4

STABILITY CHARACTERIZATION OF ADVANCED INJECTORS

Design Guide, Volume 2, Revision A
Operation of the Computer Program

Report 20672-P3D February 1971

AEROJET LIQUID ROCKET COMPANY
A DIVISION OF AEROJET-GENERAL
SACRAMENTO, CALIFORNIA

ERRATA

Design Guide for Stable H_2/O_2 Combustors
Report 20672-P2D dtd. May 1970
Volume I: Design Application

Revise values in Table 1 -- "Chamber Response as a Function of Chamber Type" (page 23) to read:

Type	($N_{chambers}$) (Fundamental and Higher Order Modes)
Cylindrical	0.70 to 1.50
Annular	1.80 to 2.60
Conical	0.90 to 1.50
Rectangular	2.0 to 2.80

ERRATA

STABILITY CHARACTERIZATION OF ADVANCED INJECTORS

Design Guide, Volume 2, Revision A
Operation of the Computer Program

Report 20672-P3D February 1971

STABILITY CHARACTERIZATION OF ADVANCED INJECTORS

DESIGN GUIDE

Volume 2, Operation of the Computer Program

Revision A

Report 20672-P3D

Prepared by:

Aerojet-General Corporation
Liquid Rocket Division

Contract NAS 8-20672

Phase III, Final Report

Prepared for

Marshall Space Flight Center
Huntsville, Alabama

Prepared by

R. C. Waugh

R. C. Waugh

Approved by

W. B. Bakken

W. B. Bakken

Approved by

J. M. McBride

J. M. McBride

Approved by

N. E. Van Huff

N. E. VanHuff

FOREWORD

This volume is submitted as partial fulfillment of contract provisions of Contract NAS 8-20672 and meets the requirements of Supplemental Agreement No. 13 of that contract. The NASA project engineer was Mr. R. J. Richmond.

Volume I of this report is written specifically for the designer with the intent of providing combustion stability information which may be applied directly during a program's design phase. This volume (Volume II), on the other hand, is to provide the tools by which the analyst may characterize the combustion stability of a system. The combustion model which provided the basis for this analytical technique is the Sensitive Time Lag theory first developed by Dr. L. H. Crocco and later extended by Dr. F. H. Reardon. This volume provides description of a computer program, written in FORTRAN V, which may be utilized in characterizing a variety of combustion systems. The program appearing in this edition has been updated to include a high Mach number analysis, hence the original edition of Vol. II is superceded by this revision.

The work on this edition was conducted by the Engine Components Department, Aerojet Liquid Rocket Company, with Dr. N. E. Van Huff, program manager; Mr. J. M. McBride project manager; and Mr. W. W. Howard and Mr. W. B. Bakken, project engineers.

Special acknowledgement is given to Mr. R. C. Waugh for his contribution in the development of analytical models, Mr. D. P. Dudley for programing and conversion of the computer program, Mr. R. K. Turner for the reduction of combustion stability theory into design criteria and analysis and correlations of test data, and Mr. R. J. Wondra and Mr. N. B. Davidson for preparation and editing the material in this report.

CONTENTS

Page

I. Problem

II. Programing

III. Deck Setup

APPENDICES

TABLES

FIGURES

I. PROBLEM

N71-20416

A. ABSTRACT

This program solves the combustion instability problem using the sensitive time lag theory. This amounts basically to solving the non-homogeneous Helmholtz equation for the condition of neutral stability. The non-homogeneous terms in the Helmholtz equation account for the mean flow and combustion effects. The program considers longitudinal and transverse modes of oscillation. It includes effects of non-uniform injection, velocity effects, and non-linear combustion response.

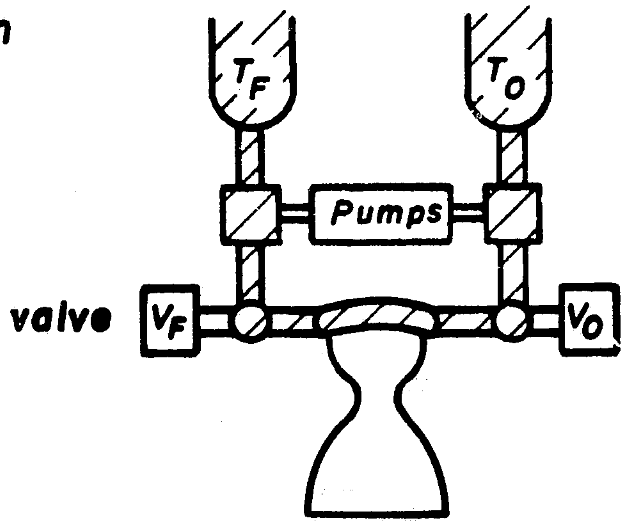
The program is divided into several subprograms which can either be run together or separately. These subprograms include analyses of nozzle admittance, concentrated combustion, and distributed combustion with a transverse mode -- as well as those which calculate describing functions for combustion response and nonuniform injection parameters.

B. TECHNICAL DESCRIPTION

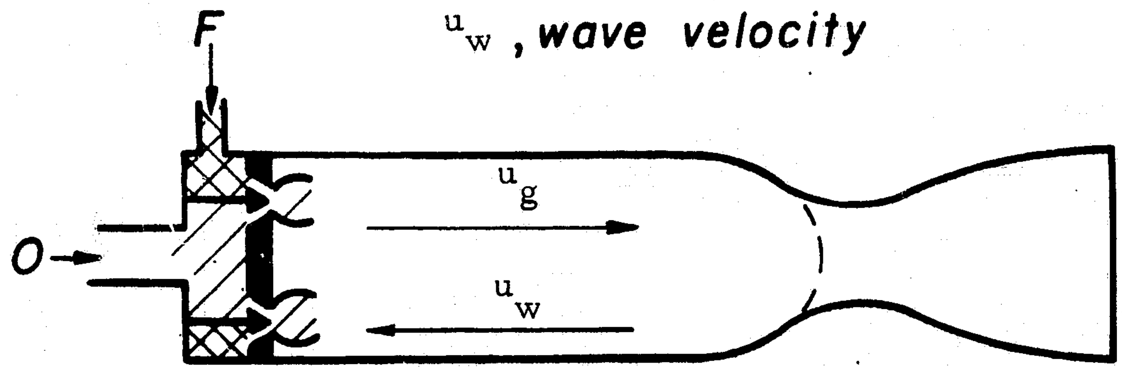
1. The Nature of Combustion Instability

It is well known that the processes occurring within a liquid rocket combustion chamber are never entirely smooth. Even when the mean operating conditions are constant, fluctuations around these mean values occur in all of the quantities that characterize the flow. The nature of the fluctuations can vary widely from one combustor to another and in a single combustor for different operating conditions. If the fluctuations are random and of small amplitude, this unsteadiness is referred to as "combustion noise". With random fluctuations of large amplitude, the operation of the rocket is said to be "rough", and the functioning of the system of which the rocket is a part may be impaired.

Type	*Characteristic coupling*
1. Chugging (low frequency)	*(1) feed system* *(2) combustion chamber*



2. Entropy wave	*(1) injector (mixture ratio)* *(2) chamber* ; u_g *, gas velocity* ; u_w *, wave velocity*



3. Hi-frequency	*(1) combustion process* *(2) chamber geometry (wave travel time)*

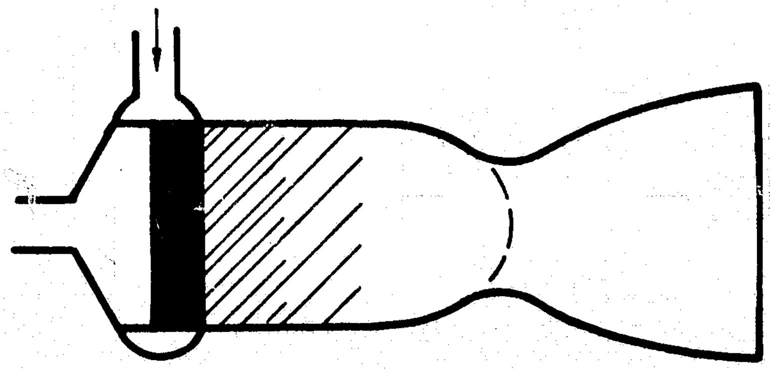


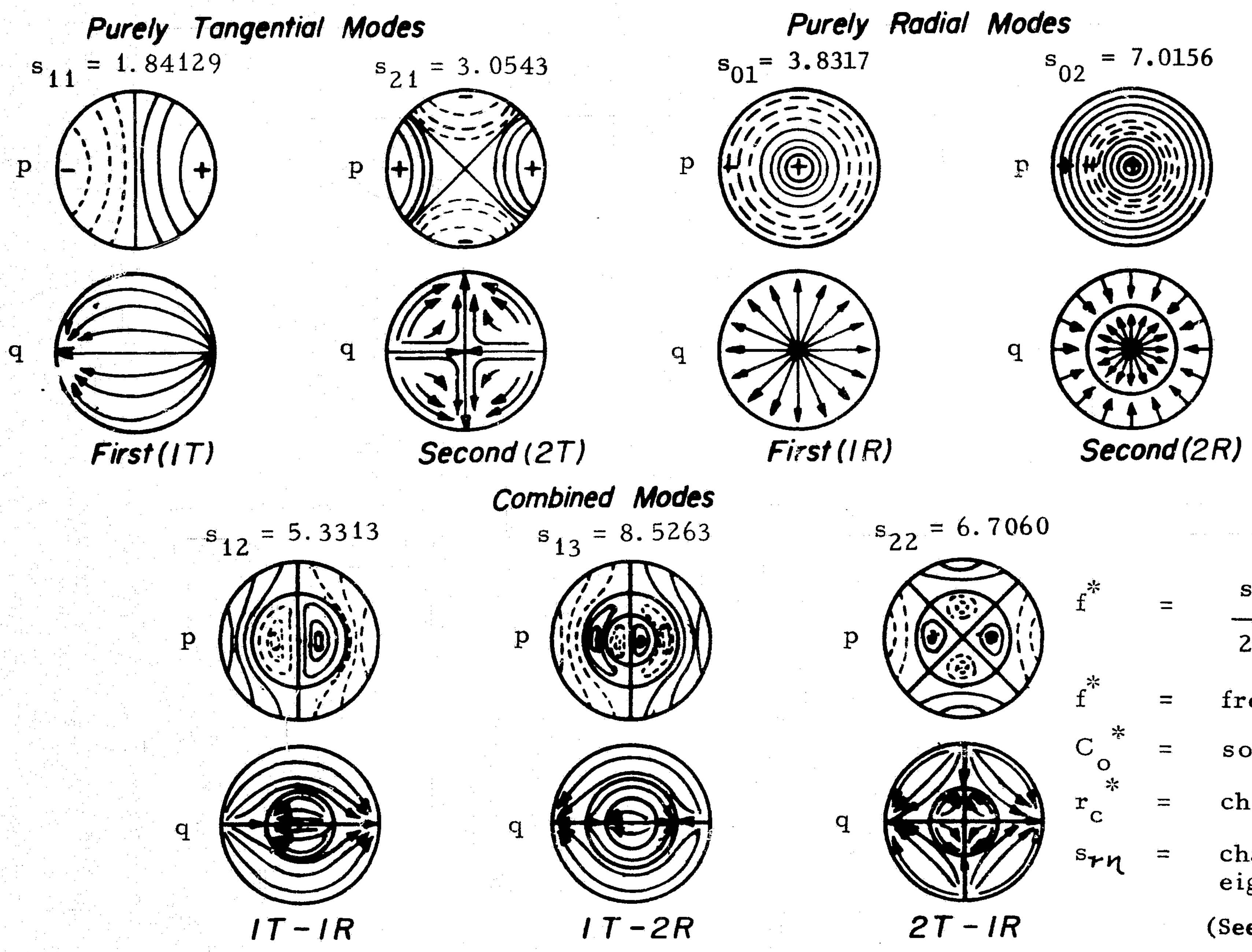
Figure 1 -- Liquid Propellant Combustion Instability

I, B, Technical Description (cont.)

Much more serious than rough operation is the problem of combustion instability, also termed unstable combustion, oscillatory combustion, or resonant combustion. Whereas rough combustion refers to random fluctuations, combustion instability consists of organized oscillations that are maintained and amplified by the combustion process itself. The various types of combustion instability can be classified roughly into three categories: low frequency, intermediate frequency, and high frequency. However, the classification is not based simply on frequency alone. Just as electrical and mechanical systems respond to specific frequencies depending on the type of coupling, so also liquid rocket systems exhibit representative frequency and amplitude patterns.

The basic coupling mechanisms for the three general types of combustion instability found in liquid propellant rocket engines are illustrated in Figure 1. For the low frequency ("chugging") type, interaction between the propellant feed system and the combustion chamber places the frequency generally less than 200 Hertz. The coupling is effected by the oscillating propellant feed rates. In the case of intermediate frequency combustion instability (sometimes referred to as entropy wave instability), the injector characteristics (especially the internal injector manifolding and orifice impedances) account for part of the interaction, with the mean gas flow and pressure wave propagation in the combustion chamber completing the process. Typical frequencies are in the several hundred Hertz range.

In this report, attention will be focused on the third type of combustion instability, namely, high frequency instability. This type depends upon a coupling between the combustion processes and flow oscillations in the combustion chamber. Such coupling requires no input from the feed system, although it is possible for the feed system to have an effect when the combustion chamber is large and acoustic frequencies are reduced to several hundred cycles per second. Normally, the frequencies to be expected are in the thousand Hertz range for most current engines.



$$f^* = \frac{s_{\tau\eta}\ C_o^{*}\ ^{12}}{2\pi\ r_c^{*}}$$

f^* = frequency

C_o^* = sound speed

r_c^* = chamber radius

$s_{\tau\eta}$ = characteristic eigen value

(See Table I)

Figure 2 -- Pressure, p, and Velocity, q, Characteristics of Several Transverse Modes

TABLE I

$s_{\nu\eta}$ FOR A CYLINDRICAL CHAMBER

For transverse modes the frequency of oscillation is given by:

$$f^* = \frac{s_{\nu\eta}\, C_o^* \times 12}{2\,\pi\, r_c^*}$$

Mode	ν	η	$s_{\nu\eta}$
1st radial	0	1	3.8317
2nd radial	0	2	7.0156
3rd radial	0	3	10.1734
1st tangential	1	1	1.8413
2nd tangential	2	1	3.0543
3rd tangential	3	1	4.2012
4th tangential	4	1	5.3175
5th tangential	5	1	6.4154
6th tangential	6	1	7.5013
1T, 1R combined	1	2	5.3313
2T, 1R combined	2	2	6.7060
3T, 1R combined	3	2	8.0151
4T, 1R combined	4	2	9.2825
1T, 2R combined	1	3	8.5263

For a longitudinal mode the frequency can be estimated by:

$$f^* = \frac{C_o^* \times 12}{2\,(L_c^* + 2/3\, L_N^*)}\,(1 - M_m^2)$$

where: L_N^* = length of nozzle in inches.

M_m = average Mach number in nozzle-chamber combination

The combustion chamber geometry is an important factor in high frequency combustion instability because the possible frequencies depend on the internal geometry. Since it is effectively closed at one end by the injector and has a choked-flow exhaust nozzle at the other end, the chamber acts acoustically much as a double closed-end cavity. For instance, for purely longitudinal modes, an approximate point of effective reflection in the nozzle can be determined theoretically; between that point and the injector face standing patterns of acoustic waves can be established. Similarly in the transverse plane, tangential modes of the spinning or standing types, as well as radial modes, may be established. Frequencies correspond approximately to those of the acoustic modes, either the fundamental or higher harmonics. Modes containing combinations of tangential, radial and longitudinal oscillations may also exist, each characterized by its own frequency. Pressure and velocity patterns for several transverse modes are illustrated in Figure 2. However, despite the similarity of the modes and the closeness of frequencies, the continuous generation of gases produces effects that do not exist in a closed chamber. In a closed chamber, the only source of damping originates from the friction on the walls. This source of damping is active of course also in the combustion chamber, but it plays a very modest role compared to other, more powerful sources of damping. Indeed, the very existence of the nozzle produces damping in the case of pure or combined longitudinal modes because the reflection of waves from the convergent (subsonic) portion of the nozzle departs from that of the ideal closed end. For purely transverse waves this source of damping is missing, and actually is replaced by a slight source of amplification.

The most important source of damping, however, is related to the process of gas generation itself, and consists of two parts of approximately equal importance. The most obvious comes from the fact that, from the conditions of the steady propellant injection flow (steady, of course, only if the feed system perturbations are disregarded) the combustion gases must have acquired, at the moment of generation, the perturbed momentum corresponding to the oscillatory flow. The acquisition of this momentum demands a certain work which must be absorbed from the system. The fact that momentum exchanges due to the drag of the droplets can take place prior to the moment of generation can only add additional damping.

The other, more subtle source of damping comes from the fact that at the moment of generation the volume of the propellant must change from its practically negligible liquid volume, to the full volume of the burned gases. To this change of volume corresponds a certain "pumping work" proportional to the local instantaneous pressure. Hence, more work must be absorbed from the system when the pressure is higher, and less when it is lower, thus providing an effective damping mechanism.

The significance of the preceding discussion is that, from the point of view of instability, each combustion system is characterized by certain well-defined proper frequencies at which the gases can oscillate in well-defined modes, and by certain damping mechanisms which absorb energy from the oscillating system. It is clear that self-sustained oscillations can exist (and instability appear) only if the combustion process is able to generate, at any one of the proper frequencies, enough feedback combustion energy (in excess of the steady-state conditions) to restore continually the amount which is being lost.

Suppose, for a moment, that the amount of combustion feedback energy is independent of frequency. Then the only important factor in the balance would be the energy damping corresponding to each mode, that (or those) mode(s) becoming unstable which correspond(s) to the lowest level of damping. Experience shows that this is not the case, and that, for a given injector, the selection of the unstable model is governed by frequency. This has been shown to be the case both longitudinal and for transverse forms of instability. It has been demonstrated experimentally that the frequency at which combustion oscillations become organized is more closely related to the combustion process (the driving mechanism) than it is to the chamber geometry (the resonating chamber). When a chamber that became unstable in the fundamental mode was lengthened so that the frequency of the fundamental mode now became a harmonic mode, the combustion oscillations continued to organize at the same frequency. Since the physical length of the harmonic wall length provides considerably more viscous damping than that of the fundamental mode, it would seem logical that the oscillations would organize in the fundamental mode. It is not our purpose to discuss here the details of the transition from one mode to the other (it does not take place suddenly), but rather to point out that the transition occurs in such a way as to maintain the oscillation frequency within a well-defined narrow range.

The only logical interpretation of this observation is that combustion energy must be fed into the chamber in a narrow range around one, well defined, frequency. The reason for the energy being confined to this narrow range is complicated -- it depends on several geometrical, physical, and chemical conditions -- but it's existence is the only logical explanation for the driving frequency being of sufficient magnitude to overcome the viscous damping associated with the increased wave length of the harmonic frequencies.

The empirically observed existence, for a given combustion system, of one narrow frequency range in which instability can appear can be interpreted by stating that, among the features characterizing the response of a given combustion system to oscillatory conditions, one can single out a "characteristic time" simply proportional to the reciprocal of frequency, such that only when its ratio to the oscillation period is around a certain value, maximum feedback can be generated and, possibly, instability produced.

This behavior has similarities to that of a resonant system, which is able to oscillate at certain natural frequencies with amplitudes that depend on the value of the frequency of the exciting forces compared to those of the natural frequencies. In the combustion instability problem also the exciting force due to the unsteady combustion processes is characterized by its own frequency (or its characteristic time), and maximum ampltidues are to be expected where there is coincidence with one of the proper frequencies. However, there is an important difference, residing in the fact that the exciting force is not independently applied from the outside, but is produced (in a sort of feedback loop) by the oscillations themselves, with the result that considerations of stability appear and take fundamental importance, and that the eventual oscillatory situation is determined by nonlinear effects. Nevertheless the terms "resonance" and "resonant" are often applied also to this case.

It appears that the knowledge of the characteristic time associated with a particular combustion system would be of primary importance in the design of new rockets, since it determines in which mode instability is able to appear. Such a knowledge could, for instance, allow the choice of the chamber and injector geometry in such a way that all of the proper frequencies would be too high to become unstable. Unfortunately, even assuming the characteristic time to be known, the choice of the propellant combination, as well as the chamber and basic injection system geometry, has always been fixed in the past during the early stages of development programs based on other stringent requirements of size, weight and performance, and there is very little chance that even in the future the designers may base their designs only on stability requirements. Therefore, a more sophisticated approach to the problem of stability is necessary, in which the second condition for appearance of instability, that of the energy balance, is also taken into consideration. In other words, even if it is impossible to avoid having some of the proper frequencies fall in the range where they may become unstable for the given combustion system, one should make sure that for the corresponding modes the combustion feedback is not sufficient to balance the damping.

Clearly there are two ways in which this balance can be improved in favor of stability: one is the depression of the combustion feedback, the other the increase of the damping. Of the two solutions, the second has been favored in recent times because it is better understood and hence easily controlled. Such damping devices as baffles or acoustic liners have been introduced, to the cost of more or less profound design complications, and have been very effective in producing substantial levels of damping, due to the dissipation caused by the devices. At the same time these devices also entail a change of the proper frequencies, and, in view of the previous discussion, this change can also have an effect on the balance.

On the other hand, the solution of depressing the combustion feedback has not been used in any consistent and systematic fashion, but only by looking for injector designs which, fortuitously or nearly so, result in the most stable operation on the test stand.

It may be added that a third, and the most effective solution, clearly consists in applying simultaneously the two solutions above, by (1) using an injector as stable as possible, and (2) adding a certain amount of extra damping to attain whatever safety margin is required.

The difficulty with a systematic application of the second and third solutions has been that knowledge of the behavior of combustion systems under oscillatory conditions is still very limited. Thus, it is impossible, starting from the fundamental physico-chemical processes involved, to predict the value of the characteristic time, the magnitude of the energy feedback, or -- with any certainty -- even the direction in which these quantities are affected by a change in design.

However, a less fundamental approach exists, which may still help the designer considerably, that is based upon the idea of establishing empirical correlations between the characteristic time and the energy feedback on one side and, on the other side, a certain number of appropriate combinations of parameters characterizing the injector geometry, the propellant mixture and the operating conditions. The feasibility of such an approach is offered by the sensitive time lag concept, introduced by Crocco in 1951 (Ref. 1). Although originally formulated with the sole intent of gaining an insight into the essential features of the high frequency instability phenomenon, the sensitive time lag combustion model was later found to predict accurately the quantitative behavior of the system in quite a few cases. In fact, this was beyond the expectations for such a heuristic approach.

This is indeed the primary advantage of the sensitive time lag concept, that the complexity of the actual combustion process can be avoided through the use of a very small number of lumped parameters. This is not to say that phenomena, such as droplet breakup, vaporization, mixing, chemical reactions, etc., are not important in determining stability and performance characteristics, but rather that, lacking such specific knowledge, the general nature of the coupling between the chamber conditions and the combustion process may still be described.

