MARSHALL GRANT

IN-20-CR

125782

108 P.

FINAL REPORT

for

SHUTTLE ROCKET BOOSTER COMPUTATIONAL FLUID DYNAMICS
NAG8-629

by

T. J. Chung and O. Y. Park

Department of Mechanical Engineering
The University of Alabama in Huntsville
Huntsville, AL 35899

prepared for

April 1988

ABSTRACT

This is the final report for the shuttle rocket
booster computational fluid dynamics. The basic
formulations and preliminary results were submitted in
November, 1987. Additional results and a revised and
improved computer program listing are reported herein.
Numerical calculations for the flame zone of solid
propellants are carried out using the Galerkin finite
elements, with perturbations expanded to the zeroth, first,
and second orders. The results indicate that amplification
of oscillatory motions does indeed prevail in high
frequency regions. For the second order system, the trend
is similar to the first order system for low frequencies,
but instabilities may appear at frequencies lower than
those of the first order system. The most significant
effect of the second order system is that the admittance is
extremely oscillatory between moderately high frequency
ranges.

CHAPTER I

INTRODUCTION

Determination of oscillatory pressure and velocity variations from their mean values in solid propellant combustion chambers has been the subject of study for many years. Because of computational difficulties, however, rigorous and accurate methods of solution still remain in various stages of development. Often details of physics are obscured in complicated computational strategies. In view of the fact that reliable experimental measurements are difficult to obtain, it is crucial to possess an analytical tool to accurately predict combustion dynamics and combustion efficiency in designing successful solid propellant rocket motors.

Earlier works on combustion dynamics include Hart and McClure [1], Denison and Baum [2], Culick [3] and T'ien [4] among others. These studies are centered around one-dimensional, linear oscillatory burning. Culick [5,6], Baum and Levine [7], and Yang et al. [8] subsequently studied the nonlinear growth and limiting amplitude of acoustic waves in a combustion chamber. Nonlinear behavior as characterized by higher order perturbation expansions of governing equations has been investigated by Flandro [9] and Chung and Kim [10] in recent years. They assumed that the pressure varies sinusoidally with time, the pressure

amplitudes being a function of position.

In the present approach, the pressure is assumed to vary in the unstable state. This allows the presence of a long flame such as in double-base solid propellants, in which high pressures and high frequencies may be accommodated. Galerkin finite elements [11] are utilized to model numerically the geometries of burning surfaces and the flame thickness. The nonsteady governing equations are linearized by means of the zeroth, first, and second order perturbation expansions. The boundary conditions, including the burning surfaces and flame edges, are imposed by means of the Lagrange multipliers.

In the simple example problems demonstrated in this paper, the gaseous flame is assumed to follow the Arrhenius law with one-step forward chemical reactions. No condensed phase reaction is included in the formulation. The calculated results confirm the prediction reported by other investigators in the literature for the low frequency region. Extended studies are then carried out for the high frequency region. It has been found that amplification of oscillatory motions does prevail in the high frequency region.

The present study does not include turbulent boundary layers which may play an important role in erosive burning and combustion instability; this is the subject of future study.

CHAPTER II

ANALYSIS AND DISCUSSION OF RESULTS

To demonstrate the validity of the theory presented above, a two-dimensional rectangular geometry, shown in Fig. 1, is analyzed, in which a burning surface and flame edge can be established as boundaries. This configuration was chosen to resemble a one-dimensional behavior so that the results may be compared directly with those reported in the literature [4,9]. The constants and parameters used in this analysis are shown in Table 1.

The computations begin with finding the natural frequencies of the system. The resonance frequencies of the system can be obtained by examining the acoustic admittance or response function at each natural frequency. As mentioned earlier, the natural frequencies of the system are given by the homogeneous solution of the total matrix equation, neglecting the boundary conditions. The frequency range of interest in this computation is shown to be between $\omega = 10^{-3}$ and $\omega = 10^{2}$, corresponding to approximately 80 Hz and 8 MHz, respectively. Thus, sixteen different frequencies in this range are chosen to evaluate the present study, and these are $\omega = 10^{-4}$, 5×10^{-4}, 10^{-3}, 5×10^{-3}, 10^{-2}, 5×10^{-2}, 10^{-1}, 5×10^{-1}, 1, 5, 10, 20, 30, 40, 50, and 100.

Figure 2 shows a typical steady-state distribution of

the field variables and the reaction rate along the flame thickness. In general, these data may be obtained by solving the energy equation together with the reaction rate. Ideally, a fourth order Runge-Kutta scheme may be used for this purpose. The result will be used as the basis of further calculations for combustion instability in the unsteady-state.

Distributions of the field variables versus frequency for the first order system are shown in Figs. 3-8. These results are obtained by imposing an acoustic pressure amplitude of unity at the flame edge as the Dirichlet condition. Conventionally, the thickness of the burning zone has been assumed to be negligibly small compared with the wavelength of the acoustic oscillation in the intermediate frequency range. Thus, the uniform pressure is approximated throughout the domain of study and is assumed to vary with time only. On the contrary, since the oscillatory pressure is regarded as a spatially nonhomogeneous time-dependent source in the present study, it is possible to investigate the response of a specific solid propellant at significantly high frequencies and to find the response even in the long flame of a double-base propellant. Figure 3 shows variations of the pressure distribution versus the acoustic frequencies. It is seen that the amplitude remains constant for $\omega < 10$. Random variations of these amplitudes occur for higher frequencies. Here, the imaginary parts representing the

phase shift are zero.

Figure 4 demonstrates distributions of various
component fluctuations of the first order system at $\omega = 1$.
A pressure wave with a certain amplitude striking the solid
surface will give rise to a response in the burning rate.
Generally, the response of the burning rate is nonlinear
and has a complete Fourier series form that may not be
expressed by a single frequency component. In Fig. 4,
however, the pressure is uniform in the flame zone,
although significant variations may occur at higher
frequencies.

The temperature amplification at the surface changes
the burning rate, while the velocity at the flame edge
represents the acoustic admittance for pressure coupling,
whose magnitude and sign indicate the instability of the
system. Since the pressure remains constant, implying that
the acoustic wavelength is larger than the flame thickness
in this case, the imaginary part of the velocity approaches
a constant slope at the edge by the assumption of an
isentropic condition at the flame edge. It is interesting
to note that the trend of variations of temperature and
velocity appears to be similar, which is inversely
proportional to the variations of species and density as
expected. These results are comparable to those of T'ien
[4] and Flandro [9].

Density distributions for various frequencies are
given in Fig. 5. At $\omega < 0.5$, changes of the distributions

versus frequency are negligible; however, near $\omega = 1$ a
significant reduction of the amplitude occurs, and then for
$\omega > 1$ the amplitudes increase moderately. The
distributions in the lower frequency region are very
similar to those in the steady-state, from which the quasi-
steady assumption may be deduced. Note that the reverse
peak moves in the flame zone, which implies a change of the
reaction zone. Close to the surface, the effect of preheat
in the solid phase increases, resulting in a higher burning
rate. Positive amplitudes at the interface in each
frequency are caused by the rise in pressure and/or the
increase of the burning rate.

It is interesting to see that the amplitudes decrease
while the frequency increases up to the unity and then
increase along the frequency, from which the flame is
expected to react sensitively at special frequencies. The
negative amplitude is shown most notably at $\omega = 1$. This
may result from the net blowing effect, and the velocity at
that point should be positive (see Fig. 8).

The normal velocity distributions are shown in Fig. 8.
The profile may be classified into positive and negative
slopes, the only exception being at $\omega = 1$. The latter
contains most of the distributions in the lower frequency
region. Negative amplitudes appear mostly at the
interface, relating to an increase of the density.
Although the pressure remains constant in lower
frequencies, the velocity varies severely. At $\omega > 20$, the

variation is more significant since from that frequency the
pressure varies in the flame zone. At ω = 0.01, note that
the slope is positive even though it is a lower frequency.
This result implies instability of the system for the
quasi-steady case. At ω = 1, a positive amplitude near the
interface results from adjusting between the density
increase and the higher gasification rate. As indicated in
Fig. 4, the slopes of the imaginary parts near the flame
edge are constant.

Rearranging the real part of the velocity at the flame
edge gives the distribution of the acoustic admittance,
whose magnitude and sign indicate the amplifications or
damping ability of the flame subject to the acoustic
disturbance. Figure 9 shows the curve-smoothed acoustic
admittance and burning rate at each frequency.

At ω = 1, there is a definite increase in the burning
rate. The burning rate increases by increasing the heat
transfer effect due to closer movement of the reaction zone
to the surface. There are actually several secondary
effects such as structural change in the solid phase and
change of the chemical kinetics in the flame. These
effects cause the feedback to the solid phase, resulting in
a change of the burning rate. These effects, however, are
not considered.

Figure 9 also reveals a résonance in the condensed
phase near ω = 0.01, indicating that the system is
unstable. This verifies the result of Denison and Baum [2]

and previous laboratory measurements which have shown that the response consists generally of a single peak in the range of frequency for which the quasi-steady approximation appears to be accurate. Some negative peaks exist at the other frequencies, implying that the resonance in the gas phase tends to damp the oscillatory motion. At most higher frequencies, the system seems to be unstable.

The real part of the burning rate shows a trend similar to the acoustic admittance at the quasi-steady region, although the magnitude is significantly different. But these trends differ from each other at the higher region. Over $\omega = 100$, the profile tends to have a second mode oscillation as the pressure varies in the flame zone. It must be emphasized that the admittance function alone is not sufficient to describe the stability of the system unless the velocity coupling is concerned.

Temperature distributions are given in Fig. 6. Since the temperature is related to the density, it can be analyzed easily by comparing the results with those in Fig. 5. Accordingly, the distribution of the temperature shows the profile opposed to that of the density. Temperature rises are predominant along the reaction region for $\omega > 1$. Note that no significant temperature changes occur at the interface as imposed by the boundary conditions. Negative amplitude near the flame edge may be caused by the stagnation phenomena in which the density increases. Because of the pressure unity, the imaginary parts of the

temperature are given as reciprocals of those of the density.

The fuel consumption at a given frequency is shown in Fig. 7. Trends for species distributions are opposite to those for temperature distributions in that negative maximum occurs along the reaction region. Note that linearly diminishing the fuel near the flame edge affects the temperature changes slowly (Fig. 6).

The results of the second order perturbation are shown in Figs. 10-28. As previously indicated, each variable of the second order response to acoustic disturbance has two components: one time-dependent component that oscillates at twice the fundamental frequency and one that is time-independent. The latter represents a shift in the mean value, thus causing a shift of the mean burning rate. It should be noted that the variations for the second order system may be characterized by the sum of those two parts, with the factor $e^{i2\omega t}$ ranging between -1 and 1 and zero indicating the case of time-independence.

Figures 10-15 show the distributions in the second order time-independent case. The calculations are also conducted by imposing the pressure of unity at the flame edge as the Dirichlet condition. General trends show that shifts in the mean values are evident.

In Fig. 10, the pressure varies from $\omega = 5$, which is half of that in the first order system. For the higher frequency region, the oscillatory motion is significant,

which may affect the velocity distribution. Figure 11
shows all of the variables for $\omega = 1$, indicating that
shifts in mean values may have the affect of damping;
however, the opposite may be true for higher frequencies as
demonstrated in Figs. 12-15. The trends are roughly
similar to those of the first order illustrated in Fig. 4a,
but they have different amplitudes because of the
nonlinearities in the higher order system. In Figs. 12-15,
the distributions of field variables in lower frequencies
appear to vary negligibly along the flame thickness, while
the opposite is true for $\omega > 1$. In particular, since the
pressure varies at $\omega > 5$, the velocity changes
significantly at those frequencies, implying that the
system is unstable at higher frequencies. As mentioned
earlier, a shift of the mean burning rate is the most
significant effect in the time-independent system. This
result will be discussed in Fig. 28.

The distributions in the second order time-dependent
case are shown in Figs. 16-21. Here, the pressure also
varies from $\omega = 5$ and oscillates for higher frequencies.
Note that the amplitude variations are smaller than those
in the time-independent case. The trends seem to reduce
the total effect of the second order system.

Figures 22-27 show the variations of field variables
in the second order system by summing time-independent and
time-dependent effects, with the factor $e^{i2\omega t} = 1$ for time-
dependent effects. Note that amplitude variations of each

variable in the flame zone are significant at $\omega = 1$, which is the same result obtained for the case of the first order system. Finally, Fig. 28 reveals that the shift in burning rate versus frequency is consistently negative, as asserted by other investigators. The offset is relatively small in the quasi-steady region, but increases with oscillatory motion along the frequencies. The variations of the acoustic admittance are similar to those of the first order system, except that oscillatory deviations are more predominant at higher frequencies in both time-independent and time-dependent cases.

Parameter studies are conducted for the first order and are summarized as follows. Decreasing the density ratio β affects the variables shifted slightly to the negative direction, keeping the distribution profiles constant. Changing the latent heat of solid L exerts a negligible effect on the variables, but a very small value of L shifts the system toward instability. Increasing the surface activation energy E_s or the gas phase activation energy E reduces the magnitude of the variables, keeping the same profiles. Changes of the rate constant and viscosity effect coefficient strongly affect the system, such that every aspect discussed herein will change.

The present study could be extended to the multi-dimensional case by introducing the appropriate axial mean flow field. It is well recognized that fluctuation of the gas velocity parallel to the propellant surface affects the

burning rate in terms of velocity coupling; therefore, this quantity must be considered together with pressure coupling for a satisfactory measure of stability.

A simple calculation has been accomplished using the artificial axial flow velocity. However, difficulties of the boundary conditions could not be eliminated. A test run shows that the existence of a small amount of the axial flow reduces differences between the variable amplitudes in each frequency considered, thus leading the system toward stability.

CHAPTER III

CONCLUSION

A multi-dimensional numerical model for the premixed
flame acoustic instability is proposed and solved using the
finite element method. The governing equations are
perturbed to the second order and formulated with the
Galerkin finite elements. The results have direct bearing
on the validity of published theories of solid propellant
combustion instability at the lower frequency region where
the uniform pressure assumption is valid. Extended studies
are made on the higher frequency region and some new
results are obtained.

Under the restricted boundary conditions, the
following conclusions, based on numerical computations, are
reached:

(1) The stability characteristics for the low frequency
     range in the first order system have been verified to
     be the same as those reported in the literature. For
     example, the acoustic admittance is controlled by the
     burning rate, negative for low frequencies, whereas
     the opposite is true for high frequencies.

(2) Instabilities are likely to occur at high frequencies
     ($\omega > 10$).

(3) For the second order system, the trend is similar to
     the first order system for low frequencies, but

instabilities may appear at frequencies lower than those of the first order system.

(4) The most significant effect of the second order system is that the admittance is extremely oscillatory between $\omega = 1$ and $\omega = 10$.

REFERENCES

1. Hart, R.W. and McClure, F.T., "Combustion Instability: Acoustic Interaction with a Burning Propellant Surface", J. Chem. Phys., Vol. 30, Sept. 1959, pp. 1501-1514.

2. Denison, M.R. and Baum, E., "A Simplified Model of Unstable Burning in Solid Propellants", ARS J., Vol. 31, Aug. 1961, pp. 1112-1122.

3. Culick, F.E.C., "A Review of Calculations for Unstable Burning of a Solid Propellant", AIAA J., Vol. 6, Dec. 1968, pp. 2241-2254.

4. T'ien, J.S., "Oscillatory Burning of Solid Propellants Including Gas Phase Time Lag", Combustion Science and Technology, Vol. 5, 1972, pp. 47-54.

5. Culick, F.E.C., "Nonlinear Behavior of Acoustic Waves in Combustion Chambers-I", Acta Astronautica, Vol. 3, 1976, pp. 715-734.

6. Culick, F.E.C., "Nonlinear Behavior of Acoustic Waves in Combustion Chambers-II", Acta Astronautica, Vol. 3, 1976, pp. 735-757.

7. Baum, J.D., Levine, J.N., and Lovine, R.L., "Pulse-Triggered Instability in Solid Rocket Motors", AIAA J., Vol. 22, No. 10, Oct. 1984, pp. 1413-1419.

8. Yang, V., Kim, S.I., and Culick, F.E.C., "Third-order Nonlinear Acoustic Oscillations in Combustion Chambers, Part I", AIAA paper 87-1873, June 1987.

9. Flandro, G.A., "Nonlinear Time-Dependent Combustion of a Solid Rocket Propellant", 19th JANNAF Combustion Meeting, Oct. 1982.

10. Chung, T.J. and Kim, P.K., "Unsteady Response of Burning Surface in Solid Propellant Combustion", AIAA paper 85-0234, Jan. 1985.

11. Chung, T.J., Finite Element Analysis in Fluid Dynamics, McGraw-Hill Book Co., 1978.

12. Cantrell, R.H., Hart, R.W., and McClure, F.T., "Linear Acoustic Gains and Losses in Solid Propellant Rocket Motors", AIAA J., Vol. 2, No. 6, June 1964, pp. 1100-1105.

13. Brown, R.S., Dunlap, R., Young, S.W., and Waugh, R.C., "Vortex Shedding as a Source of Acoustic Energy in Segmented Rockets", J. of Spacecraft and Rockets, Vol. 18, July-August 1981, pp. 312-319.

14. Flandro, G.A., "Rotating Flows in Acoustically Unstable Rocket Motors", AIAA J., Vol. 2. July 1964, pp. 1305-1306.

15. Hegde, U.G. and Strahle, W.C., "The Generation of Sound by Turbulence Inside Rocket Motor Cavities", AIAA paper 84-0287, 1984.

16. Friedly, J.C. and Petersen, E.E., "Influence of Combustion Parameters on Instability in Solid Propellant Motors, Part I: Development of Model and Linear Analysis", AIAA J., Vol. 4, No. 9, Sept. 1966, pp. 1605-1610.

17. Friedly, J.C. and Petersen, E.E., "Influence of Combustion Parameters on Instability in Solid Propellant Motors, Part II: Nonlinear Analysis", AIAA J., Vol. 4, No. 11, Nov. 1966, pp. 1932-1937.
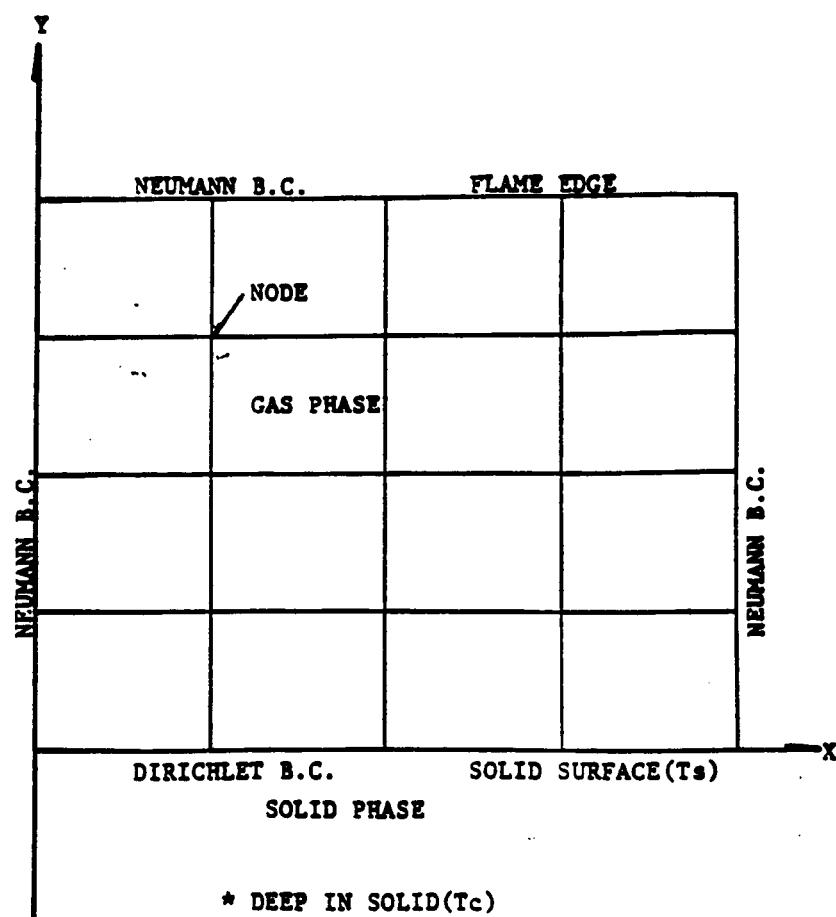
Fig. 1 Finite element geometry.
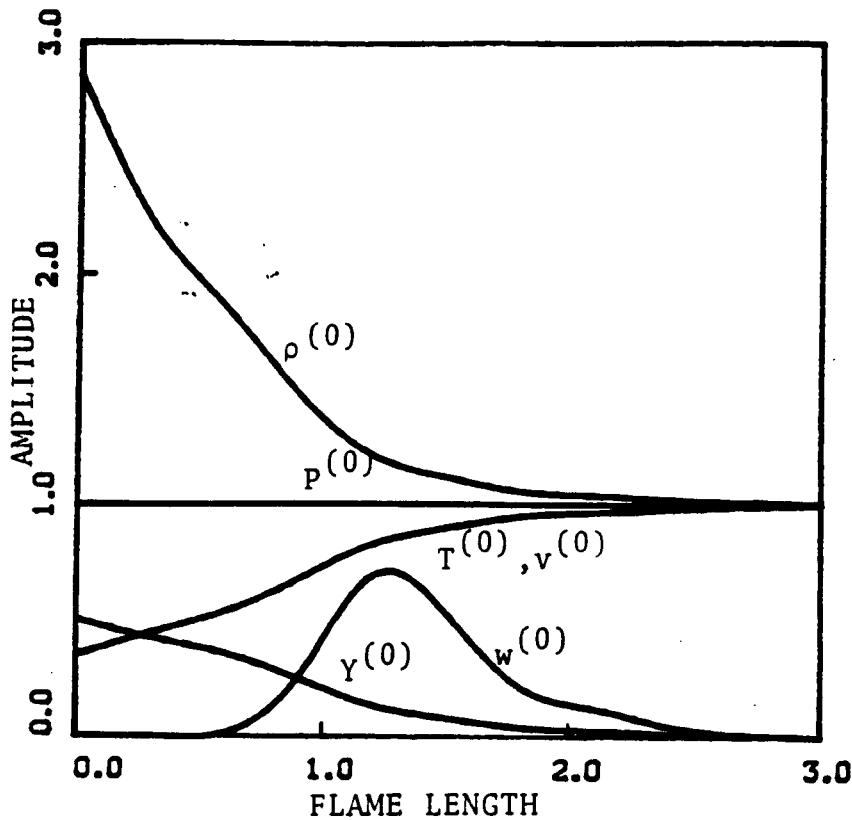
Fig. 2 Steady-state distributions of field
variables in flame zone. Parameters
used in the calculations are given
in Table 1, including Pr=1 and Mb=
0.003. Reference values for non-dim-
ensionalization are chosen from the
flame edge. (ρ :density,T:temperature,
Y:species of fuel, v:velocity, P:pre-
ssure, and w:reaction rate).
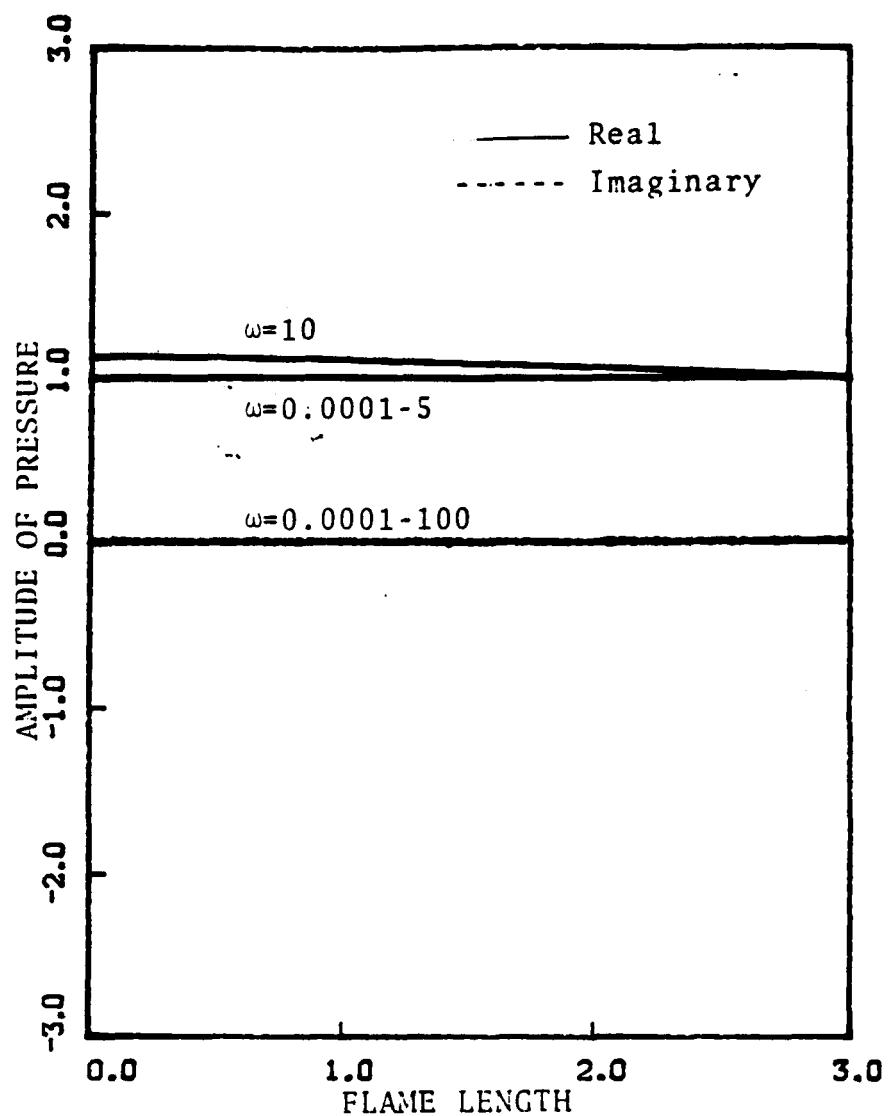
Fig. 3 First order pressure distributions vs. frequency(Dirichlet condition is imposed at the flame edge).
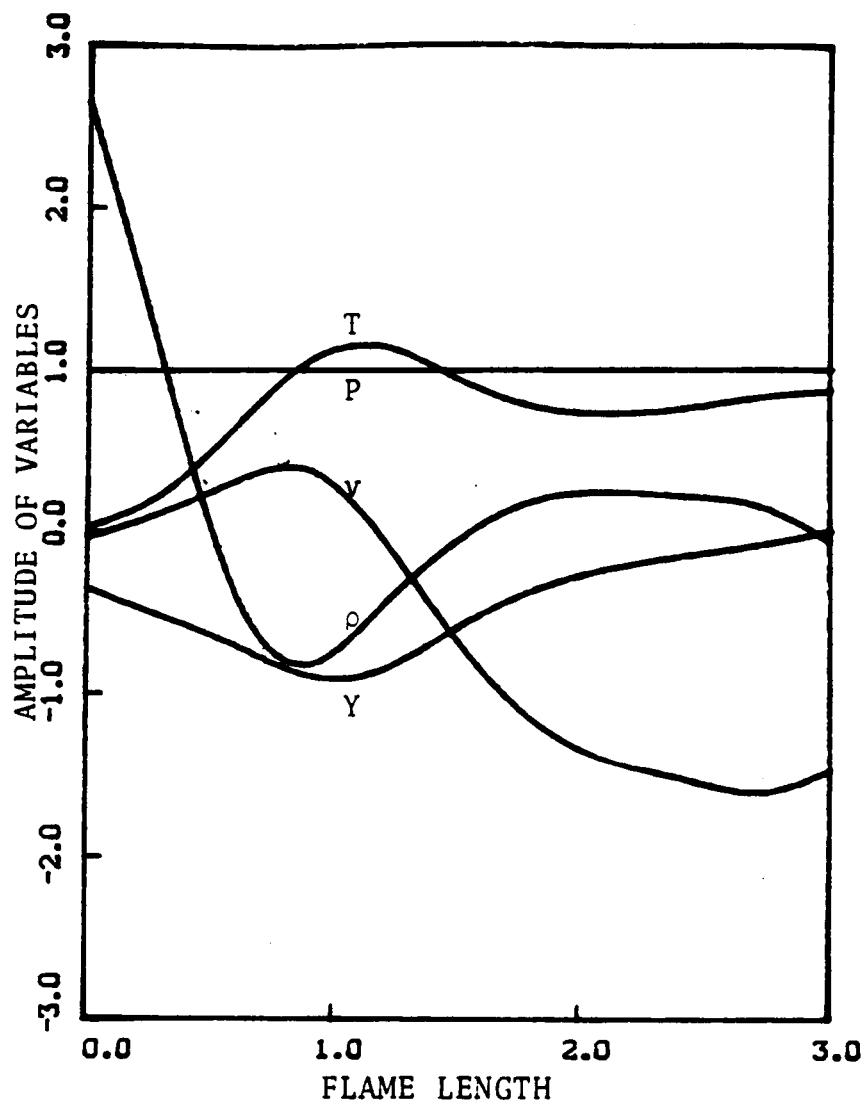
Fig. 4a   First order distributions of field
           variables at ω=1(real parts).

