>I</td>
<td>NOUTCF</td>
<td></td>
</tr>
<tr>
<td></td>
<td>6-10</td>
<td>I</td>
<td>NOZNL2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1-72</td>
<td>A</td>
<td>TITLE</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1-10</td>
<td>F</td>
<td>EN</td>
<td></td>
</tr>
<tr>
<td></td>
<td>11-20</td>
<td>F</td>
<td>TAU</td>
<td></td>
</tr>
<tr>
<td></td>
<td>21-30</td>
<td>F</td>
<td>H</td>
<td></td>
</tr>
<tr>
<td></td>
<td>31-40</td>
<td>F</td>
<td>TSTART</td>
<td></td>
</tr>
</tbody>
</table>

- If 0: coefficients are not printed out.
- If 1: only the linear coefficients are printed out.
- If 2: all the coefficients are printed out.

- If 0: nozzle nonlinearities not included.
- If 1: nozzle nonlinearities included.

Title used to label the plots.
Interaction index, $n$.
Time lag, $\tau$.
Time increment for numerical integration.
Time at which output of solution begins.
<table>
<thead>
<tr>
<th>No. of Cards</th>
<th>Location</th>
<th>Type</th>
<th>Input Item</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>41-50</td>
<td>F</td>
<td></td>
<td>TQUIT</td>
<td>Time at which output of solution ends</td>
</tr>
</tbody>
</table>
| 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: $z = 0, \theta = 0^\circ$  
If 2: $z = 0, \theta = 45^\circ$  
If 3: $z = 0, \theta = 90^\circ$ |
| 16-20       | I        |      | INTERMS    | 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 $1 \leq \text{NOUT} \leq 6$: 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 $1 \leq \text{NOUT} \leq 6$.

<table>
<thead>
<tr>
<th>No. of Cards</th>
<th>Location</th>
<th>Type</th>
<th>Input Item</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1-10</td>
<td>F</td>
<td>YHI(1)</td>
<td>Maximum ordinate for pressure plots</td>
</tr>
<tr>
<td></td>
<td>11-20</td>
<td>F</td>
<td>YHI(5)</td>
<td>Maximum ordinate for velocity plots</td>
</tr>
<tr>
<td></td>
<td>21-30</td>
<td>F</td>
<td>YIAB(1)</td>
<td>Interval for ordinate labeling of pressure plots</td>
</tr>
<tr>
<td></td>
<td>31-40</td>
<td>F</td>
<td>YIAB(5)</td>
<td>Interval for ordinate labeling of velocity plots</td>
</tr>
<tr>
<td>1</td>
<td>1-5</td>
<td>I</td>
<td>ITICY(1)</td>
<td>Number of ordinate tic marks for pressure plots</td>
</tr>
<tr>
<td></td>
<td>6-10</td>
<td>I</td>
<td>ITICY(5)</td>
<td>Number of ordinate tic marks for velocity plots</td>
</tr>
<tr>
<td></td>
<td>11-15</td>
<td>I</td>
<td>NFIRST</td>
<td>Gives the number of the first plot produced</td>
</tr>
</tbody>
</table>
|               | 16-20    | I    | NOMIT      | If 0: time-history plot produced  
|               |          |       |            | If 1: time-history plot omitted |
| 1             | 1-5      | I    | MDPLT(1)   | If 0: plot of the first mode amplitude not produced  
|               |          |       |            | If 1: plot of the first mode amplitude is produced |
|               | 6-10     | I    | MDPLT(2)   | If 0: plot of the second mode amplitude not produced  
|               |          |       |            | If 1: plot of the second mode amplitude is produced |
|               | 11-15    | I    | MDPLT(3)   | If 0: plot of the third mode amplitude not produced  
<p>|               |          |       |            | If 1: plot of the third mode amplitude is produced |</p>
<table>
<thead>
<tr>
<th>No. of Cards</th>
<th>Location</th>
<th>Type</th>
<th>Input Item</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>16-20</td>
<td>I</td>
<td>MDPLOT(1)</td>
<td>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.</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1-10</td>
<td>F</td>
<td>YHIMD</td>
<td>Maximum ordinate for mode-amplitude plots</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Interval for ordinate labeling of mode-amplitude plots</td>
</tr>
<tr>
<td></td>
<td>11-20</td>
<td>F</td>
<td>YLABMD</td>
<td>Number of ordinate tic marks for mode-amplitude plots</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Identifies complex amplitude function</td>
</tr>
<tr>
<td></td>
<td>21-25</td>
<td>I</td>
<td>ITICMD</td>
<td>Amplitude of sin(\omega t) terms in initial conditions</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Amplitude of cos(\omega t) terms in initial conditions</td>
</tr>
<tr>
<td>1</td>
<td>1-10</td>
<td>F</td>
<td>DAMP</td>
<td>Damping factor in initial condition, obtained from linear stability analysis (Appendix E of Ref. 11)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Corresponding frequency</td>
</tr>
</tbody>
</table>