According to the Crocco model, the dynamic aspects of the injection-combustion process that are of significance in high frequency instability are characterized by a time lag which is sensitive to the local, instantaneous, values of pressure, temperature, gas velocity, etc. The degree of sensitivity is measured by one or more interaction indices. The sensitive time lag, then, plays the role of the above discussed characteristic time, while the interaction indices, properly combined, hold the key to the magnitude of the energy feedback. Thus, the occurrence of high frequency combustion instability is seen to result from the matching of the sensitive combustion time lag with one of the proper frequencies of the combustion chamber, provided that the degree of sensitivity of the combustion is sufficiently large to offset the damping effects present in the chamber. The stability conditions can then be expressed only in terms of the time lag and the interaction indices, and the ways to stabilization are easily discussed.

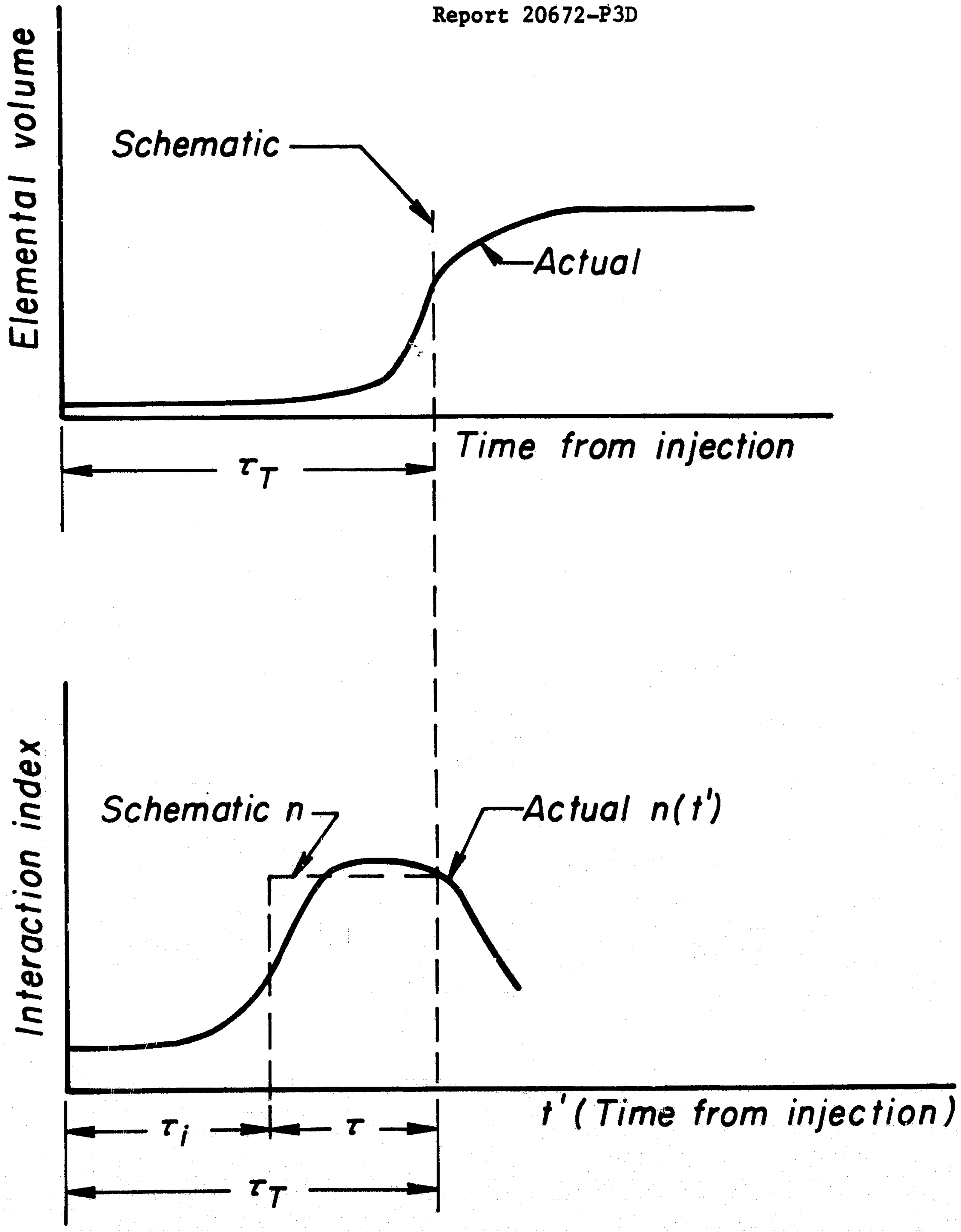


Figure 3 -- Time Lag Schematization

2. The Sensitive Time Lag Concept

The liquid propellant combustion process may be represented schematically as in Figure 3, which shows the volume of a propellant element from the instant of injection until conversion to final combustion products is accomplished. Initially, in the liquid state, the volume is small. The conversion to gaseous products takes place more or less gradually, depending on the degree of atomization, the intensity of mixing, the chemical nature of the reactants, and so on. This gradual change is replaced, for analytical simplicity, by a step function, as shown in Figure 3. This is not to say that the combustion process is thought to occur instantaneously, but that such simplification may reasonably describe the essential nature of the combustion process from a dynamic point of view. The step function approximation of the overall combustion process defines a characteristic time, called the "total time lag" for the propellant element considered and denoted by τ_T. This time, of the order of a few milliseconds, is not representative of the characteristic times of high frequency instability. Rather, the total time lag is basic to the low frequency type of combustion instability. However, the total time lag, together with the velocity history of the injected propellants, determines the space lag, that is, the location in the chamber at which combustion of the particular propellant element is taking place. It is this aspect of the total time lag that cannot be disregarded when considering high frequency instability. This point will be discussed further in a later section. It suffices to say here that the step function approximation is compatible with any combustion distribution in the chamber, since τ_T can be different for different propellant elements.

The lower diagram of Figure 3 illustrates the important concept that not all of the processes that occur during the combustion of liquid propellants are equally affected by the combustion environment. Consider first only the effect of pressure (and correlated temperature) oscillations.

Initially, only liquid streams, ligaments, or relatively large droplets are available in conditions unfavorable to combustion. It is to be expected that pressure perturbations interact only slightly with propellants in that degree of preparation. In the later portion of the preparation phase (as the nominal τ_T value is approached), droplets are small and interspersed; moreover fuel and oxidizer species are mixed. Hence, burning rates are readily influenced by instantaneous pressure changes. This increased sensitivity to disturbances in pressure and temperature is represented on the figure by a higher instantaneous "pressure interaction index", n. The portion of the total time lag that is associated with this high sensitivity can be assigned an approximately constant mean value of the interaction index and is called the "sensitive time lag", denoted by τ. The early portion is referred to as the "insensitive time lag", τ_i, and the corresponding interaction index is taken to be zero. Again, the step-function approximation is useful to reach a simple analytical description. The sensitive time lag is of the order of a few tenths of a millisecond (an order of magnitude smaller than the total time lag). It plays the role in the frequency combustion instability of the "characteristic time" discussed previously. The energy feedback resulting from pressure oscillations can be calculated in terms of τ and the interaction index, n. Similarly the energy feedback resulting from gas velocity oscillations can be calculated in terms of a velocity-sensitive time lag (which may coincide with the one already defined) and of an appropriate "velocity interaction index", and the same would apply to the interaction with other possible physical quantities.

In the schematic representation just discussed, two (or more) combustion parameters have been defined that are of key importance to further discussions of high frequency instability. The first is the sensitive time lag, τ, which places the combustion process in proper perspective with the times corresponding to the various acoustic modes associated with the chamber geometry. The second parameter, the interaction index, n (and, possibly, other parameters, representing the additional interaction indices of relevance), must exceed a certain minimum level if the damping is to be balanced and self amplifying or sustained oscillations are to be generated.

These points are further clarified (where only one interaction index is relevant) by a stability diagram using τ as the abscissa and n as the ordinate (Figure 4), relative to the first tangential (1T) mode. As already observed, the resonant behavior of the system is such that the maximum amount of energy feedback is obtained when the ratio of the characteristic time to the oscillation period (in this case of the 1T mode) is around a certain value (which for the present simple combustion model is 1/2). At the corresponding value of τ, evidently, the damping processes can be balanced with the minimum value of n_{min}. At other values of τ the effectiveness of the feedback process decreases, and hence larger values of n are required to reach the balance, the values increasing with increasing deviations of τ from the value of maximum effectiveness. The curve of Figure 4 represents, for the given mode, the n-τ combinations for which a balance between feedback and damping is obtained. If, for given τ, n is smaller than the value which produces the balance, perturbations will decay as a result of the excessive damping; if on the contrary n is larger, perturbations will amplify as a result of the excessive feedback. Hence the curve provides the stability boundary between the stable region of Figure 4 (under the curve) and the unstable region (above the curve).

A given combustion system is characterized by a certain value of the interaction index. If this value is less than n_{min} no instability at all is possible. If the interaction index is just equal to n_{min}, then instability is possible, but only if the matching of the proper time of the chamber and the characteristic combustion time is perfect. For $n > n_{min}$ the time-matching requirement is less stringent, i.e., a certain mismatch of the times will still result in instability. As the value of n increases, the amount of allowable mismatching also increases.

For other modes than the 1T mode, the stability boundary can be represented in a similar way, the scale of τ properly shifted. Putting the stability boundaries corresponding to different modes on the same plot results

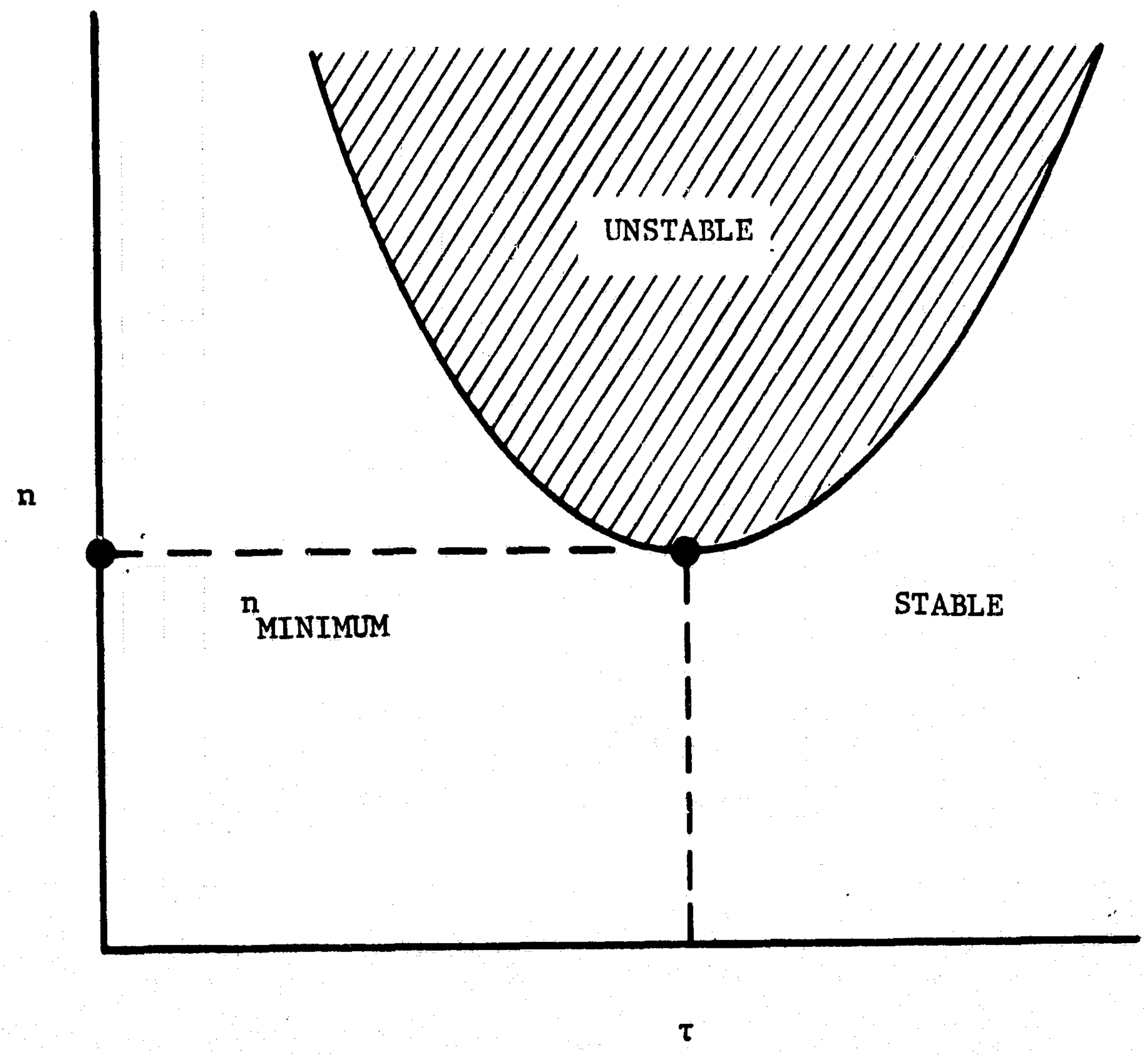


Figure 4 -- Stability Diagram, n, τ, - Plane

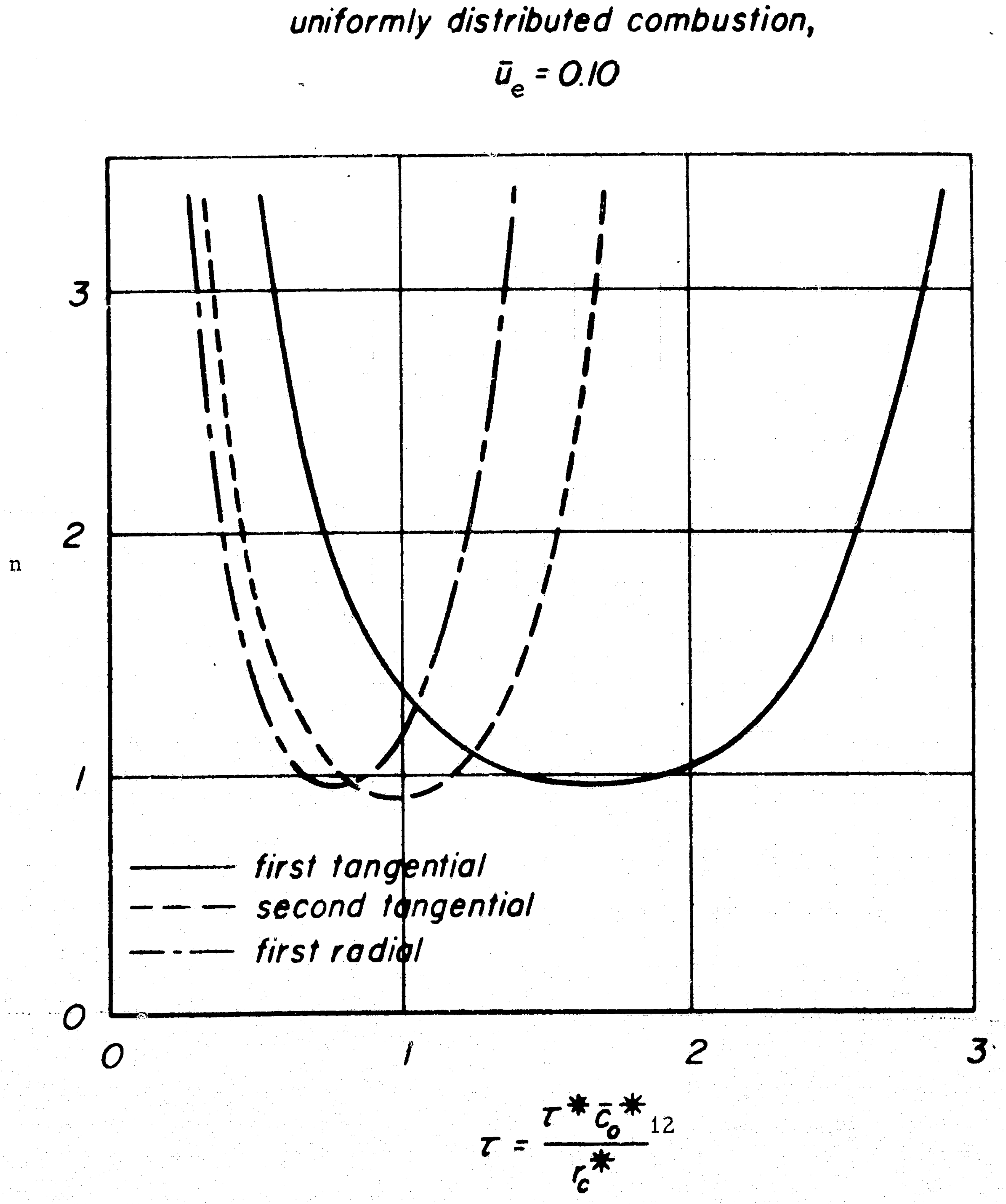


Figure 5 -- Theoretical Stability Limits for Several Purely Transverse Modes

in Figure 5, where an overall stability region, regardless of mode, is now evident. Plots of this kind are fundamental to the methods to be discussed extensively in this report. However, two remarks, to be substantiated later, should be made.

The first remark is that such a clearcut boundary between stability and instability is obtained only if the perturbations are assumed to be of small amplitude. This is indeed the assumption on which most of the theoretical developments on combustion instability have been based. Within this assumption all effects of the perturbations can be assumed to vary linearly with the perturbation amplitudes, the mathematical treatment is accordingly substantially simpler, and the (linear) stability boundary unequivocally defined, the linearly unstable region being practically interpreted as that in which oscillations grow spontaneously out of the random combustion noise. However, in real practice the perturbations are not necessarily limited to the range where all of their effects vary linearly with the amplitude, and when the contrary happens important nonlinear effects may appear. Some effects of nonlinearity can be derived theoretically, at the cost of substantial mathematical complexities. But what is important to the present qualitative discussion is that, while the linearly unstable region always remains a region of instability, the corresponding statement for the linearly stable region is not true. In other words, a system corresponding in the n-τ plane to a point of the linearly stable region may be triggered into amplifying or self-sustaining oscillations by a perturbation (for instance, a pulse) of sufficiently large amplitude. Only if the pulse remains under a certain critical level do the resulting oscillations decay - in agreement with the predictions of the linear theory. This nonlinear behavior plays an important role in rockets, and has to be taken into account when interpreting the experimental results.

The second remark is that the curves of Figures 4 and 5 depend not only on the chamber and nozzle geometries, but also on the transverse and longitudinal distributions of combustion, the first being substantially determined by the distribution of the injection flux across the injector, the second by the details of the individual combustion processes (such as atomization, evaporation, mixing and chemical reactions, recirculation flow, etc.). The significance of the total time lag τ_T in this connection has been mentioned already. Large values of τ_T, other factors being equal, will spread the combustion toward the nozzle end of the chamber. If the total time lag is too large for the chamber in question, performance will suffer because of incomplete burning prior to the nozzle entrance. Since the region of maximum combustion is closely associated with the region where propellants have reached the sensitive state in the preparation process, it is in this general location that interaction between the combustion process and the acoustic modes reaches a maximum level.

For the fundamental longitudinal mode, pressure antinodes are found at the injector and the nozzle end of the chamber, as shown in Figure 6. Higher harmonics will have additional pressure antinode locations. If combustion is uniformly distributed from one end of the chamber to the other, maximum instability coupling cannot take place since there is a region (or regions, in the harmonics) in which a pressure nodal environment is approached. If the wave is sinusoidal, a true node is found; otherwise, only an oscillation with reduced amplitude will be observed. The limiting case of a non-sinusoidal wave is that of a shock wave, the amplitude variation of which is shown in Figure 6. In the nodal region, even with proper time phasing, too little energy is available from the increased burning rate to cause the pressure oscillations to be amplified. However, if combustion is concentrated at the injector end, the best environment for energy transfer to the pressure oscillations is provided.

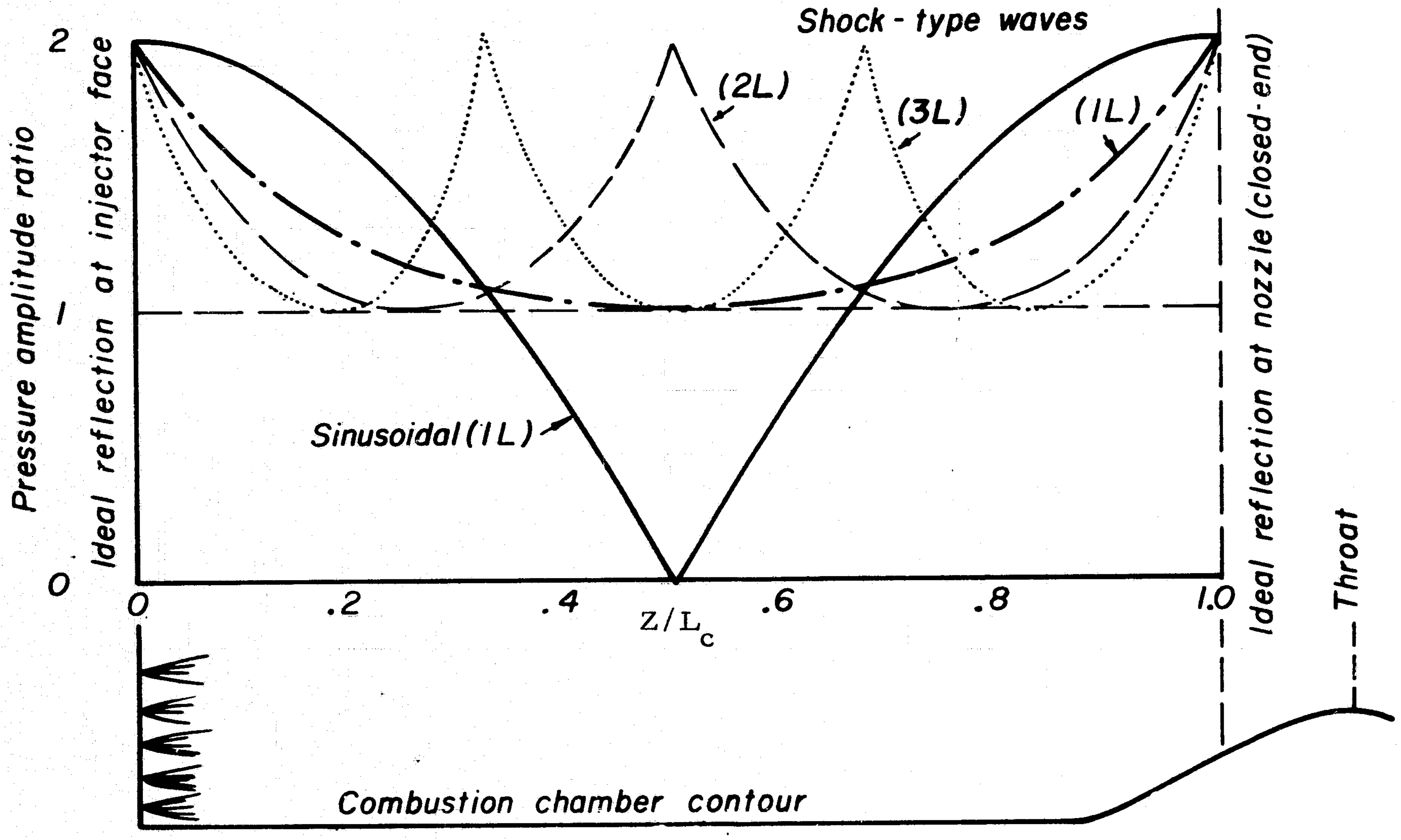


Figure 6 -- Axial Pressure Variations for Several Longitudinal Modes

I, B, Technical Description (cont.)