Fig. 4b   First order distributions of field
variables at $\omega = 1$(imaginary parts).
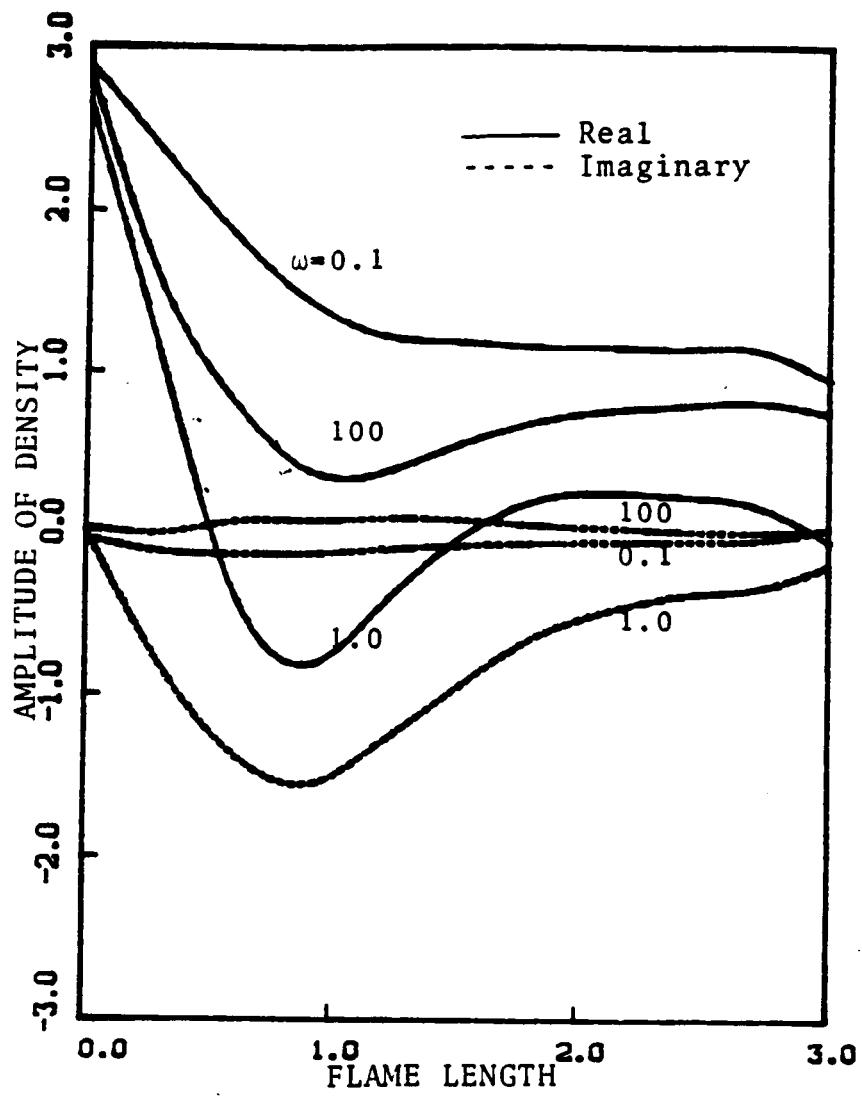
Fig. 5   First order density distributions vs.
          frequency.
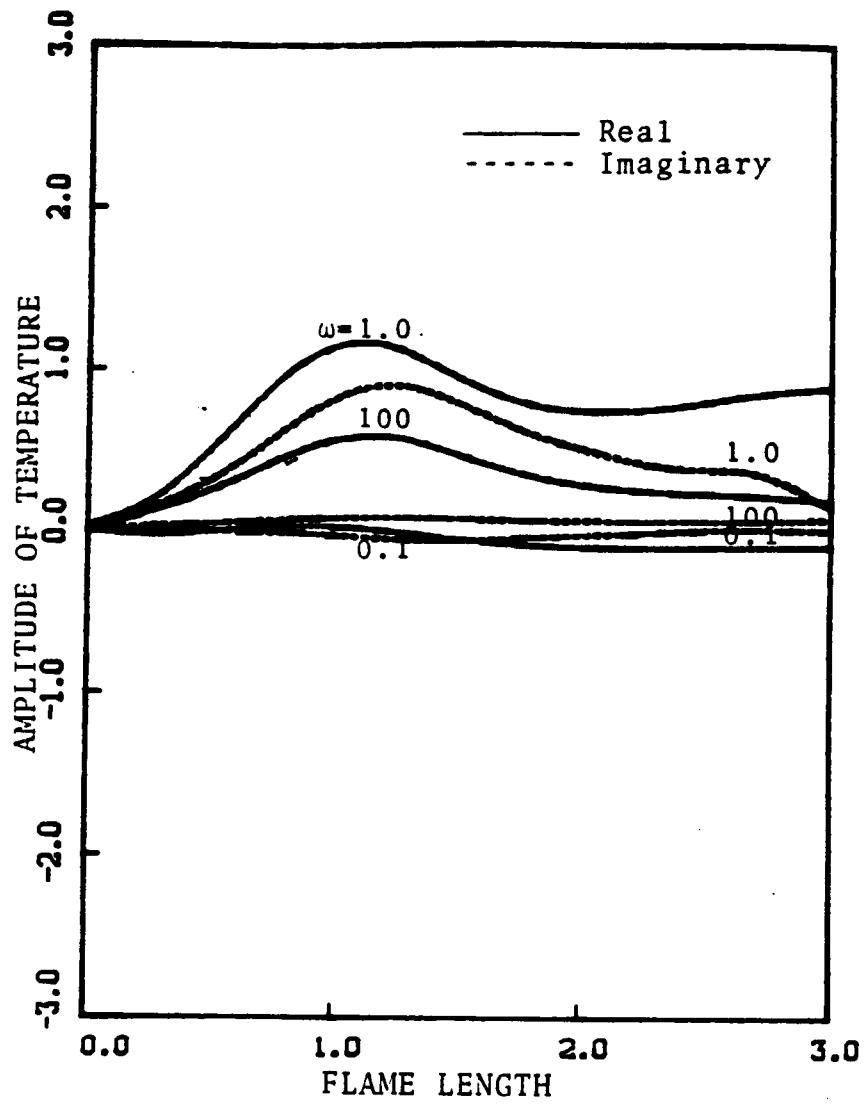
Fig. 6 First order temperature distributions vs. frequency.
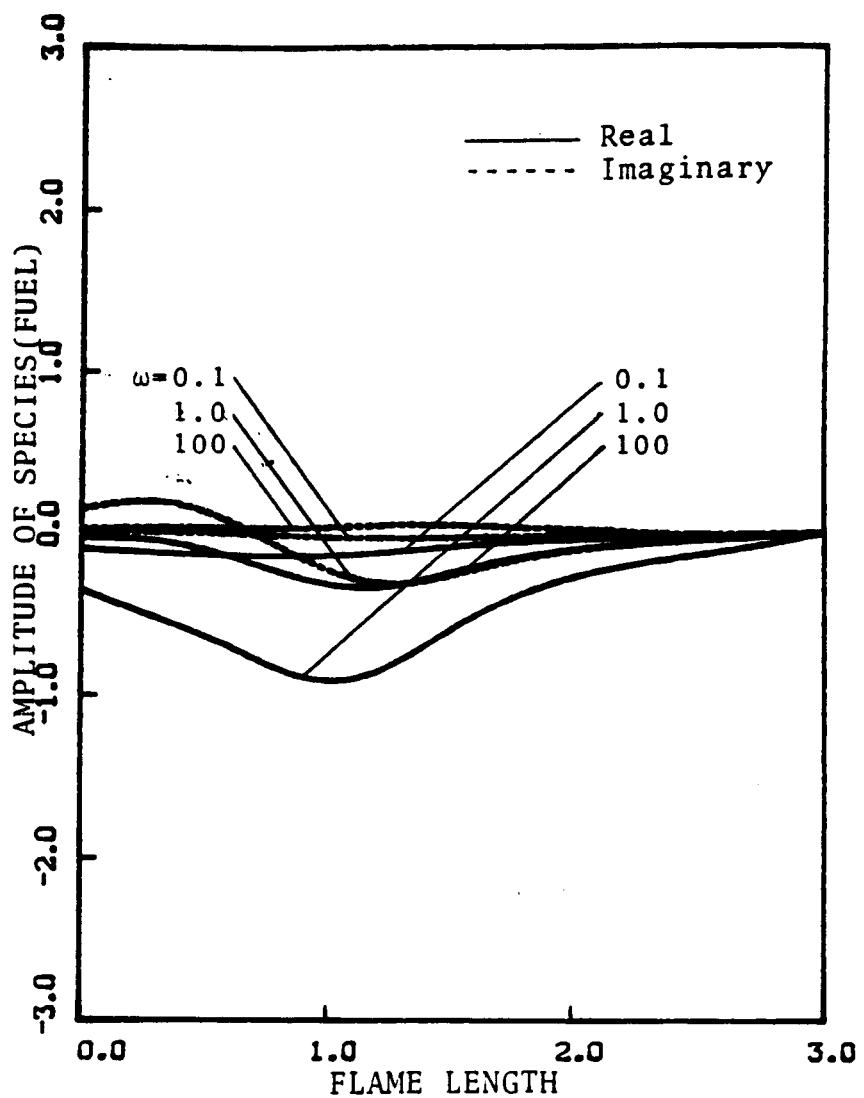
Fig. 7 First order species(fuel) distributions
vs. frequency.
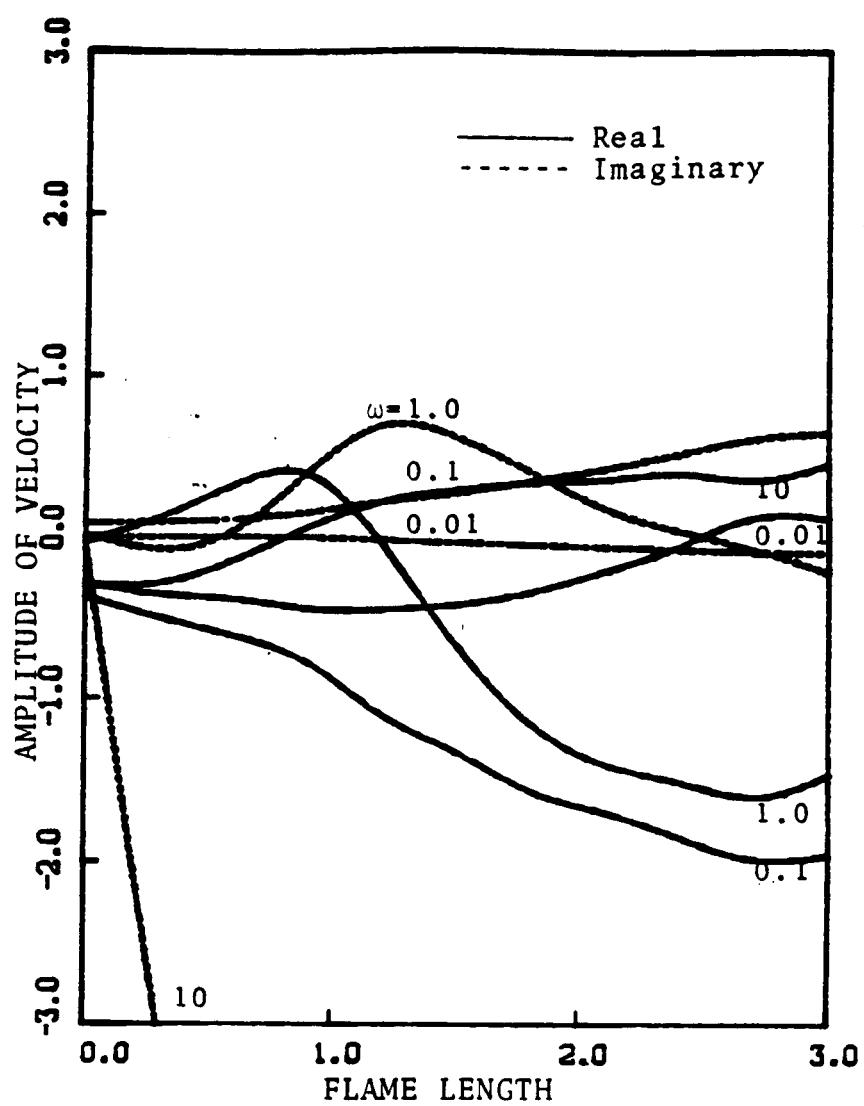
Fig. 8 First order velocity distributions
vs. frequency.
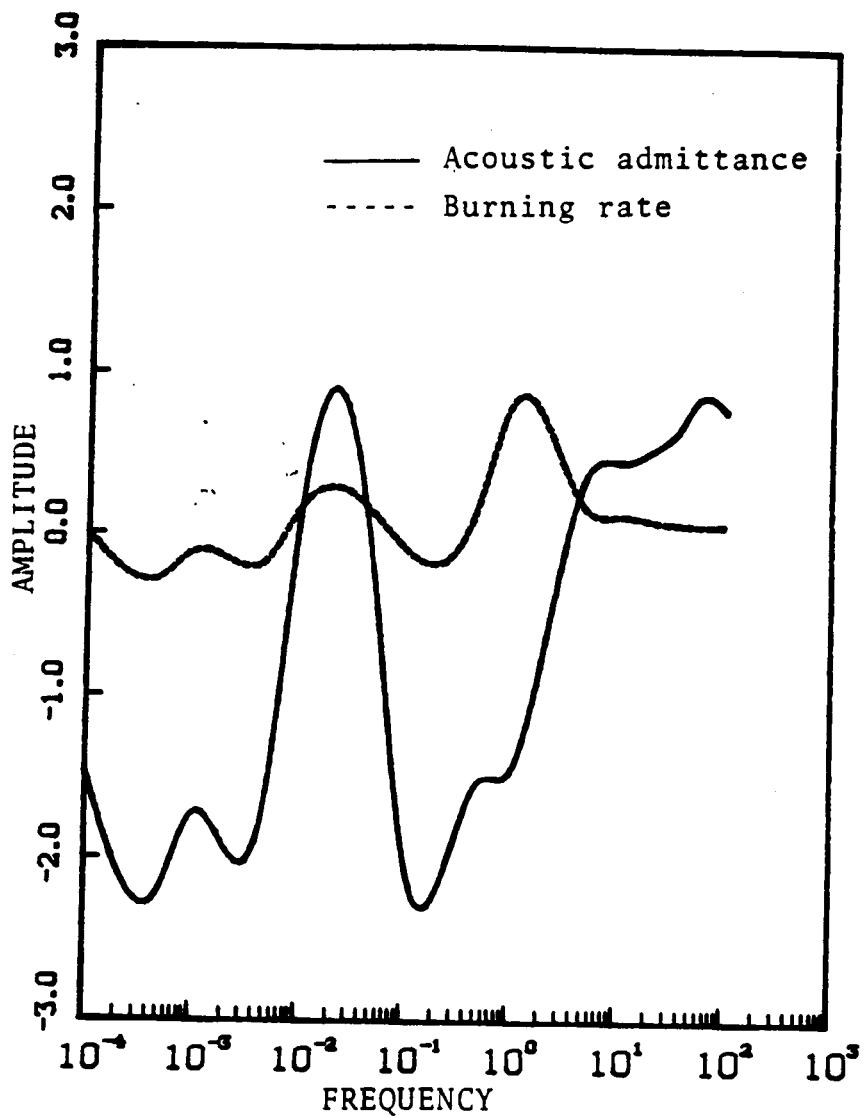
Fig. 9a First order acoustic admittance and burning rate vs. frequency. Positive peak near ω= 0.01 resembles the results of Denison and Baum(2) and T'ien(4).

Fig. 9c Real parts of acoustic admittance and burning rate vs. frequency [T'ien, 1972].