The next card is necessary only if ICTYPE = 2.
**FORTRAN Listing**

```
*************** PROGRAM LCYC3D ***********************

THIS PROGRAM CALCULATES THE NONLINEAR BEHAVIOR OF
TRANSVERSE, AXIAL, OR COMBINED LONGITUDINAL-TRANSVERSE
INSTABILITIES IN A CYLINDRICAL COMBUSTION CHAMBER WITH
UNIFORM PROPELLANT INJECTION, DISTRIBUTED COMBUSTION
PROCESS, AND A CONVENTIONAL NOZZLE. THE COMBUSTION PROCESS
IS DESCRIBED BY CHOCCO'S TIME-LAG MODEL. BOTH TRANSIENT
AND LIMIT-CYCLE SOLUTIONS ARE CALCULATED.

THE FOLLOWING INPUTS ARE REQUIRED

(1) THE CONTROL NUMBERS, NOUTCF AND NOZNL2.
(2) THE COEFFICIENTS FROM PROGRAM COEFFS3D.
(3) THE DATA DECK.

NOUTCF DETERMINES PRINTOUT OF COEFFICIENTS.
  IF NOUTCF = 0 COEFFICIENTS ARE NOT PRINTED OUT.
  IF NOUTCF = 1 LINEAR COEFFICIENTS ONLY ARE PRINTED OUT.
  IF NOUTCF = 2 ALL COEFFICIENTS ARE PRINTED OUT.

NOZNL2 DETERMINES IF THE NOZZLE NONLINEARITIES ARE TO BE INCLUDED.
  IF NOZNL2 = 0 NOZZLE NONLINEARITIES NOT INCLUDED.
  IF NOZNL2 = 1 NOZZLE NONLINEARITIES INCLUDED.

THE DATA DECK CONTAINS THE FOLLOWING INFORMATION:

TITLE OF THE RUN.

EN IS THE INTERACTION INDEX.
TAU IS THE TIME-LAG.
H IS THE INTEGRATION STEP SIZE.
TSTART IS THE TIME AT WHICH OUTPUT STARTS.
TOU1T IS THE TIME AT WHICH COMPUTATIONS ARE TERMINATED.

NTEST IS TASK CONTROL NUMBER:
  IF NTEST = 0 COMPUTE TRANSIENT BEHAVIOR.
  IF NTEST = 1 COMPUTE THE LIMIT-CYCLE BEHAVIOR.
JMODE IS THE MODE-AMPLITUDE USED TO TEST FOR LIMIT-CYCLES.

NLOC DETERMINES THE LOCATION OF THE WALL PRESSURE MAXIMA
  AND MINIMA:
  IF NLOC = 1 LOCATION IS Z = 0, THETA = 0 DEGREES.
  IF NLOC = 2 LOCATION IS Z = 0, THETA = 45 DEGREES.
  IF NLOC = 3 LOCATION IS Z = 0, THETA = 90 DEGREES.

NTERMS IS THE NUMBER OF TERMS GIVEN INITIAL VALUES.

NPZ DETERMINES HOW MANY SECONDARY STABILITY ZONES (PHANTOM
ZONES) ARE HANDLED.
  IF NPZ = 0 PHANTOM ZONES ARE RETAINED.
  IF NPZ = 1 PHANTOM ZONES ARE ELIMINATED.

NOUT IS THE OUTPUT CONTROL NUMBER.
  IF NOUT = 0 PRINTED OUTPUT ONLY.
  IF NOUT > 0 BOTH PRINTED AND PLOTTED OUTPUT. NOUT
  DETERMINES THE NUMBER OF THE LAST PLOT
  PRODUCED.

ICTYPE IS THE INITIAL CONDITION CONTROL NUMBER:
  IF ICTYPE = 1 AMPLITUDES SELECTED TO SATISFY
```

**ORIGINAL PAGE IS OF POOR QUALITY**
THE NOZZLE BOUNDARY CONDITION.
AMPLITUDES SELECTED TO ELIMINATE THE EXTRANEOUS SOLUTION.

DATA FOR SETTING UP PLOTS:

YHI(1) IS THE MAXIMUM ORDINATE FOR PRESSURE PLOTS.
YHI(5) IS THE MAXIMUM ORDINATE FOR VELOCITY PLOTS.
NOTE: THE ORDINATE SCALES FOR PRESSURE AND VELOCITY PLOTS ARE SYMMETRIC ABOUT ZERO.

YLAB IS THE INTERVAL FOR ORDINATE LABELING FOR ABOVE PLOTS.
ICY IS THE NUMBER OF ORDINATE TIC MARKS FOR ABOVE PLOTS.
NOTE: ITICY SHOULD BE NEGATIVE FOR PRESSURE AND VELOCITY PLOTS TO OBTAIN CENTERLINE.

FIRST IS THE NUMBER OF THE FIRST PLOT PRODUCED.
NOMIT DETERMINES WHETHER AMPLITUDE PLOT IS PRODUCED
  IF NOMIT = 0 AMPLITUDE PLOT IS PRODUCED.
  IF NOMIT = 1 AMPLITUDE PLOT IS OMITTED.

MDPLOT DETERMINES IF THE PLOT OF THE MODE-AMPLITUDE IS REQUIRED.
  IF MDPLOT = 0 PLOT NOT REQUIRED.
  IF MDPLOT = 1 PLOT REQUIRED.

YHIMD IS THE MAXIMUM ORDINATE FOR AMPLITUDE PLOTS.
YLABMD IS THE INTERVAL FOR ORDINATE LABELING OF AMPLITUDE PLOTS.
ITICMD IS THE NUMBER OF ORDINATE TIC MARKS.
NOTE: ITICMD SHOULD BE NEGATIVE TO OBTAIN THE CENTERLINE.

INITIAL AMPLITUDES OF F-FUNCTIONS (REMAINING CARDS)

AS(J) IS THE AMPLITUDE OF THE SINE TERM.
AC(J) IS THE AMPLITUDE OF THE COSINE TERM.

DAMP AND FREQ ARE THE DAMPING COEFFICIENT AND THE FREQUENCY FROM THE LINEAR STABILITY PROGRAM.

PARAMETER
-MX=5, MX2=10, MX4=20, MX2SQ=100
COMPLEX YNOZ(MX), BMX, C1, C2, C3, CFHI(MX), CSUM, A
COMPLEX GNOZ(MX), CA2, CI
DIMENSION L(MX), N(MX), SMX, NAME(MX), AS(MX2), AC(MX2),
  UC(250, MX4), Y(MX4), FZ(4, MX4), YF(MX4), UZ(MX4),
  CF(4, MX2, MX2), FRG1(MX2), IMF1(MX2), UMAX(500),
  Z(6), ANGLE(6), IHETA(6), CFT(6, MX2), YI(MX2),
  CFTH(6, MX2), CFZ(6, MX2), FRESS(6), AXVEL(3), YR(MX2),
  TFL0T(500), YFLOT(6, 500), DUMMY(500), DUMMY(500),
  IBUF(3000), IIT(4), ITY(7), ITY(7), ITY(7),
  ITY(7), ITY(7), IITY(6), ITY(6), UAYG(100),
  ITF(3), ITT(12), PRT(500), TIC(500), PMAX(500),
  TIMAX(500), YLO(6), YHI(6), YLAB(6), ITICY(6),
  KFREQ(MX), WPK(MAX), AIA(4), UFLOT(MX, 500), PRT(500),
  MDPLOT(4), MTITLE(4), MTITLE(4), MTITLE(4),
  "TITL(4), PRTITL(5)"

ORIGINAL PAGE IS OF POOR QUALITY.
DATA

1 /"DIMENSIONLESS TIME, T"/
2 /"INJECTOR PRESSURE PERTURBATION, \( \Theta = 0^\circ \)"/
3 /"INJECTOR PRESSURE PERTURBATION, \( \Theta = 45^\circ \)"/
4 /"NOZZLE PRESSURE PERTURBATION, \( \Theta = 90^\circ \)"/
5 /"NOZZLE AXIAL VELOCITY, \( \Theta = 0^\circ \)"/
6 /"PRESSURE PEAKS"/
7 /"AMPLITUDE OF 1 T MODE"/
8 /"AMPLITUDE OF 2 T MODE"/
9 /"AMPLITUDE OF 1 H MODE"/
10 /"PRESSURE AMPLITUDE OF 1 T 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

*************** COEFFICIENT INPUT SECTION *********************

C THIS VERSION OF LCYC3D READS THE COEFFICIENT DATA FROM
A FASTRAND FILE GENERATED BY PROGRAM COEFF3D. TO READ
THIS DATA FROM CARDS, USE READ (9, XXXX) INSTEAD OF
READ (9, XXXX) IN THIS SECTION.