The steady state combustion rate is approximately proportional to the rate of change of axial gas velocity with distance from the injector. Typical velocity profiles are shown in Figure 7. Two extremes are apparent: (1) the concentrated combustion case in which near-maximum gas velocities are produced within a short distance from the injector, as shown by curve "A", and (2) the nearly uniform combustion case in which the axial velocity curve is nearly linear, as represented by curve "B". Rocket experience has shown that the actual velocity distributions fall within these extremes. Thus, combustion concentrated near the nodal point and combustion concentrated near the antinode at the nozzle end are both unrealistic situations.

The transverse modes may also be discussed on a somewhat similar basis. Experimental measurements at a number of laboratories have shown that maximum amplitudes for the tangential modes are always found at the injector end. Thus, to reduce coupling, axially distributed combustion can offer considerable improvement as compared to the concentrated combustion case, just as was found for the longitudinal mode. In the radial direction the combustion distribution picture is more complicated. As shown in Figure 8, pressure oscillation amplitudes vary radially, and the vibration is strongly dependent on the mode. The first and second tangential modes are similar to each other but quite different from the first radial mode. To promote instability in the first tangential mode, assuming that the n, τ values are suitable, one would choose an injector design that would provide rapid burning (i.e., small τ_T, to keep the combustion near the injector face) and would have the injection orifices concentrated at the outer radii. Either spreading the combustion axially or moving the injection toward the center of the chamber would reduce the degree of coupling. If the first radial mode is also considered, then a compromise location near the half-radius point would prove to be the best injection point. This principle has been verified in an investigation of a number of injection distributions by the Aerojet-General Corporation using high-thrust hardware (Ref 2).

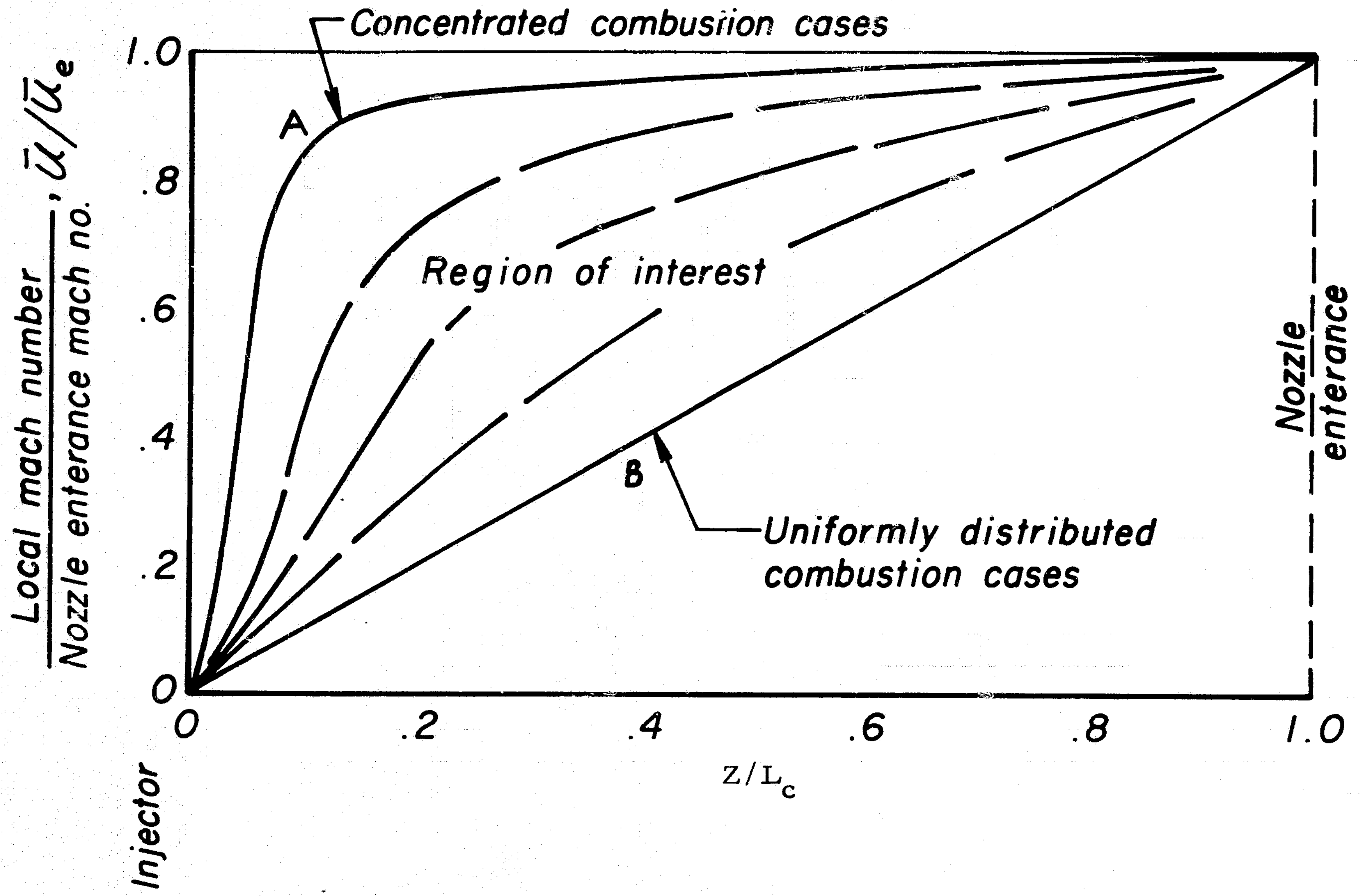


Figure 7 -- Typical Combustion Distributions

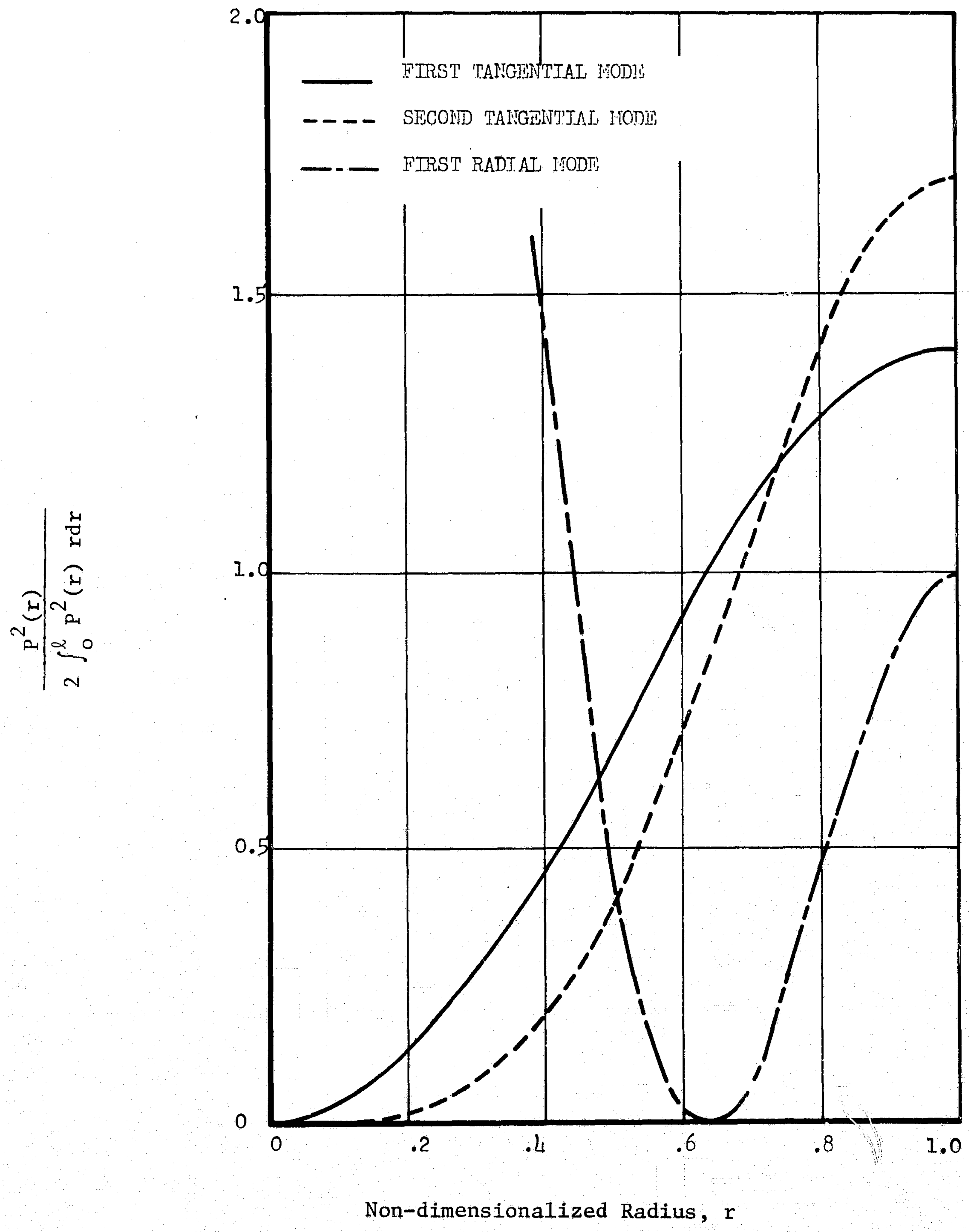


Figure 8 -- Pressure Non-Uniformity Coefficient for Concentrated Combustion

The qualitative discussion just given is based on the assumption that there is only one relevant interaction index. When more than one has to be taken into account, the effect of each additional interaction index produces a displacement of the stability boundaried in the n-τ plane, and the interpretation becomes correspondingly more difficult. Effects of this kind are found, for instance, when the combustion rates are sensitive to the transverse velocity perturbations characteristic of transverse oscillations. One can visualize important coupling mechanisms related to the inequities in the displacements of liquid drops and propellant vapors that are produced by such oscillations. The local mixture ratio can be altered, with resulting changes in the burning rate, if one of the propellants is displaced with respect to the other.

It is interesting in this connection to observe that this effect may be particularly intense in the region immediately adjacent to the injector where one can expect droplets and vapors to be present in abundance, and in very unmixed conditions. Thus, it is clear that in the use of baffles and liners there must be a consideration of the combustion distribution in the chamber. In general, it has been found that for tangential modes the oscillation amplitudes are greatest at the injection end of the chamber, combustion rates are maximum a few inches from the injector, and coupling between the combustion process and tangential waves is greatest in the outer regions of the injector face. If the chamber volume is subdivided by baffles, the dimensions of the cavities between baffles determine the period of oscillation. Combining these principles, it is concluded that if baffles are to be used to control tangential modes they must be placed at the injector end, they must protect the preparation zone found in the first few inches downstream of the injector, they must extend as close as possible to the chamber wall, and the circumferential blade spacing must be small enough to prevent cavity modes from existing. Similarly, acoustic liner orifices are most effective near the injector end of the chamber, and liner absorption characteristics must be designed to match the resonance properties of the chamber and the combustion process, as described by the sensitive time lag theory.

3. Development of the Sensitive Time Lag Theory

The historical background of the development of the sensitive time lag theory is important to the understanding of the concepts described in the preceding section. The earliest published paper on combustion instability theory was that of Gunder and Friant (Ref 3) in 1950, with a subsequent discussion by Yachter (Ref 4). Probably the most important contribution of these early treatments was the introduction of the concept of a combustion time lag (conceived independently, but not published, by other groups) between the instant of injection of a propellant element and the succeeding instant of burning, in which the propellant element is transformed into hot gas capable of contributing to the chamber pressure.

Interest at Princeton University in the problem of combustion instability in liquid propellant rocket motors was given impetus by a Bureau of Aeronautics Symposium held at the Naval Research Laboratory in December, 1950. This interest resulted in theoretical analyses by Professors Summerfield and Crocco.

Professor Summerfield's work (Ref 5) considered the effects of inertia in the propellant feed lines and the capacitance of the combustion chamber, assuming a constant combustion time lag. His analysis treated the case of low frequency oscillations for frequencies up to about 200 cps.

I, B, Technical Description (cont.)

Professor Crocco advanced the concept of the pressure dependence of the combustion time lag. His paper (Ref 1) presented the fundamentals resulting from this concept. In his paper, Crocco treated the case of low frequency instability in a bipropellant rocket, and also the case of high frequency instability with combustion concentrated at the injector end of the combustion chamber.

The analytical work on the high frequency case was continued by S. I. Cheng under the direction of Professor Crocco. His studies of the effects of the axial distribution of combustion (on the longitudinal modes) were published as his Ph.D. thesis. A thorough discussion of these and other aspects of the theory was published by Crocco and Cheng in 1956 as an AGARD monograph (Ref 6). The general theories of low and longitudinal high frequency instability, the effects of the combustion distribution, the influence of the exhaust nozzle, as well as the (scarce) experimental evidence substantiating the analyses were all discussed at length. A brief discussion of the transverse modes of combustion instability was included, and general adherence to the sensitive time lag model was predicted.

The extension to the transverse modes was initiated by S. M. Scala (Ref 7). Following Crocco's pressure dependence model, he determined the fundamental behavior of the transverse modes, including the influence of the nozzle. In addition, Scala treated the case of intermediate frequency instability, in which the coupling mechanism consists of entropy perturbations, generated by off-design mixture ratio combustion, which reflect from the nozzle as pressure waves and propagate back to the injector to cause perturbations in the injection rates.

The study of transverse instability was continued by F. H. Reardon (Ref 8), who developed several extensions to the basic theory to explain certain experimental results (which are discussed in the following section). The sensitive time lag concept was extended to include sensitivity to the transverse components of the oscillating gas velocity. The effect on the combustion rate was visualized in the oscillatory displacement of the vapors of one propellant with respect to the liquid droplets of the other. In addition, Reardon introduced an approximate treatment of the effects of nonuniform distribution of propellant injection on the transverse modes, and applied the modified theory to a sector-shaped combustor, which simulates the "pocket-mode" behavior of a baffled chamber.

I, B, Technical Description (cont.)

4. Theory

a. General Approach

The simplifying assumptions on which the mathematical treatment of combustion instability is based are the following:

(1) The substance contained in the combustion chamber is either in the form of liquid propellants, of practically zero volume, or in the form of complete combustion gases. It will be noticed that this assumption disregards the contribution of the propellants in vapor form, and of the intermediate products of combustion. Hence the assumption is equivalent to saying that not only the liquids, but also the vapors and the intermediates occupy a negligible volume compared to the final products of combustion. This assumption (first used by Crocco in Ref. 1) is actually in agreement with the step-function combustion model discussed in Section I,B,2. Even more important, it represents quite well the actual conditions in rockets where indeed, with the exception of the region immediately adjacent to the injector, practically the same gas temperature is observed throughout the chamber. Of course, it is clear that the above is true when using liquid propellants, and not for the combustion of gaseous propellants. The difference is that while in the latter case there is a constant mass flux with energy addition, in the case of liquid propellants the gas flow has a variable mass flux with energy addition being produced through an addition of mass to the gaseous flow.

(2) The combustion gases are of constant composition, they obey the perfect gas law and have constant specific heats.

(3) Frictional effects on the walls are neglected, and only those are taken into account which result in the liquid droplet drag.

I, B, Technical Description (cont.)

Also, Reynolds stresses associated with the high turbulence level caused by combustion are neglected, in spite of the fact that they may play an important role with respect to the uniformity assumption (Assumption 5).

(4) The flow of injected propellants is unaffected by oscillations in the chamber, and hence the injection flux and velocity are always the same as in steady conditions. This assumption is not necessarily verified in actual rockets where, especially for large rockets, the possibility of matching the wave propagation times in the feed lines and in the chamber may lead to interactions. However, the assumption is rather good if matching is avoided, and substantially simplifies the treatment by making the chamber behavior independent of the feed system varieties and complexities. The resulting instability problem has been termed "intrinsic instability" of the combustion chamber.

(5) The steady-state gas flow is uniform across any chamber section. This is possible partly because the previous assumption allows the boundary layer formation on the walls to be disregarded. However, it involves more. For instance, it would require the injected propellants to be uniformly distributed so as to produce no recirculation. Of course, this is not the actual situation, and the uniform flow considered in the theory should be interpreted as an average flow from which the actual flow can depart substantially if the injection is far from being uniform. For large rockets the injection systems are generally rather uniform, and hence the uniformity assumption can be quite accurate. However, it will be seen in the following that transverse stability conditions can be improved by using particular non-uniform injection systems. It is felt that the contradiction that results in these cases is not very important for not too large flow Mach numbers because of the strongly equalizing effects of the high turbulence due to combustion.

(6) In steady-state, the total energy (internal and kinetic) of the droplets remains constant. Obviously this is not exactly true, because of the heat exchanges affecting the internal energy and the droplet drag affecting the kinetic energy in ways that are not so simply related. The assumption has, however, the advantage of providing a substantial simplification of the treatment, and it is believed not to hide any of the fundamental effects.

(7) The steady-state flow in the nozzle is one-dimensional. Although not true, this assumption is known to result in very accurate predictions concerning the steady flow itself. Here, however, the same assumption is extended to the treatment of the oscillatory nozzle flow obtained when the steady flow is perturbed. It should be observed that this assumption is consistent with that of uniform steady flow in the chamber (Assumption 5).

(8) The unsteady, oscillatory quantities in the chamber and in the nozzle can be obtained by superposing small perturbations to the steady-state quantities. By "small" is meant as usual, that only "linear" (first order) terms in the perturbations are to be retained, while terms containing products or powers of perturbations (second and higher order) are to be neglected. The great advantage of this assumption is of a mathematical nature, since the resulting equations, being linear, can be treated in a much simpler way. One of the simplications is that a harmonic time dependence can be chosen, as discussed under Assumption 10. But, of course, the disadvantage is that only the linear effects can be accounted for, all the nonlinear effects being left out. Under the small perturbation assumption only the "linear" stability problem can be attacked.

(9) In the distributed combustion analysis the gas flow Mach number is always sufficiently small so that the square can be neglected compared to unity. Because of this assumption, the analysis cannot

be applied to thrust chambers in which the Mach number exceeds 0.3. Such is not an essential assumption; its relaxation leads only to complexity. This assumption does not apply to the concentrated combustion analysis.

(10) The time dependence of the perturbations can be expressed in complex form as exp (σt) where $\sigma = \lambda + i\omega$ is the same complex quantity for all perturbations. Here ω is the angular frequency and λ the amplification coefficient. As usual when using the complex representation each perturbation is obtained by multiplying the above exponential by a corresponding complex amplitude factor which is a function of the space coordinates only, and by taking the real part of the result. The assumption of exponential time dependence is common in problems of linear stability. It does not affect the generality of the result, since within the linear framework the development of the most general perturbation can be obtained from the superposition of independent components each behaving exponentially, and since, from the point of view of stability, the only thing that matters is that no single component should show amplification. Hence all the information required is the behavior of exponentially varying perturbations when the frequency is made to change over all possible values.

b. Governing Equations

(1) General Equations

The equations governing the unsteady, two-phase flow in the combustion chamber are derived from the principles of conservation of mass, momentum, and energy, and the equation of state, according to the assumptions discussed in the previous section. It is convenient to work with the equations in nondimensional form. The reference quantities for the nondimensionalization are taken as the stagnation gas properties at the injector face (pressure, temperature, density, and speed of sound) together with a reference length, which is the chamber radius for transverse modes. (The chamber length is generally used for longitudinal modes). The governing equations take the following forms:

Conservation of mass,

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{q}) = Q = - \frac{\partial \rho_L}{\partial t} - \nabla \cdot (\rho_L \vec{q}_L) \tag{1}$$

Conservation of momentum,

$$\rho \frac{\partial \vec{q}}{\partial \tau} + \rho \vec{q} \cdot \nabla \vec{q} + \frac{1}{\gamma} \nabla p = (Q + \kappa \rho_L) (\vec{q}_L - \vec{q}) \tag{2}$$

Conservation of energy,

$$\rho \frac{\partial h_s}{\partial t} + \rho \vec{q} \cdot \nabla h_s - \frac{\gamma - 1}{\gamma} \frac{\partial p}{\partial t} = Q (h_{Ls} - h_s) \tag{3}$$

Equation of state for the gas,

$$p = \rho T \tag{4}$$

Droplet drag (gas/liquid momentum interchange) using Stokes law,

$$\frac{\partial \vec{q}_L}{\partial t} + \vec{q}_L \cdot \nabla \vec{q}_L = \kappa (\vec{q} - \vec{q}_L) \tag{5}$$

And droplet energy,

$$\frac{\partial h_{Ls}}{\partial t} + \vec{q}_L \cdot \nabla h_{Ls} = 0 \tag{6}$$

I, B, Technical Description (cont.)

The symbols used in these equations are defined in Section I.D.

It is necessary to express the governing **Equations,** (1) to (6), in steady state form and perturbed form. The separation is accomplished by writing all the dependent variables, as the sum of a steady state solution and an unsteady solution. For example,

$$p = \overline{p} + p' \, e^{\sigma t} \tag{7}$$

$$u = \overline{u} + u' \, e^{\sigma t}$$

It is noted that the perturbed portion of each parameter is harmonic with respect to time.

(2) Steady State Equations

The steady state solution is assumed to be one dimensional, although the perturbations are considered in three dimensions. Therefore, the steady state vectorial components of gas velocity, q, and liquid velocity, q_L, are given by the following relationships:

$$\overline{\vec{q}} = \overline{u} \ , \ \overline{\vec{q}}_L = \overline{u}_L$$

$$\overline{v} = \overline{w} = \overline{v}_L = \overline{w}_L = 0 \tag{8}$$

where u, u_L are the axial components, v, v_L are the radial components, and w, w_L are the tangential components of $\vec{q}$, $\vec{q}_L$.

Substituting the relationships given by (7) and (8) into (1) to (6) and neglecting the time dependence yields the following system of steady state equations:

Conservation of mass

$$\frac{d}{dZ}\left(\bar{\rho}\ \bar{u}\right) = \bar{Q} = -\frac{d}{dZ}\left(\bar{\rho}_L\ \bar{u}_L\right) \tag{9a}$$

Conservation of momentum

$$\frac{d}{dZ}\left(\bar{\rho}\ \bar{u}^2\right) + \frac{d}{dZ}\left(\bar{\rho}_L\ \bar{u}_L{}^2\right) = -\frac{1}{\gamma}\frac{d\bar{p}}{dZ} \tag{9b}$$

Conservation of energy

$$\bar{\rho}\ \bar{u}\ \frac{d\bar{h}_s}{dZ} = -\bar{Q}\left(\bar{h}_s - \bar{h}_{Ls}\right) \tag{9c}$$

Equation of state (gas)

$$\bar{p} = \bar{\rho}\ \bar{T} \tag{9d}$$

Droplet drag

$$\bar{u}_L\ \frac{d\bar{u}_L}{dZ} = \kappa\left(\bar{u} - \bar{u}_L\right) \tag{9e}$$

Droplet energy

$$\frac{d\bar{h}_{Ls}}{dZ} = 0 \tag{9f}$$

It should be noted that at $Z = 0$, $u = 0$ and $\bar{p} = \bar{\rho} = 1$ because of the nondimensionalizing scheme. At $Z = L_c$, combustion is assumed to be complete so that $\bar{\rho}_L = 0$ and $\bar{Q} = 0$.