Fig. 9b Steady oscillatory mode: typical dependence of amplitude on frequency of pressure oscillation [Denison and Baum, 1961].
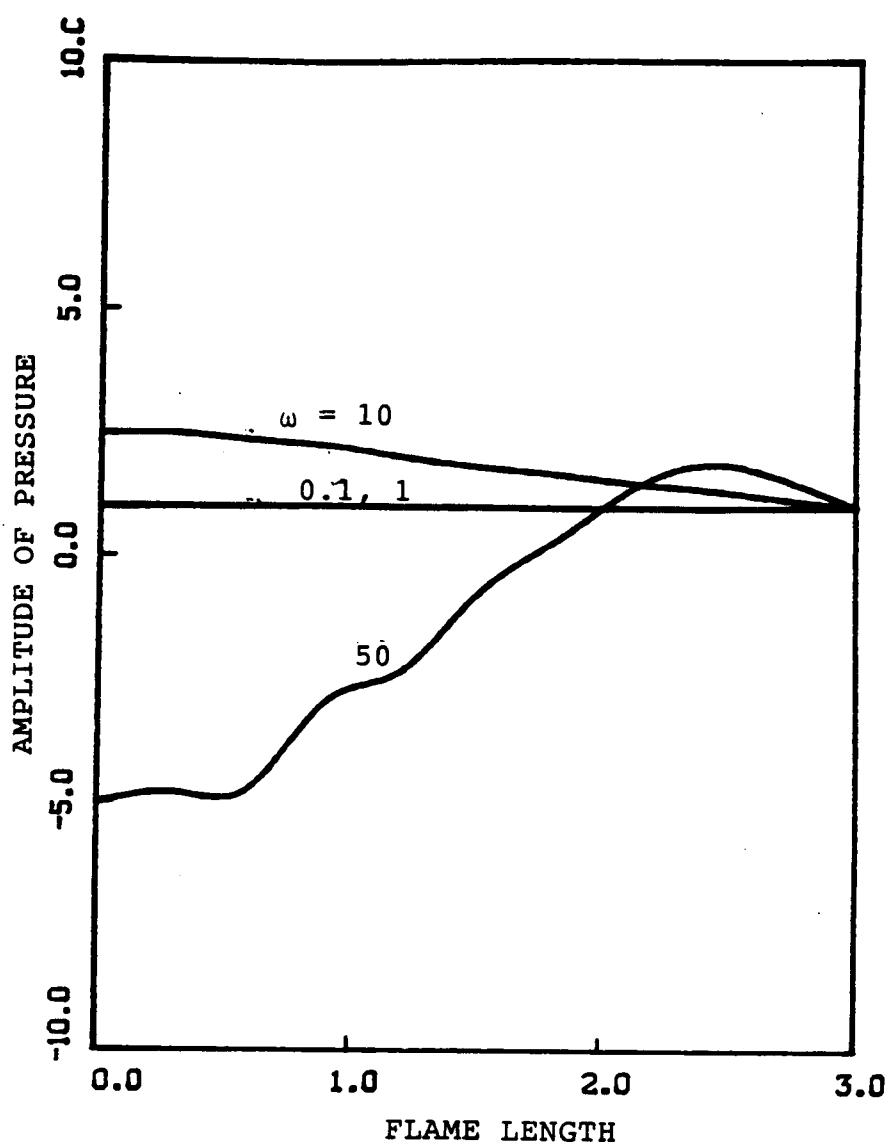
Fig. 10   Second order time-independent pressure
distributions vs. frequency.
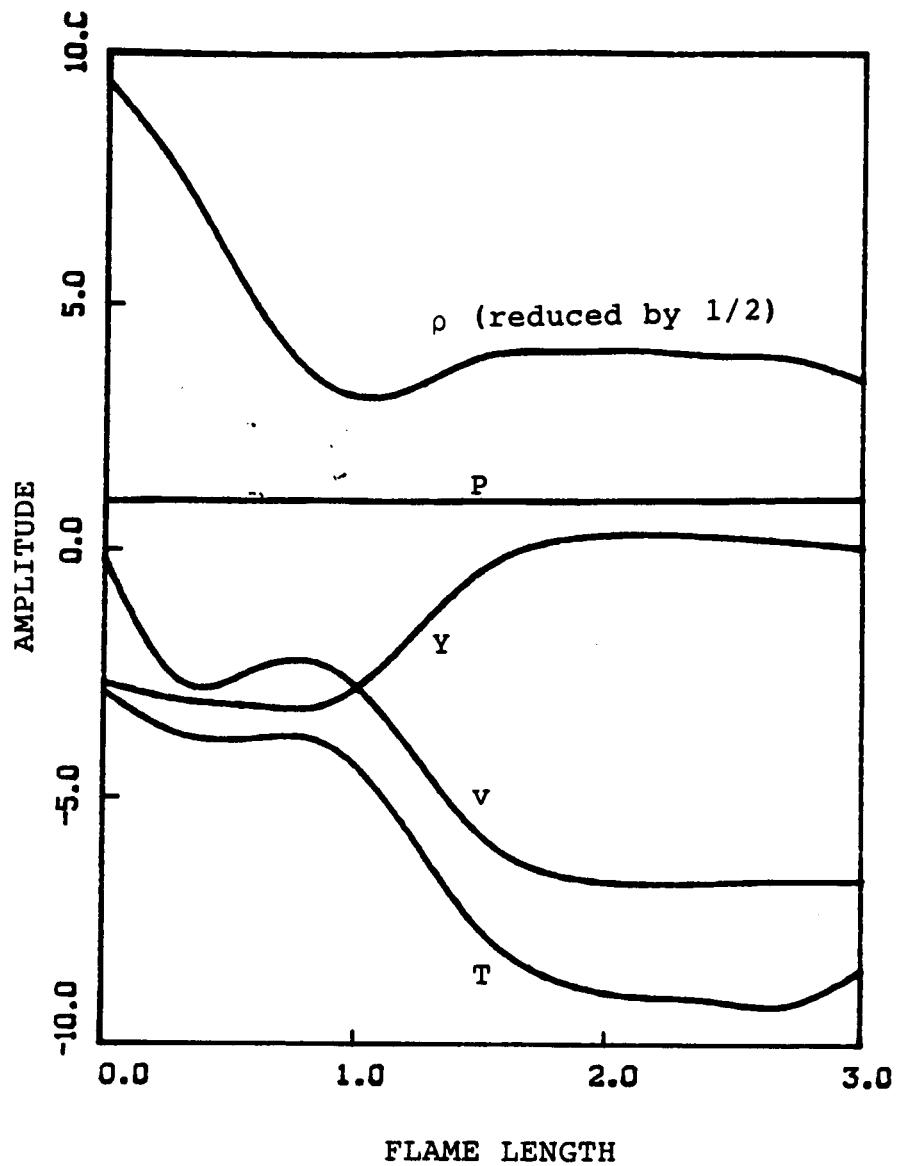
Fig. 11   Second order time-independent distri-
butions of field variables at ω = 1.
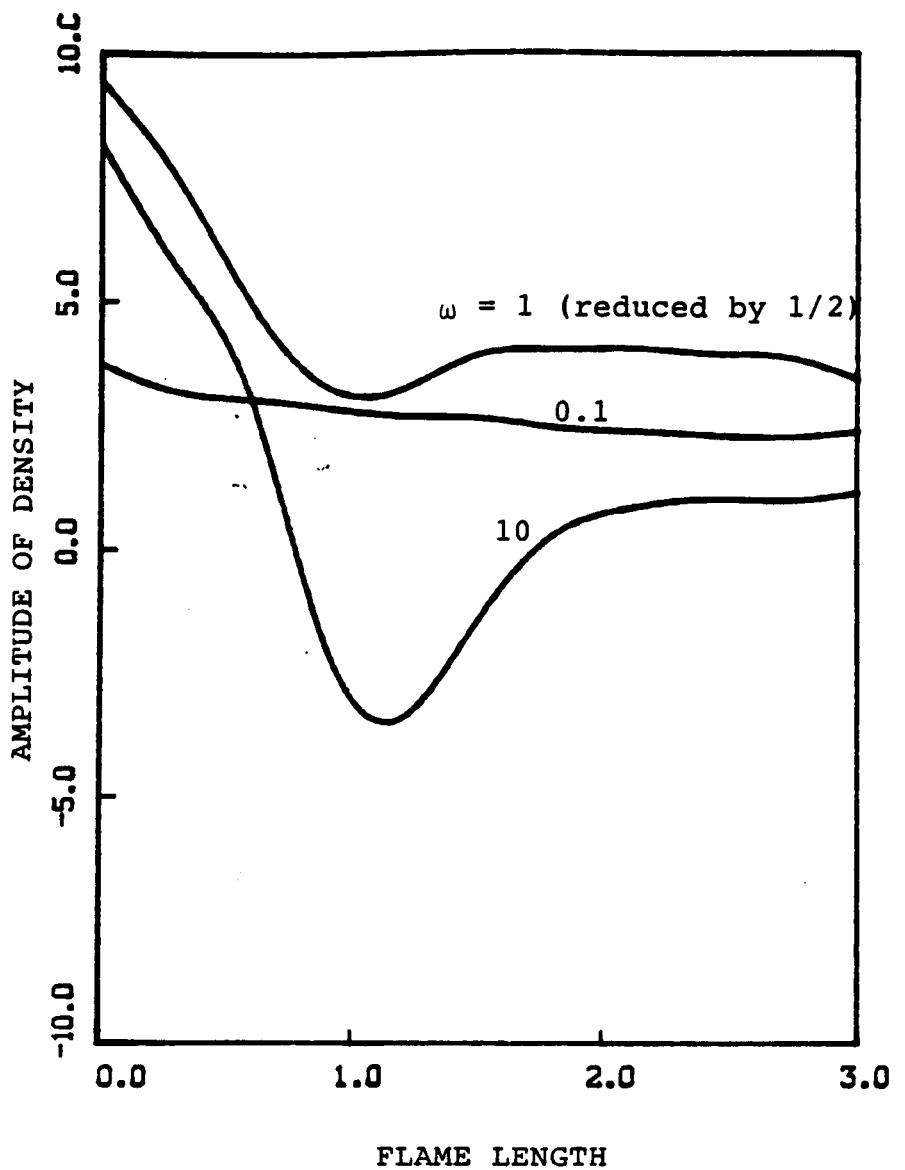
Fig. 12 Second order time-independent density
distributions vs. frequency.
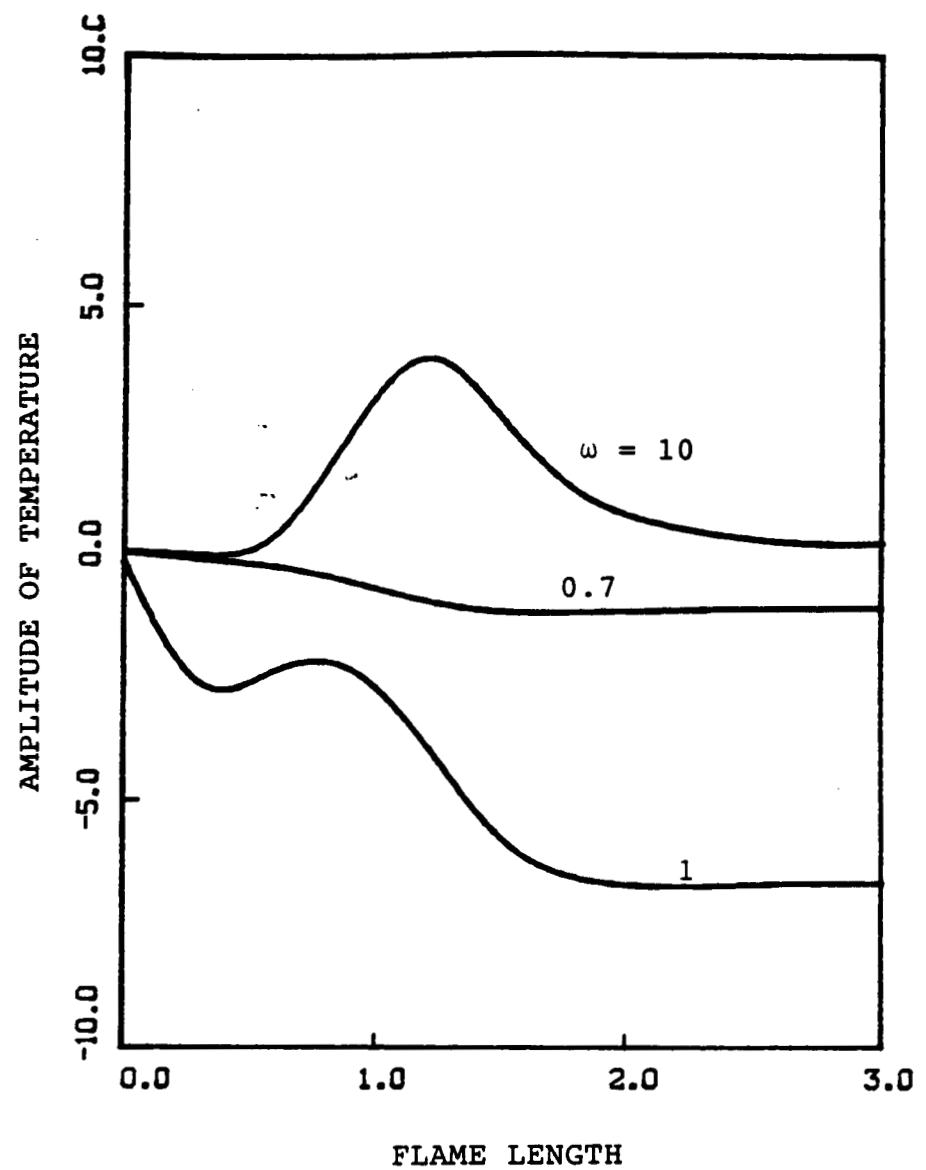
Fig. 13   Second order time-independent temperature
distributions vs. frequency.
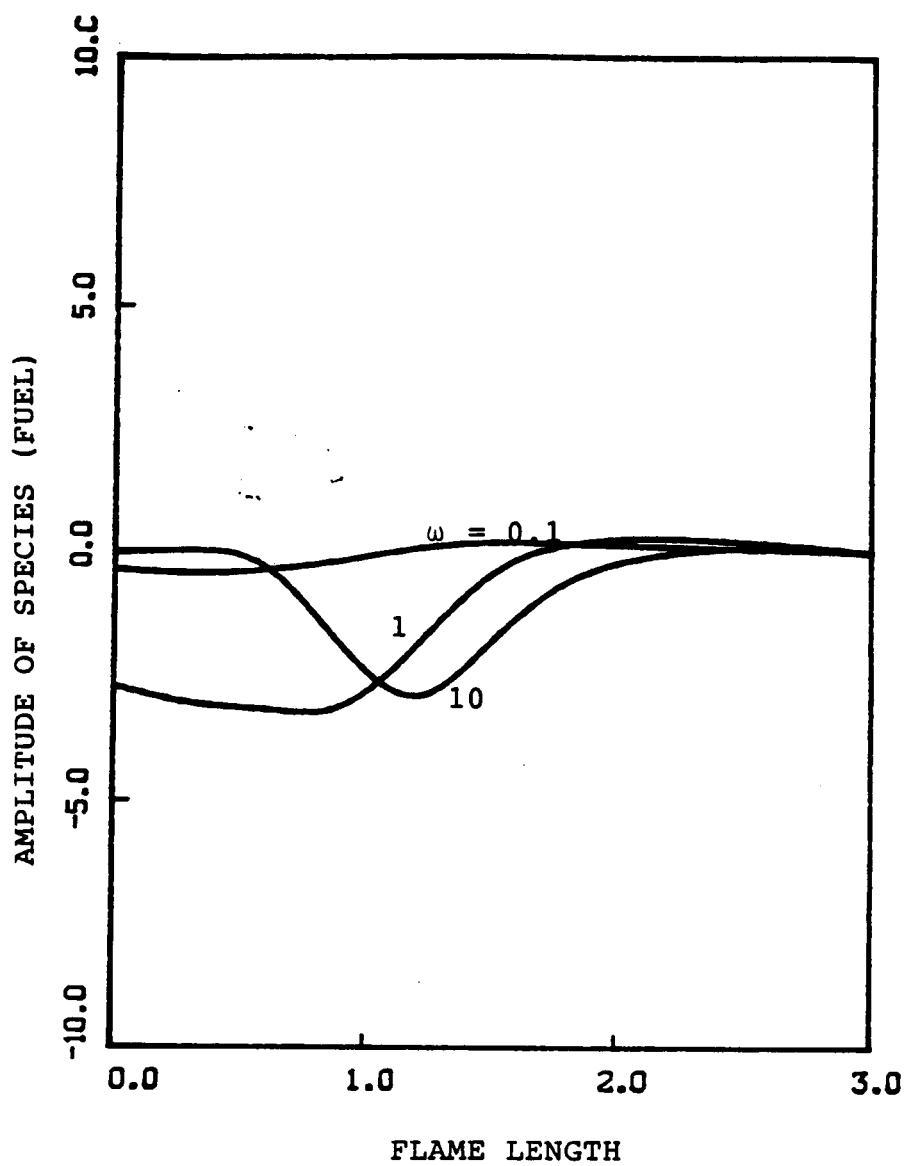
Fig. 14 Second order time-independent species
(fuel) distributions vs. frequency.
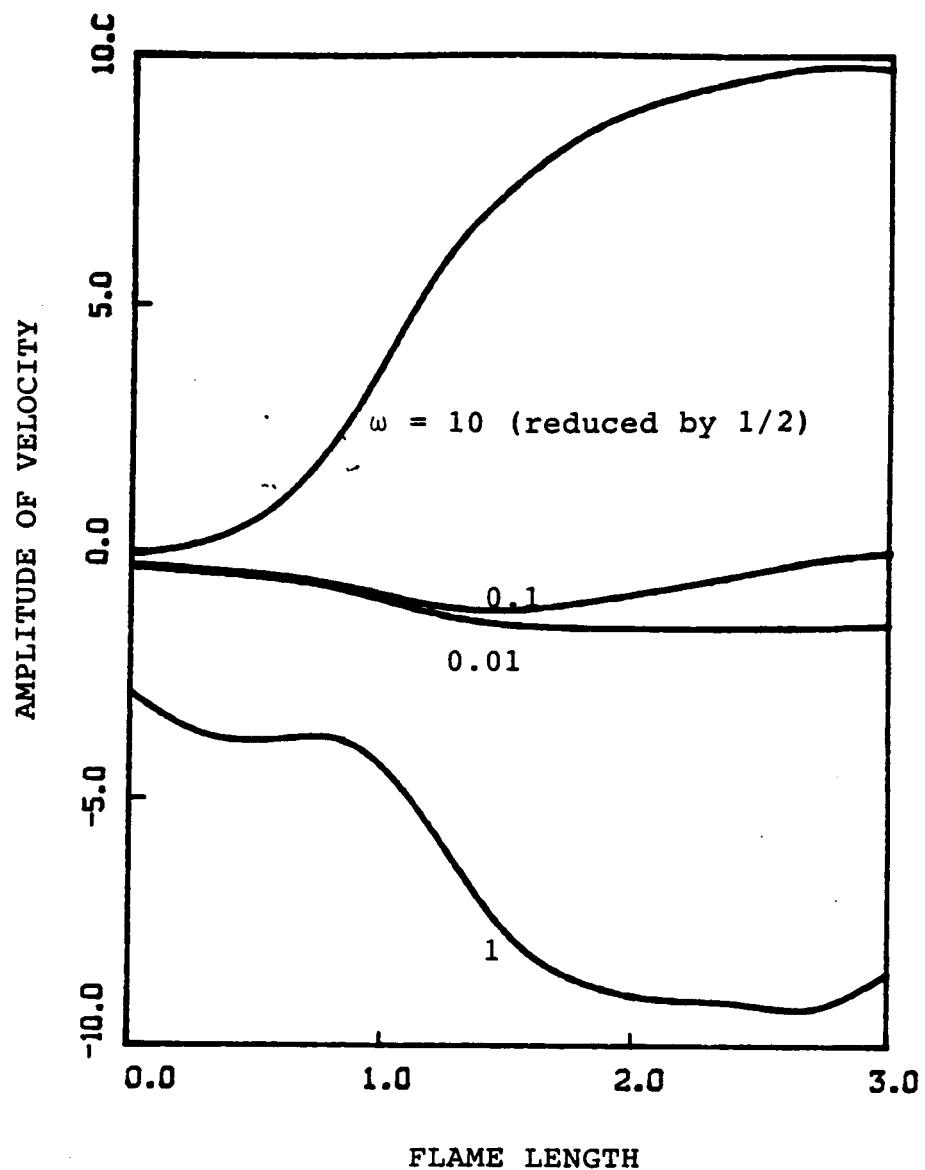
Fig. 15 Second order time-independent velocity
distributions vs. frequency.
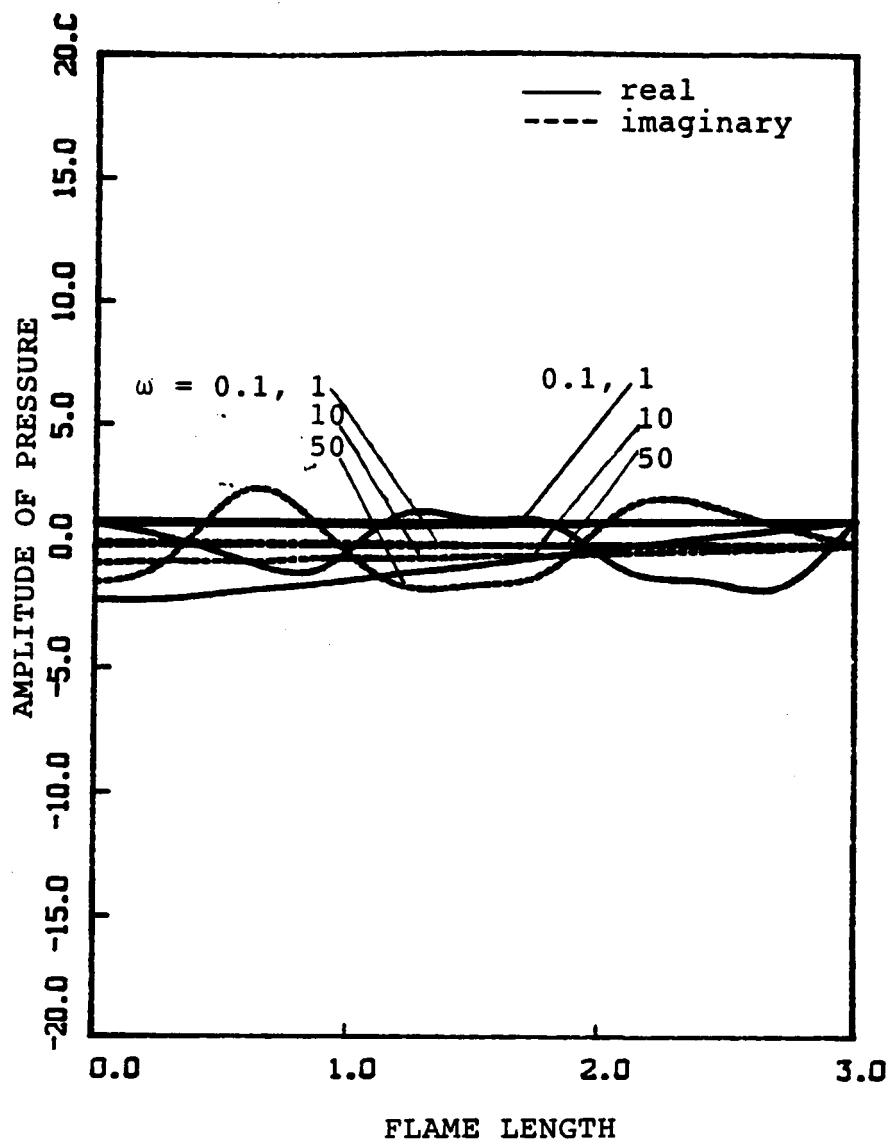
Fig. 16   Second order time-dependent pressure
distributions vs. frequency.
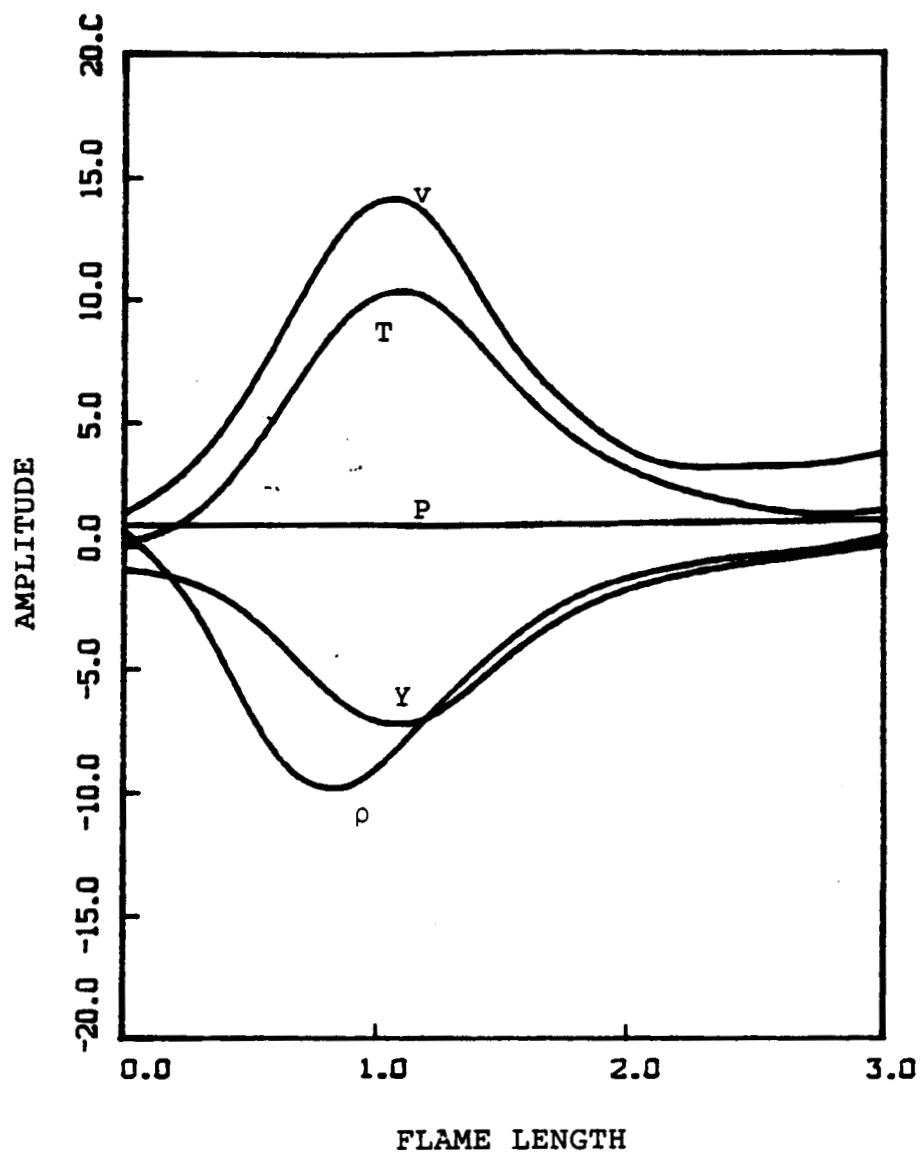
Fig. 17a   Second order time-dependent distri-
butions of field variables at $\omega = 1$
(real parts).
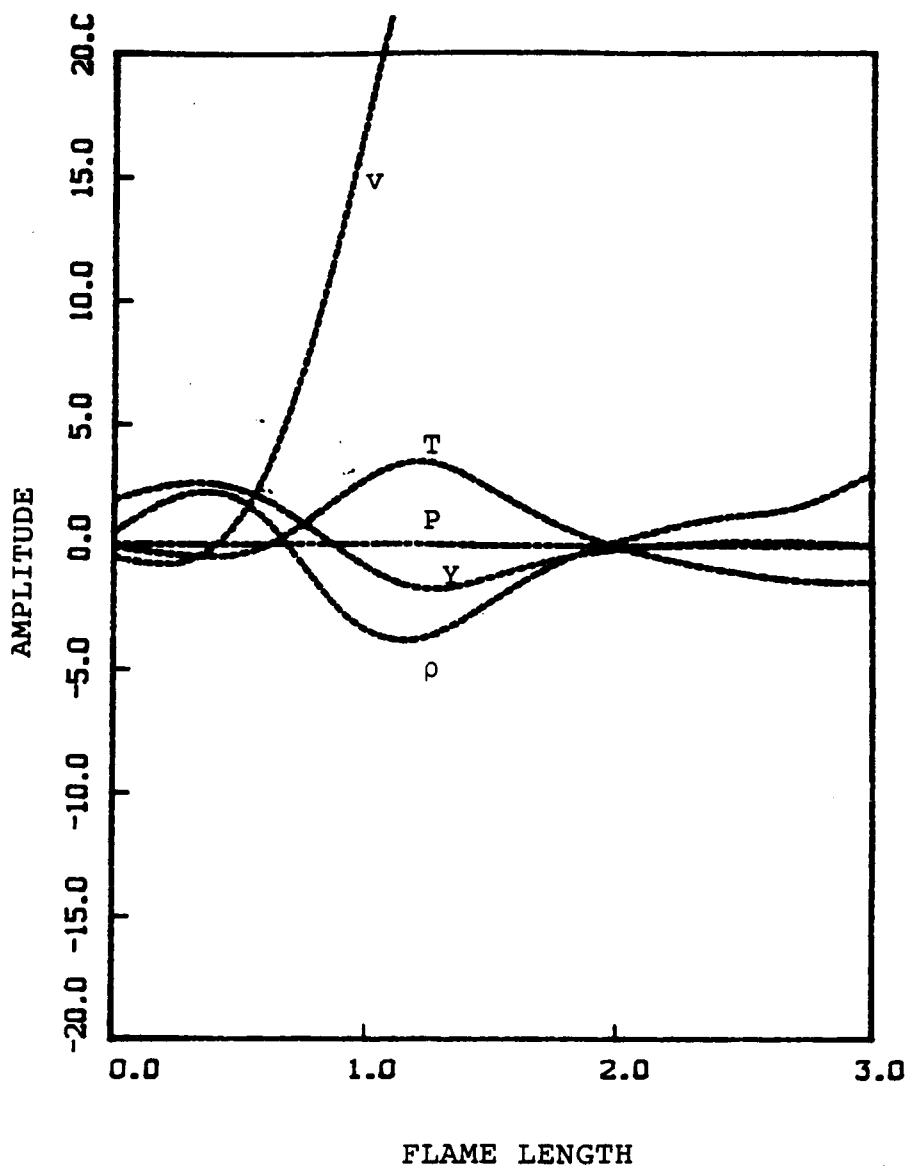
Fig. 17b   Second order time-dependent distri-
           butions of field variables at $\omega = 1$
           (imaginary parts).
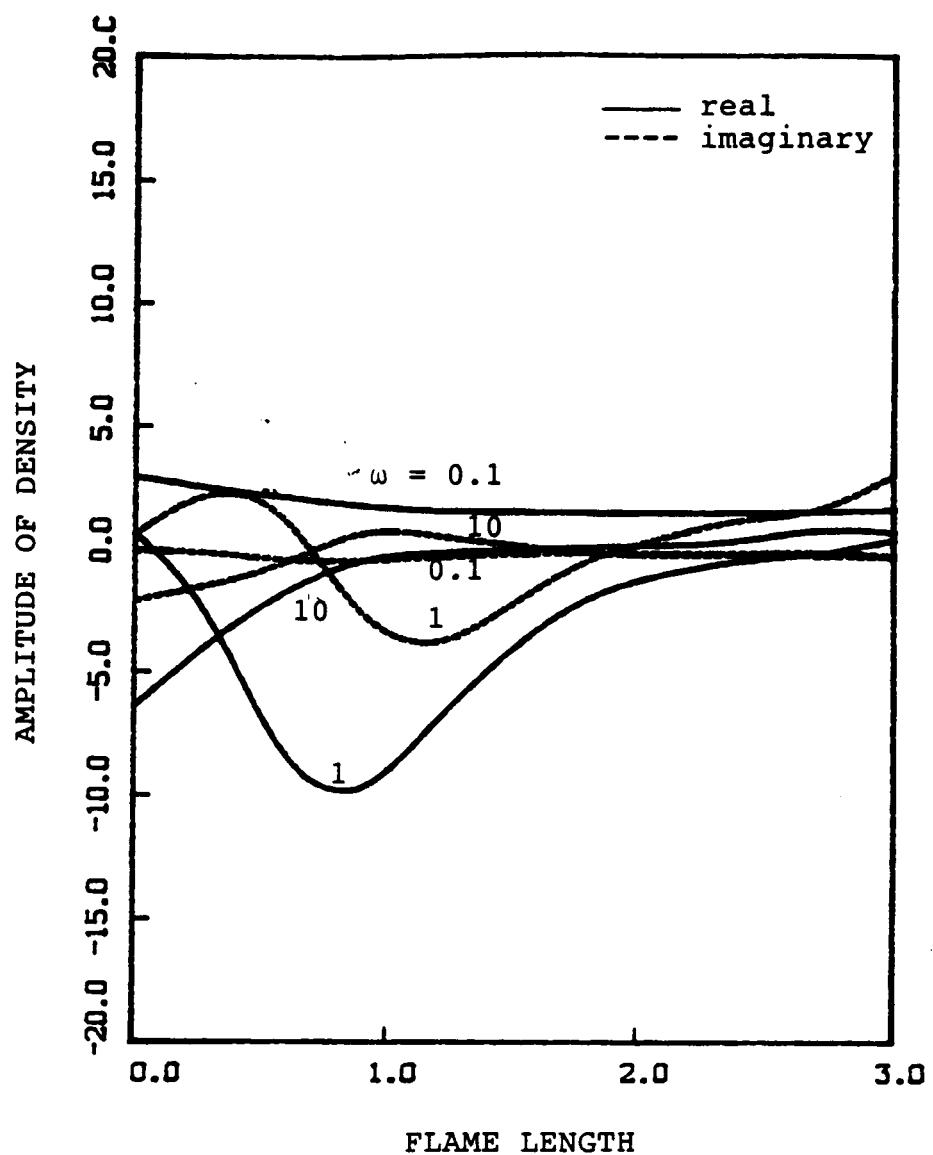
Fig. 18   Second order time-dependent density
distributions vs. frequency.