C INPUT OF MOTOR PARAMETERS AND NUMBER OF TERMS.
READ (9,5001) GAMMA, UE, ZE, ZCOME, NDROPS, NJMAX, NOZNL2
WRITE (6,6001) GAMMA, UE, ZE, ZCOME, 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
RLD = 0.5 * ZE

C

WRITE (6,6002)

C INPUT OF DESCRIPTION OF SERIES EXPANSION.
EO 10 K = 1, JMX
READ (9,5002) NJ, L(NJ), M(NJ), N(NJ), NS(NJ), SC(NJ), SJ(NJ)

C

114
WRITE (6,6003) NAME(NJ), NJ, L(NJ), M(NJ), N(NJ), NS(NJ), S(NJ), SJ(NJ)
10 CONTINUE

WRITE (6,6010)
DO 15 K = 1, JM
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, JM
READ (9,5011) J, GNOZ(J)
WRITE (6,6035) J, GNOZ(J)
820 CONTINUE
815 CONTINUE

C CALCULATE THE NUMBER OF TYPES OF LINEAR COEFFICIENT.
NCOEFF = 4
IF (NOZNL1 .EQ. 1) NCOEFF = 5
NCFM1 = NCOEFF - 1

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 ZERO NONLINEAR COEFFICIENT ARRAY.
DO 30 NJ = 1, MX2
DO 30 NP = -1, MX2
D(NJ,NP) = 0.0
30 CONTINUE

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 INPUT OF NONLINEAR COEFFICIENTS.
READ (9,5003) NLMAX
1F (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, NF, NQ, DT
IF (NOUTCF *EQ* 2) WRITE (6,6007) NJ, NF, NQ, DT
KPQ = KPQMAX(NJ)
D(NJ,KPQ) = DT
55 CONTINUE
50 CONTINUE

C *************** PRESSURE COEFFICIENT SECTION ***********************

C CALCULATE SPATIAL COORDINATES FOR PRESSURE COMPUTATION.
DO 51 NFRES = 1, 3
Z(NFRES) = 0.0
RTHEA = NFRES - 1
ANGLE(NFRES) = RTHEA * 45.0
THETA(NFRES) = RTHEA * FI/4.0
Z(NFRES + 3) = ZE
ANGLE(NFRES + 3) = ANGLE(NFRES)
THETA(NFRES + 3) = THETA(NFRES)
51 CONTINUE

C CALCULATE COEFFICIENTS FOR PRESSURE TIME HISTORIES.
DO 53 NFRES = 1, 6
DO 53 J = 1, MX2
NP = (2 * J) - 1
Z1 = Z(NFRES)
ANG = THETA(NFRES)
CALL PHICFS(J,Z1,ANG,C1,C2,C3)
IF (NFRES *EQ* 4) CPHIT(J) = C1
CFTCNFRES,NF) = REAL(C1)
CFTCNFRES,NP+1) = -AIMAG(C1)
CFTHCNFRES,NF) = REAL(C2)
CFTHCNFRES,NP+1) = -AIMAG(C2)
CFZCNFRES,NF) = REAL(C3)
CFZCNFRES,NP+1) = -AIMAG(C3)
53 CONTINUE

C C1 = (0.0,1.0)
CAXI = GAMMA * CCOSH(C1 * B(1) * ZE)
CAXII = REAL(CAXI)
CAXII = AIMAG(CAXI)

C OUTPUT OF COEFFICIENTS FOR PRESSURE TIME HISTORIES.
WRITE (6,6020)
DO 56 NFRES = 1, 6
WRITE (6,6014)
DO 56 J = 1, MX2
C *************** DATA INPUT SECTION ***************************************
C READ (5*,5000) TITLE
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 READ COMBUSTION AND CONTROL PARAMETERS.
   READ (5,5006, END = 300) EN, TAU, H, ISTART, TGUIT
C READ CONTROL NUMBERS.
   READ (5,5008) NIEST, JMODE, NLOC, NTERM, NFZ, NOUT, ICTYFE
   JMODE = (2 * JMODE) - 1
   IF (NOUT .GT. 0) GO TO 825
   IF (NOUT .EQ. 0) GO TO 9
   KFREQ = 5(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
C READ DATA FOR SETTING UP PLOTS.
   READ (5,5009) YHI(1), YHI(5), YLPBI), YLAB(5)
   READ (5,5014) MDPLOT
   MDPLTL = 0
   DO 320 K = JMX, 1, -1
      MDPLTL = MDPLTL + MDPLOT(K)
320 CONTINUE
C IF (MDPLTL .LE. 0) GO TO 9
   READ (5,5015) YHIMD, YLAPMD, ITICMD
   YLOMD = - YHIMD
C *************** INITIAL AMPLITUDES SECTION *****************************
C 9 DO 58 K = 1, NTERM
C INPUT INITIAL AMPLITUDES FOR F-FUNCTIONS.
   READ (5,5007) J, AST, ACT
   NJ = (2 * J) - 1
   AS(NJ) = AST
AC(NJ) = ACT

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)
IMP1(NJ) = 0.0
GO TO 586
584 LONG = L(J)
SMN = S(J)
READ (5, 5099) DAMP, FREQ
IMP1(NJ) = DAMP
FRQ1(NJ) = FREQ
586 CONTINUE
FRQ1(NJ+1) = FRQ1(NJ)
IMP1(NJ+1) = IMP1(NJ)

IF (ICTYPE .EQ. 2) GO TO 582
CALCULATE INITIAL AMPLITUDES FOR G-FUNCTIONS.

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

NFRES = 4
IF (NS(J) .EQ. 1) NFRES = 6

A1 = (1.0 + GYRU)*CFZ(NFRES, NJ+1)
1 - GYIF*CFZ(NFRES, NJ+1)
A2 = GYRF*CFZ(NFRES, NJ+1) + GYIU*CFZ(NFRES, NJ+1)
A3 = -(1.0 + GYRU)*CFZ(NFRES, NJ) + GYIF*CFZ(NFRES, NJ)
A4 = GYRF*CFZ(NFRES, NJ) + GYIU*CFZ(NFRES, NJ)

DET = A1*A1 + A2*A2
IF (.DET .LT. 0.0000001) GO TO 583
R1 = A3*AC(NJ) - A4*ASC(NJ)
R2 = -A4*AC(NJ) - A3*AS(NJ)