(3) The Perturbation Equations: Wave Equations

The perturbed equations are considered in three dimensional form in cylindrical coordinates and the resulting system of equations is linearized. That is, the perturbed quantities are considered to be small, and therefore, all products of two or more perturbed terms are considered to be zero. Substituting Equations (7) and (8) into Equations (1) to (6) yields the following system of equations.

$$\begin{aligned}(\sigma + \frac{d\bar{u}}{dZ}) \rho' + \bar{u} \frac{\partial \rho'}{\partial Z} + \frac{d\bar{\rho}}{dZ} u' + \bar{\rho} \Big(\frac{\partial v'}{\partial r} + \frac{v'}{r} \\ + \frac{1}{r} \frac{\partial w'}{\partial \theta} + \frac{\partial u'}{\partial Z}\Big) = \bar{Q} (\mathcal{P} p' + \mathcal{R} v' + \mathcal{T} w')\end{aligned} \tag{10a}$$

where

$$\mathcal{P} = n\,(1-e^{-\sigma\tau}),\ \mathcal{R} = 1_r\,(1-e^{-\sigma\tau}) = \frac{1r}{n} \mathcal{P},\ \text{and}\ \mathcal{T} = 1_\theta\,(1-e^{-\sigma\tau}) = \frac{1_\theta}{n} \mathcal{P}$$

Conservation of axial momentum

$$\begin{aligned}(\sigma\bar{u} + 2\bar{u} \frac{d\bar{u}}{dZ}) \rho' + (\sigma\bar{u}_L + 2\bar{u}_L \frac{d\bar{u}_L}{dZ}) \rho_L' + \bar{u}^2 \frac{\partial \rho'}{\partial Z} + \bar{u}^2{}_L \frac{\partial \rho'{}_L}{\partial Z} \\ + (\sigma\bar{\rho} + 2\bar{\rho} \frac{d\bar{u}}{dZ} + 2\bar{u} \frac{d\bar{\rho}}{dZ}) u' + (\sigma\bar{\rho}_L + 2\bar{\rho} \frac{d\bar{u}_L}{dZ} + 2\bar{u}_L \frac{d\bar{\rho}_L}{dZ}) u_L' \\ + 2\bar{\rho}\,\bar{u} \frac{\partial u'}{\partial Z} + 2\bar{\rho}_L\,\bar{u}_L \frac{\partial u_L'}{\partial Z} + \bar{\rho}\,\bar{u} \Big(\frac{\partial v'}{\partial r} + \frac{v'}{r} + \frac{1}{r} \frac{\partial w'}{\partial \theta}\Big) \\ + \bar{\rho}_L\,\bar{u}_L \Big(\frac{\partial v_L'}{\partial r} + \frac{v_L'}{r} + \frac{1}{r} \frac{\partial w_L'}{\partial \theta}\Big) = - \frac{1}{\gamma} \frac{\partial p'}{\partial Z}\end{aligned} \tag{10b}$$

I, B, Technical Description (cont.)

Conservation of radial momentum

$$(\sigma\bar{\rho} + \bar{\rho}\,\frac{d\bar{u}}{dZ} + \bar{u}\,\frac{d\bar{\rho}}{dZ})\; v' + (\sigma\bar{\rho}_L + \bar{\rho}_L\,\frac{d\bar{u}_L}{dZ} + \bar{u}_L\,\frac{d\bar{\rho}_L}{dZ})\; v'_L$$

$$+ \bar{\rho}\,\bar{u}\,\frac{\partial v'}{\partial Z} + \bar{\rho}_L\,\bar{u}_L\,\frac{\partial v'_L}{\partial Z} = -\frac{1}{\gamma}\,\frac{\partial p'}{\partial r} \qquad (10c)$$

Conservation of tangential momentum

$$(\sigma\bar{\rho} + \bar{\rho}\,\frac{d\bar{u}}{dZ} + \bar{u}\,\frac{d\bar{\rho}}{dZ})\; w' + (\sigma\bar{\rho}_L + \bar{\rho}_L\,\frac{d\bar{u}_L}{dZ} + \bar{u}_L\,\frac{d\bar{\rho}_L}{dZ})\; w'_L$$

$$+ \bar{\rho}\,\bar{u}\,\frac{\partial w'}{\partial Z} + \bar{\rho}_L\,\bar{u}_L\,\frac{\partial w'_L}{\partial Z} = -\frac{1}{\gamma r}\,\frac{\partial p'}{\partial \theta} \qquad (10d)$$

Conservation of energy and equation of state, combined

$$(\sigma\bar{\rho} + \bar{Q} - u\,\frac{d\bar{\rho}}{dZ})\;\rho' + \bar{\rho}\,\bar{u}\,\frac{\partial \rho'}{\partial Z} - (\gamma-1)\;\bar{\rho}\,\bar{u}\;(\sigma\bar{\rho} + \bar{Q} + \bar{\rho}\,\frac{d\bar{u}}{dZ})\; u'$$

$$- (\gamma-1)\;(\bar{\rho}\,\bar{u})^2\,\frac{\partial u'}{\partial Z} = (\sigma\,\frac{\bar{\rho}}{\gamma} + \bar{Q} - \bar{u}\,\frac{d\bar{\rho}}{dZ})\; p' + \bar{\rho}\,\bar{u}\,\frac{\partial p'}{\partial Z} \qquad (10e)$$

Droplet dynamics

defining $K = \frac{\kappa}{\kappa+\sigma}$

and $\xi = (\kappa+\sigma)\int_0^Z \frac{dZ}{\bar{u}_L}$

then

$$u'_L = Ku' - \frac{K}{\bar{u}_L}\, e^{-\xi}\left[\int_0^Z \frac{d\bar{u}_L}{dZ}\, e^{\xi} u'\; dZ + \int_0^Z \bar{u}_L\, e^{\xi}\,\frac{\partial u'}{\partial Z}\; dZ\right] \qquad (10f)$$

I, B, Technical Description (cont.)

$$v'_L = K\,(1-e^{-\xi})\; v' - K\, e^{-\xi} \int_0^Z e^{\xi} \frac{\partial v'}{\partial Z}\, dZ \tag{10g}$$

$$w'_L = K\,(1-e^{-\xi})\; w' - K\, e^{-\xi} \int_0^Z e^{\xi} \frac{\partial w'}{\partial Z}\, dZ \tag{10h}$$

$$\begin{aligned}(\sigma + \frac{d\bar{u}}{dZ})\, \rho' + (\sigma + \frac{d\bar{u}_L}{dZ})\, \rho'_L + \bar{u}\, \frac{\partial \rho'}{\partial Z} + \bar{u}_L\, \frac{\partial \rho'_L}{\partial Z} \\ + \bar{\rho} \left[\frac{\partial v'}{\partial r} + \frac{v'}{r} + \frac{1}{r}\frac{\partial w'}{\partial \theta} + \frac{\partial u'}{\partial Z}\right] + \bar{\rho}_L \left[\frac{\partial v'_L}{\partial r} + \frac{v'_L}{r} \right. \\ \left. + \frac{1}{r}\frac{\partial w'_L}{\partial \theta} + \frac{\partial v'_L}{\partial Z}\right] + \frac{d\bar{\rho}}{dZ}\, u' + \frac{d\bar{\rho}_L}{dZ}\, u'_L = 0\end{aligned} \tag{10i}$$

In general, separation of variables is not possible with these equations. A solution can be obtained by writing each quantity in a series such that each successive term in the series is less than its predecessor. Therefore, the pressure perturbation is written in the form

$$p' = p_0 + p_1 + p_2 + p_3 + \ldots\, p_n + \ldots \tag{11}$$

where p_0 can assume any magnitude within the restrictions of the analysis. Then $p_1 = 0\,(u_e \cdot p_0)^*$, $p_2 = 0\,(\bar{u}_e^2 \cdot p_0)$ and $p_n = 0\,(\bar{u}_e^n \cdot p_0)$ where $\bar{u}_e$ is essentially the chamber Mach number. Applying this approach to Equations (10a) through (10i), collecting terms of like order, and solving for the pressure results in wave equations for p_0 and p_1:

$$\sigma^2\, p_0 - \nabla^2\, p_0 = 0 \tag{12a}$$

---

* This notation is used to indicate the order of magnitude of a given parameter.

I, B, Technical Description (cont.)

$$\sigma^2 \, p_1 - \nabla^2 \, p_1 = \sigma\gamma \, \overline{Q} \, \Big(\mathcal{P} \, p_0 - \mathcal{R} \frac{1}{\sigma\gamma\overline{\rho}} \, \frac{\partial p_0}{\partial r}$$

$$- \mathcal{T} \frac{1}{\sigma\gamma\overline{\rho}\theta} \, \frac{\partial p_0}{\partial \theta} \Big) - \sigma \left[2 \, \frac{d\overline{u}}{dZ} + (\gamma-1) \, \frac{\overline{Q}}{\overline{\rho}} + \sigma \, \frac{\kappa}{\kappa+\sigma} \, \frac{\rho_L}{\overline{\rho}} \right] p_0 \tag{12b}$$

For a cylindrical chamber the solutions of (12a) and (12b) are given by

$$p_0 \, (z, \, r, \, \theta) = P_{oo} \, \cosh \, (\Omega \, Z) \psi \, (r) \, \Theta \, (\theta) \tag{13a}$$

$$p_1 \, (z, \, r, \, \theta) = P_{oo} \, (\Gamma_{11} \, \mathcal{P} + \Gamma_{10}) \, \psi \, (r) \, \Theta \, (\theta)' \tag{13b}$$

where

$$\psi \, (r) = J_\nu \, (s_{\nu\eta} \cdot r) \tag{13c}$$

$$\Theta \, (\theta) \begin{cases} \cos \, \nu \, \theta & \text{STANDING MODE} \\ e^{-\nu\theta} & \text{SPINNING MODE} \end{cases} \tag{13d}$$

and J_ν represents the Bessel function of the first kind of order ν (see Table I)

Also,

$$\Gamma_{11} = \frac{1}{\Omega} \, \Big(-\sigma\gamma \, A_{\nu\eta} + \frac{1_r}{n} \, B_{\nu\eta} + \frac{1\,1_\theta}{n} \, C_{\nu\eta}\Big) \int_0^Z \overline{Q} \, (\zeta) \, \sinh \, [\Omega \, (Z-\zeta)] \, d\zeta \tag{13e}$$

where

$$\Omega^2 = \sigma^2 + s_{\nu\eta}^2$$

the factors $A_{\nu\eta}$, $B_{\nu\eta}$ and $C_{\nu\eta}$ are discussed in Section I.B.4.b.(5), p. 61.

I, B, Technical Description (cont.)

$$\Gamma_{10} = \frac{\sigma}{\Omega} \left\{ (\gamma-1) \int_0^Z \overline{Q}\,(\zeta)\ \sinh\ [\Omega\ (Z-\zeta)]\ d\zeta \right. \tag{13f}$$

$$+ 2 \int_0^Z \frac{d\overline{u}}{dZ}\,(\zeta)\ \sinh\ [\Omega\ (Z-\zeta)]\ d\zeta$$

$$\left. + \frac{\kappa\sigma}{\kappa+\sigma} \int_0^Z \frac{\overline{\rho}_L}{\overline{\rho}}\,(\zeta)\ \sinh\ [\Omega\ (Z-\zeta)]\ d\zeta \right\}$$

These solutions depend on the following boundary conditions:

$Z = 0, \quad u' = 0$

$r = 1, \quad v' = 0$

$r = 0, \quad v' < \infty$

To complete the analysis, the combustion terms must be considered in detail, and the boundary condition at the nozzle entrance (viz., the nozzle admittance condition) must be specified.

The preceding analysis has been for distributed combustion and has been limited to low Mach numbers. The extension of this analysis of high Mach numbers is so difficult it has not been successfully completed. An analysis at high Mach number has been performed using the simplifying assumption that the combustion is concentrated at a plane. Using this assumption, the conservation equations across the combustion front can be written as follows: (The + indicates before combustion and the - indicates after combustion)

Conservation of mass

$$u'_+ + \dot{m}'_b = \rho'_- \bar{u}_- + \bar{\rho}_- u'_- \tag{14a}$$

Conservation of momentum

$$\frac{p'_+ - p'_-}{\gamma} = \bar{u}_-^{\,2} \rho'_- + 2\bar{\rho}_- \bar{u}_- u'_- - \dot{m}'_b \bar{u}_L \tag{14b}$$

Conservation of energy

$$\bar{\rho}_+ \bar{h}_+ u'_+ + \dot{m}'_b \left(\bar{h}_L + (\gamma-1) \frac{\bar{u}_L^{\,2}}{2}\right) = (\gamma-1) \left(\frac{\bar{u}_-^{\,3}}{2} \rho'_- + \bar{\rho}_- \frac{\bar{u}_-^{\,2}}{2} u'_- + \bar{\rho}_- \bar{u}_-^{\,2} u'_-\right)$$

$$+ \bar{h}_- \bar{u}_- \rho'_- + \bar{h}_- \bar{\rho}_- u'_- + \bar{\rho}_- \bar{u}_- h'_- \tag{14c}$$

Equation of state

$$p'_- = \rho'_- + T'_- \tag{14d}$$

Definition of enthalpy

$$h' = T' \tag{14e}$$

Upstream of the combustion front the mean velocity is zero. If the interaction between the gas and drops is neglected, the dynamics of the region upstream of the combustion is governed by the homogeneous Helmholtz equation:

$$\nabla^2 p' - \sigma^2 p' = 0 \tag{14f}$$

Downstream of the combustion front a nozzle admittance condition exists.

For an annular chamber the modification to the cylindrical chamber analysis imposed by the annular geometry enters through the solution for ψ (r). The differential equation for ψ (r) is the classic Bessel equation which results in the following general solution:

$$\psi_{\nu\eta} (r) = C_1 J_\nu (s_{\nu\eta} \cdot r) + C_2 Y_\nu (s_{\nu\eta} \cdot r) \tag{15}$$

where J_ν is the Bessel function of the first kind, Y_ν is the Bessel function of the second kind, and $s_{\nu\eta}$ (where ν specifies the order of the Bessel equation) is the transverse acoustic mode number. Specifically, this means that at all chamber walls

$$\frac{d\psi_\nu}{dr} = 0 \tag{16}$$

Applying this condition to (16) yields

$$\psi'_\nu = 0 = J'_\nu (s_{\nu\eta} \cdot r_c) + \frac{C_2}{C_1} Y'_\nu (s_{\nu\eta} \cdot r_c) \tag{17}$$

at the outer wall, $r_c = 1$, equation (17) reduces to

$$J'_\nu (s_{\nu\eta}) + \frac{C_2}{C_1} Y'_\nu (s_{\nu\eta}) = 0 \ . \tag{18}$$

For the cylindrical chamber case, equation (18) reduces to

$$J' (s_{\nu\eta}) = 0 \tag{19}$$

since B must be zero because Y_ν becomes infinite at r = 0. The solution of equation (19) serves to define the transverse acoustic mode number, $s_{\nu\eta}$, for cylindrical combustion chambers.

I, B, Technical Description (cont.)

The annular chamber has its inner wall located in the range $0<r_i<1$. Therefore, for the inner wall, equation (17) is written:

$$J'_\nu (s_{\nu\eta} \cdot r) + \frac{C_2}{C_1} Y'_\nu (s_{\nu\eta} \cdot r) = 0, \ @ \ r = r_i \qquad (20)$$

whereas at the outer wall, $r = 1$ and equation (18) is still applicable. Solution of (18) and (20) simultaneously yields

$$J'_\nu (s_{\nu\eta}) \, Y'_\nu (s_{\nu\eta} \cdot R) - J'_\nu (s_{\nu\eta} \cdot R) \, Y'_\nu (s_{\nu\eta}) = 0 \qquad (21)$$

The solution of equation (21) defines the transverse acoustic mode number, $s_{\nu\eta}$, for annular chambers. Fortunately, equation (21) has been solved in Reference (9) and the values of $s_{\nu\eta}$ are listed as a function of R,where $R = r_i/r_o$, on Table II.

Thus the use of an annular chamber will alter the frequencies of the transverse modes. The extent of the alteration is illustrated in Figure 9. In addition, radial injection distribution effects will be minimized by the use of the annular chamber so that in most cases the distribution coefficients $A_{\nu\eta}$, $B_{\nu\eta}$, and $C_{\nu\eta}$ can be assumed to be unity.

TABLE II

BESSEL FUNCTION VALUE $(s_{\nu\eta})_{ann}$ FOR TANGENTIAL MODES IN ANNULAR CHAMBERS *

Annular Chamber Geometry		Tangential Modes					
(Radius) Inner / (Radius) Outer		1T	2T	3T	4T	5T	6T
0.910	$(s_{\nu\eta})_{ann} =$	1.04802	2.09602	3.14401	4.19197	5.23989	6.28778
0.833		1.092	2.1846	3.27672	4.368	5.4588	6.5496
0.667		1.209	2.412	3.61	4.80	5.98	7.14
0.500		1.35	2.68	3.98	5.18	6.34	7.45
0.400		1.41	2.85	4.10	5.27	6.40	7.5
0.333		1.54	2.91	4.17	5.31	6.42	7.5
0.286		1.60	2.99	4.18	5.31	6.42	7.5
0.250		1.64	3.00	4.18	5.31	6.42	7.5
0.222		1.67	3.02	4.18	5.31	6.42	7.5
0.200		1.70	3.025	4.18	5.31	6.42	7.5

NOTE: 1. Figure 9 gives the frequency ratio viz. $\frac{(s_{\nu\eta})_{annular}}{(s_{\nu\eta})_{cylindrical}}$ for different chamber geometries.

2. Calculation of annular chamber tangential frequencies

$$\left[f^* = \frac{C_o^* \, (s_{\nu\eta})_{annular}^{12}}{2\pi \, r_c^*} \right]$$

---

* Bridge and Angrist, "Math. of Computation, 16, 78," April 1962.

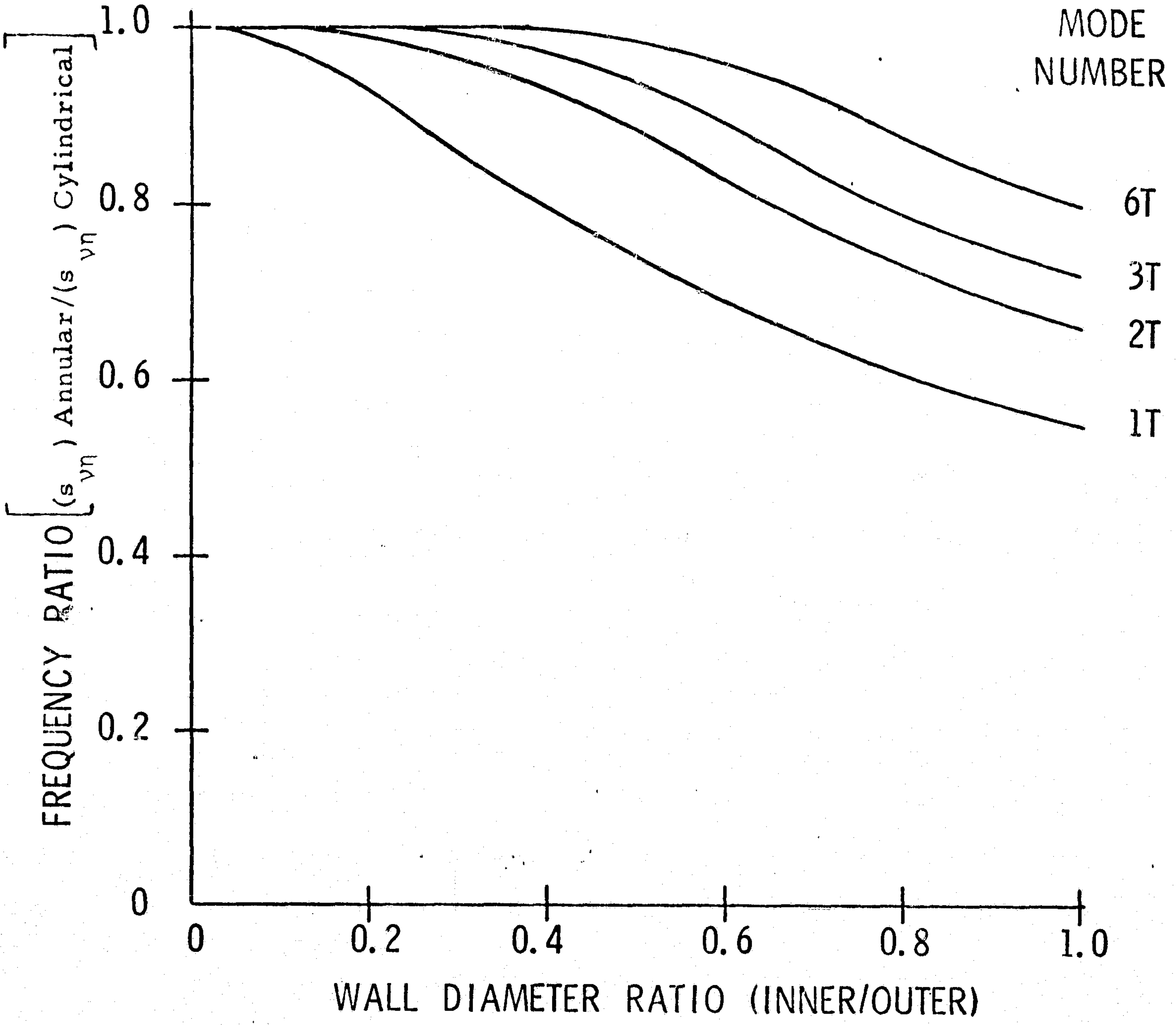


Figure 9 -- Tangential Mode Acoustic Frequencies for Annular Chambers

I, B, Technical Description (cont.)

(4) Combustion Response

(a) General Formulation

In Equation 10a combustion terms are written. It is necessary now to provide a mathematical formulation for the combustion terms in the flow equations in terms of the chamber conditions. As discussed previously our quantitative understanding of the actual combustion processes is not sufficient to provide a mathematical model. Fortunately the heuristic formulation based on the sensitive time lag concept seems to provide a good representation of the actual combustion response. The relation between time lag and burning rate is immediately found by considering that the fraction of propellant burning at a certain station in a time interval dt must have been injected during the interval $d(t - \tau_T)$. If, then $\dot{m}_b$ and $\dot{m}_i$ are the corresponding burning and injection rates we must have

$$\dot{m}_b dt = \dot{m}_i \, d(t - \tau_T)$$

In steady-state τ_T does not vary with time, and hence, if the injection rate is unaffected by the oscillations

$$\bar{\dot{m}}_b = \bar{\dot{m}}_i$$

From these two relations we obtain the fractional perturbation of the burning rate in the form

$$\frac{\dot{m}_b'}{\bar{\dot{m}}_b} = \frac{\dot{m}_b - \bar{\dot{m}}_b}{\bar{\dot{m}}_b} = - \frac{\partial \tau_T}{\partial_t} = \frac{-d\tau}{dt} \qquad (22)$$

I, B, Technical Description (cont.)

where, in accordance with the definition of sensitive time lag, the variation of τ_T with time is entirely due to the variation of τ, and where τ has been assumed to be the same for all propellant elements, and hence independent of the space coordinates.