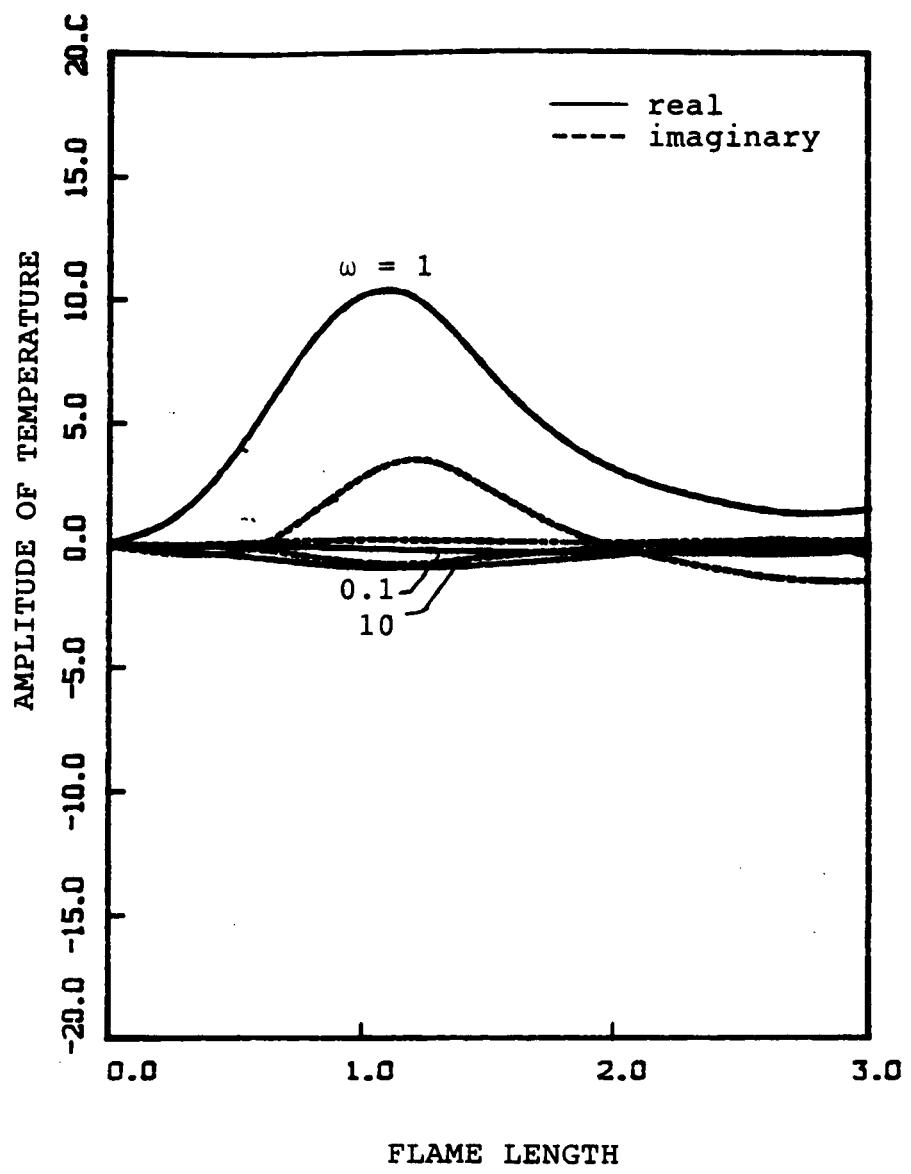
Fig. 19 Second order time-dependent temperature distributions vs. frequency.
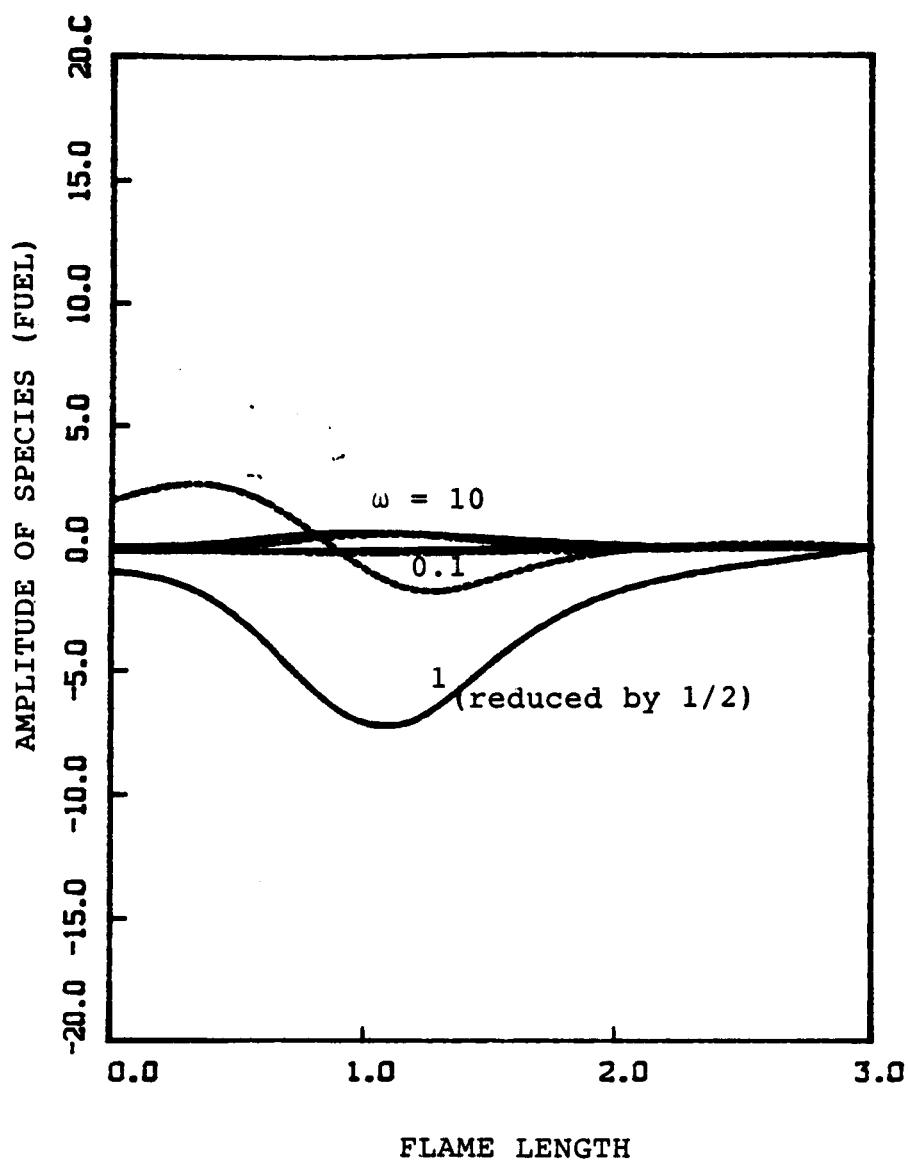
Fig. 20   Second order time-dependent species
            (fuel) distributions vs. frequency.
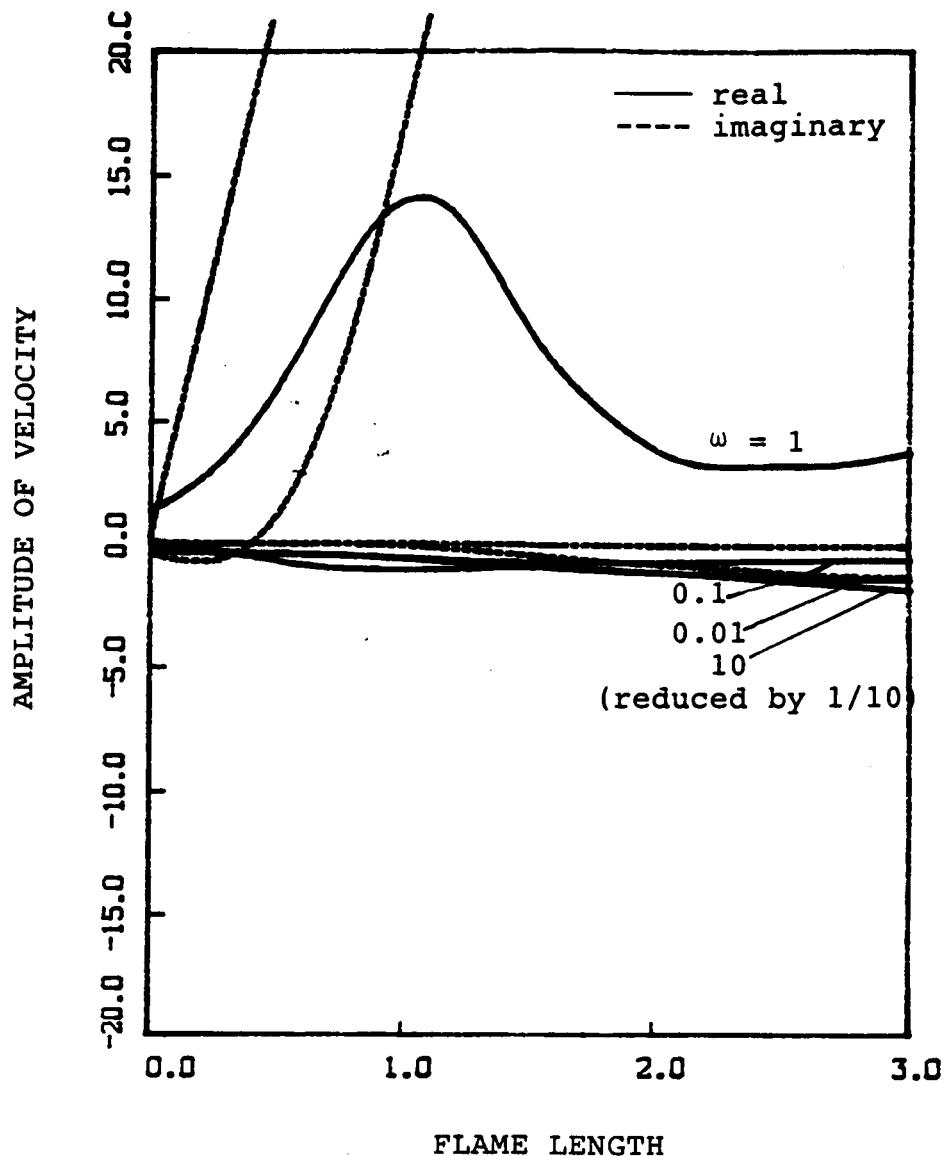
Fig. 21 Second order time-dependent velocity
distributions vs. frequency.
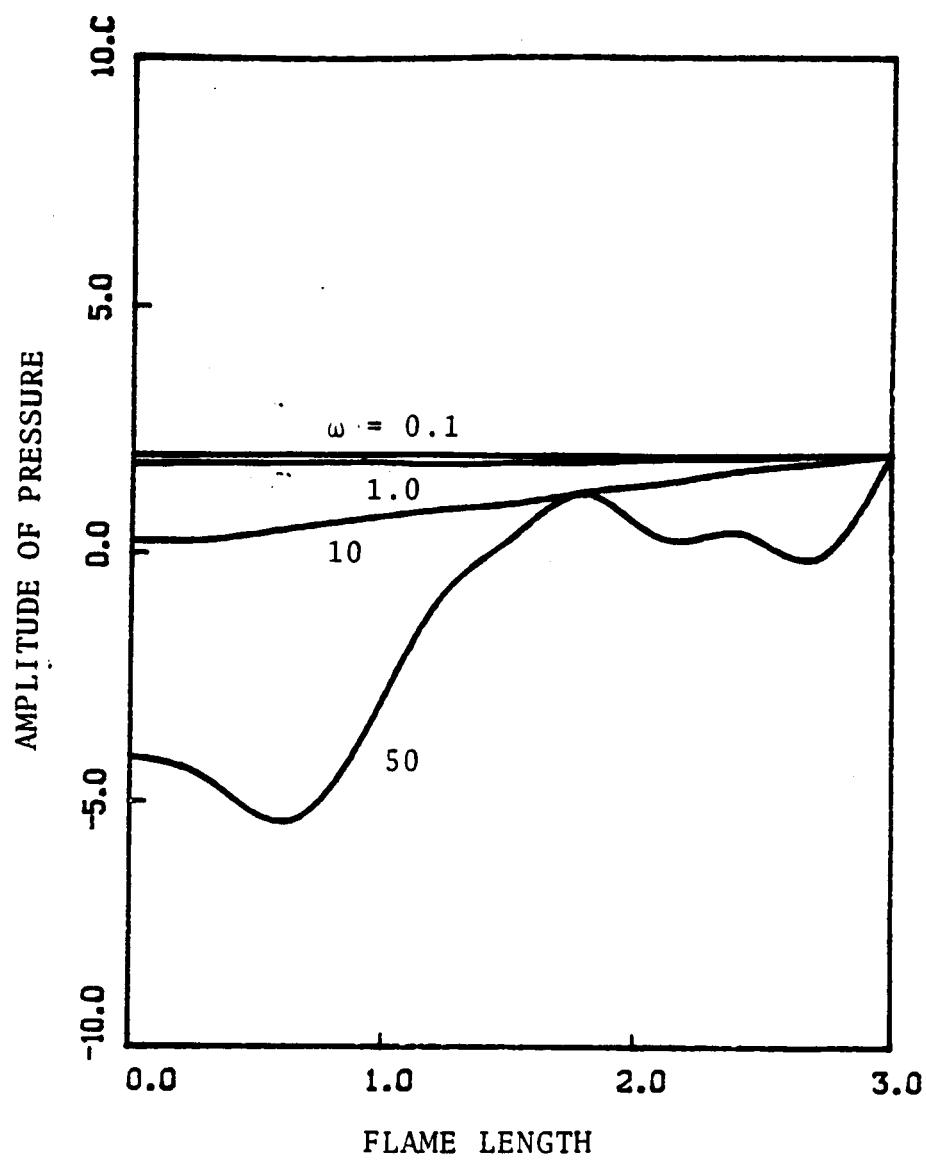
Fig. 22 Second order pressure distributions
         vs. frequency. Both time-dependent
         and time-independent coefficients
         are added for denoting the maximum
         deviation from mean value.
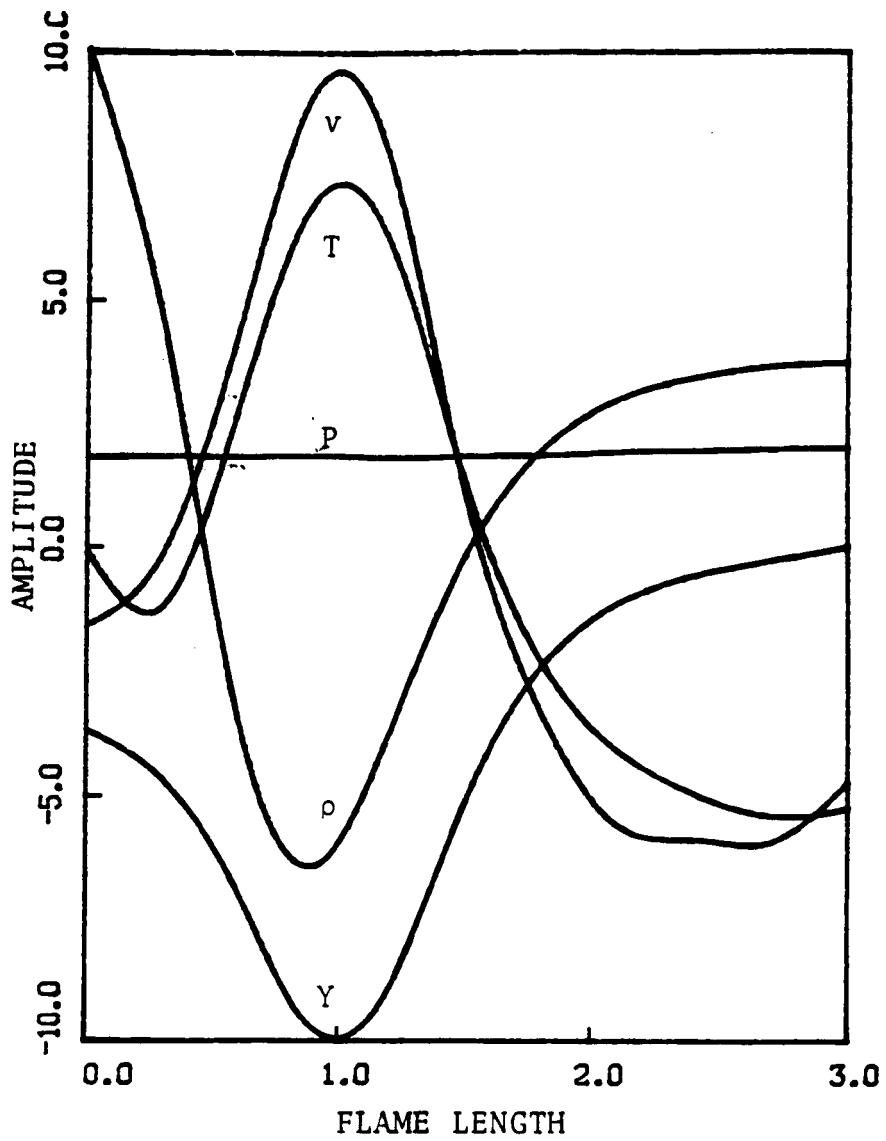
Fig. 23 Second order distributions of field
variables at ω = 1.
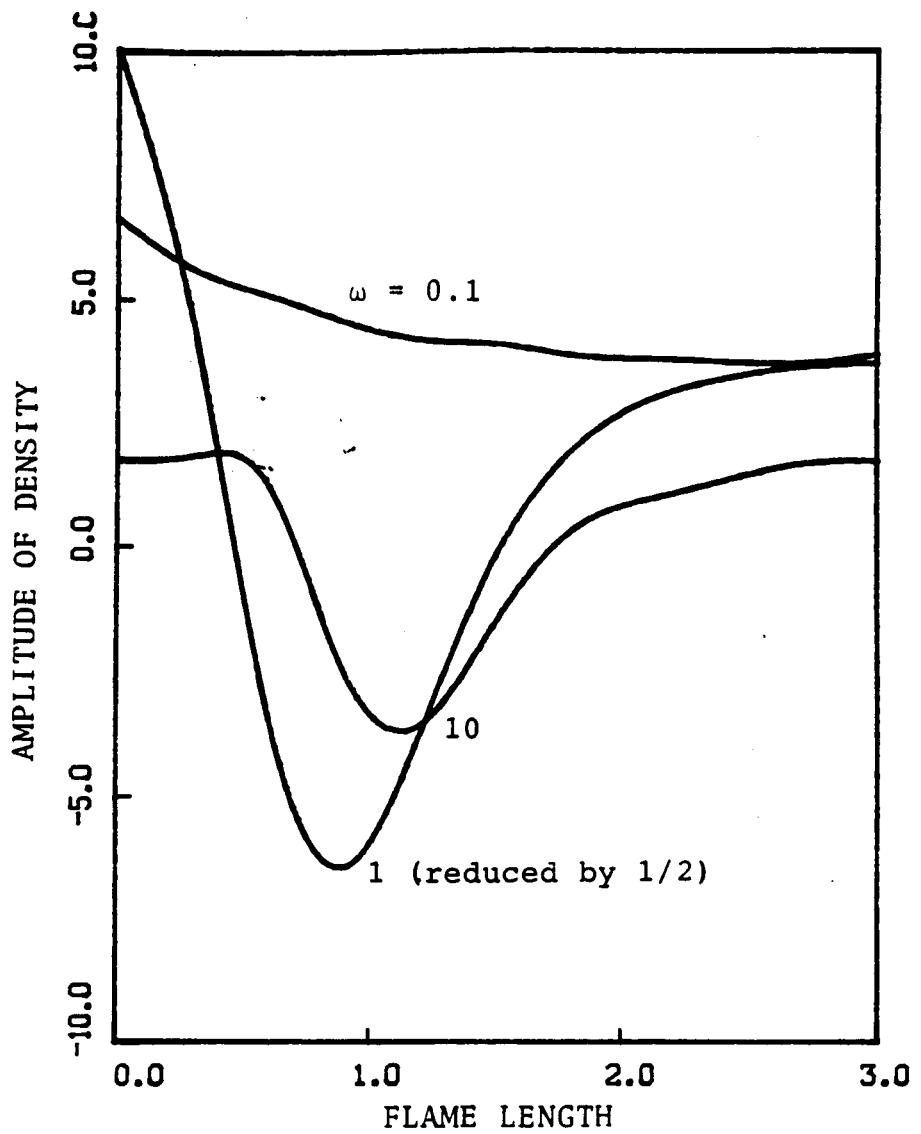
Fig. 24 Second order density distributions
vs. frequency.
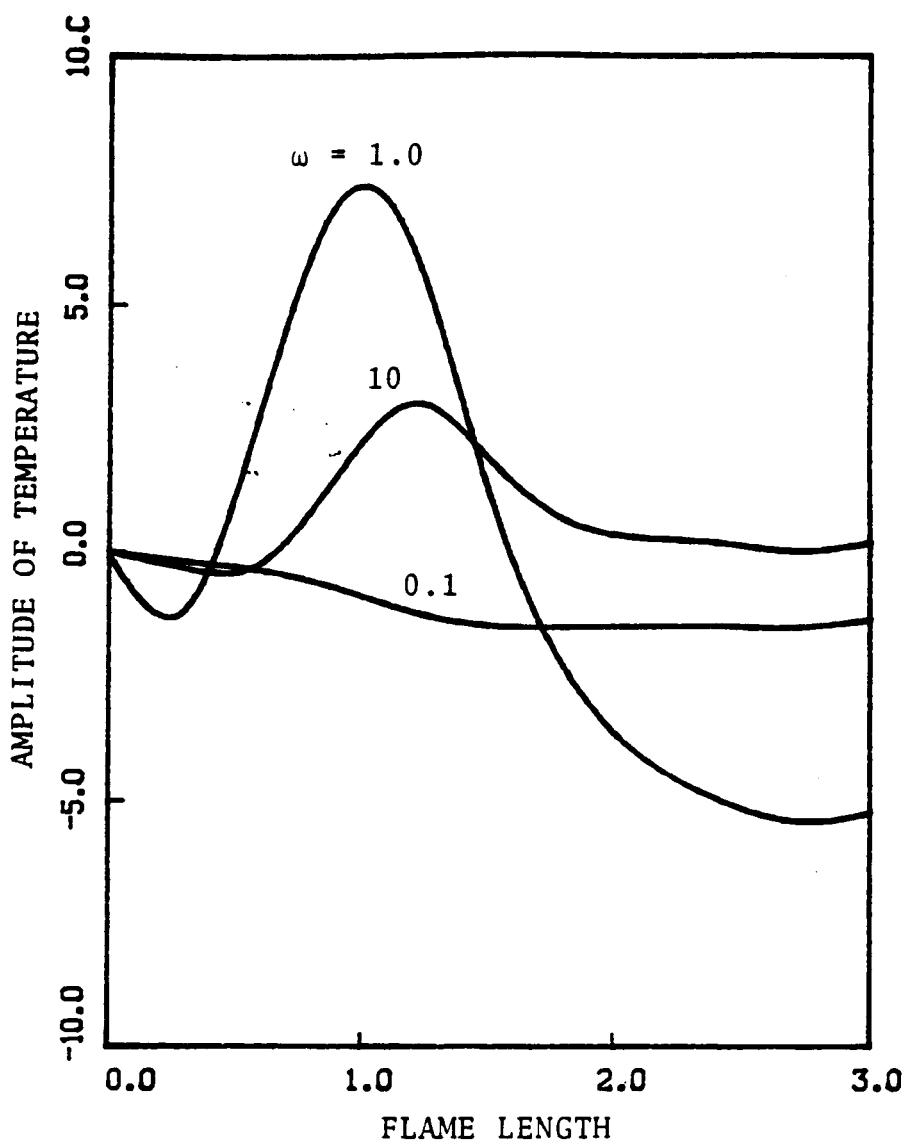
Fig. 25   Second order temperature distri-
butions vs. frequency.
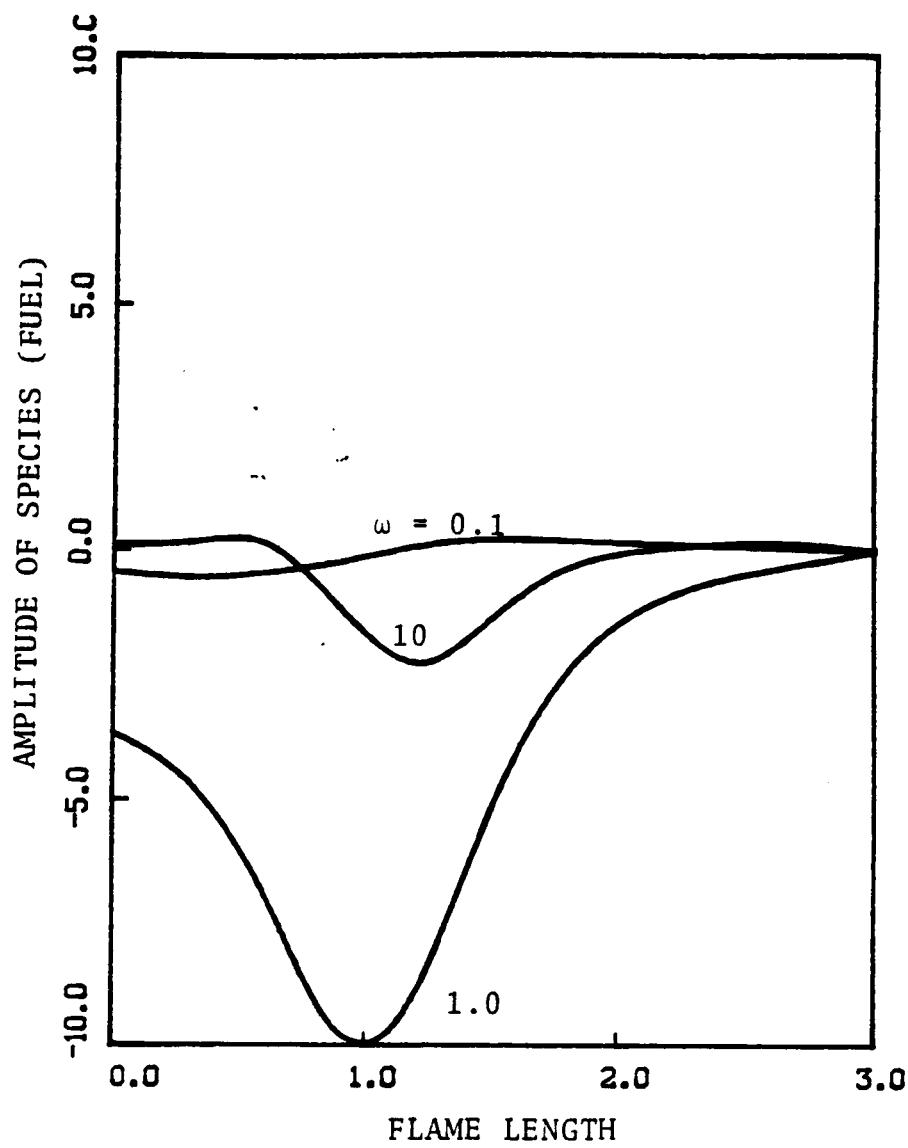
Fig. 26   Second order species (fuel) distri-
butions vs. frequency.
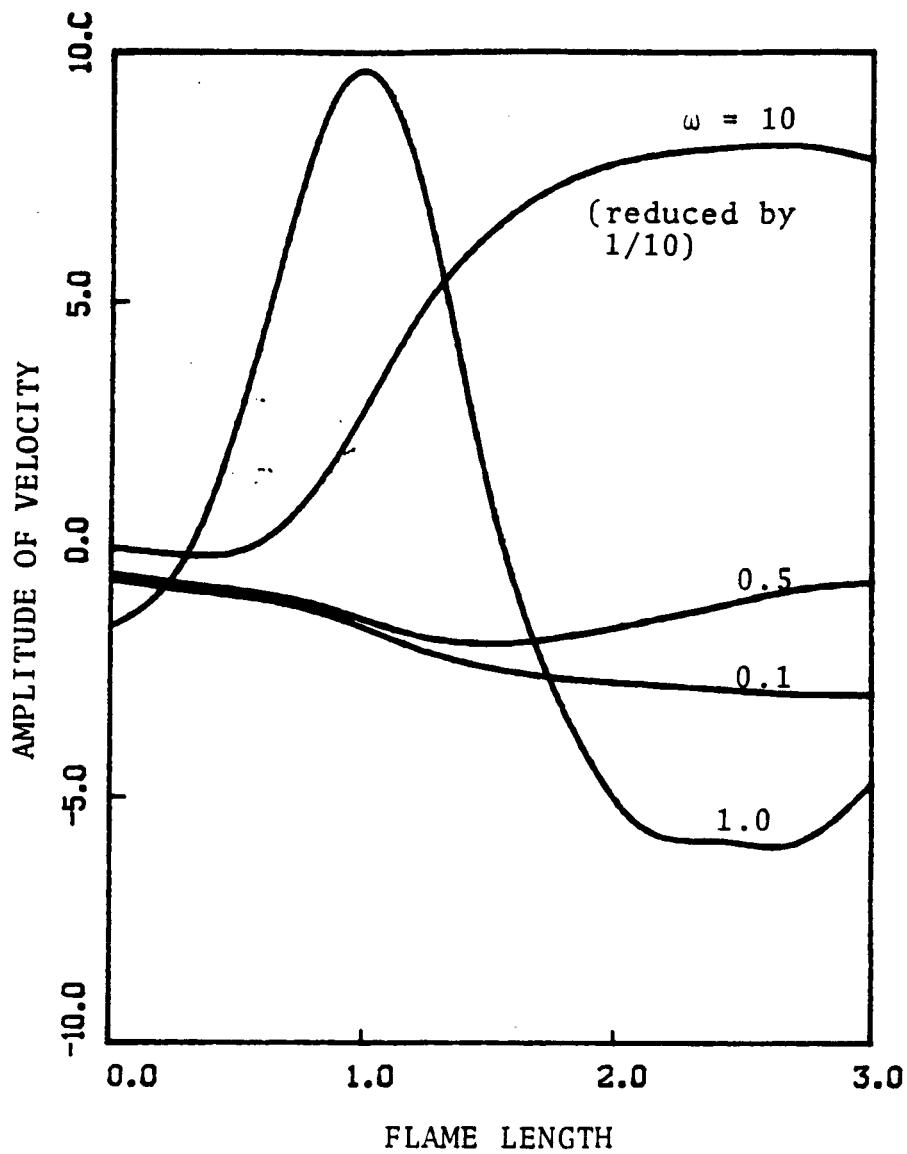
Fig. 27 Second order velocity distributions
vs. frequency.
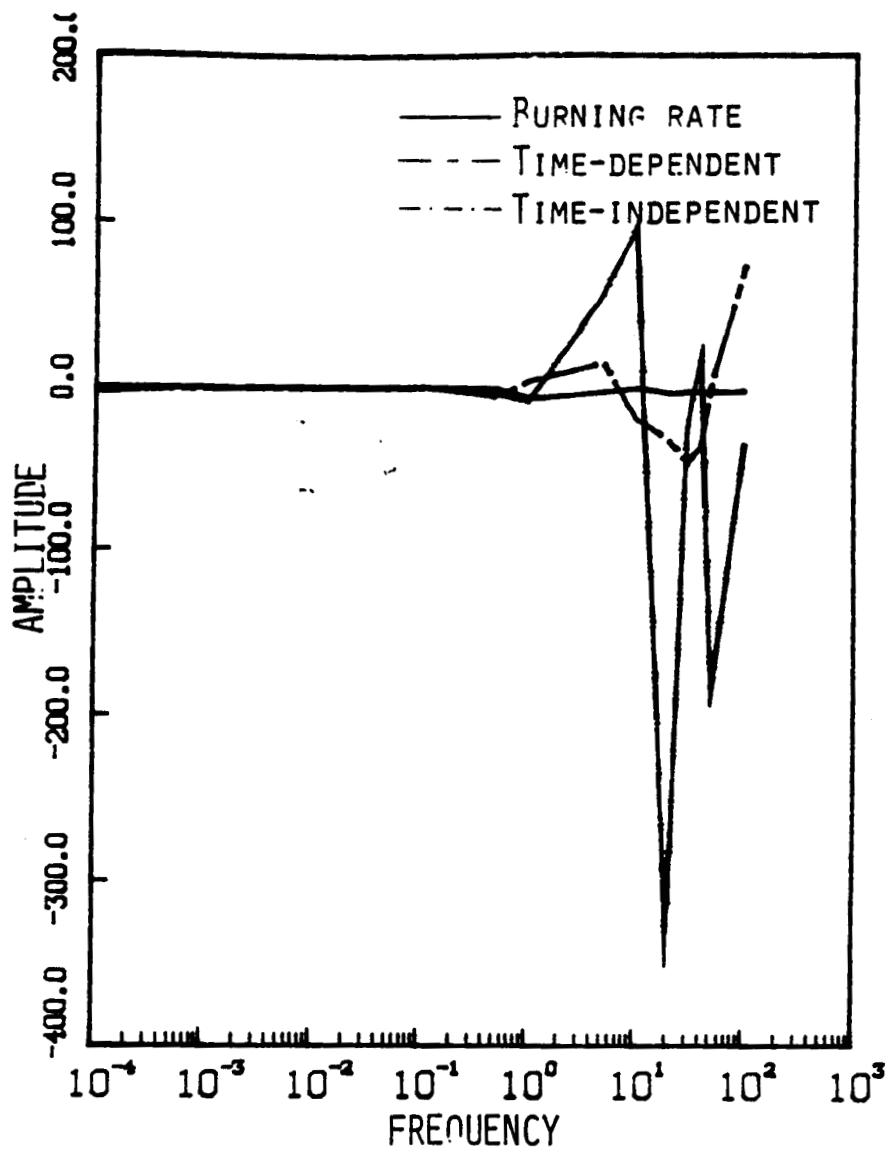
Fig. 28 Offset burning rate and acoustic
admittances of the second order time
-dependent and time-independent sys-
tems. Negative value of the burning
rate at the frequency region verifies
the past investigations.