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

FL = 0.0
GO TO 586

582 ARG = FRQ1(NJ) * TAU
FSIN = SIN(ARG)
FCOS = 1.0 - COS(ARG)
FS0 = FRQ1(NJ) * FRQ1(NJ)
DSQ = IMP1(NJ) * IMP1(NJ)
A1 = DSQ - FSQ + IMP1(NJ) * (CP(2,NJ,NJ)
1 . - EN * CF(3,NJ,NJ) * FCOS
+ EN * CF(3,NJ,NJ) * FRG1(NJ) * FSIN
3 + CP(1,NJ,NJ)
A2 = (2*0 * IMP1(NJ) + CP(2,NJ,NJ)
1 - EN * CP(3,NJ,NJ) * FCOS) * FRG1(NJ)
2 - EN * CP(3,NJ,NJ) * IMP1(NJ) * FSIN
A3 = CP(2,NJ,NJ+1) * IMP1(NJ) + CP(1,NJ,NJ+1)
A4 = CF(2,NJ,NJ+1) * FRG1(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 OUTPUT OF INITIAL AMPLITUDES.
C
WRITE (6,6016)
DO 590 J = 1, NJMAX
IF (AS(J)) 591, 592, 591
592 IF (AC(J)) 591, 592, 591
591 WRITE (6,6017) J, DMPI(J), FRG1(J), AC(J), AS(J)
590 CONTINUE
C
IF (NTEST *EQ. 0) WRITE (6,6025)
IF (NTEST *EQ. 1) WRITE (6,6026)
C
IF (NPZ *EQ. 1) WRITE (6,6028)
C
IF (NOUT *EQ. 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, JMAX
NJ = (2 * J) - 1
RL = L(J)
AX = RL * FI/ZE
AXS2 = AX * AX
SS2 = S(J) * S(J)
OMEGA = SQRT(SS2 + AXS2)
TAUCUT(NJ) = 2.0 * FI/OMEGA
TAUCUT(NJ+1) = TAUCUT(NJ)
602 CONTINUE
C
DO 604 NJ = 1, NJMAX
DO 604 NF = 1, NJMAX
IF (TAU .GT. TAU_CUT(NP)) CF3, NJ, NF) = 0.0
604 CONTINUE
C
C COMPUTE LINEAR COEFFICIENTS FOR GIVEN VALUES OF EN AND TAU.
605 DO 60 NJ = 1, NJ_MAX
   DO 60 NF = 1, NF_MAX
   CT = CF1, NJ, NF)
   IF (CT) 61, 62, 61
   61 KPMAX(1, NJ) = KPMAX(1, NJ) + 1
   KP = KPMAX(1, NJ)
   IC(1, NJ, KP) = NF
   CC1, NJ, KP) = CT
   62 CT = CF2, NJ, NF) - EN * CF3, NJ, NF)
       IF (CT) 63, 64, 63
   63 KPMAX(2, NJ) = KPMAX(2, NJ) + 1
   KP = KPMAX(2, NJ)
   IC(2, NJ, KP) = NF
   CC2, NJ, KP) = CT
   64 CT = EN * CF3, NJ, NF)
       IF (CT) 65, 66, 65
   65 KPMAX(3, NJ) = KPMAX(3, NJ) + 1
   KP = KPMAX(3, NJ)
   IC(3, NJ, KP) = NF
   CC3, NJ, KP) = CT
   66 IF (N0ZNL2 .NE. 1) GO TO 60
   67 IF (CT) 61, 62, 61
   60 CONTINUE
C
C *************** STEP-SIZE COMPUTATION *****************************************
C
NDIV = 1.0 + TAU/H
RN = NDIV
H = TAU/EN
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, NJ_MAX
   JP = J + NJ_MAX
   IF (AC(J)) 751, 753, 751
IF (AS(J)) 751, 752, 751
U(I,J) = 0.0
U(I,JP) = 0.0
GO TO 75
ARG = FRQ1(J) * TIME
FSIN = SIN(ARG)
FCOS = COS(ARG)
FEXP = EXP(DMFI(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 + DMFI(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
B(J) = U(I,J)
703 CONTINUE
UBAR = 0.0
IF (NPRES *GT. 3) UBAR = UE
UMS = 0.0
IF ((NPRES.GT.3) .AND. (NPRES.LT.4)) UMS = UE/(2.E+ZCOME)
CALL PFSVEL(UBAR, UMS, Y, P, VTH, VZ)
PRESS(NPRES) = P
IF (NPRES *GT. 3) AXVEL(NPRES - 3) = VZ
704 CONTINUE
FRS(I) = FRESS(NLOG)
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 = CMFLX(FT, GT)
CSUM = CSUM + YNOZ(J) * CPHIT(J) * A
710 CONTINUE
SUM = REAL(CCSUM)
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 *************** INITIALIZE CONTROL NUMBERS ***********************
C LINE = 8
K = 0
MAXNO = 0
MAXP = 0
IF (NOT = EQ. 0) GO TO 100
JFLOT = 0
TMIN = TSTART
TMAX = TSTART + TDEL
YLO(J) = -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
*************** NUMERICAL CALCULATIONS SECTION ****************

C
100 I = NPI
C
C RUNGE-KUTTA INTEGRATION SCHEME
105 NSTEP = (I - NPI + (LAST - NPI) * K)
RSTEF = NSTEP
TIME = RSTEF * H
TI(I) = TIME
DO 110 J = 1, NJMAX
JF = J + NJMAX
RV(J,1) = U(I-NDIVIJ)
RV(J,4) = U(I-NDIV+1,JF)
RV(J,2) = 0.375*V(JI) + 0.75*RV(J,4) - 0.125*U(I-NDIV+2,JF)
RV(J,3) = RV(J,2)
110 CONTINUE
IF (NOZNL2 *NE. 1) GO TO 10
835
II = 1,4
DO 840 J = 1,JMAX
JODD = 2*J + 1
JEVEN = 2*J
EXTRA(JODD,II) = COS(WK(JI)*TZ)
EXTRA(JEVEN,II) = SIN(WK(JI)*TZ)
840 CONTINUE
835 CONTINUE
DO 120 J = 1, NU
Y(J) = U(I,J)
120 CONTINUE
CALL RHS(NUI,Y,YP)
DO 130 J = 1, NU
FZ(J+1) = 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 RH5(NU, IJ, UZ, YF)
DO 148 J = 1, NU
FZ(IJ, J) = YF(J)
148 CONTINUE
DO 150 J = 1, NU
U1(I+1, J) = Y(J) + (FZ(1, J) + 2.0* (FZ(2, J) + FZ(3, J)) + FZ(4, J)) * H6
150 CONTINUE

C CALCULATE PRESSURE TIME HISTORIES.
DO 154 NFRES = 1, 6
DO 152 J = 1, NJMAX
COEF(1, J) = CFTH(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/(ZETZCONE)
CALL PRESVEL(UBAR, UMS, Y, P, UTH, UZ)
PRES(NPRES) = P
IF (NPRES .GT. 3) AXVEL(NPRES = 3) = UZ
154 CONTINUE
PRES(I) = PRESS(NLOC)