Turning to the evaluation of $d\tau/dt$, a satisfactory mathematical description is obtained if one imagines that during the sensitive time lag certain preparatory processes, which need not be more precisely defined, take place at a rate depending on the local, instantaneous values of quantities representing the state and the motion of the gas and droplets. When these preparatory processes, integrated over the duration of the sensitive time lag, reach a certain fixed level, the conversion into hot gases takes place abruptly. It is clear, then, that when the state and the motion conditions vary, also the duration of the sensitive (and hence, the total) time lag will vary, resulting in a variable rate of gas production.

The quantitative formulation follows at once. If the rate of the preparatory processes is given by a function $f\ (p, T, v, \ldots.)$ of the pertinent values of the pressure, temperature, any representative velocity v (for instance the radial gas velocity), and possibly other quantities representing the conditions in the chamber, the sensitive time lag τ for an element burning at time t will be given by the equation

$$\int_{t-\tau}^{t} f\ (p, T, v, \ldots.)dt_1 = \text{const.}$$

where t_1 represents the burning time. Here the values of p, T, v, ... must be evaluated not only at time t_1, but also at the position where the particular propellant element finds itself at that time. Since the above relation must be satisfied also in steady operation, indicating the steady-state quantities with a superimposed bar, we must have

I, B, Technical Description (cont.)

$$\int_{t-\tau}^{t} f\,(p,\, T,\, v,\, \dots .)\; dt_1 = \int_{t-\bar{\tau}}^{t} f(\bar{p},\, \bar{T},\, \bar{v},\, \dots .)dt_1 \,. \quad (23)$$

Now we introduce the perturbations, such that

$$p = \bar{p} + p',\ T = \bar{T} + T',\ v = \bar{v} + v' \ \dots .$$

and expand the rate function in a Taylor series

$$f(p,\, T,\, v,\, \dots .) = \bar{f} + \bar{f}_p\, p' + \bar{f}_T T' + \bar{f}_v\, v' + \dots .$$

where $\bar{f} = f(\bar{p},\, \bar{T},\, \bar{v},\, \dots .)$ and similarly for the partial derivatives $\bar{f}_p$, $\bar{f}_T$, $\bar{f}_v$ of f with respect to the subscripts. Observe that, under the small perturbations assumption, the Taylor series must be stopped after the first order terms.

If it is assumed that the temperature is a function only of the pressure (for instance, through the isentropic defining relation), $T' = p'\,(dT/dp)$. Then, defining the nondimensional interaction indices n, 1, ... as

$$n = \frac{\bar{f}_p + \bar{f}_T\,(dT/dp)}{\bar{f}},\ 1 = \frac{\bar{f}_v}{\bar{f}},\ \dots . \qquad (24)$$

$$f(p,\, T,\, v,\, \dots .) = \bar{f}\,(1 + n\,p' + 1v' + \dots .)$$

and Equation (23) can be written in the form

I, B, Technical Description (cont.)

$$\int_{t-\tau}^{t-\bar{\tau}} \bar{f}\,(1 + np' + l v' + \ldots.,)dt_1 + \int_{t-\bar{\tau}}^{t} \bar{f}(1 + np' + l v' + \ldots.)dt_1$$

$$= \int_{t-\tau}^{t} \bar{f}\,dt_1$$

Here the integration interval at the L.H.S. of Equation (23) has been split into two parts. The first interval, from $t - \tau$ to $t - \bar{\tau}$ is of duration $\bar{\tau} - \tau$ and hence of the order of the perturbation of the time lag. Hence, compared with the other two integrals, the first integral is of the order of a perturbation. As a result in its evaluation one can disregard in the integrand the terms containing the perturbations which, in view of the small perturbation assumption, would result in a negligible second order contribution. Then, simplifying, the above equation becomes

$$\int_{t-\tau}^{t} \bar{f}\,dt_1 = \int_{t-\bar{\tau}}^{t} \bar{f}(np' + lv' + \ldots.)dt_1$$

In the combustion zone $\bar{p}$ is approximately constant, and if v is the radial gas velocity, so that $\bar{v} = 0$, then $\bar{f}\,(\bar{p}, \bar{v})$ is a constant, and so are n and l.

Then we obtain simply

$$\bar{\tau} - \tau = n \int_{t-\bar{\tau}}^{t} p'\,dt_1 + l \int_{t-\bar{\tau}}^{t} v' dt_1 + \ldots$$

or, differentiating with respect to t

I, B, Technical Description (cont.)

$$-\frac{d\tau}{dt} = n\left[p'(t) - p'(t - \bar{\tau})\right] + l\left[v'(t) - v'(t - \bar{\tau})\right] + \ldots. \qquad (25)$$

Again, it must be specified that while p'(t) is evaluated at the conversion instant t at the location where the conversion takes place, $p'(t - \bar{\tau})$ must be evaluated not only at time $t - \bar{\tau}$, but also at the location where the propellant was at that time. However, the displacement of the propellant during the time $\bar{\tau}$ produces an effect of second order in Equation (25). As a reasonable approximation, therefore, both p'(t) and $p'(t - \bar{\tau})$ can be evaluated at the station when the conversion into burned gases takes place. And, of course, the same applies to the velocity effect in Equation (25), and to other possible effects.

In Equation (25) only the pressure sensitivity and the radial velocity sensitivity are explicitly considered. Concerning the last it must be added that other components of the gas velocity can be treated in exactly the same fashion. If one is interested, for instance, in the effect of the transverse non-uniformity of the gas composition, then also the tangential velocity component is relevant, and correspondingly another velocity sensitive term must appear in Equation (25), which becomes, in the absence of other interactions

$$-\frac{d\tau}{dt} = n\left[p'(t) - p'(t - \bar{\tau})\right] + l_r\left[v'(t) - v'(t - \bar{\tau})\right]$$

$$+ l_\theta\left[w'(t) - w'(t - \bar{\tau})\right] \qquad (26)$$

It must be observed that, actually, when only the effects of the nonuniform gas composition on the burning rate are sought, what counts is the displacement of the gases with respect to the droplets, rather than the relative velocity. This can be formalized by writing, for instance, instead of Equation (26).

$$-\frac{d\tau}{dt} = n\left[p'(t) - p'(t-\bar{\tau})\right] + m_r\left[\delta'_r(t) - \delta'_r(t-\bar{\tau})\right]$$

$$+ m_\theta\left[\delta'_\theta(t) - \delta'_\theta(t-\bar{\tau})\right] \qquad (27)$$

where m_r and m_θ are two displacement indices relative to the radial and tangential displacements δ_r, δ_θ respectively. The two formulations (26) and (27) are closely correlated, because of the relations existing between velocities and displacements.

If the time dependence of the perturbations in Equations (26) and (27) is taken to be exp (st), the combined effect is:

$$-\frac{d\tau}{dt} = -(\mathcal{P}p' + \mathcal{R}_v v' + \mathcal{T}_v w') \text{ or } -(\mathcal{P}p' + \mathcal{R}_\delta \delta'_r + \mathcal{T}_\delta \delta'_\theta) \qquad (28)$$

where the quantities defined by

$$\frac{\mathcal{P}}{n} = \frac{\mathcal{R}}{l_r} = \frac{\mathcal{T}}{l_\theta} = \frac{\mathcal{R}_\delta}{m_r} = \frac{\mathcal{T}_\delta}{m_\theta} = 1 - e^{-\sigma\bar{\tau}}, \quad (\sigma = \lambda + i\omega) \qquad (29)$$

are to be interpreted as feedback factors. In this case the relation between velocities and displacements is simply $v' = \sigma\delta'_r$, $w' = \sigma\delta'_\theta$ so that the same expression can be used for $-\frac{d\tau}{dt}$ for velocity or displacement effects if $\mathcal{R}_\delta$ is equivalent to $\sigma\mathcal{R}$ and $\mathcal{T}_\delta$ is equivalent to $\sigma\mathcal{T}$.

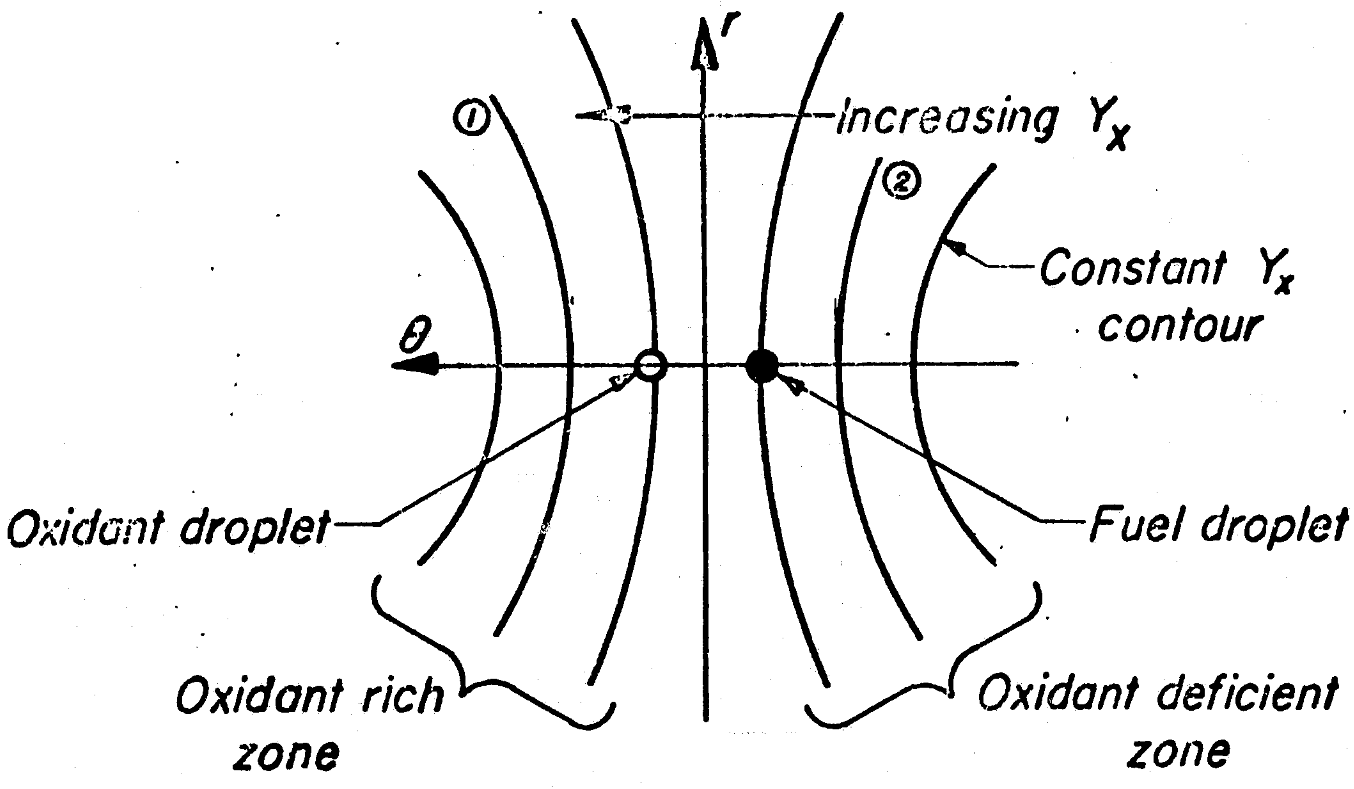
(b) Velocity Sensitive Combustion

It is desirable to take a closer look at the dynamic aspects of velocity sensitive combustion. The most significant velocity effects are those resulting from the radial and tangential components of the gas velocity perturbation. In the case of purely transverse modes, the longitudinal velocity perturbation component is always much smaller than the transverse components. In addition, the longitudinal component vanishes at the injector face and has its smallest magnitude in the early combustion zone, the region which appears to have the greatest significance for transverse modes. It is possible that in certain combustion chambers the axial spreading of the combustion will result in sizable longitudinal velocity oscillations for higher order longitudinal or combined transverse-longitudinal modes. However, in such cases, the pressure perturbation will become correspondingly small in that region, thus, the decreased pressure effect will cancel the increased velocity effect. In the present analysis, therefore, only the effects of the transverse velocity oscillations will be considered in the combustion response.

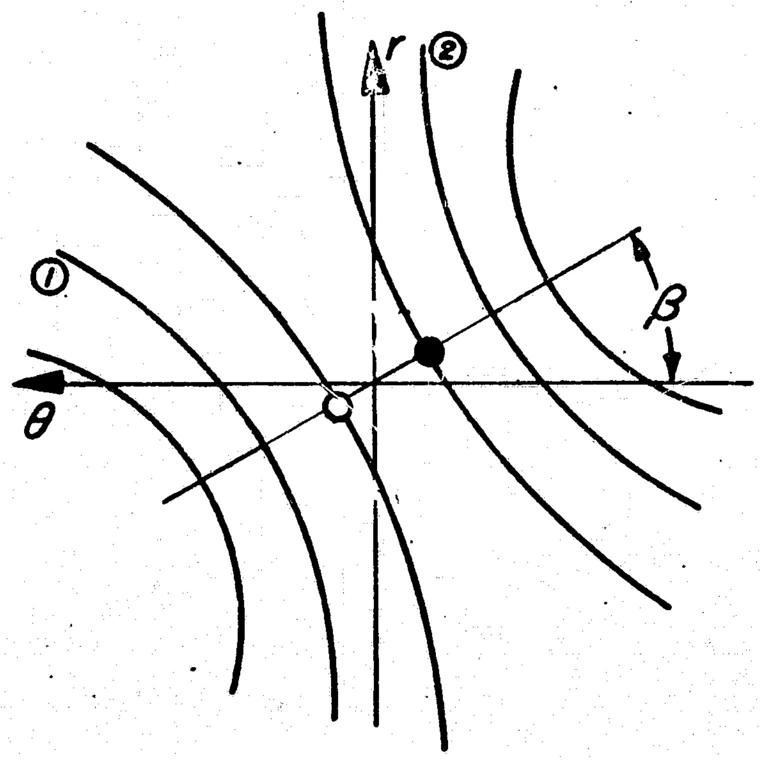
Of the various intermediate processes occurring during the combustion of liquid bipropellants, those most sensitive to velocity are the vaporization of the liquid droplets and the mixing of the vaporized propellants that must precede chemical reaction. The theoretical study of unsteady vaporization by Wieber and Mickelsen (Ref. 10) indicates that the evaporation rate is dependent on the absolute magnitude of the relative velocity between droplet and gas; therefore, the vaporization velocity effect is seen to be essentially nonlinear, and cannot be treated within the framework of a linearized theory. On the other hand, the mixing of the propellants by the oscillating velocities may be linearized, and gives rise to important modifications of the stability behavior of a combustor. Although no detailed description of such a complex phenomenon is now possible, the following discussion illustrates one process by which the burning rate may be caused to oscillate by an oscillating transverse gas velocity.

Consider first the mixture of gaseous combustion products, vaporized propellants (oxidizer and fuel), and liquid propellant droplets at some axial station downstream from a fuel-on-oxidizer impinging doublet injector element. Since liquid mixing is imperfect, some stratification will exist in the mixture. For concreteness, assume that the line of centers of the doublet is aligned tangentially (i.e., normal to a radius). Then the stratification is almost entirely in the tangential direction, as shown schematically in Figure 10 by the lines of constant mass fraction of vaporized oxidizer in the mixture Y_x. The exact shape of the constant Y_x contours will be dependent on the injector design, operating conditions, and propellant characteristics. Because of the turbulence in the combustion chamber, the stratification pattern shown represents only a mean condition.

As a droplet evaporates, the vapor diffuses away and must mix with the other vaporized propellant in the propellant proportions for chemical reaction. In a rocket combustor, the transport and mixing are most likely to be carried out by turbulence rather than by molecular diffusion. The overall burning rate of a fuel-rich droplet will, therefore, be a function of the amount of oxidizer vapor near the droplet. In the presence of small, periodic, tangential gas velocity oscillations ($w' e^{i\omega t}$) the gaseous mixture will be displaced relative to the droplets, causing oscillations of the local mass fractions of both oxidizer and fuel. Since a fuel droplet is in an oxidant-deficient region, a velocity perturbation which increases the oxidizer fraction in the vicinity of the droplet will increase the contribution of that droplet to the overall burning rate. The opposite is true for an oxidizer-rich droplet subject to the same perturbation, since an oxidizer fraction increase corresponds to a fuel fraction decrease. Thus, the effects of the same velocity perturbation on the two droplets will tend to cancel, unless the propellants have significantly different vaporization rates. In the latter case, at any axial station, there will be a greater number of droplets of the less-volatile propellant, and summation



(a) Spray produced by tangentially oriented injector spud



(b) Spray produced by rotated spud

Figure 10 -- Sprays Produced by Various Orientations of the Injector Spud

of the velocity effect over all of the droplets in the spray will result in a net contribution to the burning rate. This contribution will clearly depend on the amplitude of the velocity as well as its direction. For small perturbations, and for the doublet spray shown in Figure 10a, the burning rate contribution can be written in the form

$$f' = 1w'e^{i\omega t}$$

where 1 is a velocity interaction index analogous to the pressure interaction index defined by Crocco. In the case of an arbitrarily oriented spray, such as shown in Figure 10b, the burning rate perturbation due to velocity effects becomes

$$f' = (1_r v' + 1_\theta w')e^{i\omega t} \qquad (30)$$

so that, in general, two velocity indices are necessary.

It is clear that this linearized expression will not be valid for all types of injection patterns. For example, approximately linear effects can be expected with a fuel-on-oxidizer doublet and for a like-on-like pattern if the spacing between unlike fans is sufficiently small. However, for large spacings, nonlinear velocity effects must be taken into consideration.

At present, the magnitudes of the velocity indices cannot be calculated because of the lack of quantitative knowledge of the processes involved in liquid propellant combustion under turbulent conditions.

I, B, Technical Description (cont.)

It is also possible to formulate the above discussion in terms of displacement interaction indices. Letting the radial and transverse components of the displacements be $\delta'_r e^{i\omega t}$ and $\delta'_\theta e^{i\omega t}$, the net combustion process rate perturbation can be written

$$f' = (m_r\delta'_r + m_\theta\delta'_\theta)e^{i\omega t} \tag{31}$$

It is clear that $m_r = i\omega l_r$ $m_\theta = i\omega l_\theta$. Thus, the displacement indices present at 90° phase shift with respect to the velocity indices.

The analysis of the effects of velocity (or displacement) sensitivity on the stability of a combustor is considerably simplified by assuming that the velocity effects occur during the same time interval (the sensitive time lag) as the pressure effects. In this case, the burning rate perturbation becomes

$$Q' = \overline{Q}(\mathcal{P}p' + \mathcal{R}v' + \mathcal{T}w') \tag{32}$$

where

$$\mathcal{P} = n\,(1-e^{-\sigma\tau})$$
$$\mathcal{R} = l_r\,(1-e^{-\sigma\tau})$$
$$\mathcal{T} = l_\theta\,(1-e^{-\sigma\tau})$$

In Equation (32), additional simplifications have been introduced by assuming that all propellant elements have equal mean sensitive time lags, and that the space lag associated with the sensitive time lag is a negligible fraction of the wave length. In general, of course, the mean sensitive time lag varies from one propellant element to another. Crocco and Cheng have shown that this nonuniformity of the sensitive time lag leads to increased stability of the combustor. Therefore, the assumption of a uniform time lag produces a conservative stability prediction.

It would be possible to generalize the burning rate expression to allow for different time lags for pressure and velocity effects. However, both mathematical and physical considerations indicate the desirability of the simpler formulation.

(c) Approximate Treatment of Nonlinear Combustion Response

Nonlinearities associated with oscillatory combustion chamber operation can derive from two sources: (1) the fluid mechanical behavior of the gases in the chamber, and (2) the dynamics of the combustion process. It is clear that significant interactions between the two kinds can also occur. The studies of Priem and Guentert have shown that combustion process nonlinearities can be important even for oscillation amplitudes less than 20% of the mean chamber pressure. Thus, it is worthwhile to consider nonlinearity of the combustion response while retaining the linearized fluid mechanical analysis with its attendant simplification.

To insert the nonlinear combustion dynamics into the framework of the linear theory, some method of equivalent linearization must be used. The method selected in this analysis is the "describing function" method.

When a sinusoidal signal is input to a nonlinear element, the output will not, in general, be sinusoidal. Fourier analysis of the output will reveal many frequency components, among which is one (the fundamental) that corresponds to the frequency of the input signal.

For the analysis of stability, only the fundamental frequency component is required, as shown by Reardon (Ref 8). An equivalent linear transfer function for the nonlinear element can be defined as the ratio of the fundamental component of the output to the input. Thus, if

$$I(t) = Ie^{i\omega_f t}$$

is the input, and

$$O(t) = O_1(\omega_f)e^{i\omega_f t} + \sum_{j \neq 1} O_j e^{ij\omega_f t} \qquad (33)$$

is the output, the equivalent linear transfer function is

$$TF = \frac{O_1(\omega_f)}{I}$$

For nonlinearities that can be treated by this method, linear behavior is obtained for limiting values of the input (e.g., $I \to 0$ or $I \to \infty$). It is convenient to define a "describing function" F as

$$F(\omega) = TF/(TF)_{LIM} \qquad (34)$$

where $(TF)_{LIM}$ is the limiting linear transfer function. Thus, a linear analysis can be extended to include isolated nonlinear effects by replacing the linear transfer function of the nonlinear element by $F(\omega) \cdot (TF)_{LIM}$.