APPENDIX F

COMPUTER PROGRAM LISTING

```
C  SPCI.3
C      THIS IS SPCI(SOLID PROPELLANT COMBUSTION INSTABILITY)
C      PROGRAM TO SOLVE THE COMBUSTION INSTABILITY SUBJECT
C    · TO ACOUSTIC PRESSURE DISTURBANCE IN THE SOLID PRO-
C      PELLANTS.
C*******************************************************
C*      TO SOLVE THE 1ST AND 2ND PERTURBATION SYSTEM ON   *
C*      2-DIM.NON-LIN.TIME-DEPEND.COMBUSTION OF A SOLID    *
C*      ROCKET PROPELLANTS FOR FINDING                     *
C*      OUT NATURAL FREQUENCE AND RESPOND FUNCTION          *
C*******************************************************
C
C      A: TOTAL MATRIX (COMPLEX)
C      JC,V: FOR MATH PACK SUBROUTINE CGJR
C      CGJR: CALCULATION OF TOTAL MATRIX EQN (MATH-PACK)
C
C      ITYPE -1 : EIGENVALUE PROBLEM
C      ITYPE -* : INSTABILITY CALCULATION
C      IORDER-1 : FIRST ORDER SYSTEM ONLY
C      IORDER-* : SECOND ORDER CALCULATION
C      ITIME -1 : 2ND ORDER TIME INDEPENDENT
C      ITIME -* : 2ND ORDER TIME DEPENDENT
C      ITRT : NUMBER OF CALCULATION W.R.T. FREQUENCE
C
C
       VIRTUAL   A
C
       PARAMETER NW-5,NL-11,L4-4,NV-6
       PARAMETER ND-NW*NL,NE-(NW-1)*(NL-1),NB-NW*4
       PARAMETER NN-ND*NV,NQ-NN+NB,MC-NQ+1
C
       IMPLICIT DOUBLE PRECISION (A-H,O-Z)
       COMPLEX A(NQ,MC),V
       COMPLEX FST(NN)
C
       INTEGER JC(NQ)
C
       COMMON/FIRST/FST
C
C
       V-CMPLX(4.0,0.0)
       ITYPE - 2
       IORDER- 2
       ITIME - 1
         OME - 1.0
C
       CALL GEOMET(WD,HT)
       CALL DATAIN
       CALL STEADY
       CALL ASSM1(A,ITYPE,OME)
C
       IF (ITYPE.EQ.1) THEN
```

```
          CALL EIGEN(A)
          ELSE
          CALL BNDRY1(A,OME,HT)
          ENDIF
C
C-----SOLUTION BY CGJR SUBROUTINE
C
          CALL CGJR(A,MC,NQ,NQ,MC,$801,JC,V)
C
C----- RESERVE 1ST ORDER SOLUTION
C
          DO 100 I=1,NN
          FST(I)=A(I,MC)
          WRITE(10,*)  FST(I)
    100 CONTINUE
C
          CALL OUTPUT(1,FST)
C
          IF(IORDER.EQ.1) GO TO 5000
          CALL ASSM2(A,ITIME)
          CALL BNDRY2(A,OME,ITIME,HT)
          CALL CGJR(A,MC,NQ,NQ,MC,$801,JC,V)
C
C----- RESERVE 2ND ORDER SOLUTION
C
          DO 200 I=1,NN
          FST(I)=A(I,MC)
          WRITE(20,*)  FST(I)
    200 CONTINUE
C
          CALL OUTPUT(2,FST)
C
          PRINT*,JC(2)
    801   PRINT*,'++ERROR(1ST OR 2ND ORDER) ++',JC(1)
C
   5000 CONTINUE
C
          STOP
          END
C
```

```fortran
      SUBROUTINE GEOMET(WD,HT)
C
C     ********************************************************
C     SUBROUTINE GEOMET (PREPARATION FOR CALCULATION DOMAIN)
C     ********************************************************
C
C
      PARAMETER NW=5,NL=11,L4=4
      PARAMETER ND=NW*NL,NE=(NW-1)*(NL-1)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/DIMEN/ IE(NE,L4),X(ND),Y(ND)
C
      WIDE=6.0D+0
      HITE=3.0D+0
C
C-----(1) DEFINE GLOBAL NODES
C
      DO 10 I=1,NE
      IE(I,1)=(I-1)/(NL-1)+I+1
      IE(I,2)=IE(I,1)+NL
      IE(I,3)=IE(I,2)-1
      IE(I,4)=IE(I,1)-1
   10 CONTINUE
C
C-----(2) DEFINE GLOBAL COORD. X AND Y
C
      WD=WIDE/(NW-1)
      HT=HITE/(NL-1)
      K=0
      DO 20 I=1,NW
      DO 20 J=1,NL
      K=K+1
      X(K)=(I-1)*WD
      Y(K)=HITE-(J-1)*HT
      IF ( Y(K).LE.0.001) THEN
          Y(K)=0.0
      ENDIF
   20 CONTINUE
C
      WRITE(6,900)
  900 FORMAT(1H1,1X,'GLOBAL NODE, COORD., INPUT DATA'//)
C
      DO 500 I=1,NE
  500 WRITE(6,910) I,(IE(I,K),K=1,L4)
  910 FORMAT(1H ,3X,'NE=',I3,5X,4I5)
      PRINT*, 'X=',(X(I),I=1,ND-NL+1,NL)
      PRINT*, 'Y=',(Y(I),I=1,NL)
C
      RETURN
      END
C
```

```fortran
      SUBROUTINE DATAIN
C
C     ****************************************************
C     SUBROUTINE DATAIN( INPUT DATA )
C     ****************************************************
C        TC - TEMPERATURE AT COLD SIDE PROPELLANT
C        TS - TEMPERATURE AT SURFACE
C        HO - HEAT OF COMBUSTION PER UNIT MASS
C        EO - GAS PHASE ACTIVATION ENERGY
C        ES - SURFACE ACTIVATION ENERGY
C        QL - LATENT HEAT OF VAPORIZATION
C      DEL1 - KS*CP/K*CS
C      DEL2 - K/KS
C        BE - DENSITY RATIO(RS/RO)
C       GAM - SPECIFIC RATIO
C        PR - PRANDTL NUMBER
C        BM - MACH NUMBER
C
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/GIVEN/TC,TS,HO,EO,ES,QL,DEL1,
     &             DEL2,BE,GAM,PR,BM
C
        TC=0.15D+0
        TS=0.35D+0
        HO=1.30D+0
        EO=1.0D+01
        ES=4.00D+0
        QL=0.15D+0
      DEL1=1.00D+0
      DEL2=1.00D+0
        BE=1.0D+03
       GAM=1.2D+00
        PR=1.0D+00
        BM=0.3D-02
C
      WRITE(6,914)
      PRINT*, 'TS, TC, HO, ES, EO, QL, DEL1, DEL2, GAM, BE, PR, BM'
      PRINT*,  TS,TC,HO,ES,EO,QL,DEL1,DEL2,GAM,BE,PR,BM
  914 FORMAT(//,1X,'INPUT CONSTANTS:')
C
      RETURN
      END

C
```

```
      SUBROUTINE STEADY
C
C     ***********************************************************
C     SUBROUTINE STEADY( ZEROTH ORDER )
C     ***********************************************************
C     UC : CORE VELOCITY FOR 2-D FLOW FIELD
C     RC : DISTANCE FROM SURFACE FOR 2-D FLOW FIELD
C     RZG: STEADY STATE DENSITY
C     UZG: STEADY STATE X-VELOCITY
C     VZG: STEADY STATE Y-VELOCITY
C     TZG: STEADY STATE TEMPERATURE
C     SZG: STEADY STATE SPECIES(FUEL)
C     PZG: STEADY STATE PRESSURE
C     WZG: STEADY STATE REACTION RATE
C
C
      PARAMETER NW=5,NL=11,L4=4
      PARAMETER ND=NW*NL,NE=(NW-1)*(NL-1)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/ZERO/RZG(ND),UZG(ND),VZG(ND),TZG(ND),
     &           SZG(ND),PZG(ND),WZG(ND)
      COMMON/GIVEN/TC,TS,HO,EO,ES,QL,DEL1,
     &           DEL2,BE,GAM,PR,BM
      COMMON/DIMEN/IE(NE,L4),X(ND),Y(ND)
C
C*    DATA TZG/1.00D+0,9.50D-1,8.50D-1,5.50D-1,3.50D-1/
C     DATA TZG/1.00D+0,9.95D-1,9.93D-1,9.90D-1,9.70D-1,
C    &         9.50D-1,9.20D-1,8.00D-1,6.25D-1,4.50D-1,3.50D-1/
      DATA TZG/1.00D+0,9.90D-1,9.80D-1,9.65D-1,9.50D-1,
     &         9.00D-1,8.30D-1,6.80D-1,5.40D-1,4.50D-1,3.50D-1/
C     DATA TZG/1.00D+0,9.95D-1,9.90D-1,9.85D-1,9.80D-1,
C    &         9.72D-1,9.65D-1,9.57D-1,9.50D-1,9.25D-1,9.00D-1,
C    &         8.65D-1,8.30D-1,7 50D-1,6.80D-1,6.10D-1,5.40D-1,
C    &         4.95D-1,4.50D-1,4.00D-1,3.50D-1/
C
      UC=0.0D+0
      RC=5.0D+0
C
      DO 100 J=2,NW
      DO 100 I=1,NL
      K=I+(J-1)*NL
      TZG(K)=TZG(I)
 100  CONTINUE
C
      DO 200 I=1,ND
       RZG(I)=1.0/TZG(I)
       UZG(I)=UC*(1.0-EXP(-Y(I)/RC))
       VZG(I)=TZG(I)
       SZG(I)=1.0/HO*(1.0-TZG(I))
       PZG(I)=1.0
       WZG(I)=2827880.0*((1.0-TZG(I))/TZG(I))**2
```

```fortran
C           WZG(I)=2565000.0*((1.0-TZG(I))/TZG(I))**2
C           WZG(I)=2848208.85*((1.0-TZG(I))/TZG(I))**2
     &           *EXP(-EO/TZG(I))
 200  CONTINUE
C
      WRITE(6,900)
      WRITE(6,910)
      DO 500 I=1,ND
      WRITE(6,920) I,RZG(I),UZG(I),VZG(I),TZG(I),SZG(I),PZG(I),WZG(I)
 500  CONTINUE
 900  FORMAT(1H1, 5X,'ZEROTH ORDER(STEADY) SOLUTION'//)
 910  FORMAT(5X,'NODE',6X,'RO',11X,'U',12X,'V',
     &           12X,'T',12X,'Y',12X,'P',12X,'W')
 920  FORMAT(5X,I5,7(E10.3,3X))
C
      RETURN
      END
C
```

```
      SUBROUTINE ASSM1(A,ITYPE,OME)
C
C***************************************************************
C        SUBROUTINE ASSM1
C***************************************************************
C     INTPOL : INTERPOLATION AND ITS DERIVATIVES
C     CGJR : CALCULATION OF TOTAL MATRIX EQN (MATH-PACK)
C
      PARAMETER NW=5,NL=11,L4=4,NV=6
      PARAMETER ND=NW*NL,NE=(NW-1)*(NL-1),NB=NW*4
      PARAMETER NN=ND*NV,NQ=NN+NB,MC=NQ+1
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMPLEX A(NQ,MC)
C
      COMMON/GRAD/ PNNL(L4,L4), PXYL(L4,L4), PNXL(L4,L4),
     &             PNYL(L4,L4), PDDL(L4,L4),
     &             PXXL(L4,L4), PYYL(L4,L4)
      COMMON/LOCAL1/AC2(L4,L4),   AC3(L4,L4),   AC4(L4,L4),
     &              AC5(L4,L4),   AC6(L4,L4),   AC7(L4,L4),
     &             AMX1(L4,L4),  AMX2(L4,L4),  AMX3(L4,L4),
     &             AMX4(L4,L4),  AMX5(L4,L4),  AMX6(L4,L4),
     &             AMX9(L4,L4), AMX11(L4,L4), AMX12(L4,L4),
     &             AMY1(L4,L4),  AMY2(L4,L4),  AMY3(L4,L4),
     &             AMY4(L4,L4),  AMY5(L4,L4),  AMY6(L4,L4),
     &             AMY9(L4,L4), AMY11(L4,L4), AMY12(L4,L4),
     &              AE1(L4,L4),   AE2(L4,L4),   AE3(L4,L4),
     &              AE4(L4,L4),   AE5(L4,L4),   AE6(L4,L4),
     &              AE7(L4,L4),   AE8(L4,L4),   AE9(L4,L4),
     &             AE10(L4,L4),  AE11(L4,L4),  AE12(L4,L4),
     &             AE14(L4,L4),  AE15(L4,L4),
     &             ASP1(L4,L4),  ASP2(L4,L4),  ASP3(L4,L4),
     &             ASP4(L4,L4),  ASP5(L4,L4),  ASP6(L4,L4),
     &             ASP7(L4,L4),  ASP8(L4,L4),  ASP9(L4,L4),
     &            APRS1(L4,L4)
      COMMON/DIMEN/IE(NE,L4),X(ND),Y(ND)
      COMMON/ZERO/RZG(ND),UZG(ND),VZG(ND),TZG(ND),
     &            SZG(ND),PZG(ND),WZG(ND)
      COMMON/GIVEN/TC,TS,HO,EO,ES,QL,DEL1,
     &             DEL2,BE,GAM,PR,BM
      COMMON/COEFF/C1Z(ND,ND),C2Z(ND,ND),C3Z(ND,ND),C4Z(ND,ND),
     &             X1Z(ND,ND),X2Z(ND,ND),X3Z(ND,ND),X4Z(ND,ND),
     &             X5Z(ND,ND),
     &             Y1Z(ND,ND),Y2Z(ND,ND),Y3Z(ND,ND),Y4Z(ND,ND),
     &             Y5Z(ND,ND),
     &             E1Z(ND,ND),E2Z(ND,ND),E3Z(ND,ND),E4Z(ND,ND),
     &             E5Z(ND,ND),E6Z(ND,ND),E7Z(ND,ND),
     &             S1Z(ND,ND),S2Z(ND,ND),S3Z(ND,ND),S4Z(ND,ND),
     &             S5Z(ND,ND),S6Z(ND,ND),PRES1(ND,ND),
     &             ANXZ(ND,ND),ANYZ(ND,ND)
C
C
```

```
      GMB2=1.0/(GAM*BM**2)
C*    PR3=1.0/3.0*PR
      PR3=0.0
C
C-----(1) INITIALIZATION
C
      DO 10 J=1,ND
      DO 10 I=1,ND
      ANXZ(I,J)=0.0
      ANYZ(I,J)=0.0
C
      C1Z(I,J)=0.0
      C2Z(I,J)=0.0
      C3Z(I,J)=0.0
      C4Z(I,J)=0.0
      X1Z(I,J)=0.0
      X2Z(I,J)=0.0
      X3Z(I,J)=0.0
      X4Z(I,J)=0.0
      X5Z(I,J)=0.0
      Y1Z(I,J)=0.0
      Y2Z(I,J)=0.0
      Y3Z(I,J)=0.0
      Y4Z(I,J)=0.0
      Y5Z(I,J)=0.0
      E1Z(I,J)=0.0
      E2Z(I,J)=0.0
      E3Z(I,J)=0.0
      E4Z(I,J)=0.0
      E5Z(I,J)=0.0
      E6Z(I,J)=0.0
      E7Z(I,J)=0.0
      S1Z(I,J)=0.0
      S2Z(I,J)=0.0
      S3Z(I,J)=0.0
      S4Z(I,J)=0.0
      S5Z(I,J)=0.0
      S6Z(I,J)=0.0
      PRES1(I,J)=0.0
 10   CONTINUE
C
C-----(2) GLOBAL SUMMATION FOR EACH ELEMENT
C
      DO 700 LE=1,NE
C
C-----INTERPOLATION FUNCION AND ITS DERIVATIVES
C
      CALL INTPOL(LE,1,ITIME)
C
C
      DO 500 M=1,L4
      J=IE(LE,M)
      DO 500 N=1,L4
```

```
          I=IE(LE,N)
C
C-----(3) BLOCK ASSM:TERM BY TERM GLOBAL ASSME
C
      ANXZ(I,J)=ANXZ(I,J)+PNXL(N,M)
      ANYZ(I,J)=ANYZ(I,J)+PNYL(N,M)
C
       C1Z(I,J)= C1Z(I,J)+PNNL(N,M)
       C2Z(I,J)= C2Z(I,J)+ AC2(N,M)+ AC3(N,M)
       C3Z(I,J)= C3Z(I,J)+ AC4(N,M)+ AC5(N,M)
       C4Z(I,J)= C4Z(I,J)+ AC6(N,M)+ AC7(N,M)
C
       X1Z(I,J)= X1Z(I,J)+AMX1(N,M)
C    &              +GMB2*(AMX2(N,M)+AMX3(N,M))
       X2Z(I,J)= X2Z(I,J)+AMX4(N,M)
       X3Z(I,J)= X3Z(I,J)+AMX5(N,M)+AMX6(N,M)
     &            +PR*PDDL(N,M)+PR3*PXXL(N,M)
      X4Z(I,J)= X4Z(I,J)+AMX9(N,M)+PR3*PXYL(M,N)
       X5Z(I,J)= X5Z(I,J)+AMX11(N,M)+AMX12(N,M)
C
       Y1Z(I,J)= Y1Z(I,J)+AMY1(N,M)
C    &              +GMB2*(AMY2(N,M)+AMY3(N,M))
       Y2Z(I,J)= Y2Z(I,J)+AMY4(N,M)
       Y3Z(I,J)= Y3Z(I,J)+AMY5(N,M)+AMY9(N,M)
     &            +PR*PDDL(N,M)+PR3*PYYL(N,M)
      Y4Z(I,J)= Y4Z(I,J)+AMY6(N,M)+PR3*PXYL(N,M)
       Y5Z(I,J)= Y5Z(I,J)+AMY11(N,M)+AMY12(N,M)
C
       E1Z(I,J)= E1Z(I,J)+ AE1(N,M)
     &               -2.0*HO*AE4(N,M)
C    &          +RGAM*(AE2(N,M)+ AE3(N,M))-2.0*HO*AE4(N,M)
       E2Z(I,J)= E2Z(I,J)+AE5(N,M)
C    &            +RGAM*(AE6(N,M)+AE7(N,M))
       E3Z(I,J)= E3Z(I,J)+AE8(N,M)
C    &             +RGAM*(AE9(N,M)+AE10(N,M))
       E4Z(I,J)= E4Z(I,J)+AE11(N,M)
       E5Z(I,J)= E5Z(I,J)+AE12(N,M)
     &            +PDDL(N,M)-HO*EO*AE14(N,M)
       E6Z(I,J)= E6Z(I,J)+AE15(N,M)
       E7Z(I,J)= E7Z(I,J)+ AE8(N,M)
C
       S1Z(I,J)= S1Z(I,J)+ASP1(N,M)+2.0*ASP2(N,M)
       S2Z(I,J)= S2Z(I,J)+ASP3(N,M)
       S3Z(I,J)= S3Z(I,J)+ASP4(N,M)
       S4Z(I,J)= S4Z(I,J)+ASP5(N,M)
       S5Z(I,J)= S5Z(I,J)+ASP6(N,M)
       S6Z(I,J)= S6Z(I,J)+ASP7(N,M)
     &            +PDDL(N,M)+ASP9(N,M)*2.0
C
      PRES1(I,J)= PRES1(I,J)+APRS1(N,M)
  500 CONTINUE
  700 CONTINUE
C
```

```fortran
C-----(4) CONSTRUCTION OF GLOBAL MATRIX
C
      CALL GMATRX(A,OME,ITYPE)
C
      RETURN
      END
C
```

```fortran
      SUBROUTINE GMATRX(A,OME,ITYPE)
C
C******************************************************
C       SUBROUTINE GMATRX( GLOBAL MATRIX BEFORE APPLYING
C                          BOUNDARY CONDITIONS )
C******************************************************
C
C
      PARAMETER NW=5,NL=11,L4=4,NV=6
      PARAMETER ND=NW*NL,NE=(NW-1)*(NL-1),NB=NW*4
      PARAMETER NN=ND*NV,NQ=NN+NB,MC=NQ+1
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMPLEX A(NQ,MC)
C
      COMMON/COEFF/ C1Z(ND,ND),C2Z(ND,ND),C3Z(ND,ND),C4Z(ND,ND),
     &              X1Z(ND,ND),X2Z(ND,ND),X3Z(ND,ND),X4Z(ND,ND),
     &              X5Z(ND,ND),
     &              Y1Z(ND,ND),Y2Z(ND,ND),Y3Z(ND,ND),Y4Z(ND,ND),
     &              Y5Z(ND,ND),
     &              E1Z(ND,ND),E2Z(ND,ND),E3Z(ND,ND),E4Z(ND,ND),
     &              E5Z(ND,ND),E6Z(ND,ND),E7Z(ND,ND),
     &              S1Z(ND,ND),S2Z(ND,ND),S3Z(ND,ND),S4Z(ND,ND),
     &              S5Z(ND,ND),S6Z(ND,ND),PRES1(ND,ND),
     &              ANXZ(ND,ND),ANYZ(ND,ND)
      COMMON/DIMEN/IE(NE,L4),X(ND),Y(ND)
      COMMON/ZERO/RZG(ND),UZG(ND),VZG(ND),TZG(ND),
     &            SZG(ND),PZG(ND),WZG(ND)
      COMMON/GIVEN/TC,TS,HO,EO,ES,QL,DEL1,
     &             DEL2,BE,GAM,PR,BM
C
C
C-----(1) BLOCK RI : CONSTRUCTION OF GLOBAL MATRIX
C
      ND2=2*ND
      ND3=3*ND
      ND4=4*ND
      ND5=5*ND
C
      RGAM=(GAM-1.0)/GAM
      GMB2=1.0/(GAM*BM**2)
C
      DO 1000 J=1,ND
      M1=J+ND
      M2=J+ND2
      M3=J+ND3
      M4=J+ND4
      M5=J+ND5
      DO 1000 I=1,ND
      N1=I+ND
      N2=I+ND2
      N3=I+ND3
```

```
            N4=I+ND4
            N5=I+ND5
C
            IF(ITYPE.EQ.1) THEN
            CRZI= C1Z(I,J)
            XUZI= X2Z(I,J)
            YVZI= Y2Z(I,J)
            ETZI= E4Z(I,J)
            EPZI=-C1Z(I,J)*RGAM
            ELSE
            CRZI= C1Z(I,J)*OME
            XUZI= X2Z(I,J)*OME
            YVZI= Y2Z(I,J)*OME
            ETZI= E4Z(I,J)*OME
            EPZI=-C1Z(I,J)*OME*RGAM
            SSZI= S5Z(I,J)*OME
            ENDIF
C
C     CRZI= C1Z(I,J)*OME
            CRZR= C2Z(I,J)
            CUZR= C3Z(I,J)
            CVZR= C4Z(I,J)
C
            XRZR= X1Z(I,J)
C     XUZI= X2Z(I,J)*OME
            XUZR= X3Z(I,J)
            XVZR= X4Z(I,J)
            XPZR=ANXZ(I,J)*GMB2
C
            YRZR= Y1Z(I,J)
C     YVZI= Y2Z(I,J)*OME
            YVZR= Y3Z(I,J)
            YUZR= Y4Z(I,J)
            YPZR=ANYZ(I,J)*GMB2
C
            ERZR= E1Z(I,J)
            EUZR= E2Z(I,J)
            EVZR= E3Z(I,J)
C     ETZI= E4Z(I,J)*OME
            ETZR= E5Z(I,J)
            ESZR=-E6Z(I,J)*2.0*HO
C     EPZI=-C1Z(I,J)*RGAM*OME
C
            SRZR= S1Z(I,J)
            SUZR= S2Z(I,J)
            SVZR= S3Z(I,J)
            STZR= S4Z(I,J)*EO
C     SSZI= S5Z(I,J)*OME
            SSZR= S6Z(I,J)
C
CC    PRZR2=PRES1(I,J)
CC    PTZR2=  E4Z(I,J)
CC    PPZR2= -C1Z(I,J)
```

```
C
      IF(I.EQ.J) THEN
      PRZR2=  TZG(I)
      PTZR2=  RZG(I)
      PPZR2= -1.00
      ELSE
      PRZR2=0.000
      PTZR2=0.000
      PPZR2=0.000
      ENDIF
C
C-----(2) BLOCK COEFF
C
      A(I,J)  =CMPLX(CRZR,CRZI)
      A(I,M1) =CMPLX(CUZR,0.0)
      A(I,M2) =CMPLX(CVZR,0.0)
      A(I,M3) =CMPLX(0.0,0.0)
      A(I,M4) =CMPLX(0.0,0.0)
      A(I,M5) =CMPLX(0.0,0.0)
C
      A(N1,J) =CMPLX(XRZR,0.0)
      A(N1,M1)=CMPLX(XUZR,XUZI)
      A(N1,M2)=CMPLX(XVZR,0.0)
      A(N1,M3)=CMPLX(0.00,0.0)
      A(N1,M4)=CMPLX(0.0,0.0)
      A(N1,M5)=CMPLX(XPZR,0.0)
C
      A(N2,J) =CMPLX(YRZR,0.0)
      A(N2,M1)=CMPLX(YUZR,0.0)
      A(N2,M2)=CMPLX(YVZR,YVZI)
      A(N2,M3)=CMPLX(0.00,0.0)
      A(N2,M4)=CMPLX(0.0,0.0)
      A(N2,M5)=CMPLX(YPZR,0.0)
C
      A(N3,J) =CMPLX(ERZR,0.0)
      A(N3,M1)=CMPLX(EUZR,0.0)
      A(N3,M2)=CMPLX(EVZR,0.0)
      A(N3,M3)=CMPLX(ETZR,ETZI)
      A(N3,M4)=CMPLX(ESZR,0.0)
      A(N3,M5)=CMPLX(0.00,EPZI)
C
      A(N4,J) =CMPLX(SRZR,0.0)
      A(N4,M1)=CMPLX(SUZR,0.0)
      A(N4,M2)=CMPLX(SVZR,0.0)
      A(N4,M3)=CMPLX(STZR,0.0)
      A(N4,M4)=CMPLX(SSZR,SSZI)
      A(N4,M5)=CMPLX(0.00,0.0)
C
      A(N5, J)=CMPLX(PRZR2,   0.0)
      A(N5,M1)=CMPLX(  0.0,   0.0)
      A(N5,M2)=CMPLX(  0.0,   0.0)
      A(N5,M3)=CMPLX(PTZR2,   0.0)
      A(N5,M4)=CMPLX(  0.0,   0.0)
```

```
       A(N5,M5)=CMPLX(PPZR2,  0.0)
C
 1000 CONTINUE
C
C-----(3) BLOCK SPACE : PREPERATION FOR LAGRANGE MULTIPLIER
C                        APPLICATION
C-----ZERO SETS FOR TOTAL MATRIX
C
       DO 100 J=1,NQ
       DO 110 I=NN+1,NQ
       A(I,J)=CMPLX(0.0,0.0)
  110 CONTINUE
       A(J,MC)=CMPLX(0.0,0.0)
  100 CONTINUE
C
C-----(4) BLOCK RSV : RESERVATION FOR HIGHER ORDER CALCULATION