C CALCULATE VALUES OF NOZZLE E.C.
CSUM = (0.0, 0.0)
DO 650 J = 1, JMAX
JP = NJMAX + (2 * J) - 1
FT = Y(JP)
GT = Y(JP+1)
A = CMPLX(FT, GT)
CSUM = CSUM + YN0Z(J) * CFHIT(J) * A
650 CONTINUE
SUM = REAL(CSUM)
YPHI = -GAMMA * SUM

C DETERMINE MAXIMA AND MINIMA OF PRINCIPAL MODE-AMPLITUDE
C FUNCTION FOR USE IN DETERMINING LIMIT-CYCLE BEHAVIOR.
IF (U(I, JMODE) * U(I+1, JMODE)) 170, 170, 160
170 PDEN = U(I, JMODE) - U(I+1, JMODE)
IF (PDEN) 171, 160, 171
171 PP = U(I, JMODE)/PDEN
PA = (PP - 1.0) * PP * 0.5
PB = 1.0 - (PP * PP)
PC = (PP + 1.0) * PP * 0.5
MAXNO = MAXNO + 1
IMAX(MAXNO) = PA*UI-I-1, JMODE) + PB*UI, JMODE) + PC*UI+1, JMODE)
IF (MAXNO .GE. 500) 60 TO 250
160 CONTINUE
C DETERMINE MAXIMUM AND MINIMUM PRESSURE AT LOCATION SPECIFIED
C BY NLOC.
LPL - PRS(I) - PRS(I-1)

173  FNUM = FRHS(I-2) - FRHS(I)

174  FDEN = 2.0 * (PRH(I-2) + PRH(I) - 2.0 * PRH(I-1))

IF (FDEN) 174, 175, 174

175  CONTINUE

FA = (PP - 1.0) * PP * 0.5

FB = 1.0 - (FF * FF)

FC = (PP + 1.0) * FF * 0.5

MAXP = MAXP + 1

TIMAX(MAXP) = T(I-1) + FF*H

IF (MAXP .GE. 500) GO TO 250

C

IF (NTEST .LE. 1) GO TO 155

IF (TIME .LT. TSTART) GO TO 155

IF ((NOUT .EQ. 0) .OR. (NOUT .GT. 6)) GO TO 156

C

FILL INJECTOR PRESSURE ARRAYS FOR PLOTTING (THETA = 0

1001 DO J = 1, 3

YFLOT(J, JPLOT) = PRESS(J)

1001 CONTINUE

C

FILL NOZZLE PRESSURE ARRAY FOR PLOTTING (THETA = 0

YPLOT(4, JPLOT) = PRESS(4)

C

FILL NOZZLE AXIAL VELOCITY ARRAY FOR PLOTTING (THETA = 0

YPLOT(5, JPLOT) = AXVEL(1)

C

FILL NOZZLE B.C. ARRAY FOR PLOTTING (THETA = 0

YPLOT(6, JPLOT) = YFHI

C

IF (MDPLOT .EQ. 0) GO TO 156

C

FILL MODE AMPLITUDE ARRAYS FOR PLOTTING

DO 322 J = 1, JMX

322 CONTINUE

C

ORIGINAL PAGE IS OF POOR QUALITY
PRIT(JPLOT) = CAXI*U(I,J1T1) - CAXII*U(I,J1T2)

C 1000 NUM = JPLOT
C FLOT TIME HISTORIES.
DO 1020 NFPLOT = NFIRST, NOUT
C
JFPLOT = 0
C ASSIGN FLOTTING PARAMETERS.
YMIN = YLOC(NFPLOT)
YMAX = YHI(NFPLOT)
NTICY = ITICY(NFPLOT)
DELY = YLAB(NFPLOT)
C ELIMINATE POINTS THAT ARE OUT OF THE OCALINE RANGE.
C 1010 J = 1, NUM
IF ((YFPLOT(NFPLOT,J) .LT. YMIN) .OR. (YFPLOT(NFPLOT,J) .GT. YMAX))
1  GO TO 1010
JFPLOT = JFPLOT + 1
DUMMYT(JFPLOT) = TFPLOT(J)
DUMMYY(JFPLOT) = YFPLOT(NFPLOT,J)
1010 CONTINUE
C IF (JFPLOT .EQ. 0) GO TO 1020
C DO 1020 (1011, 1012, 1013, 1014, 1015, 1016), NFPLOT
C FLOT INJECTOR PRESSURE AT THETA = 0 DEGREES.
1011 CALL GRAPHS(IBUF, 3000, 4, JFPLOT, 51, NTICY, TMAX, YMAX, TMIN, YMIN, 
1  IITT, IITY, 21, 41, DUMTYT, DUMMYY, 2, 0, DELY, TITLE)
1020 CONTINUE
C FLOT INJECTOR PRESSURE AT THETA = 45 DEGREES.
1012 IF (M(JNODE) .EQ. 0) GO TO 1020
CALL GRAPHS(IBUF, 3000, 4, JFPLOT, 51, NTICY, TMAX, YMAX, TMIN, YMIN, 
1  IITT, IITY, 21, 42, DUMTYT, DUMMYY, 2, 0, DELY, TITLE)
1020 CONTINUE
C FLOT INJECTOR PRESSURE AT THETA = 90 DEGREES.
1013 IF (M(JNODE) .EQ. 0) GO TO 1020
CALL GRAPHS(IBUF, 3000, 4, JFPLOT, 51, NTICY, TMAX, YMAX, TMIN, YMIN, 
1  IITT, IITY, 31, 42, DUMTYT, DUMMYY, 2, 0, DELY, TITLE)
1020 CONTINUE
C FLOT NOZZLE PRESSURE AT THETA = 0 DEGREES.
1014 CALL GRAPHS(IBUF, 3000, 4, JFPLOT, 51, NTICY, TMAX, YMAX, TMIN, YMIN, 
1  IITT, IITY, 41, 39, DUMTYT, DUMMYY, 2, 0, DELY, TITLE)
1020 CONTINUE
C FLOT NOZZLE AXIAL VELOCITY AT THETA = 0 DEGREES.
1015 CALL GRAPHS(IBUF, 3000, 4, JFPLOT, 51, NTICY, TMAX, YMAX, TMIN, YMIN, 
1  IITT, IITY, 51, 32, DUMTYT, DUMMYY, 2, 0, DELY, TITLE)
GO TO 1020

C PLOT NOZZLE B.C. AT THEIA = 0 DEGREES.