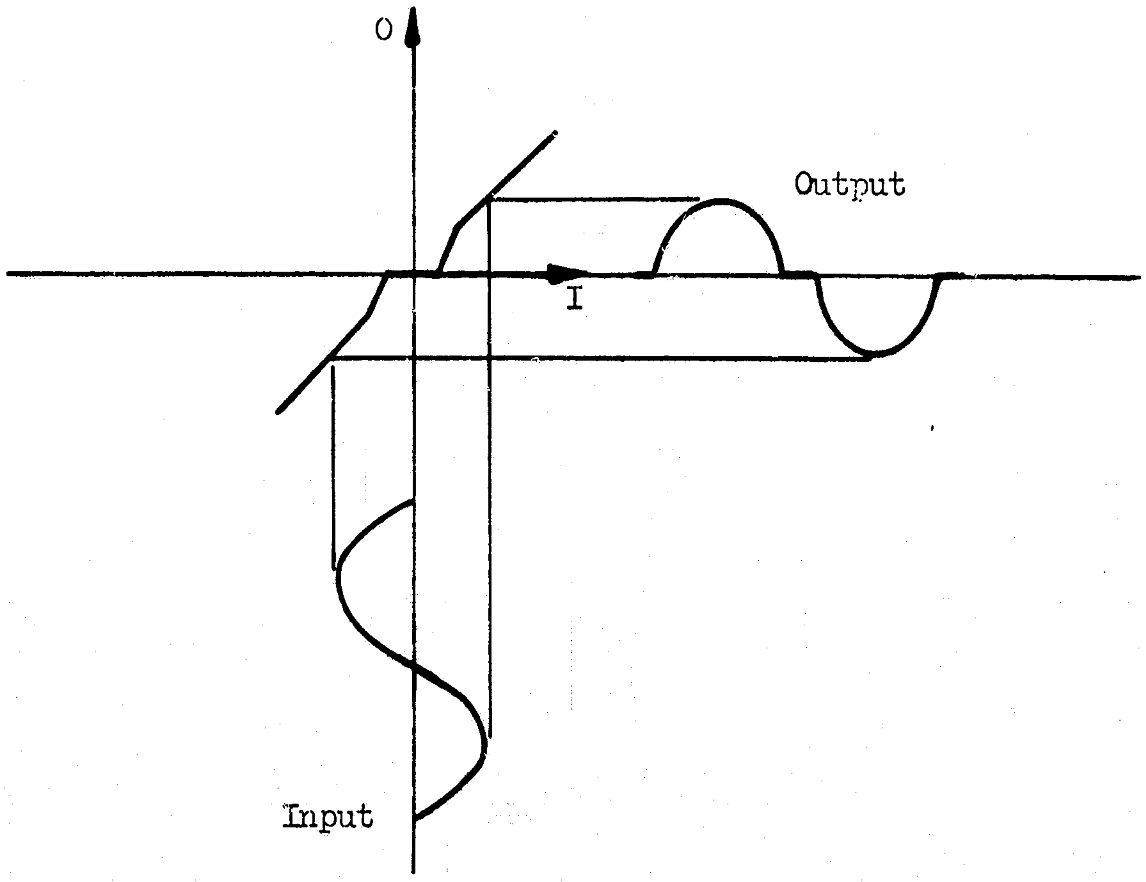
In applying this approach to the combustion instability problem, it is assumed that the only significant nonlinearities are those associated with the response of the combustion process to pressure and velocity perturbations. Three describing functions are required, corresponding to the combustion response to pressure, radial velocity, and tangential velocity perturbations. Thus, the combustion rate perturbation becomes

$$Q' = \overline{Q} \left[F_P(p') \mathcal{P}p' + F_R(v') \mathcal{R}v' + F_T (w') \mathcal{T}w' \right] \qquad (35)$$

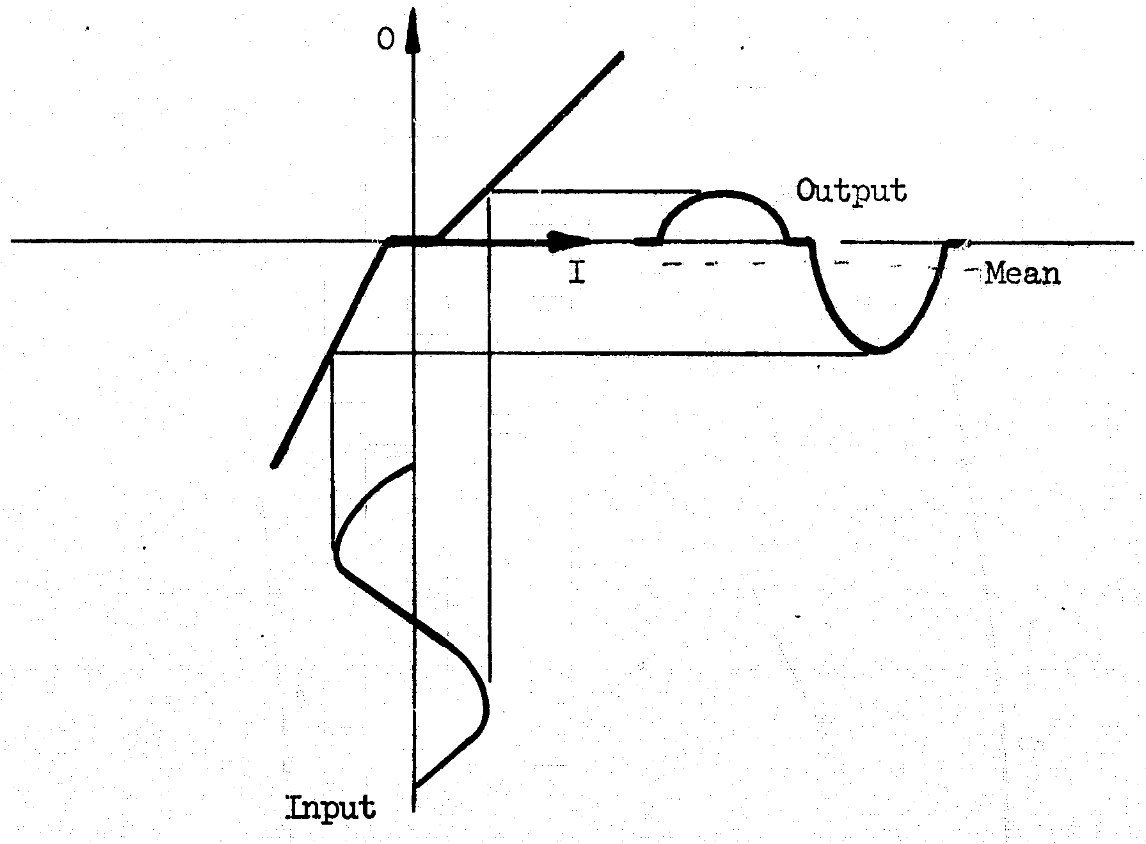
In the above expression, the dependence of the describing functions on the perturbation (input) amplitude is shown explicitly. This dependence on amplitude introduces complications into the solution of the perturbation equations. In general, the input amplitude is a function of the axial, as well as the transverse, space coordinate. To obtain a solution, it is necessary to introduce the additional simplification of neglecting the axial variation of perturbation amplitude in the evaluation of the describing function. For purely transverse modes, this is not an unreasonable approximation, and it breaks down significantly only for higher-order longitudinal modes. The error made by using the perturbation amplitudes at the injector face will be small.

To calculate the describing functions, it is necessary to know the shape of the combustion response to each perturbation. Since it is assumed that the effects are independent of each other, the burning rate perturbation can be written

$$\frac{Q'}{\overline{Q}} = \phi_p (P') + \phi_R (v') + \phi_T(w')$$



(a) Response Function with Odd-Symmetry



(b) Asymmetric Response Function

Figure 11 -- Examples of Nonlinear Response Functions

The procedure for calculating each describing function is the same; therefore, it is necessary only to discuss one, say, the pressure-effect describing function.

The input perturbation is $p' = Po \cos\chi$ where $\chi = \omega t$ and, from Fourier analysis the fundamental term of the output series is

$$\phi_P (P_o) = \frac{1}{\pi} \int_{C-\pi}^{C+\pi} \phi_P(P_o \cos\chi) e^{-i\chi} d\chi$$

Thus, the describing function is given by

$$F_P = \frac{1}{\pi P_o (TF)_{LIM}} \int_{C-\pi}^{C+\pi} \phi_P(P_o \cos\chi) e^{-i\chi} d\chi \qquad (36)$$

In the expressions above, C is an arbitrary constant, and $(TF)_{LIM}$ is a suitable normalizing factor, such that $F \to 1$ for equivalent linear operation. The choice of $(TF)_{LIM}$ depends on the characteristics of each nonlinear response function, and a general rule does not appear feasible.

The describing function method applies well to nonlinearities with odd symmetry (Figure 11a). It is not applicable to response functions with even symmetry, such as the velocity effect on vaporization, since there is no contribution to the fundamental term of the Fourier series. An intermediate case is that of asymmetric response function (Figure 11b). In this case, there will be a significant contribution to the fundamental oscillation, but a change in the mean burning rate as well. This change in the mean burning rate occurs only during the sensitive portion of the total time lag and so will have a negligible influence on the steady state solution.

I, B, Technical Description (cont.)

In general, the describing function is complex; that is, the nonlinear combustion response introduces a phase shift as well as an amplitude change. However, for response functions with odd symmetry, the fundamental component of the output is in phase with the input, so that the describing function is real.

(5) Effect of Non-Uniform Injection

The non-uniformity of mass injection in the r and θ direction can be characterized by a burning rate distribution function $\mu(r, \theta)$ which is the ratio of the local injection density to the mean injection density so that

$$\pi = \int_0^1 \int_0^{2\pi} \mu r dr d\theta .$$

When $\overline{Q}$ is multiplied by μ in Equations 12b, terms on the right hand side are known functions of r and θ, as well as Z. The solution of this type of inhomogeneous differential equation can be expressed as a Bessel-Fourier series as follows:

$$p_1 = (A_{\nu\eta} P_{a\nu\eta} + B_{\nu\eta} P_{b\nu\eta} + C_{\nu\eta} P_{c\nu\eta} + P_d) \Psi_{\nu\eta} \theta_\nu +$$

$$\sum_{\substack{p \neq \nu \\ q \neq \eta}} \sum (A_{pq} P_{apq} + B_{pq} P_{bpq} + C_{pq} P_{cpq}) \Psi_{pq} \theta_p$$

where the subscript a indicates pressure effects (associated with $\mathcal{P}$) b radial velocity effects (associated with $\mathcal{R}$), c tangential velocity effects (associated with $\mathcal{T}$) and d damping effects (which are independent of r and θ).

I, B, Technical Description (cont.)

When the orthogonality property of the functions Ψ and θ is used on the system characteristic equation the coefficients $A_{\nu\eta}$, $B_{\nu\eta}$ and $C_{\nu\eta}$ are found to be:

$$A_{\nu\eta} = \frac{\int_0^1 \int_0^{2\pi} \mu(r,\theta)\ \Psi^2_{\nu\eta}\ \theta_\nu \theta_\nu^*\ r\ dr\ d\theta}{\int_0^1 \int_0^{2\pi} \Psi_{\nu\eta}^2\ \theta_\nu\ \theta_\nu^*\ r\ dr\ d\theta}$$

$$B_{\nu\eta} = \frac{\int_0^1 \int_0^{2\pi} \mu(r,\theta)\ \frac{d\Psi_{\nu\eta}}{dr}\ \Psi_{\nu\eta}\ \theta_\nu\ \theta_\nu^*\ r\ dr\ d\theta}{\int_0^1 \int_0^{2\pi} \Psi_{\nu\eta}^2\ \theta_\nu\ \theta_\nu^*\ r\ dr\ d\theta} \tag{37}$$

$$C_{\nu\eta} = \frac{\int_0^1 \int_0^{2\pi} \mu(r,\theta)\ \Psi^2(r,\theta)\ \frac{d\theta_\nu}{d\theta}\ \theta_\nu^*\ dr\ d\theta}{\int_0^1 \int_0^{2\pi} \Psi_{\nu\eta}^2\ \theta_\nu\ \theta_\nu^*\ r\ dr\ d\theta}$$

In section I,B,4,b (7) it is seen how these coefficients affect the characteristic equation. It should be noted that, if the energy addition follows a non-linear relationship, a describing function must also be included in the above coefficients.

(6) Nozzle Admittance

In any rocket combustion instability analysis, it is desirable to apply a boundary condition at the nozzle entrance to describe the effect of the nozzle upon wave motion in the combustion chamber. In a linearized analysis, this boundary condition is written in the form of an admittance relation; that is, a linear relation between the perturbations of two thermodynamic properties and of the velocity components. The coefficients in this relation are termed admittance coefficients and are calculated by means of an analysis of the oscillatory flow in the nozzle. In this section, the analysis and numerical integration which lead to the determination of these coefficients are discussed.

The divergent portion of the supercritical nozzle need not be analyzed; all that is pertinent is the subsonic flow in the convergent portion since any disturbances to the supersonic flow cannot propagate upstream through the throat. Therefore, disturbances in the subsonic portion of the nozzle and in the chamber are neither affected nor caused by disturbances in the supersonic region. (The opposite, however, is not true.)

To date, two types of nozzles have been analyzed: axisymmetric designs and two-dimensional designs. The axisymmetric case is presently the one of the most practical significance and is the one to be discussed here. The two-dimensional case applies to thin annular chambers and to certain experimental configurations. The analyses of the two cases are similar; details of both are given in References 11 and 12.

The unperturbed, or steady-state, flow is considered to be one-dimensional in order to simplify the analysis. The perturbed flow, however, may be three-dimensional. The combustion process is

I, B, Technical Description (cont.)

assumed to be completed before the flow enters the nozzle so that there are no source terms in the differential equations of motion. The equations do allow for the occurrence of entropy waves and vorticity waves in the nozzle due to the combustion chamber.

The three-dimensional coordinate system (Figure 12) employs the values of the velocity potential ϕ and the stream function ψ of the unperturbed flow in addition to the azimuthal angle, with

$$\bar{q} = \frac{\partial \phi}{\partial s}$$

$$r \rho \bar{q} = \frac{\partial \psi}{\partial n}$$

where s is the streamline direction and n is the direction normal to the streamline. Since the value of the stream function is a constant at the nozzle walls where the boundary conditions are applied, separation of variables is allowed.

Under the usual assumption of small-amplitude oscillations, linear partial differential equations are obtained that govern the perturbations. These equations are separated under the assumption that the nozzle is sufficiently long that the cosine of the semi-angle of convergence may be approximated by unity. The time and azimuthal dependencies are given by sinusoidal functions. The radial dependencies are given in terms of Bessel functions of the first kind and their derivatives. The axial dependencies are related to the solution to a certain second-order linear ordinary differential equation with complex-variable coefficients which can only be obtained in exact form by numerical integration.

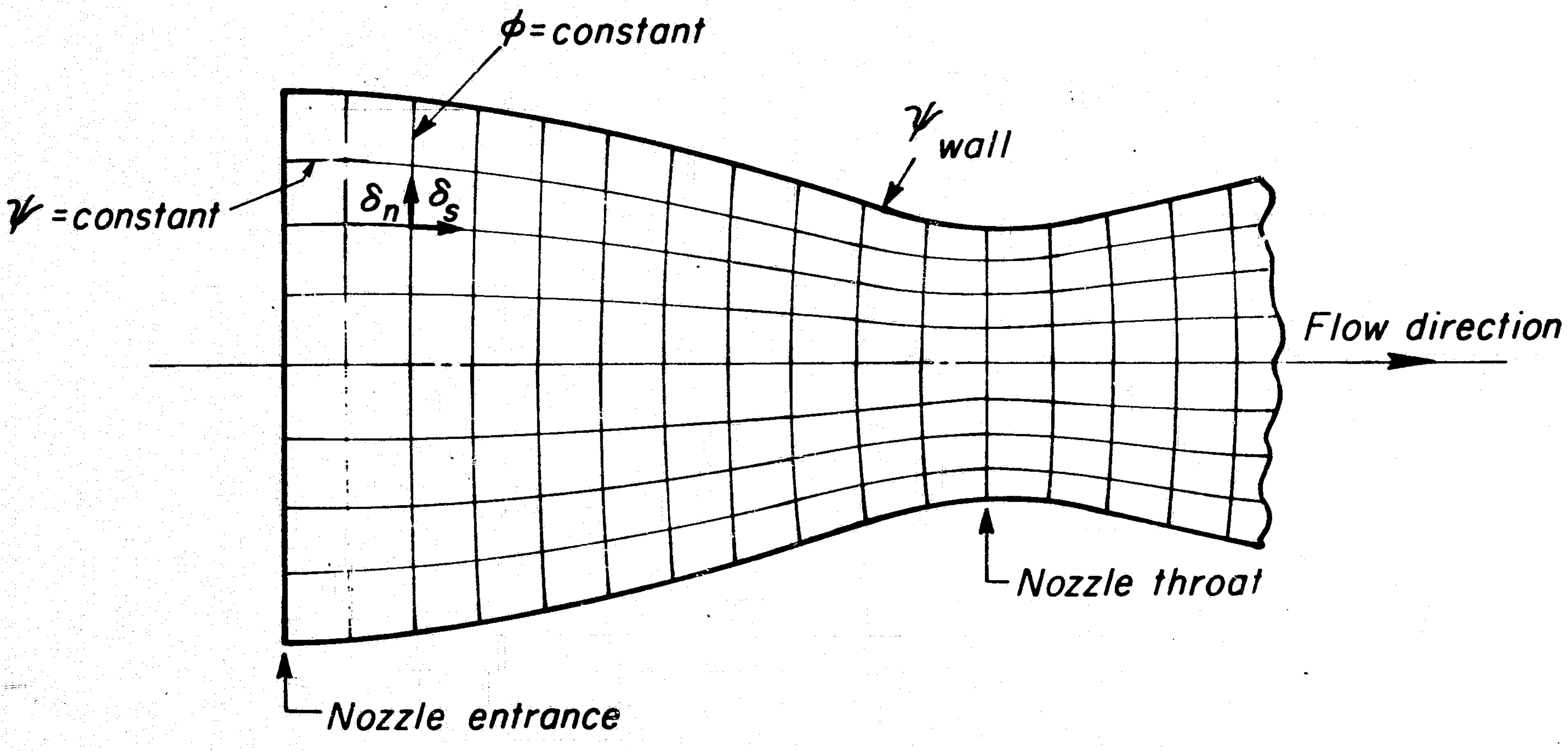


Figure 12 -- Exhaust Nozzle Coordinate System

This second-order differential equation is singular at the throat; one of the homogeneous solutions will be regular there and the other one will be singular. The singular solution is cast away. This procedure has been demonstrated to be equivalent to disallowing perturbations to propagate upstream from the supersonic portion of the nozzle (Ref 6).

It has been shown that the admittance coefficients are functions of the solutions to certain first order equations that are obtained by reduction of the original second-order equation. So, while it would be necessary to integrate the second-order equation in order to determine the variation of the flow properties, it is not necessary for the purpose of determining the admittance coefficients. Since the interest lies in the prediction of global stability characteristics and not in the details of the flow itself, only the equations immediately needed to determine the admittance coefficient will be presented. The derivations and additional analyses may be found in Reference 11.

The admittance coefficients for a given geometry are determined as functions of the axial coordinate or, equivalently, of the local mean-flow Mach number. This implies that, when the admittance coefficient at the nozzle entrance is desired, the axial coordinate at the entrance or the entrance Mach number must be known before the admittance coefficients can be determined.

The linear admittance condition is given by

$$U + \mathcal{A}P + \mathcal{B}V + \mathcal{C}S = 0 \tag{38}$$

where U, P, V, and S are the axial dependencies of the nondimensional perturbations of axial velocity, pressure, radial velocity, and entropy, respectively.

The annular nozzle analysis used in this program is based on Reference 11. It contains certain inconsistencies which should be aired. The current analysis is an approximation to an annular nozzle analysis which is expected to be exact for a certain case.

The cylindrical nozzle analysis postulates the existence of streamlines and a velocity potential. The equations of motion are written in terms of stream functions and velocity potentials. The limits of integration in the radial direction must be written in terms of these streamlines. For a conventional nozzle these limits are the nozzle centerline and the outer wall, which are both streamlines. To make an annular analysis from the conventional nozzle analysis with a minimum of fuss, a different streamline can be chosen for the inner boundary. A new corresponding value of $s_{\nu\eta}$ can also be determined. It is clear that once the outer boundary is chosen the number of possible inner boundaries is severely limited to one which is a constant fraction of the outer radius.

The nozzle admittance analysis requires a table of the square of the velocity versus the velocity potential. For a general annular nozzle inner and outer contour this can be calculated by first calculating area ratios versus distance, then velocities versus distance, and the velocity potential from velocity and distances. As this program is written, the table of velocity squared versus velocity potential is calculated for a very general annular nozzle configuration. It should be clear from the discussion above that the configuration for which the admittance is obtained is not in general the configuration originally input.

Basic changes to the nozzle admittance analysis are needed to rectify this limitation.

(7) Characteristic Equation

Solution of the perturbation equations gives the pressure perturbation in the form of a Bessel-Fourier series:

$$p' = P_{oo}\,[P_{\nu\eta}(Z)\,\Psi_{\nu\eta}(r)\,\theta_{\nu}(\theta) + \sum_{\substack{p \neq \nu \\ q \neq \eta}} \sum P_{pq}(Z)\,\psi_{p\bar{f}}(r)\,\theta_{p}(\theta)]$$

In this expression the indices ν and η refer to the fundamental term, that is, to the oscillatory mode under consideration. The other terms in the series account for the distortion introduced by flow, injection distribution, velocity effects, and nonlinear combustion response. For values of the indices p, q different from ν, η, each term in the series includes an integration constant. However, the integration constant for the fundamental term can be shown to be of order M^3, and so can be neglected in the present analysis. The constant P_{oo} represents the perturbation amplitude level; in this linearized analysis, the amplitude has no effect on the stability solution.

Since the perturbation solution is obtained in the form of a series, the nozzle admittance boundary condition must be applied term by term. For each p, q $\neq$ ν, η, the nozzle boundary condition can be used to determine the integration constant. Application of the remaining condition,

$$U_{\nu\eta}(Z_e) + \mathcal{A}P_{\nu\eta}(Z_e) + \mathcal{B}V_{\nu\eta}(Z_e) + \mathcal{C}S(Z_e) = 0 \qquad (39)$$

results in an eigenvalue problem. That is, this equation is the characteristic equation for the eigenvalues $\sigma = \lambda + i\omega$. For a given combustor geometry and for a given value of the combustion parameters n, τ, l_r and l_θ, the characteristic equation can, in principle, be used to determine the frequency of oscillation ω and the growth rate λ of the perturbation amplitude.

However, since the coefficients of the characteristic equation are function of the variable σ, it is more convenient to regard σ as one of the independent variables. Considerable simplification results if $\lambda = 0$, that is, $\sigma = i\omega$, which is interpreted physically as an oscillation, the amplitude of which neither grows nor decays with time. The neutral condition is clearly the boundary between stable and unstable operation, and is sometimes referred to as the stability limit.

For neutral oscillations, regarding the frequency as an independent variable, the characteristic equation becomes a relation between the combustion parameters. Since the equation is complex, two of the combustion parameters can be determined in terms of the other two. It is natural to select the sensitive time lag τ and the pressure interaction index n as the dependent variables, because these parameters are significant to all modes of oscillation. Following this procedure, the characteristic equation can be written in the form:

$$n\,(1 - e^{-i\omega\tau}) = \frac{h\ (\omega)}{A_{\nu\eta} + \frac{i}{\gamma\omega}\,(B_{\nu\eta}\,\frac{l_r}{n} + C_{\nu\eta}\,\frac{l_\theta}{n})} = \tilde{h}_r + i\,\tilde{h}_i\ . \qquad (40)$$

The solution is

$$n\,(\omega, \frac{l_r}{n}, \frac{l_\theta}{n}) = \frac{\tilde{h}_r^{\,2} + \tilde{h}_i^{\,2}}{2\,\tilde{h}_r} \qquad (41)$$

I, B, Technical Description (cont.)

$$\tau\ (\omega, \frac{l_r}{n}, \frac{l_\theta}{n}) = \tan^{-1} (\frac{\tilde{h}_i}{n - \tilde{h}_r}) \qquad (42)$$

where τ is determined to within an additive constant $\frac{2\pi}{\omega}$.