C
       DO 2000 J=1,MC
       DO 2000 I=1,NQ
          WRITE(7,7777) A(I,J)
 2000 CONTINUE
 7777 FORMAT(5X,E27.18,E27.18)
C
       RETURN
       END
C
```

```
      SUBROUTINE ASSM2(A,ITIME)
C
C********************************************
C      SUBROUTINE ASSM2
C********************************************
C
C
      PARAMETER NW=5,NL=11,L4=4,NV=6
      PARAMETER ND=NW*NL,NE=(NW-1)*(NL-1),NB=NW*4
      PARAMETER NN=ND*NV,NQ=NN+NB,MC=NQ+1
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMPLEX A(NQ,MC),FST(NN)
      COMPLEX CF ,XF ,YF ,EF ,SF ,PF ,
     &        GC ,GX ,GY ,GE ,GS ,GP ,
     &        ACG,AXG,AYG,AEG,ASG,APG
C
      COMMON/LOCAL2/ ACG(L4),AXG(L4),AYG(L4),
     &               AEG(L4),ASG(L4),APG(L4)
      COMMON/DIMEN/IE(NE,L4),X(ND),Y(ND)
      COMMON/FIRST/FST
C
      DIMENSION CF(ND),XF(ND),YF(ND),EF(ND),SF(ND),PF(ND)
C
      CLOSE(7)
       OPEN(7)
      DO 2000 J=1,MC
      DO 2000 I=1,NQ
         READ(7,7777) A(I,J)
 2000 CONTINUE
 7777 FORMAT(5X,E27.18,E27.18)

C
      ND2=2*ND
      ND3=3*ND
      ND4=4*ND
      ND5=5*ND
C
C-----(1) INITIALIZATION
C
      DO 100 J=1,ND
C
      CF(J)=CMPLX(0.0,0.0)
      XF(J)=CMPLX(0.0,0.0)
      YF(J)=CMPLX(0.0,0.0)
      EF(J)=CMPLX(0.0,0.0)
      SF(J)=CMPLX(0.0,0.0)
      PF(J)=CMPLX(0.0,0.0)
 100    CONTINUE
C
      DO    700 LE=1,NE
C
```

```
C
C-----(2) INTEGRATION BY GAUSSIAN QUADRATURE
C
      CALL INTPOL(LE,2,ITIME)
C
C-----(3) BLOCK ASSME : GLOBAL ASSEMBLING OF SOURCE TERM
C
      DO 200 M=1,L4
      I=IE(LE,M)
C
      CF(I)=CF(I)+ACG(M)
      XF(I)=XF(I)+AXG(M)
      YF(I)=YF(I)+AYG(M)
      EF(I)=EF(I)+AEG(M)
      SF(I)=SF(I)+ASG(M)
      PF(I)=PF(I)+APG(M)
C
  200 CONTINUE
  700 CONTINUE
C
C-----(4) BLOCK COEFF: COEFFICIANT OF SOURCE TERM
C
      DO 300 J=1,ND
C
      M1=J+ND
      M2=J+ND2
      M3=J+ND3
      M4=J+ND4
      M5=J+ND5
C
      GC=-0.5*CF(J)
      GX=-0.5*XF(J)
      GY=-0.5*YF(J)
      GE=-0.5*EF(J)
      GS=-0.5*SF(J)
      GP=-0.5*PF(J)
C
      A(J ,MC)= GC
      A(M1,MC)= GX
      A(M2,MC)= GY
      A(M3,MC)= GE
      A(M4,MC)= GS
      A(M5,MC)= GP
C*    A(M5,MC)=-TF(J)*RF(J)*0.5
  300 CONTINUE
C
C-----(5) BLOCK RSV:RESERVATION FOR NEXT CALCULATION
C
      DO 400 J=1,ND
      M1=J+ND
      M2=J+ND2
      M3=J+ND3
      M4=J+ND4
```

```fortran
      M5=J+ND5
      DO 400 I=1,ND
      N1=I+ND
      N2=I+ND2
      N3=I+ND3
      N4=I+ND4
      N5=I+ND5
C
C-----(5-A) SUB-BLOCK RSV1:CHANGE OF COEFF. FOR 2ND TIME INDEPENDENT
C
      IF(ITIME.EQ.1) THEN
      A(I ,J )=CMPLX(REAL(A(I ,J )),0.0 )
      A(N1,M1)=CMPLX(REAL(A(N1,M1)),0.0 )
      A(N2,M2)=CMPLX(REAL(A(N2,M2)),0.0 )
      A(N3,M3)=CMPLX(REAL(A(N3,M3)),0.0 )
      A(N4,M4)=CMPLX(REAL(A(N4,M4)),0.0 )
C
C-----(5-B) SUB-BLOCK RSV2:CHANGE OF COEFF. FOR 2ND TIME DEPENDENT
C
      ELSE
      A(I ,J )=CMPLX(REAL(A(I ,J )),2.0*AIMAG(A(I ,J )))
      A(N1,M1)=CMPLX(REAL(A(N1,M1)),2.0*AIMAG(A(N1,M1)))
      A(N2,M2)=CMPLX(REAL(A(N2,M2)),2.0*AIMAG(A(N2,M2)))
      A(N3,M3)=CMPLX(REAL(A(N3,M3)),2.0*AIMAG(A(N3,M3)))
      A(N4,M4)=CMPLX(REAL(A(N4,M4)),2.0*AIMAG(A(N4,M4)))
      ENDIF
C
 400  CONTINUE
C
      DO 3000 J=1,MC
      DO 3000 I=1,NQ
         WRITE(8,7777) A(I,J)
 3000 CONTINUE
C
      RETURN
      END
C
```

```
      SUBROUTINE INTPOL(LE,IORDER,ITIME)
C
C     **************************************************
C   . SUBROUTINE INTPOL (INTERPOLATION FUNCION)
C     **************************************************
C      INTPL... INTERPOLATION AND ITS DERIVATIVES
C      DJ: DETERMINANTS OF JACOBIAN MATRIX
C      IG: NO. OF GAUSS.POINT
C
C
      PARAMETER NW=5,NL=11,L4=4,NV=6
      PARAMETER NE=(NW-1)*(NL-1),ND=NW*NL,NB=NW*4
      PARAMETER NN=ND*NV,IG=4
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
      COMMON/DJP/DJ(IG,IG),
     &           PIWX(L4,IG,IG),PIWY(L4,IG,IG),
     &           PIWN(L4,IG,IG),H(IG),W(IG)
      COMMON/GRAD/ PNNL(L4,L4), PXYL(L4,L4), PNXL(L4,L4),
     &             PNYL(L4,L4), PDDL(L4,L4),
     &             PXXL(L4,L4), PYYL(L4,L4)
      COMMON/DIMEN/IE(NE,L4),X(ND),Y(ND)
      COMMON/ZERO/RZG(ND),UZG(ND),VZG(ND),TZG(ND),
     &            SZG(ND),PZG(ND),WZG(ND)
      COMMON/LZERO/RZ(L4),UZ(L4),VZ(L4),TZ(L4),SZ(L4),PZ(L4),WZ(L4)
C
      DIMENSION A4(4),B4(4),BN(L4),CN(L4)
C
C
      CALL GAUSS(IG,H,W)
C
C-----BLOCK LOCAL
C
      DO 100 I=1,L4
          J=IE(LE,I)
      RZ(I)=RZG(J)
      UZ(I)=UZG(J)
      VZ(I)=VZG(J)
      TZ(I)=TZG(J)
      SZ(I)=SZG(J)
      PZ(I)=PZG(J)
      WZ(I)=WZG(J)
 100  CONTINUE
C
C-----BLOCK JACOB : JACOBIAN CALCULATION
C
      N1=IE(LE,1)
      N2=IE(LE,2)
      N3=IE(LE,3)
      N4=IE(LE,4)
      A4(1)=+X(N1)+X(N2)+X(N3)+X(N4)
```

```fortran
      A4(2)=-X(N1)+X(N2)+X(N3)-X(N4)
      A4(3)=-X(N1)-X(N2)+X(N3)+X(N4)
      A4(4)=+X(N1)-X(N2)+X(N3)-X(N4)
      B4(1)=+Y(N1)+Y(N2)+Y(N3)+Y(N4)
      B4(2)=-Y(N1)+Y(N2)+Y(N3)-Y(N4)
      B4(3)=-Y(N1)-Y(N2)+Y(N3)+Y(N4)
      B4(4)=+Y(N1)-Y(N2)+Y(N3)-Y(N4)
C
      BN(1)=-1.0
      BN(2)= 1.0
      BN(3)= 1.0
      BN(4)=-1.0
      CN(1)=-1.0
      CN(2)=-1.0
      CN(3)= 1.0
      CN(4)= 1.0
C
      DO 200 N=1,L4
      DO 200 M=1,L4
      PNNW=0.0
      PXYW=0.0
      PNXW=0.0
      PNYW=0.0
      PXXW=0.0
      PYYW=0.0
C
      DO 300 I=1,IG
      DO 300 J=1,IG
      PXPN=0.25*(A4(3)+A4(4)*H(I))
      PYPN=0.25*(B4(3)+B4(4)*H(I))
      PXPZ=0.25*(A4(2)+A4(4)*H(J))
      PYPZ=0.25*(B4(2)+B4(4)*H(J))
C
      DJ(I,J)=-PXPN*PYPZ+PYPN*PXPZ
C
      PHIX=0.25*(1.0+BN(M)*H(I))
      PHIY=0.25*(1.0+CN(M)*H(J))
      PPPZ=BN(M)*PHIY
      PPPN=CN(M)*PHIX
      PIWX(M,I,J)=( PYPN*PPPZ-PYPZ*PPPN)/DJ(I,J)
      PIWY(M,I,J)=(-PXPN*PPPZ+PXPZ*PPPN)/DJ(I,J)
C
      PIWN(M,I,J)=4.0*PHIX*PHIY
C
      WWDJ=W(I)*W(J)*DJ(I,J)
C
      PNNW=PNNW+PIWN(N,I,J)*PIWN(M,I,J)*WWDJ
      PXYW=PXYW+PIWX(N,I,J)*PIWY(M,I,J)*WWDJ
      PNXW=PNXW+PIWN(N,I,J)*PIWX(M,I,J)*WWDJ
      PNYW=PNYW+PIWN(N,I,J)*PIWY(M,I,J)*WWDJ
      PXXW=PXXW+PIWX(N,I,J)*PIWX(M,I,J)*WWDJ
      PYYW=PYYW+PIWY(N,I,J)*PIWY(M,I,J)*WWDJ
C
```

```fortran
 300  CONTINUE
      PNNL(N,M)=PNNW
      PXYL(N,M)=PXYW
      PNXL(N,M)=PNXW
      PNYL(N,M)=PNYW
      PXXL(N,M)=PXXW
      PYYL(N,M)=PYYW
      PDDL(N,M)=PXXW+PYYW
 200  CONTINUE
C
      IF (IORDER.EQ.1) THEN
      CALL LASS1
      ELSE
      CALL LASS2(LE,OME,ITIME)
      ENDIF
C
      RETURN
      END
C
```

```fortran
      SUBROUTINE LASS1
C
C     ***************************************************
C     SUBROUTINE LASS1 ( LOCAL ASSEMBLING OF GENERAL TERM )
C     ***************************************************
C
C
      PARAMETER NW=5,NL=11,L4=4,NV=6
      PARAMETER NE=(NW-1)*(NL-1),ND=NW*NL
      PARAMETER IG=4,NN=ND*NV
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/DJP/DJ(IG,IG),
     &          PIWX(L4,IG,IG),PIWY(L4,IG,IG),
     &          PIWN(L4,IG,IG),H(IG),W(IG)
      COMMON/GRAD/ PNNL(L4,L4), PXYL(L4,L4), PNXL(L4,L4),
     &          PNYL(L4,L4), PDDL(L4,L4),
     &          PXXL(L4,L4), PYYL(L4,L4)
      COMMON/LOCAL1/AC2(L4,L4),  AC3(L4,L4),  AC4(L4,L4),
     &          AC5(L4,L4),  AC6(L4,L4),  AC7(L4,L4),
     &          AMX1(L4,L4), AMX2(L4,L4), AMX3(L4,L4),
     &          AMX4(L4,L4), AMX5(L4,L4), AMX6(L4,L4),
     &          AMX9(L4,L4),AMX11(L4,L4),AMX12(L4,L4),
     &          AMY1(L4,L4), AMY2(L4,L4), AMY3(L4,L4),
     &          AMY4(L4,L4), AMY5(L4,L4), AMY6(L4,L4),
     &          AMY9(L4,L4),AMY11(L4,L4),AMY12(L4,L4),
     &          AE1(L4,L4),  AE2(L4,L4),  AE3(L4,L4),
     &          AE4(L4,L4),  AE5(L4,L4),  AE6(L4,L4),
     &          AE7(L4,L4),  AE8(L4,L4),  AE9(L4,L4),
     &          AE10(L4,L4), AE11(L4,L4), AE12(L4,L4),
     &          AE14(L4,L4), AE15(L4,L4),
     &          ASP1(L4,L4), ASP2(L4,L4), ASP3(L4,L4),
     &          ASP4(L4,L4), ASP5(L4,L4), ASP6(L4,L4),
     &          ASP7(L4,L4), ASP8(L4,L4), ASP9(L4,L4),
     &          APRS1(L4,L4)
      COMMON/LZERO/ RZ(L4),UZ(L4),VZ(L4),TZ(L4),
     &          SZ(L4),PZ(L4),WZ(L4)
C
      DIMENSION   ZRO(IG,IG),ZROX(IG,IG),ZROY(IG,IG),
     &            ZU(IG,IG),   ZV(IG,IG),  ZUV(IG,IG),
     &            ZTM(IG,IG),ZTMX(IG,IG),ZTMY(IG,IG),
     &            ZRU(IG,IG),  ZRV(IG,IG),ZRUX(IG,IG),
     &            ZRUY(IG,IG),
     &            ZRVX(IG,IG),ZRVY(IG,IG),ZUU1(IG,IG),
     &            ZUU2(IG,IG),
     &            ZUT(IG,IG),ZRTX(IG,IG),ZRTY(IG,IG),
     &            ZWDT(IG,IG),ZWSP(IG,IG), ZWT(IG,IG),
     &            ZRSX(IG,IG),ZRSY(IG,IG),ZUVS(IG,IG),
     &            ZTUV(IG,IG),ZTRX(IG,IG),ZTRY(IG,IG),
     &            ZTRO(IG,IG), ZTU(IG,IG), ZTV(IG,IG)
C
C
```

```
C-----(1) BLOCK ZERO : ZEROTH ORDER TERM EVALUATION
C
      DO 100 I=1,IG
      DO 100 J=1,IG
      PNUZ=0.0
      PXUZ=0.0
      PYUZ=0.0
      PNVZ=0.0
      PXVZ=0.0
      PYVZ=0.0
      PNRZ=0.0
      PXRZ=0.0
      PYRZ=0.0
      PNTZ=0.0
      PXTZ=0.0
      PYTZ=0.0
      PXSZ=0.0
      PYSZ=0.0
      PWDT=0.0
      PWSP=0.0
      PWT =0.0
C
      DO 110 M=1,L4
      PN=PIWN(M,I,J)
      PX=PIWX(M,I,J)
      PY=PIWY(M,I,J)
C
      PNUZ=PNUZ+PN*UZ(M)
C*    PXUZ=PXUZ+PX*UZ(M)
      PYUZ=PYUZ+PY*UZ(M)
      PNVZ=PNVZ+PN*VZ(M)
C*    PXVZ=PXVZ+PX*VZ(M)
      PYVZ=PYVZ+PY*VZ(M)
      PNRZ=PNRZ+PN*RZ(M)
C*    PXRZ=PXRZ+PX*RZ(M)
      PYRZ=PYRZ+PY*RZ(M)
      PNTZ=PNTZ+PN*TZ(M)
C*    PXTZ=PXTZ+PX*TZ(M)
      PYTZ=PYTZ+PY*TZ(M)
C*    PXSZ=PXSZ+PX*SZ(M)
      PYSZ=PYSZ+PY*SZ(M)
      PWDT=PWDT+PN*WZ(M)/TZ(M)**2
      PWT =PWT +PN*WZ(M)*TZ(M)
      IF (ABS(SZ(M)).LE.1.0E-04) THEN
          PWSP=PWSP
      ELSE
          PWSP =PWSP +PN*WZ(M)/SZ(M)
      ENDIF
C
 110  CONTINUE
C
      ZRO(I,J)=PNRZ
      ZROX(I,J)=PXRZ
```

```
            ZROY(I,J)=PYRZ
              ZU(I,J)=PNUZ
              ZV(I,J)=PNVZ
             ZUV(I,J)=PXUZ+PYVZ
             ZTM(I,J)=PNTZ
            ZTMX(I,J)=PXTZ
            ZTMY(I,J)=PYTZ
             ZRU(I,J)=PNRZ*PNUZ
             ZRV(I,J)=PNRZ*PNVZ
            ZRUX(I,J)=PNRZ*PXUZ
            ZRUY(I,J)=PNRZ*PYUZ
            ZRVX(I,J)=PNRZ*PXVZ
            ZRVY(I,J)=PNRZ*PYVZ
            ZUU1(I,J)=PNUZ*PXUZ+PNVZ*PYUZ
            ZUU2(I,J)=PNUZ*PXVZ+PNVZ*PYVZ
            ZRTX(I,J)=PNRZ*PXTZ
            ZRTY(I,J)=PNRZ*PYTZ
             ZUT(I,J)=PNUZ*PXTZ+PNVZ*PYTZ
            ZWDT(I,J)=PWDT
            ZWSP(I,J)=PWSP.
             ZWT(I,J)=PWT
            ZRSX(I,J)=PNRZ*PXSZ
            ZRSY(I,J)=PNRZ*PYSZ
            ZUVS(I,J)=PNUZ*PXSZ+PNVZ*PYSZ
             ZTU(I,J)=PNTZ*PNUZ
             ZTV(I,J)=PNTZ*PNVZ
            ZTUV(I,J)=PNTZ*(PXUZ+PYVZ)
            ZTRO(I,J)=1.0
C           ZTRO(I,J)=PNTZ*PNRZ
            ZTRX(I,J)=PNTZ*PXRZ
            ZTRY(I,J)=PNTZ*PYRZ
  100   CONTINUE
C
C-----(2) BLOCK INTEG : GAUSSIAN INTEGRATION
C
        DO 120 N=1,L4
        DO 120 M=1,L4
           SC2=0.0
           SC3=0.0
           SC4=0.0
           SC5=0.0
           SC6=0.0
           SC7=0.0
          SMX1=0.0
          SMX2=0.0
          SMX3=0.0
          SMX4=0.0
          SMX5=0.0
          SMX6=0.0
          SMX9=0.0
         SMX11=0.0
         SMX12=0.0
          SMY1=0.0
```

```fortran
      SMY2=0.0
      SMY3=0.0
      SMY4=0.0
      SMY5=0.0
      SMY6=0.0
      SMY9=0.0
     SMY11=0.0
     SMY12=0.0
       SE1=0.0
       SE2=0.0
       SE3=0.0
       SE4=0.0
       SE5=0.0
       SE6=0.0
       SE7=0.0
       SE8=0.0
       SE9=0.0
      SE10=0.0
      SE11=0.0
      SE12=0.0
      SE14=0.0
      SE15=0.0
      SSP1=0.0
      SSP2=0.0
      SSP3=0.0
      SSP4=0.0
      SSP5=0.0
      SSP6=0.0
      SSP7=0.0
      SSP9=0.0
     SPRS1=0.0
C
      DO 130 K=1,IG
      DO 130 L=1,IG
      WWDJ=W(K)*W(L)*DJ(K,L)
      PNN =PIWN(N,K,L)*PIWN(M,K,L)*WWDJ
      PNX =PIWN(N,K,L)*PIWX(M,K,L)*WWDJ
      PNY =PIWN(N,K,L)*PIWY(M,K,L)*WWDJ
C
       SC2= SC2+(PNX *ZU(K,L)+PNY*ZV(K,L))
       SC3= SC3+PNN *ZUV(K,L)
       SC4= SC4+PNN*ZROX(K,L)
       SC5= SC5+PNX *ZRO(K,L)
       SC6= SC6+PNN*ZROY(K,L)
       SC7= SC7+PNY *ZRO(K,L)
C
      SMX1=SMX1+PNN*ZUU1(K,L)
      SMX2=SMX2+PNX *ZTM(K,L)
      SMX3=SMX3+PNN*ZTMX(K,L)
      SMX4=SMX4+PNN *ZRO(K,L)
      SMX5=SMX5+(PNX*ZRU(K,L)+PNY*ZRV(K,L))
      SMX6=SMX6+PNN*ZRUX(K,L)
      SMX9=SMX9+PNN*ZRUY(K,L)
```

```
      SMX11-SMX11+PNN*ZROX(K,L)
      SMX12-SMX12+PNX *ZRO(K,L)
C
      SMY1-SMY1+PNN*ZUU2(K,L)
      SMY2-SMY2+PNY *ZTM(K,L)
      SMY3-SMY3+PNN*ZTMY(K,L)
      SMY4-SMY4+PNN *ZRO(K,L)
      SMY5-SMY5+(PNX*ZRU(K,L)+PNY*ZRV(K,L))
      SMY6-SMY6+PNN*ZRVX(K,L)
      SMY9-SMY9+PNN*ZRVY(K,L)
      SMY11-SMY11+PNN*ZROY(K,L)
      SMY12-SMY12+PNY *ZRO(K,L)
C
      SE1-  SE1+PNN *ZUT(K,L)
      SE2-  SE2+PNX *ZTU(K,L)+PNY*ZTV(K,L)
      SE3-  SE3+PNN*ZTUV(K,L)
      SE4-  SE4+PNN* ZWT(K,L)
      SE5-  SE5+PNN*ZRTX(K,L)
      SE6-  SE6+PNN*ZTRX(K,L)
      SE7-  SE7+PNX*ZTRO(K,L)
      SE8-  SE8+PNN*ZRTY(K,L)
      SE9-  SE9+PNN*ZTRY(K,L)
      SE10-SE10+PNY*ZTRO(K,L)
      SE11-SE11+PNN *ZRO(K,L)
      SE12-SE12+PNX *ZRU(K,L)+PNY*ZRV(K,L)
      SE14-SE14+PNN*ZWDT(K,L)
      SE15-SE15+PNN*ZWSP(K,L)
C
      SSP1-SSP1+PNN*ZUVS(K,L)
      SSP3-SSP3+PNN*ZRSX(K,L)
      SSP4-SSP4+PNN*ZRSY(K,L)
C
      SPRS1-SPRS1+PNN*ZTM(K,L)
C
 130  CONTINUE
C
      SSP2-SE4
      SSP5-SE14
      SSP6-SE11
      SSP7-SE12
      SSP9-SE15
C
C-----(3) BLOCK LTERM : LOCAL COEFFICIENT
C
      AC2(N,M)- SC2
      AC3(N,M)- SC3
      AC4(N,M)- SC4
      AC5(N,M)- SC5
      AC6(N,M)- SC6
      AC7(N,M)- SC7
C
      AMX1(N,M)-SMX1
      AMX2(N,M)-SMX2
```

```fortran
          AMX3(N,M)=SMX3
          AMX4(N,M)=SMX4
          AMX5(N,M)=SMX5
          AMX6(N,M)=SMX6
          AMX9(N,M)=SMX9
         AMX11(N,M)=SMX11
         AMX12(N,M)=SMX12
C
          AMY1(N,M)=SMY1
          AMY2(N,M)=SMY2
          AMY3(N,M)=SMY3
          AMY4(N,M)=SMY4
          AMY5(N,M)=SMY5
          AMY6(N,M)=SMY6
          AMY9(N,M)=SMY9
         AMY11(N,M)=SMY11
         AMY12(N,M)=SMY12
C
          AE1(N,M)= SE1
          AE2(N,M)= SE2
          AE3(N,M)= SE3
          AE4(N,M)= SE4
          AE5(N,M)= SE5
          AE6(N,M)= SE6
          AE7(N,M)= SE7
          AE8(N,M)= SE8
          AE9(N,M)= SE9
         AE10(N,M)=SE10
         AE11(N,M)=SE11
         AE12(N,M)=SE12
         AE14(N,M)=SE14
         AE15(N,M)=SE15
C
         ASP1(N,M)=SSP1
         ASP2(N,M)=SSP2
         ASP3(N,M)=SSP3
         ASP4(N,M)=SSP4
         ASP5(N,M)=SSP5
         ASP6(N,M)=SSP6
         ASP7(N,M)=SSP7
         ASP9(N,M)=SSP9
C
        APRS1(N,M)=SPRS1
C
  120   CONTINUE
C
        RETURN
        END
C
```

```
      SUBROUTINE LASS2(LE,OME,ITIME)
C
C     *************************************************
C     SUBROUTINE LASS2( LOCAL ASSEMBLING OF SOURCE TERM )
C     *************************************************
C
C
      PARAMETER NW=5,NL=11,L4=4,NV=6
      PARAMETER NE=(NW-1)*(NL-1),ND=NW*NL
      PARAMETER IG=4,NN=ND*NV
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMPLEX FST(NN)
      COMPLEX  RF,UF,VF,TF,SF,
     &         PNRF,PXRF,PYRF,  PNUF,PXUF,PYUF,
     &         PNVF,PXVF,PYVF,  PNTF,PXTF,PYTF,
     &         PNSF,PXSF,PYSF,
     &         FRO  ,FRUV ,FRU  ,FRV  ,F011X,F011Y,
     &         F101X,F101Y,F110X,F110Y,FRT1X,FRT1Y,
     &         FRT2X,FRT2Y,FPTM ,FRVT1,FRVT2,FRVT3,
     &         FRUS1,FRUS2,FRUS3,FPRSP,FPWT2,FPWRT,
     &         FPWST,FPWRS,FPWR2,FPWS2,FTUVR,
     &         CG ,XG ,YG ,EG ,SG ,PG ,
     &         ACG,AXG,AYG,AEG,ASG,APG
C
      COMMON/DJP/DJ(IG,IG),
     &           PIWX(L4,IG,IG),PIWY(L4,IG,IG),
     &           PIWN(L4,IG,IG),H(IG),W(IG)
      COMMON/LOCAL2/ ACG(L4),AXG(L4),AYG(L4),
     &               AEG(L4),ASG(L4),APG(L4)
      COMMON/DIMEN/IE(NE,L4),X(ND),Y(ND)
      COMMON/FIRST/FST
      COMMON/GIVEN/TC,TS,HO,EO,ES,QL,DEL1,
     &             DEL2,BE,GAM,PR,BM
C
      COMMON/LZERO/ RZ(L4),UZ(L4),VZ(L4),TZ(L4),
     &              SZ(L4),PZ(L4),WZ(L4)
      DIMENSION     RF(L4),UF(L4),VF(L4),
     &              TF(L4),SF(L4)
C
      DIMENSION     FRO(IG,IG),  FRUV(IG,IG),
     &              FRU(IG,IG),   FRV(IG,IG),
     &              F011X(IG,IG),F011Y(IG,IG),
     &              F101X(IG,IG),F101Y(IG,IG),
     &              F110X(IG,IG),F110Y(IG,IG),
     &              FRT1X(IG,IG),FRT1Y(IG,IG),
     &              FRT2X(IG,IG),FRT2Y(IG,IG),
     &               FPTM(IG,IG),FRVT1(IG,IG),
     &              FRVT2(IG,IG),FRVT3(IG,IG),
     &              FRUS1(IG,IG),FRUS2(IG,IG),
     &              FRUS3(IG,IG),FPRSP(IG,IG),
     &              FPWT2(IG,IG),FPWRT(IG,IG),
```

```
     &                    FPWST(IG,IG),FPWRS(IG,IG),
     &                    FPWR2(IG,IG),FPWS2(IG,IG),FTUVR(IG,IG)
C
C
C-----(1) BLOCK LOCAL : DEFINE LOCAL VARIABLES
C
      GMB2=0.0
      RGAM=(GAM-1.0)/GAM
      RGAM=0.0
      QGAM=1.0-2.0*RGAM
C
      DO 200 I=1,L4
            J=IE(LE,I)
C
      N1=J+ND
      N2=J+ND*2
      N3=J+ND*3
      N4=J+ND*4
      N5=J+ND*5
C
      RF(I)=FST(J)
      UF(I)=FST(N1)
      VF(I)=FST(N2)