1016 CALL GRAPHS(IBUF,3000,4,JFLOT,51,NTICM,MAX,YMAX,TMIN,YMIN,
1 ITITYO,21,44,DUMMY,T,DUMMY,2.0,DELY,TITLE)

C 1020 CONTINUE

C IF (MDFLTL*EQ.0) 60 TO 330
DO 324 NJLOT = 1, JMX
IF (MDFLOT(NLOT) *EQ. 0) 60 TO 324
JFLOT = 0
DO 328 J123 = 1, JLOT
IF (NJLOT *EQ. 1) MTITL(J123) = MTITL(J123)
IF (NJLOT *EQ. 2) MTITL(J123) = MTITL2(J123)
IF (NJLOT *EQ. 3) MTITL(J123) = MTITL3(J123)
328 CONTINUE

C DO 326 J = 1, NUM
IF ((UFLOT(NLOT,J) + LT* YLORD) OR+ (UFLOT(NLOT,J) + GT* YHIMD)) 60 TO 326
JFLOT = JFLOT + 1
DUMMY(JJLOT) = TLOT(J)
DUMMY(JJLOT) = UFLOT(NLOT,J)
326 CONTINUE

C IF (JFLOT *EQ. 0) GO TO 324

C PLOT AMPLITUDES OF DIFFERENT MODES.
CALL GRAPHS(IBUF,3000,4,JFLOT,51,NTICM,MAX,YHIMD,TMIN,
1 YLORD,ITTY,MTITL,21,44,DUMMY,T,DUMMY,2.0,YLARY,R,TITLE)
324 CONTINUE

C IF (MDFLOT(4) *EQ. 0) 60 TO 330
JLOT = 0
DO 332 J = 1, NUM
IF ((FRIT(J) + LT* YLORD) OR+ (FRIT(J) + GT* YHIMD)) 60 TO 332
JFLOT = JFLOT + 1
DUMMY(JJFLOT) = TLOT(J)
DUMMY(JJFLOT) = FRIT(J)
332 CONTINUE

C IF (JLOT *EQ. 0) GO TO 330

C PLOT PRESSURE AMPLITUDE OF 17 MODE.
CALL GRAPHS(IBUF,3000,4,JFLOT,51,NTICM,MAX,YHIMD,TMIN,
1 YLORD,ITTY,FRITL,21,44,DUMMY,T,DUMMY,2.0,YLARY,R,TITLE)
330 CONTINUE

C REASSIGN PLOTTING PARAMETERS FOR NEXT SET OF PLOTS.
JLOT = 0
TMIN = TMAX
TMAX = TMAX + TLEL

C ****************** TIME HISTORY PRINTED OUTPUT SECTION ******************

156 WRITE (6,6011) NSTEF, TIME, (PRESS(J), J = 1,6)
\[
1 \quad \text{(AXVEL(J), J = 1, 3), YPHI}
\]

177 IF (TIME .GT. TQUIT) GO TO 250
179 IF (LINE .LT. 52) GO TO 155
182 WRITE (6*6013)
184 WRITE (6*6022) (ANGLE(J), J = 1, 6), (ANGLE(J), J = 1, 3)
LINE = 4
155 I = I + 1
157 IF (I .LT. LAST) GO TO 105

*************** LIMIT-CYCLE SECTION ******************************

C TEST FOR LIMIT CYCLE.
K = K + 1
190 IF ((NTEST .EQ. 0) .OR. (MAXNO .LT. 80)) GO TO 190
255 USTAB T = TIME/2.0
257 LST = LSTART + 2
258 TM = LST
259 TSTOP = TM + TSTOP
250 CONTINUE
GO TO 100

*************** PRESSURE MAXIMA AND MINIMA PRINTOUT ***************

250 WRITE (6*6023) Z(NLOC), ANGLE(NLOC), MAXP
LINE = 4
255 JST = 1, MAXF, 8
JSTART = JST
JSTOP = JST + 7
IF (JSTOP .GT. MAXF) JSTOP = MAXF
WRITE (6,6024) (FMAX(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

************** PRESSURE MAXIMA FLOATING SECTION **************

C
Determine largest value of FMAX.
AMFPAX = 0
DO 260 J = 1, MAXF
IF (FMAX(J) .LT. AMFPAX) GO TO 260
AMFPAX = FMAX(J)
260 CONTINUE
C
Range of float and coordinate labeling.
ITM = AMFPAX + 1.0
AMFMAX = ITM
ITM = ITM + TIMAX(MAXP)/50.0
TMAX = ITM * 50
DELY = TMAX/10.0
DELX = AMFPAX/10.0
C
Eliminate negative values.
JPLCT = 0
DO 262 J = 1, MAXF
IF (FMAX(J)) 262, 264, 264
264 JPLCT = JPLCT + 1
DUMMY(JPLCT) = TIMAX(J)
DUMMY(JPLCT) = FMAX(J)
262 CONTINUE
C
Plot values.
CALL GRAPHS(IBUF,3000,4,JPLCT,101,101,TMAX,AMFPAX,0.0,0.0,
1
IT,IT,F,21,14,DUMMY,DUMMY,DELX,DELY,TITLE)
C
GO TO 5
C
Turn off plotting routine.
300 IF (NFT .EQ. 1) CALL SHFARG
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 (15,2F10.0)
5008 FORMAT (7I5)
5009 FORMAT (7F10.0)
5010 FORMAT (15,F10.5)
5011 FORMAT (15,2F10.5)
5012 FORMAT (F10.0)
5013 FORMAT (2I5)
5014 FORMAT (4F9.5)
5015 FORMAT (2F10.0)
5099 FORMAT (2F10.0)