A typical solution for n (ω) and τ (ω) for assumed values of the velocity indices is shown in Figure 13. This solution applies at the stability limits, where $\lambda = 0$. It can be seen that for any value of τ only one value of n is consistent with neutral oscillations. For the same τ, a larger n corresponds to instability ($\lambda > 0$) and a smaller n to stability ($\lambda < 0$). In the case of transverse modes, for values of n in the range of interest (≤ 2), the frequency varies over a very narrow range, and is very nearly equal to the corresponding acoustic frequency. For longitudinal modes, both the frequency range and the departure from the acoustic-mode frequency are somewhat larger. The narrow frequency range result is related directly to the fact that high frequency instability involves the interaction of the combustion chamber. The chamber acoustics are somewhat modified by the presence of the exhaust nozzle and the mean gas flow, but a good approximation of the resonant frequencies can be made without reference to the combustion effects.

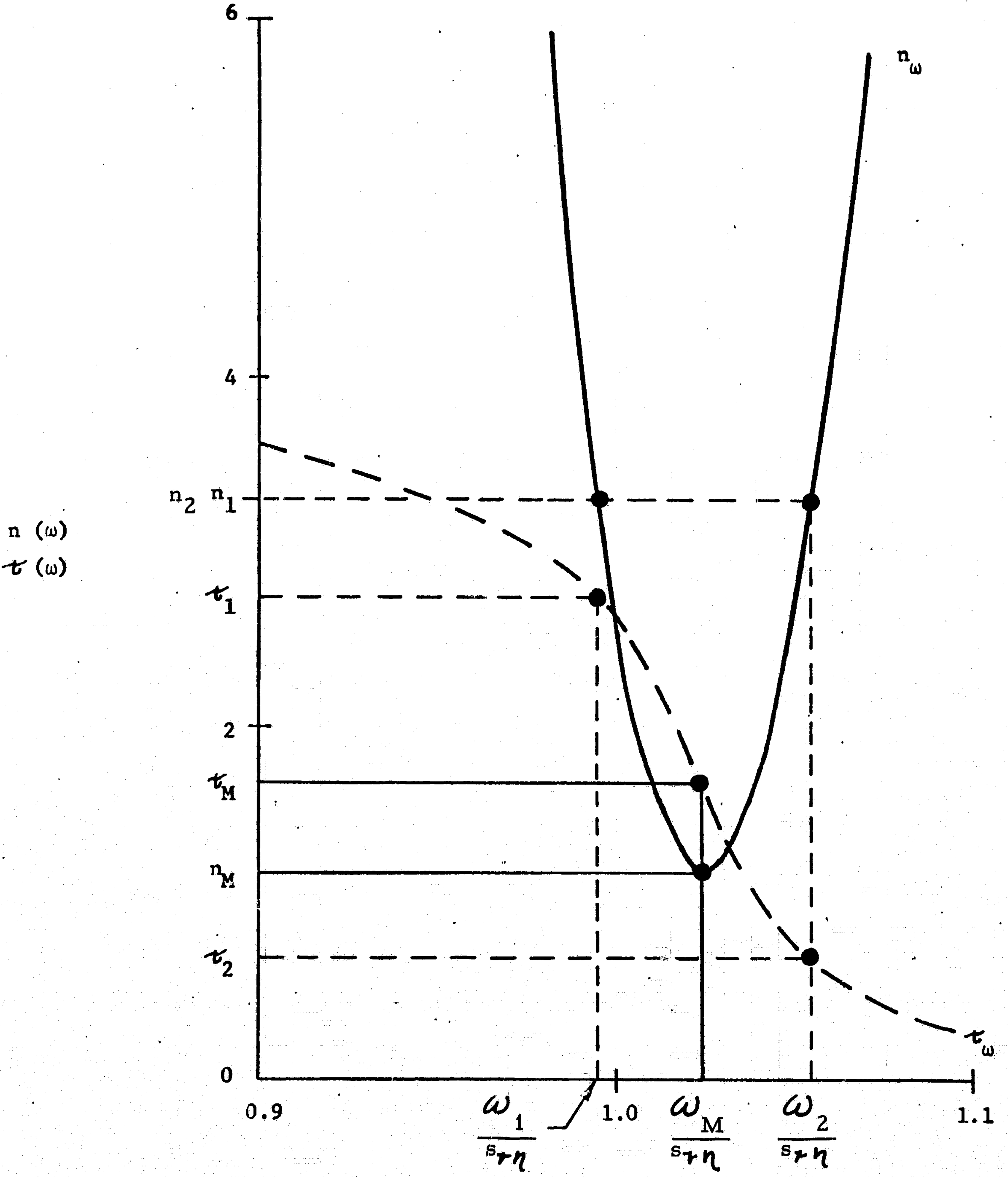


Figure 13 -- Typical Solutions of η(ω) and τ(ω)

I, Problem (cont.)

C. EQUATIONS

1. **Program A-Distributed Combustion Analysis**

Input = $s_{\nu\eta}$, γ, $D_1(Z)$, r_c^*, L_c^*, C_o^*, $\bar{u}_{Lo}^*$, κ, $\bar{u}_e$, $\mathcal{E}$

$$\bar{u} = \frac{D_1\ \bar{u}_e}{D_1(Z=L_c)}$$

Solve $$\frac{d\ \bar{u}_L}{DZ} = \frac{\kappa[\bar{u} - \bar{u}_L]}{\bar{u}_L}$$ for $\bar{u}_L$

$$\bar{\rho} = [1 + \frac{\gamma-1}{2}\bar{u}^2]^{-\frac{1}{\gamma-1}}$$

$$\rho_L = [\bar{u}_e\ (1 + \frac{\gamma-1}{2}\bar{u}_e{}^2)^{-\frac{1}{\gamma-1}} - \frac{\bar{\rho}\ \bar{u}}{\bar{u}_L}]$$

$$\overline{Q} = \frac{[(1-\gamma\bar{u}^2)\bar{\rho}\ \frac{d\bar{u}}{DZ} - \gamma\bar{u}\ \bar{\rho}_L\ \kappa\ (\bar{u}-\bar{u}_L)]}{1 + \gamma\bar{u}(\bar{u} - \bar{u}_L)}$$

$$\Omega = (s_{\nu\eta}{}^2 - \omega^2)^{1/2}$$

I, C, Equations

$$Y_1 = \Omega \sinh (\Omega Z_e) - i\omega E \cosh (\Omega Z_e)$$

$$Y_2 = - \int_o^{Z_e} Q_1(Z) \cosh [\Omega (Z_e - Z)] \, dZ + \frac{i\omega E}{\Omega} \int_o^{Z_e} Q_1(Z) \sinh [\Omega (Z_e - Z)] \, dZ$$

$$Y_3 = \int_o^{Z_e} Q_2 (Z) \cosh [\Omega (Z_e - Z)] \, dZ - \frac{i\omega E}{\Omega} \int_o^{Z_e} Q_z(z) \sinh [\Omega (Z_e - Z)] \, dZ$$

$$Q_1 = - \sigma \left[(\gamma+1) \overline{Q} + \frac{K \sigma}{K+\sigma} \rho_L \right]$$

$$Q_2 = \sigma\gamma \overline{Q}$$

$$h = \frac{Y_1 + Y_2}{Y_3}$$

I, C, Equations (cont.)

2. Program B-Concentrated Combustion Analysis

The perturbation equations in this program are solved using matrices as shown here.

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} & a_{16} \\ a_{21} & & & & & \\ a_{31} & & & & & \\ a_{41} & & & & & \\ a_{51} & & & & & \\ a_{61} & & & & & \end{bmatrix} \begin{bmatrix} p'_- \\ \rho'_- \\ v'_- \\ p'_+ \\ h'_- \\ s'_- \end{bmatrix} + \dot{m}'_b \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \end{bmatrix} = 0$$

where

$a_{14} = Y$

$a_{12} = - \bar{v}_-$

$a_{13} = - \bar{\rho}_-$

$a_{21} = - \frac{1}{\gamma}$

$a_{22} = - \bar{v}_-^2$

$a_{23} = - \bar{\rho}_- \ \bar{v}_-$

$a_{24} = + \frac{1}{\gamma}$

$a_{32} = - (\gamma-1) \frac{\bar{v}_-^3}{2} - \bar{h}_- \ \bar{v}_-$

$a_{33} = - (\gamma-1) \frac{3}{2} \bar{\rho}_- \ \bar{v}_-^2 - \bar{h}_- \ \bar{\rho}_-$

$a_{34} = Y \ \bar{\rho}_+ \ \bar{h}_+$

$a_{35} = - \bar{\rho}_- \ \bar{v}_-$

$a_{41} = 1$

$a_{42} = -1$

$a_{45} = -1$

$a_{51} = \frac{\gamma-1}{\gamma \ \bar{\rho}_-}$

$a_{55} = -1$

$a_{56} = \bar{T}_-$

$a_{61} = E$ (when $S_{\nu\eta} = 0$ $A_{61} = A\gamma$)

$a_{63} = \gamma$

$a_{66} = C\gamma$

I, C, Equations (cont.)

$$Y = -\frac{(\sigma^2 + S_{\nu\eta}^{\ 2})}{\sigma\gamma} \tanh \left[(\sigma^2 + S_{\nu\eta}^{\ 2})^{1/2} x \right] \quad \text{where } \sigma = j\omega$$

All other a's are zero.

Also,

$$b_1 = 1,\ b_2 = \bar{v}_L,\ b_3 = \bar{h}_L + (\gamma - 1)\frac{\bar{v}_L^{\,2}}{2},\ b_4 = 0,\ b_5 = 0,\ b_6 = 0.$$

The a's and b's must be evaluated from the following input parameters:

$$S_{\nu\eta},\ \omega^*,\ x^*,\ E(\omega),\ C(\omega),\ \bar{v}_-,\ \bar{v}_L^*,\ \bar{c}_o^*,\ r_c^*,\ \bar{w}^*,\ \gamma,\ \bar{p}_o^*$$

as well as the conditions:

$$\bar{p}_+ = 1,\ \bar{\rho}_+ = 1,\ \bar{h}_+ = 1,\ \bar{h}_L = 1$$

The following equations can be used to determine other variables:

$$\bar{v}_L = \frac{\bar{v}_L^*}{\bar{c}_o^*}\ ;\ g^* R_o^* T_o^* = \frac{\bar{c}_o^{*2}}{\gamma}\ ; \qquad \bar{\rho}_o^* = \frac{\bar{P}_o^*}{g^* R_o^* \bar{T}_o^*}\ ;$$

$$\bar{\rho}_- = \frac{\bar{w}^*}{\bar{\rho}_o^* g^* \pi r_c^{*2}\ \bar{c}_o^*\ \bar{v}_-}; \qquad \bar{p}_- = 1 - \gamma(\bar{\rho}_-\ \bar{v}_-^2 - \bar{\rho}_-\ \bar{v}_-\ \bar{v}_L);$$

$$\bar{h}_- = 1 + \frac{\bar{v}_L^2}{2} - \frac{\bar{v}_-^2}{2}\ ; \qquad \omega = \frac{\omega^*\ r_c^*}{\bar{c}_o^*}; \qquad x = \frac{x^*}{r_c^*}\ ;\ \bar{T}_- = \bar{h}_-.$$

L_c^* replaces r_c^* for longitudinal modes in the definition of x and ω.

I, C, Equations (cont.)

After the matrix equation is multiplied by the inverse of matrix (a), it represents a series of equations relating each perturbation variable to $\dot{m}_b'$. The first and fourth of these relate p_-' and p_+' to m_b'. A characteristic equation is formed by combining these equations.

$$\dot{m}_b' = \bar{\rho}_- \bar{v}_- \left[\frac{p_+' + p_-'}{2 \bar{p}_-} \right] \eta \left(1-e^{-i\omega\tau}\right) ,$$

$$p_+' = K_1 m_b' ,$$

$$p_-' = K_2 m_b' ;$$

$$\frac{2 \bar{p}_-}{\bar{\rho}_- \bar{v}_- (K_1 + K_2)} = h$$

I, C, Equations (cont.)

3. Program C - Nozzle Admittance

Nozzle admittance is calculated for an axisymmetric conventional or annular nozzle with an inner wall radius r_i and outer wall radius r_o. These radii are functions of axial distance Z from the nozzle throat which are defined as follows:

$$r_o^* (Z^*) = R^*_{ATo} + R^*_{CTo} - \sqrt{R^{*2}_{CTo} - Z^{*2}} \quad , \quad 0 \leq Z^* \leq Z^*_{1o}$$

$$\text{where } Z^*_{1o} = R^*_{CTo} \sin \alpha_o$$

$$r_o^* (Z^*) = R^*_{ST1o} + \frac{Z^* - Z^*_{1o}}{Z^*_{2o} - Z^*_{1o}} (R^*_{ST2o} - R^*_{ST1o}), \quad Z^*_{1o} < Z^* \leq Z^*_{2o}$$

$$\text{where } R^*_{ST1o} = R^*_{ATo} + R^*_{CTo} (1 - \text{Cos } \alpha_o)$$

$$Z^*_{2o} = Z^*_{1o} + (R^*_{ST2o} - R^*_{ST1o}) \text{ Cot } \alpha_o$$

$$R^*_{ST2o} = R^*_{ACo} - R^*_{CCo} (1 - \text{Cos } \alpha_o)$$

$$r_o^*(Z^*) = R^*_{ACo} - R^*_{CCo} + \sqrt{R^{*2}_{CCo} - (Z^*_{3o} - Z^*)^2} \quad , \quad Z^*_{2o} < Z^* \leq Z^*_{3o}$$

$$\text{where } Z^*_{3o} = Z^*_{2o} + R^*_{CCo} \sin \alpha_o$$

I, C, Equations (cont.)

$$r_o^*\ (Z^*) = R_{ACo}^* \quad , \quad Z_{3o}^* \leq Z^*$$

$$r_i^*\ (Z^*) = R_{ATi}^* + R_{CTi}^* - \sqrt{R_{CTi}^{*\,2} - Z^{*2}} \quad , \quad 0 \leq Z^* \leq Z_{1i}^* \ , \quad R_{CTi}^* > 0$$

$$r_i^*\ (Z) = R_{ATi}^* + R_{CTi}^* + \sqrt{R_{CTi}^{*\,2} - Z^{*2}} \quad , \quad 0 \leq Z^* \leq Z_{2i}^* \quad R_{CTi}^* < 0$$

$$r_i^*\ (Z^*) = 0 \ , \qquad R_{ATi}^* = 0 \quad R_{ACi}^* = 0$$

where $Z_{1i}^* = R_{CTi}^* \sin \alpha_i$

$$r_i^*\ (Z^*) = R_{ST1i}^* + \frac{Z^* - Z_{1i}^*}{Z_{2i}^* - Z_{1i}^*} (R_{ST2i}^* - R_{ST1i}^*) \quad Z_{1i}^* < Z^* \leq Z_{2i}^*$$

where $R_{ST1i}^* = R_{ATi}^* + R_{CTi}^* (1 - \text{Cos}\ \alpha_i)$

$$Z_{2i}^* = Z_{1i}^* + (R_{ST2i}^* - R_{ST1i}^*) \text{ Cot } \alpha_i$$

$$R_{ST2i}^* = R_{ACi}^* - R_{CCi}^* (1 - \text{Cos}\ \alpha_i)$$

I, C, Equations (cont.)

$$r^*_i\,(Z^*) = R^*_{ACi} - R^*_{CCi} + \sqrt{R^{*2}_{CCi} - (Z^*_{3i} - Z^*)^2} \quad , \qquad Z^*_{2i} < Z^* \leq Z^*_{3i}$$

$$R^*_{CCi} > 0$$

$$r^*_i\,(Z^*) = R^*_{ACi} - R^*_{CCi} - \sqrt{R^{*2}_{CCi} - (Z^*_{3i} - Z^*)^2} \quad , \qquad Z^*_{2i} < Z^* \leq Z^*_{3i}$$

$$R^*_{CCi} < 0$$

where $Z^*_{3i} = Z^*_{2i} + R^*_{CCi} \sin \alpha_i$

R^*_{ATo}, R^*_{ATi}, R^*_{CTo}, R^*_{CTi}, R^*_{ACo}, R^*_{ACi}, R^*_{CCo}, R^*_{CCi}, α_o and α_i are input data defined by Figures 14 and 15.

From the nozzle radius calculate the nozzle area a*.

$$a^*(Z^*) = \pi\,[r^{*2}_o(Z^*) - r^{*2}_i\,(Z^*)]$$

Mach number, M(Z), can be obtained by creating a table of M versus a* from the equation.

$$\frac{a^*(Z)}{a^*(Z^*=0)} = \frac{1}{M(Z^*)}\,\left[\left(\frac{2}{\gamma+1}\right)\left(1 + \frac{\gamma-1}{2}\,M^2(Z^*)\right)\right]^{\frac{\gamma+1}{2(\gamma-1)}}$$

The velocity divided by the speed of sound at the throat, $\bar{q}$, (commonly called M*) is given by

$$\bar{q}\,(Z)^2 = \frac{\frac{\gamma+1}{2}\,M^2(Z^*)}{1 + \frac{\gamma-1}{2}\,M^2(Z^*)}$$

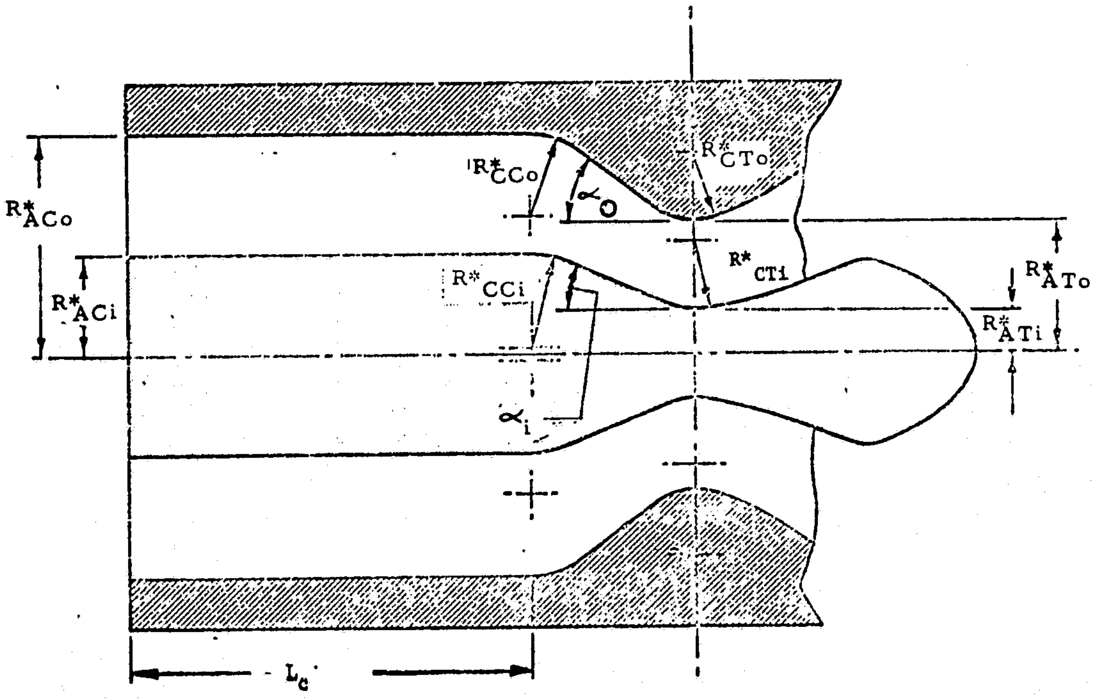


Figure 14 -- Computer Input Parameters -- Annular Chamber

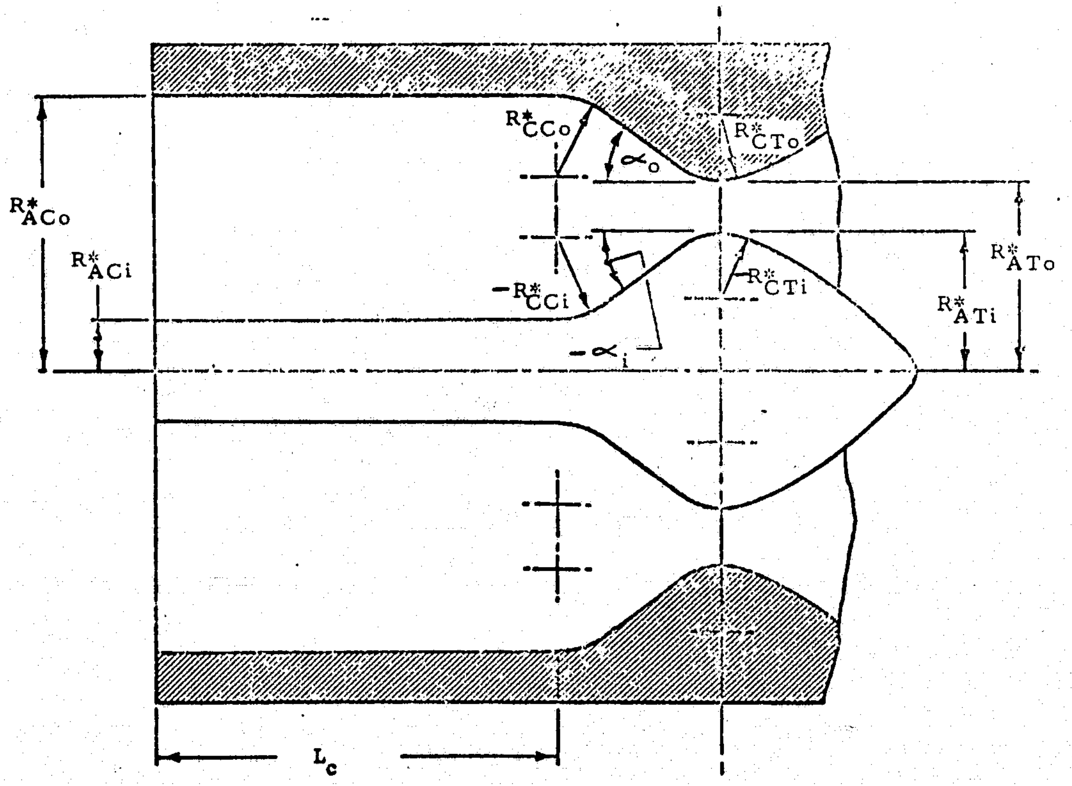


Figure 15 -- Computer Input Parameters -- Annular Chamber

I, C, Equations (cont.)