      TF(I)=FST(N3)
      SF(I)=FST(N4)
C
      IF ( REAL(SF(I)).LE.1.0E-05) THEN
      SF(I)=0.0
      ENDIF
 200  CONTINUE
C
C-----(2) INITIALIZATION
C
      DO 210 I=1,IG
      DO 210 J=1,IG
C
      PNRZ=0.0
      PNUZ=0.0
      PXUZ=0.0
      PYUZ=0.0
      PNVZ=0.0
      PXVZ=0.0
      PYVZ=0.0
      PNTZ=0.0
      PXTZ=0.0
      PYTZ=0.0
      PNSZ=0.0
      PXSZ=0.0
      PYSZ=0.0
C
      PNUF=CMPLX(0.0,0.0)
      PXUF=CMPLX(0.0,0.0)
      PYUF=CMPLX(0.0,0.0)
```

```
          PNVF-CMPLX(0.0,0.0)
          PXVF-CMPLX(0.0,0.0)
          PYVF-CMPLX(0.0,0.0)
          PNRF-CMPLX(0.0,0.0)
          PXRF-CMPLX(0.0,0.0)
          PYRF-CMPLX(0.0,0.0)
          PNTF-CMPLX(0.0,0.0)
          PXTF-CMPLX(0.0,0.0)
          PYTF-CMPLX(0.0,0.0)
          PNSF-CMPLX(0.0,0.0)
          PXSF-CMPLX(0.0,0.0)
          PYSF-CMPLX(0.0,0.0)
C
          PWT2-0.0
          PWRT-0.0
          PWST-0.0
          PWRS-0.0
          PWR2-0.0
          PWS2-0.0
C
C-----(3) BLOCK SUM : SUMMATION
C
          DO 220 M-1,L4
          PN-PIWN(M,I,J)
          PX-PIWX(M,I,J)
          PY-PIWY(M,I,J)
C
          PNRZ-PNRZ+PN*RZ(M)
          PNUZ-PNUZ+PN*UZ(M)
C*        PXUZ-PXUZ+PX*UZ(M)
          PYUZ-PYUZ+PY*UZ(M)
          PNVZ-PNVZ+PN*VZ(M)
C*        PXVZ-PXVZ+PX*VZ(M)
          PYVZ-PYVZ+PY*VZ(M)
          PNTZ-PNTZ+PN*TZ(M)
C*        PXTZ-PXTZ+PX*TZ(M)
          PYTZ-PYTZ+PY*TZ(M)
C*        PXSZ-PXSZ+PX*SZ(M)
          PYSZ-PYSZ+PY*SZ(M)
C
          PNUF-PNUF+PN*UF(M)
          PXUF-PXUF+PX*UF(M)
          PYUF-PYUF+PY*UF(M)
          PNVF-PNVF+PN*VF(M)
          PXVF-PXVF+PX*VF(M)
          PYVF-PYVF+PY*VF(M)
          PNRF-PNRF+PN*RF(M)
          PXRF-PXRF+PX*RF(M)
          PYRF-PYRF+PY*RF(M)
          PNTF-PNTF+PN*TF(M)
          PXTF-PXTF+PX*TF(M)
          PYTF-PYTF+PY*TF(M)
          PNSF-PNSF+PN*SF(M)
```

```fortran
      PXSF=PXSF+PX*SF(M)
      PYSF=PYSF+PY*SF(M)
C
      PWT2=PWT2+PN*(EO*WZ(M)*(0.5*EO/TZ(M)-1.0))/TZ(M)**3
      PWRT=PWRT+PN*(2.0*EO*WZ(M)/TZ(M))
      PWR2=PWR2+PN*WZ(M)/RZ(M)**2
      IF (ABS(SZ(M)).LE.1.0E-04) THEN
      PWST=PWST
      PWRS=PWRS
      PWS2=PWS2
      ELSE
      PWST=PWST+PN*(2.0*EO*WZ(M)/(SZ(M)*TZ(M)**2))
      PWRS=PWRS+PN*(4.0*WZ(M)/(RZ(M)*SZ(M)))
      PWS2=PWS2+PN*WZ(M)/SZ(M)**2
      ENDIF
C
  220 CONTINUE
CC
      IF(ITIME.EQ.1) THEN
C
C-----(4) BLOCK TI : TIME INDEPENDENT SOURCE
        FRO(I,J)=CONJG(PXRF)*PNUF+CONJG(PYRF)*PNVF
       FRUV(I,J)=CONJG(PNRF)*(PXUF+PYVF)
        FRU(I,J)=CONJG(PNRF)*PNUF
        FRV(I,J)=CONJG(PNRF)*PNVF
      F011X(I,J)=PNRZ*(CONJG(PNUF)*PXUF+CONJG(PNVF)*PYUF)
      F011Y(I,J)=PNRZ*(CONJG(PNUF)*PXVF+CONJG(PNVF)*PYVF)
      F101X(I,J)=CONJG(PNRF)*(PNUZ*PXUF+PNVZ*PYUF)
      F101Y(I,J)=CONJG(PNRF)*(PNUZ*PXVF+PNVZ*PYVF)
      F110X(I,J)=CONJG(PNRF)*(PNUF*PXUZ+PNVF*PYUZ)
      F110Y(I,J)=CONJG(PNRF)*(PNUF*PXVZ+PNVF*PYVZ)
      FRT1X(I,J)=CONJG(PXRF)*PNTF
      FRT1Y(I,J)=CONJG(PYRF)*PNTF
      FRT2X(I,J)=CONJG(PNRF)*PXTF
      FRT2Y(I,J)=CONJG(PNRF)*PYTF
       FPTM(I,J)=CONJG(PNRF)*PNTF
      FRVT1(I,J)=PNRZ*(CONJG(PNUF)*PXTF+CONJG(PNVF)*PYTF)
      FRVT2(I,J)=CONJG(PNRF)*(PNUZ*PXTF+PNVZ*PYTF)
      FRVT3(I,J)=CONJG(PNRF)*(PNUF*PXTZ+PNVF*PYTZ)
      FRUS1(I,J)=PNRZ*(CONJG(PNUF)*PXSF+CONJG(PNVF)*PYSF)
      FRUS2(I,J)=CONJG(PNRF)*(PNUZ*PXSF+PNVZ*PYSF)
      FRUS3(I,J)=CONJG(PNRF)*(PNUF*PXSZ+PNVF*PYSZ)
      FPRSP(I,J)=CONJG(PNRF)*PNSF
      FPWT2(I,J)=PWT2*CONJG(PNTF)*PNTF
      FPWRT(I,J)=PWRT*CONJG(PNRF)*PNTF
      FPWST(I,J)=PWST*CONJG(PNSF)*PNTF
      FPWRS(I,J)=PWRS*CONJG(PNRF)*PNSF
      FPWR2(I,J)=PWR2*CONJG(PNRF)*PNRF
      FPWS2(I,J)=PWS2*CONJG(PNSF)*PNSF
      FTUVR(I,J)=PNTZ*(CONJG(PXRF)*PNUF+CONJG(PYRF)*PNVF)
     &          +PNTZ*(CONJG(PNRF)*PXUF+CONJG(PNRF)*PYVF)
C
C
```

```fortran
      ELSE
C
C-----(5) BLOCK TD : TIME DEPENDENT SOURCE
CC
        FRO(I,J)=PXRF*PNUF+PYRF*PNVF
       FRUV(I,J)=PNRF*(PXUF+PYVF)
        FRU(I,J)=PNRF*PNUF
        FRV(I,J)=PNRF*PNVF
      F011X(I,J)=PNRZ*(PNUF*PXUF+PNVF*PYUF)
      F011Y(I,J)=PNRZ*(PNUF*PXVF+PNVF*PYVF)
      F101X(I,J)=PNRF*(PNUZ*PXUF+PNVZ*PYUF)
      F101Y(I,J)=PNRF*(PNUZ*PXVF+PNVZ*PYVF)
      F110X(I,J)=PNRF*(PNUF*PXUZ+PNVF*PYUZ)
      F110Y(I,J)=PNRF*(PNUF*PXVZ+PNVF*PYVZ)
      FRT1X(I,J)=PXRF*PNTF
      FRT1Y(I,J)=PYRF*PNTF
      FRT2X(I,J)=PNRF*PXTF
      FRT2Y(I,J)=PNRF*PYTF
       FPTM(I,J)=PNRF*PNTF
      FRVT1(I,J)=PNRZ*(PNUF*PXTF+PNVF*PYTF)
      FRVT2(I,J)=PNRF*(PNUZ*PXTF+PNVZ*PYTF)
      FRVT3(I,J)=PNRF*(PNUF*PXTZ+PNVF*PYTZ)
      FRUS1(I,J)=PNRZ*(PNUF*PXSF+PNVF*PYSF)
      FRUS2(I,J)=PNRF*(PNUZ*PXSF+PNVZ*PYSF)
      FRUS3(I,J)=PNRF*(PNUF*PXSZ+PNVF*PYSZ)
      FPRSP(I,J)=PNRF*PNSF
      FPWT2(I,J)=PWT2*PNTF*PNTF
      FPWRT(I,J)=PWRT*PNRF*PNTF
      FPWST(I,J)=PWST*PNSF*PNTF
      FPWRS(I,J)=PWRS*PNRF*PNSF
      FPWR2(I,J)=PWR2*PNRF*PNRF
      FPWS2(I,J)=PWS2*PNSF*PNSF
      FTUVR(I,J)=PNTZ*(PXRF*PNUF+PYRF*PNVF)
     &           +PNTZ*(PNRF*PXUF+PNRF*PYVF)
C
      ENDIF
C
 210  CONTINUE
C
C-----(6) BLOCK LSOURCE
C
      DO 230 N=1,L4
      CG=CMPLX(0.0,0.0)
      XG=CMPLX(0.0,0.0)
      YG=CMPLX(0.0,0.0)
      EG=CMPLX(0.0,0.0)
      SG=CMPLX(0.0,0.0)
      PG=CMPLX(0.0,0.0)
C
      DO 240 I=1,IG
      DO 240 J=1,IG
      WWDJ=W(I)*W(J)*DJ(I,J)
        PN=PIWN(N,I,J)*WWDJ
```

```fortran
C
      CG=CG+PN*(  FRO(I,J)+ FRUV(I,J))
      XG=XG+PN*(F011X(I,J)+F101X(I,J)+F110X(I,J)+
     &     GMB2*(FRT1X(I,J)+FRT2X(I,J))+CMPLX(0.0,1.0)*OME*FRU(I,J))
      YG=YG+PN*(F011Y(I,J)+F101Y(I,J)+F110Y(I,J)+
     &     GMB2*(FRT1Y(I,J)+FRT2Y(I,J))+CMPLX(0.0,1.0)*OME*FRV(I,J))
      EG=EG+PN*(FRVT1(I,J)+FRVT2(I,J)+FRVT3(I,J)-
     &       HO*(FPWT2(I,J)+FPWRT(I,J)+FPWST(I,J)+
     &           FPWRS(I,J)+FPWR2(I,J)+FPWS2(I,J))+
     &     QGAM*CMPLX(0.0,1.0)*OME*FPTM(I,J)+ RGAM*FTUVR(I,J))
      SG=SG+PN*(FRUS1(I,J)+FRUS2(I,J)+FRUS3(I,J)+
     &          FPWT2(I,J)+FPWRT(I,J)+FPWST(I,J)+
     &          FPWRS(I,J)+FPWR2(I,J)+FPWS2(I,J)+
     &       OME*CMPLX(0.0,1.0)* FPRSP(I,J))
      PG=PG+PN*FPTM(I,J)
C
 240  CONTINUE
C
      ACG(N)=CG
      AXG(N)=XG
      AYG(N)=YG
      AEG(N)=EG
      ASG(N)=SG
      APG(N)=PG
 230  CONTINUE
C
      RETURN
      END
C
```

```fortran
      SUBROUTINE GAUSS(IG,H,W)
C
C
C     **********************************************
C     SUBROUTINE GAUSS ( GAUSSIAN QUADRATURE INTEGRATION )
C     **********************************************
C     IG: NO. OF GAUSS.POINT
C     H,W: ABSCISSAE AND WEIGHTING COEFF. OF GAUSS. POINTS
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION H(IG), W(IG)
C
      IF    (IG.EQ.1) THEN
      H(1)=0.0D+00
      W(1)=2.0D+00
      ELSEIF(IG.EQ.2) THEN
      H(1)=0.577350269189626D+00
      H(2)=-H(1)
      W(1)=1.0D+00
      W(2)=1.0D+00
      ELSEIF(IG.EQ.3) THEN
      H(1)=0.774596669241483D+00
      H(2)=0.0D+00
      H(3) =-H(1)
      W(1)=0.555555555555555D+00
      W(2)=0.888888888888888D+00
      W(3)=W(1)
      ELSEIF(IG.EQ.4) THEN
      H(1)=0.861136311594053D+00
      H(2)=0.339981043584856D+00
      H(3)=-H(2)
      H(4)=-H(1)
      W(1)=0.347854845137454D+00
      W(2)=0.652145154862546D+00
      W(3)=W(2)
      W(4)=W(1)
      ELSEIF(IG.EQ.5) THEN
      H(1)=0.906179845938664D+00
      H(2)=0.538469310105683D+00
      H(3)=0.0D+00
      H(4)=-H(2)
      H(5)=-H(1)
      W(1)=0.236926885056189D+00
      W(2)=0.478628670499366D+00
      W(3)=0.568888888888889D+00
      W(4)=W(2)
      W(5)=W(1)
      ELSEIF(IG.EQ.6) THEN
      H(1)=0.932469514203152D+00
      H(2)=0.661209386466265D+00
      H(3)=0.238619186083197D+00
      H(4)=-H(3)
      H(5)=-H(2)
```

```fortran
          H(6)=-H(1)
          W(1)=0.171324492379170D+00
          W(2)=0.360761573048139D+00
          W(3)=0.467913934572691D+00
          W(4)=W(3)
          W(5)=W(2)
          W(6)=W(1)
C
          ENDIF
C
          RETURN
          END
C
```

```fortran
      SUBROUTINE BNDRY1 (A,OME,HT)
C
C     ************************************************************
C     SUBROUTINE BNDRY1( BOUNDARY CONDITION APPLICATION )
C     ************************************************************
C
C
      PARAMETER NW=5,NL=11,NV=6
      PARAMETER ND=NW*NL,NE=(NW-1)*(NL-1),NB=NW*4
      PARAMETER NN=ND*NV,NQ=NN+NB,MC=NQ+1
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMPLEX A(NQ,MC),B(NB,NN),C(NN)
      COMPLEX FST(NN)
C
      COMMON/FIRST/FST
C
C
      CALL BCEQN(B,OME,1,1,HT)
C
      DO 100 J=1,NN
      DO 100 I=1,NB
            K=I+NN
      A(K,J)=B(I,J)
      A(J,K)=A(K,J)
 100  CONTINUE
C
      CALL DIRICH(A,1)
      CALL  NEUMN(C,1)
C
      DO 110 I=1,NN
      A(I,MC)=A(I,MC)+C(I)
 110  CONTINUE
C
      RETURN
      END
C
```

```fortran
      SUBROUTINE BNDRY2 (A,OME,ITIME,HT)
C
C     ************************************************
C     SUBROUTINE BNDRY2( BOUNDARY CONDITION APPLICATION )
C     ************************************************
C
C
      PARAMETER NW=5,NL=11,L4=4,NV=6
      PARAMETER ND=NW*NL,NE=(NW-1)*(NL-1),NB=NW*4
      PARAMETER NN=ND*NV,NQ=NN+NB,MC=NQ+1
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMPLEX A(NQ,MC),B(NB,NN),C(NN)
      COMPLEX FST(NN)
C
      COMMON/FIRST/FST
C
C
      CALL BCEQN(B,OME,2,ITIME,HT)
C
      DO 200 I=1,NB
      DO 200 J=1,NN
            K=I+NN
      A(K,J)=B(I,J)
      A(J,K)=A(K,J)
  200 CONTINUE
C
      CALL DIRICH(A,2)
      CALL  NEUMN(C,2)
C
      DO 210 I=1,NN
      A(I,MC)=A(I,MC)+C(I)
  210 CONTINUE
C
      RETURN
      END
C
```

```
      SUBROUTINE BCEQN(B,OME,IORDER,ITIME,HT)
C
C     ***************************************************
C     SUBROUTINE BCEQN (PREPARATION FOR BOUNDARY RELATIONS)
C     ***************************************************
C
C
      PARAMETER NW=5,NL=11,L4=4,NV=6
      PARAMETER ND=NW*NL,NE=(NW-1)*(NL-1),NB=NW*4
      PARAMETER NN=ND*NV,NQ=NN+NB,MC=NQ+1
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMPLEX B(NB,NN),FST(NN),RAMDA1,RAMDA2,C3,C4,C5,
     &        TERM1,TERM2,CG9
      COMMON/ZERO/RZG(ND),UZG(ND),VZG(ND),TZG(ND),
     &            SZG(ND),PZG(ND),WZG(ND)
      COMMON/GIVEN/TC,TS,HO,EO,ES,QL,DEL1,
     &            DEL2,BE,GAM,PR,BM
      COMMON/FIRST/FST
C
C
      ND2=ND*2
      ND3=ND*3
      ND4=ND*4
      ND5=ND*5
C
      K1W=NW
      K2W=2*NW
      K3W=3*NW
      K4W=4*NW
      RGAM=(GAM-1.0)/GAM
C
      HOME=1.0/OME/HT
      TS1=1.0/TS
      TS2=1.0/TS**2
      C1=ES*TS2
      ET1=-(ES*TS1+1.0)
      C2=(TS-TC)/DEL1
      AAT2=SQRT(+0.5+0.5*SQRT(1.0+(4.0*OME*BE*DEL1)**2))
      AAT3=0.5*(4.0*OME*BE*DEL1)/AAT2
C
      RAMDA1=0.5/DEL1 * DCMPLX(1.0+AAT2,AAT3)
      TERM1=(AAT2+1.0+C1*C2*AAT3/(OME*BE))/
     &      (2.0*DEL1*DEL2)+C1*QL
      TERM2=(AAT3-(AAT2+1.0)*C1*C2/(OME*BE))/(2.0*DEL1*DEL2)+
     &      C1*C2/(OME*BE*DEL1*DEL2)
C*    TERM1=RAMDA1*(1.0-DCMPLX(0.0,1.0)*C1*C2/OME/BE)/DEL2
C*    &      +DCMPLX(0.0,1.0)*C1*C2/OME/BE/DEL1/DEL2
C*    &      +C1*QL
C*    CG9=0.0
      TERMR=TERM1*HT+1.0
      TERMI=TERM2*HT
```

```
C
C-----(1) FIRST ORDER B.C.
C
      IF(IORDER.EQ.1) THEN
C
      DO 100 J=1,NW
      I=1+(J-1)*NL
      K= J*NL
      TZI1=1.0/TZG(I)
      TZI2=TZI1/TZG(I)
C
C-----BLOCK TU (TEMPERATURE UPPER CONDITION)
C
      B(K1W+J,I+ND3)  =CMPLX(1.00,-HOME)
      B(K1W+J,1+I+ND3)=CMPLX(0.00,+HOME)
      B(K1W+J,I+ND5)  =CMPLX(-RGAM,0.0)
C
C-----BLOCK YL (SPECIES LOWER CONDITION)
C
      ENFF=+C1*(SZG(K-1)-SZG(K))
      B(K2W+J,K+ND3) =CMPLX(ENFF,0.0)
      B(K2W+J,K+ND4) =CMPLX(HT+1.0,0.0)
      B(K2W+J,K+ND4-1)= CMPLX(-1.0,0.0)
      B(K2W+J, MC)   =CMPLX(0.0,0.0)
C
C-----BLOCK VL (Y-VELOCITY LOWER CONDITION)
C
      B(K3W+J,K    )=CMPLX(TS**2,0.0)
      B(K3W+J,K+ND2)=CMPLX(1.0,0.0)
      B(K3W+J,K+ND3)=CMPLX(-ES/TS,0.0)
C
C-----BLOCK TL (TEMPERATURE LOWER CONDITION)
C
      B(J,K+ND3)  =CMPLX(TERMR,TERMI)
      B(J,K+ND3-1)=CMPLX(-1.0,0.0)
      B(J,   MC)  =CMPLX(0.0,0.0)
C*    B(J,K+ND3)  =TERM1*HT+1.0
C*    B(J,K+ND3-1)=CMPLX(-1.0,0.0)
C*    B(J,   MC)  =HT*CG9
C
 100  CONTINUE
C
C-----(2) SECOND ORDER TIME INDEP. B.C.
C
      ELSEIF(IORDER.NE.1.AND.ITIME.EQ.1) THEN
C
C
      OMBE=1.0/(OME*BE)
      OMB2=0.5*OMBE
      OBDD=0.5/(OME*BE*DEL1*DEL2)
      OME2=2.0*OME
      HOME2=1.0/OME2/HT
      AAT22=SQRT(+0.5+0.5*SQRT(1.0+(4.0*OME2*BE*DEL1)**2))
```

```fortran
      AAT23=0.5*(4.0*OME2*BE*DEL1)/AAT22
C
      TERM21=(AAT22+1.0+C1*C2*AAT23/(OME2*BE))/
     &      (2.0*DEL1*DEL2)+C1*QL
      TERM22=(AAT23-(AAT22+1.0)*C1*C2/(OME2*BE))/(2.0*DEL1*DEL2)+
     &      C1*C2/(OME2*BE*DEL1*DEL2)
C*    TERM1=1.0/DEL1/DEL2+C1*C2/DEL2+C1*QL
C*    CG9=RAMDA1*C1*FST(JB)*CONJG(FST(JB))*0.5*
C*    &    (1.0-DCMPLX(0.0,1.0)*C1*C2/OME/BE)
C*    &    *(1.0/(DEL1*RAMDA1)-1.0)/DEL2/(DEL1*RAMDA1-1.0)
C*    &    -C2*C3/DEL2-C3*QL
      TERM2R=TERM21*HT+1.0
      TERM2I=TERM22*HT
C
C
      DO 210 J=1,NW
      I=1+(J-1)*NL
      K= J*NL
      TZI1=1.0/TZG(I)
      TZI2=TZI1/TZG(I)
C
C-----BLOCK TU (2ND TEMP. UPPER CONDITION)
C
      B(K1W+J,I+ND3)=CMPLX(1.00,+0.0)
      B(K1W+J,1+I+ND3)=CMPLX(-1.00,+0.0)
      B(K1W+J,MC)=0.0
C
      JA=ND2+K
      JB=ND3+K
      JC=ND4+K
      ENFF=+C1*(SZG(K-1)-SZG(K))
C
C-----BLOCK YL (2ND SPECIES LOWER CONDITION)
C
      B(K2W+J,K+ND3)= CMPLX(ENFF,0.0)
      B(K2W+J,K+ND4)= CMPLX(HT+1.0,0.0)
      B(K2W+J,K+ND4-1)= CMPLX(-1.0,0.0)
      B(K2W+J, MC)    =-C1*0.5*CONJG(FST(JB))*(FST(JC-1)-FST(JC))
     &               +(0.5*C1+1.0/TS)*ENFF*FST(JB)*CONJG(FST(JB))*0.5
C
C-----BLOCK VL (2ND Y-VEL LOWER CONDITION)
C
      PRSL2=1.0
      GAX22=TS1*PRSL2
      B(K3W+J,K)=CMPLX(1.0,0.0)
      B(K3W+J,K+ND3)=CMPLX(TS2,0.0)
      B(K3W+J,MC)=DCMPLX(GAX22,0.0)-0.5*CONJG(FST(K))*FST(JB)*TS1
C
C-----BLOCK TL (2ND TEMP. LOWER CONDITION)
C
      RAMDA2=0.5/DEL1 * DCMPLX(1.0+AAT22,AAT23)
        C3=C1*(0.5*C1-TS1)*FST(JB)*CONJG(FST(JB))*0.5
        CCC2=-C2*(C1*FST(JB))**2*0.5/(OME*BE)**2/DEL1
```

```fortran
              CCC3=(RAMDA2-RAMDA1)*C1*FST(JB)*CONJG(FST(JB))*RAMDA1*0.5
     &             *(1.0-CMPLX(0.0,1.0)*C1*C2/(OME*BE))
     &             /(DEL1*RAMDA1**2-RAMDA1-CMPLX(0.0,1.0)*OME2*BE)
          TERM2R=HT*(1.0/DEL1/DEL2+C1/DEL2+C1*QL)+1.0
          TERM2I=0.0
C
          B(J,K+ND3)= CMPLX(TERM2R,TERM2I)
          B(J,K+ND3-1)= CMPLX(-1.0,0.0)
          B(J,   MC)= HT*( TC/DEL1/DEL2+C1*FST(JB)*RAMDA1
     &                *0.5*CONJG(FST(JB))
     &                   *(1.0-CMPLX(0.0,1.0)*C1*C2/OME/BE)
     &                   /(DEL2*(DEL1*RAMDA1-1.0))
     &                   *(1.0/(DEL1*RAMDA1)-1.0)
     &                   -C3*C2/DEL2-CMPLX(0.0,1.0)*C2
     &             *C1**2*0.5*FST(JB)*CONJG(FST(JB))*2.0*OBDD-C3*QL)
C*    B(J,K+ND3)=TERM1*HT+1.0
C*    B(J,K+ND3-1)= CMPLX(-1.0,0.0)
C*    B(J,MC)      =-HT*CG9
C
 210  CONTINUE
C
C-----(3) SECOND ORDER TIME DEP. B.C.
C
      ELSE
C
       OME2=2.0*OME
      HOME2=1.0/OME2/HT
      AAT22=SQRT(+0.5+0.5*SQRT(1.0+(4.0*OME2*BE*DEL1)**2))
      AAT23=0.5*(4.0*OME2*BE*DEL1)/AAT22
C
      DO 220 J=1,NW
      I=1+(J-1)*NL
      K= J*NL
      TZI1=1.0/TZG(I)
      TZI2=TZI1/TZG(I)
C
C-----BLOCK TU (2ND TEMP. UPPER CONDITION)
C
      B(K1W+J,I+ND5)=CMPLX(-RGAM,0.0)
      B(K1W+J,I+ND3)=CMPLX(1.00,+0.0)
      B(K1W+J,1+I+ND3)=CMPLX(-1.00,+0.0)
      B(K1W+J,MC)=-RGAM/GAM*DCMPLX(0.0,1.0)*OME*0.5
C
      JA=ND2+K
      JB=ND3+K
      JC=ND4+K
      ENFF=+C1*(SZG(K-1)-SZG(K))
C
C-----BLOCK YL (2ND SPECIES LOWER CONDITION)
C
      B(K2W+J,K+ND3)= CMPLX(ENFF,0.0)
      B(K2W+J,K+ND4)= CMPLX(HT+1.0,0.0)
      B(K2W+J,K+ND4-1)= CMPLX(-1.0,0.0)
```

```fortran
      B(K2W+J, MC)     =-C1*0.5*FST(JB)*(FST(JC-1)-FST(JC))
     &               +(0.5*C1+1.0/TS)*ENFF*FST(JB)*FST(JB)*0.5
C
C-----BLOCK VL (2ND Y-VEL LOWER CONDITION)
C
      B(K3W+J,K)=CMPLX(TS**2,0.0)
      B(K3W+J,K+ND2)=CMPLX(1.0,0.0)
      B(K3W+J,K+ND3)=CMPLX(-ES/TS,0.0)
      B(K3W+J,MC)=C1*(0.5*ES/TS-1.0)*FST(JB)**2*0.5
     &               +(FST(JB)-FST(JA))*FST(K)*TS*0.5
C
C-----BLOCK TL (2ND TEMP. LOWER CONDITION)
C
      RAMDA2=0.5/DEL1 * DCMPLX(1.0+AAT22,AAT23)
      C3=C1*(0.5*C1-TS1)*FST(JB)*FST(JB)*0.5
      C4=-C2**(C1*FST(JB))**2*0.5
     &    /(2.0*OME**2*BE**2*DEL1)
      C5=RAMDA1*(RAMDA2-RAMDA1)*C1*FST(JB)*
     &   FST(JB)*(1.0-DCMPLX(0.0,1.0)*C2*C1/OME/BE)*0.5
     &   /(DEL1*RAMDA1**2-RAMDA1-DCMPLX(0.0,1.0)*2.0*OME*BE)
C
      TERM1=RAMDA2*(1.0-DCMPLX(0.0,1.0)*C1*C2
     &       *0.5/OME/BE)/DEL2 + DCMPLX(0.0,1.0)
     &       *0.5*C1*C2/OME/BE/DEL1/DEL2
     &       +C1*QL
      CG9=C2*C3*0.5/OME/BE/DEL2*(1.0/DEL1-RAMDA2)
     &    +(RAMDA2*C4+C5-C4/DEL1)/DEL2-C3*QL
C
      B(J,K+ND3)= TERM1*HT+1.0
      B(J,K+ND3-1)= CMPLX(-1.0,0.0)
      B(J,MC)      = HT*CG9
C
 220  CONTINUE
C
      ENDIF
C
      RETURN
      END
C
```

```
      SUBROUTINE DIRICH(A,IORDER)
C
C     ***********************************************
C     SUBROUTINE DIRICH (PREPARATION FOR DIRICHLET CONDITION)
C     ***********************************************
C
C
      PARAMETER NW=5,NL=11,NV=6
      PARAMETER ND=NW*NL,NE=(NW-1)*(NL-1),NB=NW*4