********** WHITE FORMAT Specifications **********:

C
C
C

6001 FORMAT (1H19H GAMMA = ,F5.3, 5X, 5HUE = ,F5.3,
1 5X, 5HZE = ,F8.5, 5X, 8HZCOMB = ,F5.0,
2 5X, 8HNJMAX = ,I12//)
6002 FORMAT (2X,29HNAME, J,L,M,N,N5,7X,3HSN,3X,
1 7HJM(SMN)//)
6003 FORMAT (2X,A4,5I5,2F10.5)
6004 FORMAT (1HO,29H NUMBER OF COEFFICIENTS C(I1,10H,N,NJ,NO) IS,15/)
6005 FORMAT (2X,8HHC, 11,1H,12,1H,12,4H) = ,F10.5)
6006 FORMAT (1HO,39HN NUMBER OF COEFFICIENTS D(NJ,NO,NO) IS,15/)
6007 FORMAT (2X,8HHC,12,1H,12,1H,12,4H) = ,F10.5)
6008 FORMAT (1HO,45HN COMBUSTION PARAMETERS: INTERACTION INDEX = ,F7.5,
1 12X,11HTIME-LAG = ,F7.5/2X,17H MOTOR PARAMETERS: 19X,
2 8H GAMMA = ,F7.5,2EH EXIT MACH NUMBER = ,F7.5,
3 22H LENGTH/DIAMETER = ,F7.5//)
6009 FORMAT (2X,18H INITIAL CONDITIONS//)
6010 FORMAT (1HO,5X,1HO,6X,2HY,8X,2HY1,7X,3HEFS,7X,3HEFA//)
6011 FORMAT (2X,15,F12,5,10F1.5)
6012 FORMAT (1HO)
6013 FORMAT (1H1)
6014 FORMAT (1H)
6015 FORMAT (2X,15,4F10.5)
6016 FORMAT (1HO,36H INITIAL CONDITIONS ARE OF THE FORM://
1 2X,49H(U(I,J) = AC(J)+*COS(FREQ*T) + AS(J)*SIN(FREQ*T)),
2 14H * EXP(DAMP*T)///6X,1HO,7X,7DAMPING,
3 6X,9H FREQUENCY, 10X, SHAC(J), 10X, SHAS(J)//)
6017 FORMAT (2X,15,4F10.5)
6018 FORMAT (1HO,46H COEFFICIENTS FOR COMPUTATION OF WALL PRESSURE,
1 10H WAVES // 4X,27H COEFFICIENTS IN SERIES FOR //
2 22X,5HTHETA, 10X, 4HTIME, 10X, 5HTHETA, 10X, 5HAXIAL/
3 6X, 1HO, 9X, 1HZ, 3X, 9H(Degrees), 5X, 10H DERIVATIVE,
4 5X, 10H DERIVATIVE, 5X, 10H DERIVATIVE//)
6019 FORMAT (2X,15,F10.3,F12,1,3F15.7)
6020 FORMAT (2X,17HINJECTOR PRESSURE, 14X,15H NOZZLE PRESSURE,
1 12X,11HNOZZLE AXIAL VELOCITY/3X, 4H STEF, 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 6X, 4HYPHI//)
6021 FORMAT (1H1,39H PRESSURE MAXIMA AND MINIMA AT: Z = ,F5.2,
1 11H THETA = ,F4.1/19H VALUES COMPUTED: ,I3//)
6022 FORMAT (1HO,7X,8F13.5)
6023 FORMAT (2X,17H THE TRANSIENT behavior IS calculated.,
1 2X,17H THE LIMIT-CYCLE behavior is calculated.,
6024 FORMAT (2X,17H THIS RUN PRODUCES PLOTTED OUTPUT.)
6025 FORMAT (2X,17H "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 (1HO,6X,1HJ,10X,2HGR,10X,2HGI)
6035 FORMAT (5X,15,2F12.5)
END
SUBROUTINE PHICFS(NP, Z, THETA, CT, CTH, CZ)

C THIS SUBROUTINE COMPUTES THE COEFFICIENTS NEEDED TO
C CALCULATE THE WALL PRESSURE PERTURBATION.
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 OF
C THE VELOCITY POTENTIAL.
C CTH IS THE COEFFICIENT IN THE SERIES FOR THE Theta
C DERIVATIVE OF THE VELOCITY POTENTIAL.
C CZ IS THE COEFFICIENT IN THE SERIES FOR THE AXIAL
C DERIVATIVE OF THE VELOCITY POTENTIAL.
C
PARAMETER MX = 5
COMPLEX CI, CZ, CAXI, CAXIZ, CRAD, CAZI, CAZITH,
1 COMMON /BLK2/ M(MX), NS(MX), SJ(MX), B
C
CI = (0.0, 1.0)
CZ = CMPLX(Z, 0.0)
CAXI = CCOSH(CI * B(NF) * CZ)
CAXIZ = CI * B(NF) * CSINH(CI * B(NF) * CZ)
CRAD = CMPLX(SJ(NF), 0.0)
EM = M(NP)

ARG = EM * THETA
FSIN = SIN(ARG)
FCOS = COS(ARG)
AZI = FCOS
IF (NS(NP) == 1) AZI = FSIN
AZITH = EM * FCOS
IF (NS(NP) == 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 PESVEL(UBAR, UMS, Y, F, VTH, VZ)

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)

DO 10 I = 1, 4
   SUM(I) = 0.0
10 CONTINUE

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

FLIN = SUM(1) + UBAR*SUM(3) + UMS*SUM(4)
PFL = 0.0
   IF (NLMAX .EQ. 0) GO TO 40
   DO 30 I = 1, 3
      SUMSQ(I) = SUM(I) * SUM(I)
30 CONTINUE
PFL = 0.5 * (SUMSQ(2) + SUMSQ(3) - SUMSQ(1))

40 P = -GAMMA * (FLIN + PFL)
   VTH = SUM(2)
   VZ = SUM(3)

RETURN
END
SUBROUTINE RHS(NJ,II,UI,UP)

PARAMETER MX=5, MX2=10, MX4=20, MX2SC=100

DIMENSION UCNU, UF(NJ)

COMMON RV(MX2,4), IC(MX2,MX2), D(MX2,MX2SC)

COMMON KFMAX(MX2), IC(4,MX2, MX2), KFQMAX(MX2)

COMMON /BLK3/ NUMAX, NLMAX, GAMMA, COEF(3,MX2)

COMMON /NLTERM/ N0ZNL2, EXTRA(MX2,4)

DO 10 NJ = 1, NJMAX

NJF = NJ + NJMAX

UP(NJ) = UC(JF)

SL1 = 0.0

SL2 = 0.0

SL3 = 0.0

SL4 = 0.0

SNL = 0.0

MAX = KFMAX(1,NJ)

IF MAX .EQ. 0 GO TO 25

DO 20 KP = 1, MAX

NP = IC(NJ,KP)

SL1 = SL1 + (C(1,NJ,KP) * UCNP)

20 CONTINUE

MAX = KFMAX(2,NJ)

IF MAX .EQ. 0 GO TO 35

DO 30 KP = 1, MAX

NP = IC(2,NJ,KP) + NJMAX

SL2 = SL2 + (C(2,NJ,KP) * UC(NP))

30 CONTINUE

MAX = KFMAX(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

IF (NOZNL2 .NE. 1) GO TO 55

MAX = KFMAX(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

IF (NLMAX .EQ. 0) GO TO 55

MAX = KFQMAX(NJ)

IF MAX .EQ. 0) GO TO 55

DO 50 KP = 1, MAX

NP = IDP(NJ,KP)

NOF = IDG(NJ,KP) + NJMAX

SNL = SNL + (D(NJ,KP) * UC(NF) * U(NPF))

50 CONTINUE

UP(NJF) = -(SL1 + SL2 + SL3 + SL4 + SNL)

10 CONTINUE

RETURN

END
SUBROUTINE GRAPHS(IBUF,NLOG,LDEV,NTOT,NTICX,NTICY,
1 XMAX,YMAX,XMIN,YMIN,ITITLX,ITITLY,LTITLX,LTITLY,XARRAY,
2 YARRAY,DELX,DELY,TITLE)

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 INTEO/1/
DATA ABSCIS/8./
DATA ORDINA/6./
DATA ICODE/-1/

134
DATA TOPMAR/1.0/
DATA BOTMAR/1.5/
REAL LEMAR
DATA LEMAR/1.9/
DATA RYTMAR/1.1/
DATA FACT/1.0/
DATA MAXIS/1.0/
DATA MLINE/1.0/
DATA HTLAB/.105/

C -----------------------------------------------
19 INITIAL COMPUTATION OF DERIVED PARAMETERS
AND INITIAL PLOTS CALL
20 SKIPS PRELIMINARIES FOR 2ND AND SUBSEQUENT CALLS
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
SYMLH = 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(I.)
J = 2
CALL SYMBOL (HEIGHT, 36*HEIGHT + 5.5*HEIGHT, TITLE, 270., 72)
CALL PLOT (1., -.5, -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 RESET ORIGIN
C -----------------------------------------------

XPAGE = BOTMAR + ORDINA
GO TO 2019
20
XPAGE = BOTMAR + ORDINA + TOPMAR
2019
CALL WHERE(XPAGE, YPAGE, FACT)
YPAGE = YPAGE - LEMAR
CALL PLOT(XPAGE, YPAGE, -3)
CALL FACTOR(FACT)
C ---------------------------------------------------------
C
C PROGRAM AXES AND LABELING MAXIS TIMES
C ---------------------------------------------------------
DO 100 I = 1,MAXIS
100 CALL MYAXIS
C ---------------------------------------------------------
C DRAW POINTS, OPTIONAL CENTERLINE, AND PAGE SCISSOR MLINES TIMES
C ---------------------------------------------------------
DO 200 I = 1,MLINE
200 CALL MYLINE
RETURN
C ---------------------------------------------------------
C ENTRY POINT SHPARG
C TERMINATE PLOTTING SEQUENCE
C ---------------------------------------------------------
ENTRY SHPARG
CALL WHERE(RXPAGE,RYPAGE,I)
CALL PLOT(RXPAGE,RYPAGE,999)
RETURN
C ---------------------------------------------------------
C SUBROUTINE MYAXIS (INTERNAL)
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.,ITITLE,LTITLE,HTLAB)
K = 2
12 FPX = YMAX - I * DELY
IF(ZERO) FPX = 0.
TMID = 1.
XPAGE = - I * ORDINA/IMAX - .5 * HEIGHT
IF(FP(X) 113,115,116
113 IF(NDEC - 2)) 115,114,112
114 YPAGE = STARTL@5CHAR
C ---------------------------------------------------------
GO TO 112
115 IF(NDEC = 1)117,116,112
116 YPAGE = STARTL - HEIGHT*CHAR
   GO TO 112
117 IF(ABS(FPN) = 100.)119,116,116
119 IF(ABS(FPN) = 10.)120,121,121
120 YPAGE = STARTL - 3 * HEIGHT*CHAR
   GO TO 112
121 YPAGE = STARTL - 2 * HEIGHT*CHAR
   GO TO 112
122 YPAGE = STARTL - 4 * HEIGHT*CHAR
   GO TO 112
118 IF(NDEC = 2)123,116,112
123 IF(NDEC = 1)125,124,112
124 IF(FPN = 10.)127,126,121
125 IF(FPN = 100.)122,120,126
126 IF(FPN = 1000.)121,116,128
127 IF(FPN = 10000.)116,114,114
128 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 + 1/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 25 I = 1,IMAX
      STARTL = - I * ABSCIS/IMAX
      GO TO (24,25),K
24 IF(2 * I.LT.IMAX)GO TO 25
   CALL AXLAB(270.,1T1LX,1T1LX,1T1LX,1T1LX)
   K = 2
25 XPAGE = - ASTART - ORDINA
   FPN = XMIN + 1 * DELX
   IF(ZERO)FPN = 0.
   IF(FPN)813,822,818
813 IF(NDEC = 2)815,814,23
814 YPAGE = STARTL + 16./7.* HEIGHT*CHAR
   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.)821,816,816
827  IF(FPN - 1000.)821,816,816
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
760  TMID = I.
     YPAGE = - I * ABSCIS/IMAX.
     DO 27 JJ = 1,N
800  XPAGE = - ORDINA - TICKLE * TMID
     CALL PLOT(XPAGE,YPAGE,3)
801  XPAGE = XPAGE + (TICKLE + FLD(0.36,1.NE-0) * TICKLE) * TMID
     CALL PLOT(XPAGE,YPAGE,2)
     IF(1)111,26,111
111  XPAGE = - ORDINA
     CALL PLOT(XPAGE,YPAGE,3)
811  YPAGE = YPAGE + TICSEP
     CALL PLOT(XPAGE,YPAGE,2)
812  TMID = -5
27  CONTINUE
26  CONTINUE
     RETURN

SUBROUTINE MYLINE (INTERNAL)

SUBROUTINE MYLINE
    ITOP = IFIX((ABSCIS + RYTRAR + *5)/11. * 99.)
    IBOT = IFIX(RYTRAR/11. * 99.)
    DO 17 I = 1,NTOT
807  XPAGE = (YARRAY(I) - YMAX)/(YMAX - YMIN) * ORDINA
808  YPAGE = (XMIN - XARRAY(I))/(XMAX - XMIN) * ABSCIS
817  CALL SYMBOL(XPAGE,YPAGE,SYMBLH,INTEO,270.,ICODE)
     IF(NTICY.GE.0)GO TO 22
801  XPAGE = - ORDINA/2.
802  YPAGE = - ABSCIS
     CALL PLOT(XPAGE,YPAGE,3)
    DO 18 I = IBOT,ITOP

ORIGINAL PAGE IS
OF POOR QUALITY
CALL PLOT(XPAGE,YDIT(I),3 - MOD(I,2))
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

SUBROUTINE AXLAB (INTERNAL)

SUBROUTINE AXLAB(ANGLE,IBCD,NCHAR,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,'S').NE.0) GO TO 14
14 IF(FLD(0,6,5').NE.0) 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

SUBROUTINE DENDEC (INTERNAL)

SUBROUTINE DENDEC(QMAX,DELQ,NDEC)
IF(INT(ABS(QMAX)) GE 10) GO TO 5
IF(AMOD(ABS(QMAX - DELQ),10) GE 01) GO TO 7
NDEC = 1
RETURN
5 NDEC = - 1
RETURN
7 NDEC = 2
RETURN
END
REFERENCES


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
Blg. 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
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
Attn: 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-SS41
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  
Atttn: J. C. O'Hara  
6823 St. Charles Ave  
New Orleans, IA 70118

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

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

United Aircraft Corporation  
Pratt & Whitney Aircraft Division  
Atttn: Thomas C. Mayes  
P. O. Box 2591  
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  
Atttn: Library  
P. O. Box 358  
Sunnyvale, CA 94088

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

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

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

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

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

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

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

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

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