      PARAMETER NN=ND*NV,NQ=NN+NB,MC=NQ+1
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMPLEX A(NQ,MC),DIR0,DIR1
C
      ND2=2*ND
      ND3=3*ND
      ND4=4*ND
      ND5=5*ND
C
C-----(1) FIRST ORDER DIRICHLET
C     (A) UPPER B.C.
C
      DIR1= CMPLX(1.0,0.0)
      DIR0= CMPLX(0.0,0.0)
C
C
      IF(IORDER.EQ.1) THEN
      DO 100 J=1,ND,NL
      K1=J+ND
      K2=J+ND2
      K3=J+ND3
      K4=J+ND4
      K5=J+ND5
C
      DO 110 K=1,NQ
      A(K,MC)=A(K,MC)-A(K,K4)*DIR0
      A(K,MC)=A(K,MC)-A(K,K5)*DIR1
      A(K4,K)=DIR0
      A(K5,K)=DIR0
      A(K,K4)=DIR0
      A(K,K5)=DIR0
  110 CONTINUE
C
      A(K4,K4)=DIR1
      A(K5,K5)=DIR1
      A(K4,MC)=DIR0
      A(K5,MC)=DIR1
  100 CONTINUE
C
C     (B) LOWER B.C.
C
      DO 120 J=NL,ND,NL
```

```fortran
      K1=J+ND
C
      DO 130 K=1,NQ
      A(K,MC)=A(K,MC)-A(K,K1)*DIR0
      A(K1,K)=DIR0
      A(K,K1)=DIR0
  130 CONTINUE
      A(K1,K1)=DIR1
      A(K1,MC)=DIR0
  120 CONTINUE
C
C-----(2) SECOND ORDER DIRICHLET
C
      ELSE
C     (A) UPPER B.C.
C
      DO 200 J=1,ND,NL
      K1=J+ND
      K2=J+ND2
      K3=J+ND3
      K4=J+ND4
      K5=J+ND5
C
      DO 210 K=1,NN
      A(K,MC)=A(K,MC)-A(K,K4)*DIR0
      A(K4,K)=DIR0
      A(K,K4)=DIR0
C     A(K,K5)=DIR0
  210 CONTINUE
C
      A(K4,K4)=DIR1
      A(K4,MC)=DIR0
  200 CONTINUE
C
C     (B) LOWER B.C.
C
      DO 220 J=NL,ND,NL
      K1=J+ND
      K5=J+ND5
C
      DO 230 K=1,NN
      A(K,MC)=A(K,MC)-A(K,K1)*DIR0
      A(K1,K)=DIR0
      A(K,K1)=DIR0
  230 CONTINUE
      A(K1,K1)=DIR1
      A(K1.MC)=DIR0
  220 CONTINUE
C
      ENDIF
C
      RETURN
      END
```

```fortran
C
      SUBROUTINE NEUMN(C,IORDER)
C
C     ************************************************************
C     SUBROUTINE NEUMN (PREPARATION FOR NEUMANN CONDITION)
C     ************************************************************
C
      PARAMETER NW=5,NL=11,L4=4,NV=6
      PARAMETER ND=NW*NL,NE=(NW-1)*(NL-1),NB=NW*4
      PARAMETER NN=ND*NV,NQ=NN+NB,MC=NQ+1
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMPLEX C(NN)
C
C-----(6) APPLY NEUMANN B.C
C
C        DEFINE BOUNDARY NODES
      IF(IORDER.EQ.1) THEN
      DO 100 I=1,NN
      C(I)=0.0
 100  CONTINUE
      ELSE
      DO 200 I=1,NN
      C(I)=0.0
 200  CONTINUE
      ENDIF
C      GRADIENT CALCULATION
C
C  ALL GRADIENTS ARE ASSUMED TO BE ZERO IN THIS CASE
C
      RETURN
      END
C
```

```fortran
      SUBROUTINE OUTPUT(IORDER,FST)
C
C     ***************************************************
C     SUBROUTINE OUTPUT( OUTPUT DATA )
C     ***************************************************
C      PRESS : PRESSURE BY DENSITY*TEMPERATURE
C      ADMIT : ADMITTANCE(=VELOCITY/PRESSURE)
C      RESP  : RESPONSE FUNCTION(=BURNING RATE/PRESS)
C
C
      PARAMETER NW=5,NL=11,L4=4,NV=6
      PARAMETER ND=NW*NL,NE=(NW-1)*(NL-1),NB=NW*4
      PARAMETER NN=ND*NV,NQ=NN+NB,MC=NQ+1
C
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMPLEX   FST(NN),PRESS(ND),ADMIT(ND)
C
      COMMON/ZERO/RZG(ND),UZG(ND),VZG(ND),TZG(ND),
     &            SZG(ND),PZG(ND),WZG(ND)
      COMMON/GIVEN/TC,TS,HO,EO,ES,QL,DEL1,
     &             DEL2,BE,GAM,PR,BM
C
      DIMENSION RESP(ND)
C
      ND2=2*ND
      ND3=3*ND
      ND4=4*ND
      ND5=5*ND
      ND6=6*ND
C
C-----(1) SOLUTION OUTPUT
C
      IF(IORDER.EQ.1) THEN
      WRITE(6,900)
 900  FORMAT(1H1,5X,'FIRST ORDER SOLUTION'//)
      ELSE
      WRITE(6,910)
 910  FORMAT(1H1,5X,'SECOND ORDER SOLUTION'//)
      ENDIF
      WRITE(6,920)
 920  FORMAT(5X,'NODE',7X,'DENSITY(RO)',14X,'X-VELOCITY(U)',
     &       13X,'Y-VELOCITY(V)'//)
      DO 100 I=1,ND
      WRITE(6,930) I, FST(I),FST(I+ND),FST(I+ND2)
 100  CONTINUE
C
      WRITE(6,940)
 940  FORMAT(5X,'NODE',5X,'TEMPERRATURE(T)',14X,'SPECIES(Y)',
     &       16X,'PRESSURE(P)'//)
      DO 200 I=1,ND
      WRITE(6,930) I, FST(I+ND3),FST(I+ND4),FST(I+ND5)
```

```
  200    CONTINUE
C
  930    FORMAT(3X,I5,2X,3('(',E10.3,',',E10.3,')',3X))
C
C-----(2) CALCULATE PRESSURE, ADMITTAJCE AND RESPONSE FUNCION
C
       DO 300 I-1,ND
       M-ND2+I
       K-ND3+I
       PRESS(I)-RZG(I)*FST(K)+TZG(I)*FST(I)
       ADMIT(I)-FST(M)/VZG(I)/PRESS(I)
       RESP(I)-DBLE( (RZG(I)*FST(M)+FST(I)*VZG(I))/PRESS(I) )
C      RESP(I)-REAL( (RZG(I)*FST(M)+FST(I)*VZG(I))/PRESS(I) )
  300  CONTINUE
C
       IF(IORDER.EQ.1) THEN
       WRITE(6,950)
  950  FORMAT(5X,'FIRST ORDER SOLUTION'//)
       ELSE
       WRITE(6,960)
  960  FORMAT(5X,'SECOND ORDER SOLUTION'//)
       ENDIF
       WRITE(6,970)
  970  FORMAT(5X,'NODE',7X,'PRESSURE(P)',12X,'ADMITTANCE(AD)',
      &        12X,'RESPONSE(RE)'//)
       DO 400 I-1,ND
       WRITE(6,980) I, PRESS(I),ADMIT(I),RESP(I)
  400  CONTINUE
  980  FORMAT(3X,I5,2X,2('(',E10.3,',',E10.3,')',3X),E10.3)
C
       RETURN
       END
C
```

```fortran
      SUBROUTINE EIGEN (A)
C
C     ****************************************************
C     SUBROUTINE EIGEN( BOUNDARY CONDITION APPLICATION )
C     ****************************************************
      VIRTUAL ANF,BNF,CZZ1,
C
      PARAMETER NW=5,NL=11,L4=4,NV=6
      PARAMETER ND=NW*NL,NE=(NW-1)*(NL-1),NB=NW*4
      PARAMETER NN=ND*NV,NQ=NN+NB,MC=NQ+1
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMPLEX EIGAV(NN),EIGBV(NN),EIGV(NN),ANF(NN,NN),BNF(NN,NN),
     &         CZZ1(NN,NN)
      COMPLEX A(NQ,MC)
      INTEGER ITER(NN)
C
      DO 100 J=1,NN
      DO 100 I=1,NN
      ANF(I,J)=REAL(A(I,J))
      BNF(I,J)=AIMAG(A(I,J))
 100  CONTINUE
C
C
      CALL CONVRT(NN,ANF,NN,BNF,NN,CZZ1,NN)
      CALL SOLVE(NN,ANF,NN,BNF,NN,CZZ1,NN,ITER,EIGAV,EIGBV)
C
C
      DO 200 IEG=1,NN
      IF(EIGBV(IEG).EQ.0.0) GO TO 200
      EIGV(IEG)=EIGAV(IEG)/EIGBV(IEG)
 200  CONTINUE
C
      DO 300 I=1,NN
      WRITE(6,900)  EIGV(I)
      WRITE(12,*)   EIGV(I)
 900  FORMAT(5X, '( ',E10.3,',',E10.3,' )')
 300  CONTINUE
C
      STOP
      END
C
C
C
```

```
C
C      ********************************************************
C      SUBROUTINE CONVRT (PREPARATION FOR EIGENVALUE PROBLEM)
C      ********************************************************
       SUBROUTINE CONVRT
CC
      $( N, A, NA, B, NB, X, NX )
C
           IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
      COMPLEX A(NA,N)
      COMPLEX B(NB,N)
      COMPLEX W
      COMPLEX X(NX,N)
      COMPLEX Y
      COMPLEX Z
C
      REAL C
      REAL D
C
      INTEGER I
      INTEGER II
      INTEGER IMJ
      INTEGER IM1
      INTEGER IP1
      INTEGER J
      INTEGER JM2
      INTEGER JP1
      INTEGER K
      INTEGER N
      INTEGER NA
      INTEGER NB
      INTEGER NM1
      INTEGER NM2
      INTEGER NX
C
      LUWT = 6
      NM1 = N - 1
      DO   80 I=1,NM1
      D = 0.0
      IP1 = I + 1
      DO   10 K=IP1,N
      Y = B(K,I)
      C = ABS(REAL(Y)) + ABS(AIMAG(Y))
      IF( C.LE.D )     GO TO    9
      D =     C
      II = K
    9 CONTINUE
   10 CONTINUE
      IF( D.EQ.0.0 )      GO TO   78
      Y = B(I,I)
      IF( D.LE.ABS(REAL(Y)) + ABS(AIMAG(Y)) )      GO TO   40
```

```fortran
      DO 20 J=1,N
      Y = A(I,J)
      A(I,J) = A(II,J)
      A(II,J) = Y
   20 CONTINUE
      DO 30  J=I,N
      Y = B(I,J)
      B(I,J) = B(II,J)
      B(II,J) = Y
   30 CONTINUE
   40 CONTINUE
      DO  70 J=IP1,N
      Y = B(J,I)/B(I,I)
      IF( REAL(Y).EQ.0.0  .AND.  AIMAG(Y).EQ.0.0 )      GO TO  68
      DO  50 K=1,N
      A(J,K) = A(J,K) - Y*A(I,K)
   50 CONTINUE
      DO  60 K=IP1,N
      B(J,K) = B(J,K) - Y*B(I,K)
   60 CONTINUE
   68 CONTINUE
   70 CONTINUE
      B(IP1,I) = CMPLX(0.0,0.0)
   78 CONTINUE
   80 CONTINUE
C
      DO 100 I=1,N
      DO  90 J=1,N
      X(I,J) = CMPLX(0.0,0.0)
   90 CONTINUE
      X(I,I) = CMPLX(1.0,0.0)
  100 CONTINUE
C
      NM2 = N - 2
      IF( NM2.LT.1 )      GO TO 270
      DO 260 J=1,NM2
      JM2 = NM1 - J
      JP1 = J + 1
      DO 250 II=1,JM2
      I = N + 1 - II
      IM1 = I - 1
      IMJ = I - J
C
      W = A(I,J)
      Z = A(IM1,J)
      IF( ABS(REAL(W)) + ABS(AIMAG(W)).LE.
     $    ABS(REAL(Z)) + ABS(AIMAG(Z)) )      GO TO 140
      DO 120 K=J,N
      Y = A(I,K)
      A(I,K) = A(IM1,K)
      A(IM1,K) = Y
  120 CONTINUE
      DO 130 K=IM1,N
```

```fortran
      Y = B(I,K)
      B(I,K) = B(IM1,K)
      B(IM1,K) = Y
  130 CONTINUE
  140 CONTINUE
C
      Z = A(I,J)
      IF( REAL(Z).EQ.0.0  .AND.  AIMAG(Z).EQ.0.0 )     GO TO 170
      Y = Z/A(IM1,J)
      DO 150 K=JP1,N
      A(I,K) = A(I,K) - Y*A(IM1,K)
  150 CONTINUE
      DO 160 K=IM1,N
      B(I,K) = B(I,K) - Y*B(IM1,K)
  160 CONTINUE
  170 CONTINUE
C
      W = B(I,IM1)
      Z = B(I,I)
      IF( ABS(REAL(W)) + ABS(AIMAG(W)).LE.
     $    ABS(REAL(Z)) + ABS(AIMAG(Z)) )     GO TO 210
      DO 180 K=1,I
      Y = B(K,I)
      B(K,I) = B(K,IM1)
      B(K,IM1) = Y
  180 CONTINUE
      DO 190 K=1,N
      Y = A(K,I)
      A(K,I) = A(K,IM1)
      A(K,IM1) = Y
  190 CONTINUE
      DO 200 K=IMJ,N
      Y = X(K,I)
      X(K,I) = X(K,IM1)
      X(K,IM1) = Y
  200 CONTINUE
  210 CONTINUE
C
      Z = B(I,IM1)
      IF( REAL(Z).EQ.0.0  .AND.  AIMAG(Z).EQ.0.0 )     GO TO 249
      Y = Z/B(I,I)
      DO 220 K=1,IM1
      B(K,IM1) = B(K,IM1) - Y*B(K,I)
  220 CONTINUE
      B(I,IM1) = CMPLX(0.0,0.0)
      DO 230 K=1,N
      A(K,IM1) = A(K,IM1) - Y*A(K,I)
  230 CONTINUE
      DO 240 K=IMJ,N
      X(K,IM1) = X(K,IM1) - Y*X(K,I)
  240 CONTINUE
  249 CONTINUE
C
```

```fortran
  250 CONTINUE
      A(JP1+1,J) = CMPLX(0.0,0.0)
  260 CONTINUE
  270 CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE SOLVE
     $( N, A, NA, B, NB, X, NX, ITER, EIGA, EIGB )
C
         IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
      COMPLEX S
      COMPLEX W
      COMPLEX Y
      COMPLEX Z
      COMPLEX A(NA,N)
      COMPLEX B(NB,N)
      COMPLEX X(NX,N)
      COMPLEX EIGA(N)
      COMPLEX EIGB(N)
      COMPLEX ANNM1
      COMPLEX ALFM
      COMPLEX BETM
      COMPLEX D
      COMPLEX SL
      COMPLEX DEN
      COMPLEX NUM
      COMPLEX ANM1M1
C
      REAL D0
      REAL D1
      REAL D2
      REAL E0
      REAL E1
      REAL    C
      REAL EPSA
      REAL EPSB
      REAL SS
      REAL R
      REAL ANORM
      REAL BNORM
      REAL ANI
      REAL BNI
C
      INTEGER ITER(N)
C
      NN = N
      ANORM = 0.0
      BNORM = 0.0
      DO 30 I=1,N
      ANI = 0.0
      IF( I.EQ.1 )     GO TO 10
      Y = A(I,I-1)
      ANI = ANI + ABS(REAL(Y)) + ABS(AIMAG(Y))
   10 CONTINUE
      BNI = 0.0
      DO 20 J=I,N
      ANI = ANI + ABS(REAL(A(I,J))) + ABS(AIMAG(A(I,J)))
```

```
      BNI = BNI + ABS(REAL(B(I,J))) + ABS(AIMAG(B(I,J)))
   20 CONTINUE
      IF( ANI.GT.ANORM )     ANORM = ANI
      IF( BNI.GT.BNORM )     BNORM = BNI
   30 CONTINUE
C
      IF( ANORM.EQ.0.0 )     ANORM = 1.0
      IF( BNORM.EQ.0.0 )     BNORM = 1.0
      EPSA = ANORM
      EPSB = BNORM
   40 CONTINUE
      EPSA = EPSA/2.0
      EPSB = EPSB/2.0
      C = ANORM + EPSA
      IF( C.GT.ANORM )     GO TO 40
      IF( N.LE.1 )     GO TO 320
   50 CONTINUE
      ITS = 0
      NM1 = NN - 1
   60 CONTINUE
      D2 = ABS(REAL(A(NN,NN))) + ABS(AIMAG(A(NN,NN)))
      DO 70 LB=2,NN
      L = NN + 2 - LB
      SS = D2
      Y = A(L-1,L-1)
      D2 = ABS(REAL(Y)) + ABS(AIMAG(Y))
      SS = SS + D2
      Y = A(L,L-1)
      R = SS + ABS(REAL(Y)) + ABS(AIMAG(Y))
      IF( R.EQ.SS )     GO TO 80
   70 CONTINUE
      L = 1
   80 CONTINUE
      IF( L.EQ.NN )     GO TO 320
      IF( ITS.LT.30 )     GO TO 90
      ITER(NN) = -1
      IF( ABS(REAL(A(NN,NM1))) + ABS(AIMAG(A(NN,NM1))).GT.
     $0.8*ABS(REAL(ANNM1)) + 0.8*ABS(AIMAG(ANNM1))     )     GO TO 999
   90 CONTINUE
      IF( ITS.EQ.10  .OR.  ITS.EQ.20 )     GO TO 110
C
      ANNM1 = A(NN,NM1)
      ANM1M1 = A(NM1,NM1)
      S = A(NN,NN)*B(NM1,NM1) - ANNM1*B(NM1,NN)
      W = ANNM1*B(NN,NN)*
     $    (A(NM1,NN)*B(NM1,NM1) - ANM1M1*B(NM1,NN))
      Y = (ANM1M1*B(NN,NN) - S)/2.0
      Z = CSQRT(Y*Y + W)
      IF( REAL(Z).EQ.0.0  .AND.  AIMAG(Z).EQ.0.0 )     GO TO 100
      DO = REAL(Y/Z)
      IF( DO.LT.0.0 )     Z = -Z
  100 CONTINUE
      DEN = (Y + Z)*B(NM1,NM1)*B(NN,NN)
```

```
      IF(  REAL(DEN).EQ.0.0    .AND.
     $     AIMAG(DEN).EQ.0.0 )
     $DEN = CMPLX(EPSA,0.0)
      NUM = (Y + Z)*S - W
      GO TO 120
  110 CONTINUE
      Y = A(NM1,NN-2)
      NUM = CMPLX(ABS(REAL(ANNM1)) + ABS(AIMAG(ANNM1)),
     $             ABS(REAL(Y))     + ABS(AIMAG(Y))     )
      DEN = CMPLX(1.0,0.0)
  120 CONTINUE
      IF( NN.EQ.L+1 )     GO TO 140
      D1 = ABS(REAL(A(NN,NN))) + ABS(AIMAG(A(NN,NN)))
      D2 = ABS(REAL(A(NM1,NM1))) + ABS(AIMAG(A(NM1,NM1)))
      E1 = ABS(REAL(ANNM1)) + ABS(AIMAG(ANNM1))
      NL = NN - (L + 1)
      DO 130 MB=1,NL
      M = NN - MB
      E0 = E1
      Y = A(M,M-1)
      E1 = ABS(REAL(Y)) + ABS(AIMAG(Y))
      D0 = D1
      D1 = D2
      Y = A(M-1,M-1)
      D2 = ABS(REAL(Y)) + ABS(AIMAG(Y))
      Y = A(M,M)*DEN - B(M,M)*NUM
      D0 = (D0 + D1 + D2)*( ABS(REAL(Y)) + ABS(AIMAG(Y)) )
      E0 = E0*E1*( ABS(REAL(DEN)) + ABS(AIMAG(DEN)) ) + D0
      IF( E0.EQ.D0 )     GO TO 150
  130 CONTINUE
  140 CONTINUE
      M = L
  150 CONTINUE
      ITS = ITS + 1
      W = A(M,M)*DEN - B(M,M)*NUM
      Z = A(M+1,M)*DEN
      D1 = ABS(REAL(Z)) + ABS(AIMAG(Z))
      D2 = ABS(REAL(W)) + ABS(AIMAG(W))
      LOR1 = 1
      NNORN = N
      DO 310 I=M,NM1
      J = I + 1
      IF( I.EQ.M )     GO TO 170
      W = A(I,I-1)
      Z = A(J,I-1)
      D1 = ABS(REAL(Z)) + ABS(AIMAG(Z))
      D2 = ABS(REAL(W)) + ABS(AIMAG(W))
      IF( D1.EQ.0.0 )     GO TO 60
  170 CONTINUE
      IF( D2.GT.D1 )     GO TO 190
      DO 180 K=I,NNORN
      Y = A(I,K)
      A(I,K) = A(J,K)
```

```fortran
     &      A(J,K) = Y
            Y = B(I,K)
            B(I,K) = B(J,K)
            B(J,K) = Y
  180 CONTINUE
            IF( I.GT.M )      A(I,I-1) = A(J,I-1)
            IF( D2.EQ.0.0 )      GO TO 220
            Y = CMPLX( REAL(W)/D1, AIMAG(W)/D1 )/
     $         CMPLX( REAL(Z)/D1, AIMAG(Z)/D1 )
            GO TO 200
  190 CONTINUE
            Y = CMPLX( REAL(Z)/D2, AIMAG(Z)/D2 )/
     $         CMPLX( REAL(W)/D2, AIMAG(W)/D2 )
  200 CONTINUE
            DO 210 K=I,NNORN
            A(J,K) = A(J,K) - Y*A(I,K)
            B(J,K) = B(J,K) - Y*B(I,K)
  210 CONTINUE
  220 CONTINUE
            IF( I.GT.M )      A(J,I-1) = CMPLX(0.0,0.0)
            Z = B(J,I)
            W = B(J,J)
            D1 = ABS(REAL(Z)) + ABS(AIMAG(Z))
            D2 = ABS(REAL(W)) + ABS(AIMAG(W))
            IF( D1.EQ.0.0 )      GO TO 60
            IF( D2.GT.D1 )      GO TO 270
            DO 230 K=LOR1,J
            Y = A(K,J)
            A(K,J) = A(K,I)
            A(K,I) = Y
            Y = B(K,J)
            B(K,J) = B(K,I)
            B(K,I) = Y
  230 CONTINUE
            IF( I.EQ.NM1 )      GO TO 240
            Y = A(J+1,J)
            A(J+1,J) = A(J+1,I)
            A(J+1,I) = Y
  240 CONTINUE
            DO 250 K=1,N
            Y = X(K,J)
            X(K,J) = X(K,I)
            X(K,I) = Y
  250 CONTINUE
            B(J,I) = CMPLX(0.0,0.0)
            IF( D2.EQ.0.0 )      GO TO 310
            Z = CMPLX( REAL(W)/D1, AIMAG(W)/D1 )/
     $         CMPLX( REAL(Z)/D1, AIMAG(Z)/D1 )
            GO TO 280
  270 CONTINUE
            Z = CMPLX( REAL(Z)/D2, AIMAG(Z)/D2 )/
     $         CMPLX( REAL(W)/D2, AIMAG(W)/D2 )
  280 CONTINUE
```

```fortran
      DO 290 K-LOR1,J
      A(K,I) - A(K,I) - Z*A(K,J)
      B(K,I) - B(K,I) - Z*B(K,J)
  290 CONTINUE
      B(J,I) - CMPLX(0.0,0.0)
      IF( I.LT.NM1 )     A(I+2,I) - A(I+2,I) - Z*A(I+2,J)
      DO 300 K-1,N
      X(K,I) - X(K,I) - Z*X(K,J)
  300 CONTINUE
  310 CONTINUE
      GO TO 60
  320 CONTINUE
      EIGA(NN) - A(NN,NN)
      EIGB(NN) - B(NN,NN)
      IF( NN.EQ.1 )     GO TO 330
      ITER(NN) - ITS
      NN - NM1
      IF( NN.GT.1 )     GO TO 50
      ITER(1) - 0
      GO TO 320
  330 CONTINUE
      M - N
  340 CONTINUE
      ALFM - A(M,M)
      BETM - B(M,M)
      B(M,M) - CMPLX(1.0,0.0)
      L - M - 1
      IF( L.EQ.0 )     GO TO 370
  350 CONTINUE
      L1 - L + 1
      SL - CMPLX(0.0,0.0)
      DO 360 J-L1,M
      SL - SL + B(J,M)*(BETM*A(L,J) - ALFM*B(L,J))
  360 CONTINUE
      Y - BETM*A(L,L) - ALFM*B(L,L)
      IF(  REAL(Y).EQ.0.0    .AND.
     $     AIMAG(Y).EQ.0.0 )
     $Y - CMPLX( (EPSA+EPSB)/2.0, 0.0 )
      B(L,M) - -SL/Y
      L - L - 1
  370 CONTINUE
      IF( L.GT.0 )     GO TO 350
      M - M - 1
      IF( M.GT.0 )     GO TO 340
      M - N
  380 CONTINUE
      DO 400 I-1,N
      S - CMPLX(0.0,0.0)
      DO 390 J-1,M
      S - S + X(I,J)*B(J,M)
  390 CONTINUE
      X(I,M) - S
  400 CONTINUE
```

```fortran
      M = M - 1
      IF( M.GT.0 )     GO TO 380
      M = N
410 CONTINUE
      SS = 0.0
      DO 420 I=1,N
      R = ABS(REAL(X(I,M))) + ABS(AIMAG(X(I,M)))
      IF( R.LT.SS )    GO TO 418
      SS = R
      D = X(I,M)
418 CONTINUE
420 CONTINUE
      IF( SS.EQ.0.0 )     GO TO 440
      DO 430 I=1,N
      X(I,M) = X(I,M)/D
430 CONTINUE
440 CONTINUE
      M = M - 1
      IF( M.GT.0 )     GO TO 410
999 CONTINUE
      RETURN
      END
```