The velocity potential, which is the independent variable for the equations of motion in the nozzle, is given by

$$\hat{\phi}\ (Z^*) = \frac{2\ \hat{K}}{R^*_{AT_o}} \int_o^{Z^*} \bar{q}\ (Z^*)\ dZ^* \tag{43}$$

Auxiliary functions $\hat{\zeta}$, $\hat{\xi}^{(1)}$, $\hat{\xi}^{(2)}$, f_3 which are required to define the admittance coefficients in the nozzle are given by the following differential equations

$$b\left(\frac{d\hat{\zeta}}{d\hat{\phi}} + \hat{\zeta}^2\right) = (g + ih)\ \hat{\zeta} - (j + i\ k)$$

$$\frac{d}{d\hat{\phi}}\ [(1 - \bar{q}^2)\ \hat{\xi}^{(1)}] = -\left[\hat{\zeta} - i\ \hat{\omega}\ \left(\frac{1}{2\bar{q}^2} + \frac{2}{(\gamma+1)(1-\bar{q}^2)}\right)\right] (1-\bar{q}^2)\ \hat{\xi}^{(1)}$$

$$+ \frac{2}{\gamma+1}\ \frac{\hat{s}_{\nu\eta}^{\,2}}{4}\ \frac{(\bar{c})^{\frac{2}{\gamma-1}}}{\bar{q}}$$

$$\frac{d}{d\hat{\phi}}\ [(1-\bar{q})^2\ \xi^{(2)}] = -\left[\hat{\zeta} - i\hat{\omega}\ \left(\frac{1}{2\bar{q}^2} + \frac{2}{(\gamma+1)(1-\bar{q}^2)}\right)\right] (1-\bar{q}^2)\ \hat{\xi}^{(2)}$$

$$+ \frac{2}{\gamma+1} \left[\frac{df_3}{d\hat{\phi}} + \frac{\hat{s}_{\nu\eta}^{\,2}\ \bar{c}^{\frac{2}{\gamma-1}}\ \bar{q}}{2i\hat{\omega}} \left(\frac{1-\bar{q}^2}{2\bar{q}^2} + \frac{\hat{c}^2}{\bar{q}^2}\ f_3\right)\right]$$

$$\frac{d}{d\hat{\phi}}\ (\bar{c}^2\ f_3) - \frac{i\hat{\omega}}{2\bar{q}^2}\ (\bar{c}^2\ f_3) = \frac{1}{2}\ \frac{d\bar{q}^2}{d\hat{\phi}}$$

where $\quad b = \bar{q}^2\ (\bar{c}^2 - \bar{q}^2)$

I, C, Equations (cont.)

$$g = \frac{\gamma+1}{2} \frac{(\bar{q})^2}{\bar{c}^2} \frac{d(\bar{q})^2}{d\hat{\phi}}$$

$$h = \hat{\omega} (\bar{q})^2$$

$$j = \frac{\hat{\omega}^2}{4} - \frac{\bar{q} (\bar{c})^{\frac{2}{\gamma-1}}}{4} (\hat{s}_{\nu\eta})^2$$

$$k = \frac{\gamma-1}{4} \frac{(\bar{q})^2}{\bar{c}^2} \frac{d(\bar{q})^2}{d\hat{\phi}} \hat{\omega}$$

$$\hat{\omega} = \omega/\hat{K} \qquad (44)$$

$$\hat{s}_{\nu\eta} = s_{\nu\eta}/\hat{K} \qquad (45)$$

$$\hat{K} = \left(\frac{d\bar{q}}{dZ}\right)_{Z=o} = R^*_{ATo} \left[\frac{2}{\gamma+1} \left(\frac{R^*_{ATo}}{R^*_{CTo}} - \frac{R^*_{ATi}}{R^*_{CTi}} \right) \Big/ \left(R^{*2}_{ATo} - R^{*2}_{ATi} \right) \right]^{1/2}$$

I, C, Equations (cont.)

The equations for $\hat{\xi}^{(1)}$, $\hat{\xi}^{(2)}$, and $\hat{\zeta}$ are singular at the throat, $\hat{\phi} = 0$. The solutions to these equations at the throat can be found using a series expansion about the throat. The numerical integration can be started at some small value of $\hat{\phi} = \hat{\phi}_i$. At $\hat{\phi}_i$ the initial values are:

$$\bar{c}^2 f_3 = 0.5\,\hat{\phi}_i$$

$$(1-\bar{q}^2)\,\xi^{(1)} = \frac{\frac{2}{\gamma+1}\,\frac{\hat{s}_{\nu\eta}^{2}}{4}\,\hat{\phi}_i}{1 + \frac{i\,\hat{\omega}\,2}{\gamma + 1}}$$

$$(1-\bar{q}^2)\,\xi^{(2)} = \frac{\hat{\phi}_i}{\gamma+1 + i2\hat{\omega}}$$

$$\hat{\zeta} = \alpha_o + \hat{\phi}\,\alpha_1$$

where

$$\alpha_o = \frac{\hat{\omega}^2 - \hat{s}^2_{\nu\eta} - i\,(\gamma-1)\,\hat{\omega}}{4(\frac{\gamma+1}{2} + i\hat{\omega})}$$

$$\alpha_1 = \frac{\frac{\gamma+1}{2}\,\alpha_o^2 + \frac{\gamma+1}{2}\,\alpha_o\,(\frac{\gamma+1}{2} + 2\,\tilde{b}) + \alpha_o i\hat{\omega} + \frac{\hat{s}_{\nu\eta}^{2}}{8}\,(1-\gamma) + \frac{i\omega(\gamma-1)}{4}(\frac{\gamma+1}{2} + 2\tilde{b})}{-(\gamma+1 + i\hat{\omega})}$$

and $\tilde{b}$ comes from the expansion $\bar{q}^2 = 1 + \hat{\phi} + \tilde{b}\,\hat{\phi}^2$

I, C, Equations (cont.)

The nozzle admittance coefficients can be defined from the auxiliary functions at the desired value of Z.

$$A = \frac{1}{\gamma} \left(\frac{\gamma + 1}{2}\right)^{\frac{1}{2}(\gamma+1)/(\gamma-1)} \frac{\bar{q}}{\bar{c}^{2/(\gamma-1)}} \frac{\overline{c^2}\hat{\xi}^{(1)} - \hat{\zeta}}{\overline{q^2}(\overline{c^2}\hat{\xi}^{(1)} - \hat{\zeta}) - \frac{1}{2}i\hat{\omega}} \tag{46}$$

$$B = i\hat{K} \frac{\hat{\omega}\bar{q}^{-\frac{1}{2}}}{\bar{c}^{1/(\gamma-1)}} \frac{\overline{c^2}\hat{\xi}^{(1)}}{\overline{q^2}(\overline{c^2}\hat{\xi}^{(1)} - \hat{\zeta}) - \frac{1}{2}i\hat{\omega}} \tag{47}$$

$$C = \left(\frac{2}{\gamma + 1}\right)^{\frac{1}{2}} \bar{q}\, \overline{c^2} \left[\frac{\frac{1}{2}(1 - \overline{q^2})\hat{\xi}^{(1)} - \frac{1}{2}i\hat{\omega}\xi^{(2)} + f_3\hat{\zeta}}{\overline{q^2}(\overline{c^2}\hat{\xi}^{(1)} - \hat{\zeta} - \frac{1}{2}i\hat{\omega}}\right] \tag{48}$$

$$E = \gamma A + \frac{iB}{\omega_c} \tag{49}$$

I, C, Equations (cont.)

4. Program D - Expansion of Results of Programs A & B

Take h from Program B

$$\tilde{h} = \frac{h}{A_{\nu\eta} + i\left(\frac{B_{\nu\eta}\frac{1_R}{n}}{\gamma\omega} + \frac{C_{\nu\eta}\frac{1_\theta}{n}}{\gamma\omega}\right)}$$

5. Program E - Injector Nonuniformity Coefficients

For standing mode

$$A_{\nu\eta} = \frac{\int_0^{2\pi}\int_0^1 \mu(r,\theta)\, F_P(r,\theta)\, J_\nu^{\,2}\,(s_{\nu\eta}\, r)\, \mathrm{Cos}^2\, \nu\theta\; r\, dr\, d\theta}{\int_0^{2\pi}\int_0^1 J_\nu^{\,2}(s_{\nu\eta} r)\, \mathrm{Cos}^2\, \nu\theta\; r\, dr\, d\theta} \tag{50}$$

$$B_{\nu\eta} = \frac{\int_0^{2\pi}\int_0^1 \mu(r,\theta)\, F_R\,(\omega,r,\theta)\, J_\nu'\,(s_{\nu\eta}\, r)\, J(s_{\nu\eta} r)\, \mathrm{Cos}^2\, \nu\theta\; r\, dr\, d\theta}{\int_0^{2\pi}\int_0^1 J_\nu^{\,2}\,(s_{\nu\eta}\, r)\, \mathrm{Cos}^2\, \nu\theta\; r\, dr\, d\theta} \tag{51}$$

$$C_{\nu\eta} = -\frac{\int_0^{2\pi}\int_0^1 \mu(r,\theta)\, \frac{F_T(r,\theta,\omega)}{r}\, J_\nu^{\,2}\,(S_{\nu\eta}\, r)\, \nu\, \sin\, \nu\theta\, \mathrm{Cos}\, \nu\theta\; r\, dr\, d\theta}{\int_0^{2\pi}\int_0^1 J_\nu^{\,2}\,(s_{\nu\eta}\, r)\, \mathrm{Cos}^2\, \nu\theta\; r\, dr\, d\theta} \tag{52}$$

For a spinning mode make the following replacements:

$$A_{\nu\eta} = \frac{A_{\nu\eta}}{2}$$

$$B_{\nu\eta} = \frac{B_{\nu\eta}}{2}$$

$$C_{\nu\eta} = \frac{i\, C_{\nu\eta}}{2}$$

I, C, Equations (cont.)

$$\mu(r,\theta) = \frac{\sigma(r,\theta)}{\sigma_{ave}} \tag{53}$$

$\sigma(r,\theta)$ is the injection density (weight flow per unit area)

σ_{ave} is the total weight flow divided by the total injector area

6. Program F - Stability Zone Mapping

$$n\ (1 - e^{-i\omega\tau}) = \tilde{h}$$

7. Program I - Nonlinear Combustion Response

For non-linear combustion response, input tables of burning rate versus pressure, radial velocity, and tangential velocity, ϕ_P, ϕ_R, ϕ_1. Additional data needed are: P_{oo} (amplitude of pressure oscillation being considered divided by the mean chamber pressure) TFLP, TFLR, and TFLT (the linear transfer functions for equivalent linear operation associated with the pressure, radial velocity and tangential velocity dependent nonlinear element.) A describing function is calculated at each point on the injector.

$$f_p(r,\theta) = \frac{1}{\pi\ P_o\ (TF_{LP})} \int_{-\pi}^{\pi} \phi_P(\omega\tau, r,\theta)\ \mathrm{Cos}\ (\omega\tau)\ d(\omega\tau) \tag{54}$$

where

$$\phi_P(\omega\tau,r,\theta) = \phi_P(P')$$

$$p' = P_o \cos \omega\tau$$

$$P_o = |P_{oo}\ J_\nu(s_{\nu\eta}\ r)\ \mathrm{Cos}\ \nu\ \theta|$$

$$F_R(r,\theta,\omega) = \frac{1}{\pi\ V_o(TF_{LR})} \int_{-\pi}^{\pi} \phi_R(\omega\tau,r,\theta)\ \mathrm{Cos}\ (\omega\tau)\ d(\omega\tau)$$

I, C, Equations (cont.)

where
$$\phi_R(\omega\tau, r, \theta) = \phi_R(v')$$

$$v' = V_o \text{ Cos } \omega\tau$$

$$V_o = \left| V_{oo} \left[\frac{\nu\, J_\nu(s_{\nu\eta}\, r) - s_{\nu\eta}\, J_{\nu+1}\,(s_{\nu\eta}\, r)}{r} \right] \cos \nu\theta \right|$$

$$V_{oo} = \frac{P_{oo}}{\gamma\omega}$$

$$f_T(r, \theta, \omega) = \frac{1}{\pi\, W_o (TF_{LT})} \int_{-\pi}^{\pi} \phi_T(\omega\tau, r, \theta) \text{ Cos } (\omega\tau)\, d(\omega\tau)$$

where

$$\phi_T(\omega\tau, r, \theta) = \phi_T(w')$$

$$w' = W_o \text{ Cos } \omega\tau$$

$$W_o = \left| \frac{V_{oo}\, J_\nu(s_{\nu\eta}\, r)\, \nu \sin \nu\theta}{r} \right|$$

8. Program J, Injected Mass Distribution Effects

This program calculates mass flow distribution $\mu(r,\theta)$ used in the program for calculating the $A_{\nu\eta}$, $B_{\nu\eta}$ and $C_{\nu\eta}$. It also calculates useful design parameters such as pressure drops, injection velocities, mixture ratios, and flow areas. Input consists of type and location of elements, description of each element type in terms of orifice diameters and number of orifices, and overall information including propellant densities, orifice discharge coefficients, amount of film cooling, injector radius, total flow rates and total mixture ratio.

First the total orifice area, A_T, for each propellant is calculated:

$$A_T = \sum_i 0.7853891\, d_i^2$$

I, C, Equations (cont.)

Then the area of the film coolant orifices, A_{FCT}, is calculated

$$A_{FCT} = \sum_i 0.7853891\, d_{FC_i}^2$$

Total flows of oxidizer and fuel are then calculated. The equations used depend on how the information on the film coolant is input.

If percent film coolant is given; P_{ffc}, percent fuel film coolant; P_{xfc} percent oxidizer film coolant, the total fuel flow rate in injector matrix, W_{fT}, is given by:

$$W_{fT} = \frac{W_T}{MR+1} (1 - P_{ffc})$$

The total oxidizer flow rate in the injector matrix, W_{XT}, is given by:

$$W_{XT} = W_T - \frac{W_T}{MR+1} (1 - P_{xfc})$$

If the number and size of the film cooling orifices are given in terms of the fuel hole area in the matrix A_F, oxidizer hole area in the matrix A_X, fuel film coolant hole area A_{FFC}, and oxidizer film coolant area A_{XFC}.

$$W_{fT} = \frac{W_T}{MR+1} \frac{A_{FT}}{A_{FT} + A_{FFC}}$$

$$W_{XT} = W_T - \frac{W_T}{MR+1} \frac{A_{XT}}{A_{XT} + A_{XFC}}$$

The total weight flow per element is given by:

$$W_{E_i} = \frac{A_{Fi}}{A_{FT}} W_{fT} + \frac{A_{xi}}{A_{XT}} W_{XT}$$

I, C, Equations (cont.)

W_{E_i} is used in Program E as follows:

$$\frac{W_{Ei}}{W_T} \; XM \; XN = \mu_i \text{ for each element.}$$

See Equations 51 to 53.

This program also calculates injector pressure drop ΔP:

$$\Delta P = \frac{W^2}{C_d^2 \; A^2} \frac{(144)}{\rho (64.4)}$$

where C_d is the orifice discharge coefficient, A is the orifice area in in.^{2}, W is the orifice flow rate in lbm/sec and ρ is the propellant density in lbm/ft^3. Injection velocity V is calculated from ΔP.

$$V = \sqrt{\frac{(144)(64.4) \; \Delta P}{\rho}}$$

D. DEFINITION OF TERMS

$a^*(z^*)$	Area of annular nozzle, in.^2
A_{Fi}	Area of the ith fuel element, in.^2
A_{xi}	Area of the ith oxidizer element, in.^2
A_{XFC}	Total area of oxidizer film coolant orifices, in.^2
A_{FFC}	Total area of fuel film coolant orifices, in.^2
A_{XT}	Total area of oxidizer orifices excluding film coolant, in.^2
A_{FT}	Total area of fuel orifices excluding film coolant, in.^2
A_{FCT}	Total film coolant area, in.^2
A_T	Total orifice area, in.^2
$A_{\nu\eta}$	Factor measuring the effect of nonuniform injection on stability (see Equation 37)
b	Variable used in nozzle analysis
B	Term used in longitudinal analysis
$B_{\nu\eta}$	Factor measuring the effect of nonuniform injection on stability (see Equation 37)
c	Speed of sound nondimensionalized by the stagnation speed of sound in the chamber analysis or the throat speed of sound in the nozzle analysis
C	Term used in longitudinal analysis
C_d	Injector orifice discharge coefficient
C_{STAR}	Characteristic Exhaust Velocity, ft/sec
C_o^*	Dimensional stagnation gas speed of sound, feet per second
C_{th}^*	Dimensional speed of sound at the nozzle throat, feet per second
C_1, C_2	Constants in radial dependent factor of pressure in an annular chamber
$C_{\nu\eta}$	Factor measuring the effect of nonuniform injection on stability (see Equation 37) (complex number, $C_{\nu\eta R} + i\, C_{\nu\eta I}$)
d_i	Diameter of the ith orifice, in.
d_{FC_i}	Diameter of the ith film in coolant orifice, in.
D	Term used in longitudinal analysis
$D_1(z)$	Combustion distribution used in transverse analysis

I, D, Definition of Terms (cont.)

e	Base the natural logarithms
E	Term used in longitudinal analysis
f^*	Dimensional frequency in Hertz
f_3	Auxiliary function in nozzle analysis (complex)
f_p	$\frac{\partial f}{\partial p}$
f_T	$\frac{\partial f}{\partial T}$
f_v	$\frac{\partial f}{\partial v}$
$f(\ )$	Rate of preparatory processes before burning
F	Term used in longitudinal analysis
$F(\omega)$	Describing function defined by Equation 34
F_p	Describing function for pressure oscillations
F_R	Describing function for velocity oscillations in the radial direction
F_T	Describing function for velocity oscillations in the tangential direction
g	Variable used in nozzle analysis
$\hat{h}$	Variable used in nozzle analysis
h	Part of characteristic equation which includes all acoustic effects (Equation 40) can also be viewed as a chamber admittance (complex number, $h = h_r + i\ h_i$)
h_i	Imaginary part of h
h_r	Real part of h
$\tilde{h}$	Includes acoustic and nonuniform injection effects in the characteristic equation (see Equation 40)
$\tilde{h}_i$	Imaginary part of $\tilde{h}$
$\tilde{h}_r$	Real part of $\tilde{h}$
h_s	Total enthalpy of gas nondimensionalized by $R^*\ T_o^*$
i	Unit imaginary number $\sqrt{-1}$
I	Input to a nonlinear element
I_1	Term used in longitudinal analysis

I, D, Definition of Terms (cont.)

j	Variable used in nozzle analysis
J	Term used in the longitudinal analysis
$J_\nu(\ )$	Bessel function of the first kind of order ν
$J'_\nu(\ )$	Derivative with respect to the argument of the Bessel function of the first kind of order ν
k	Variable used in nozzle analysis
$\hat{K}$	Velocity gradient at the throat, nondimensional (see Equation 45)
K	Factor involving droplet drag on frequency defined in Equation 10f
K_1	Term used in longitudinal analysis
l_r	Velocity interaction index measuring the amount of interaction between gas velocity in the radial direction and the burning rate
l_θ	Velocity interaction index measuring the amount of interaction between gas velocity in the tangential direction and the burning rate
L	Term used in longitudinal analysis
$L_c{}^*$	Chamber length, in.
$\dot{m}_b$	Mass burning rate, lbf-sec/ft
$\dot{m}_i$	Mass injection rate, lbf-sec/ft
m_r	Displacement interaction index measuring the amount of interaction between gas displacement in the radial direction and the burning rate
m_θ	Displacement interaction index measuring the amount of interaction between gas displacement in the tangential direction and the burning rate
M	Mach number
M_1	Term used in longitudinal analysis
M_e	Mach number at nozzle entrance
MR	Mixture ratio
n	Pressure interaction index measuring the amount of interaction between pressure oscillations and burning rate oscillations
n_{min}	Minimum value of the pressure interaction index for a particular mode
N	Term used in the longitudinal analysis

I, D, Definition of Terms (cont.)

O	Output of a nonlinear element
O_1	Component of output of a nonlinear element at the input frequency ω_f
O_j	Component of output of a nonlinear element of a frequency $j\omega_f$: $j = 2, 3, 4,$ etc.
p	Pressure nondimensionalized by the stagnation gas pressure
p_o	Zeroth order solution for pressure. Defined by Equations 11 and 12a
p_1	First order solution for pressure defined by Equations 11 and 12b
P	Z dependent factor in pressure perturbation
P_o	Amplitude of pressure oscillation lbf/in^2
P_{oo}	Constant coefficient measuring amplitude of pressure defined in Equation 13a
P_{ffc}	Percent fuel film coolant
P_{xfc}	Percent oxidizer film coolant
$\vec{q}$	Velocity nondimensionalized by the stagnation gas speed of sound, C_o^* in chamber. In the nozzle it is nondimensionalized by the speed of sound at the throat
Q	Burning rate per unit volume nondimensionalized by $\frac{\rho_o^* C_o^*}{r_c^*}$ or $\frac{\rho_o^* C_o^*}{L_c^*}$ for transverse or longitudinal modes.
r	Coordinate in the radial direction. Nondimensionalized by the chamber radius r_c^*
r_c^*	Dimensional chamber radius in inches
r_d^*	Droplet radius, feet
r_i^*	Inner radius of an annular chamber, feet
r_o^*	Outer radius of an annular chamber, feet
R^*	Gas constant, ft-lbf/lbm-°R
R	Ratio of inner radius and outer radius of an annular nozzle
R^*_{ACi}	Radius of inner contour of an annular chamber at the chamber entrance, inches (see Figures 14,15)
R^*_{ACo}	Radius of outer contour of an annular chamber at the chamber entrance, inches (see Figures 14,15)

I, D, Definition of Terms (cont.)

R^*_{ATi}	Radius of the throat of the inner contour of an annular nozzle, inches (see Figures 14,15)
R^*_{ATo}	Radius of the throat of the outer contour of an annular nozzle, inches (see Figures 14,15)
R^*_{CCi}	Radius of curvature at the chamber entrance of the inner contour of an annular nozzle, inches (see Figures 14,15)
R^*_{CCo}	Radius of curvature at the chamber entrance of the outer contour of an annular nozzle, inches (see Figures 14,15)
R^*_{CTi}	Radius of curvature at the throat of an inner contour of an annular nozzle, inches (see Figures 14,15)
R^*_{CTo}	Radius of curvature at the throat of the outer contour of an annular nozzle, inches (see Figures 14,15)
R^*_{ST1i}	Radius of inner contour of annular nozzle at transition between throat curvature and tangent, inches $[r_i^* (Z^*_{1i})]$
R^*_{ST1o}	Radius of outer contour of annular nozzle at transition between throat curvature and tangent, inches $[r_o^* (Z^*_{10})]$
R^*_{ST2i}	Radius of inner contour of annular nozzle at transition between chamber curvature and tangent, inches $[r_i^* (Z^*_{2i})]$
R^*_{ST2o}	Radius of outer contour of annular nozzle at transition between chamber curvature and tangent, $[r_o^* (Z^*_{20})]$
S	Z dependent factor in entropy perturbation
$s_{\nu\eta}$	Mode number for a transverse mode defined by Equations 19 thru 21 tabulated for a cylindrical chamber in Table I
$\hat{s}_{\nu\eta}$	Mode number used in nozzle analysis (see Equation 45)
t	time nondimensional as τ_i
T	Temperature nondimensionalized by T_o^*
T_1	Term in longitudinal analysis
T_o^*	Stagnation gas temperature, °R
TF	Equivalent linear transfer function (see Equation 33)
$(TF)_{LIM}$	Limiting linear transfer function
u	Gas velocity in the axial direction nondimensionalized by C_o^*
$\bar{u}_e$	Mean value of the gas velocity at the nozzle entrance nondimensionalized by the stagnation gas speed of sound
u_L	Axial velocity of liquid drops, ft/sec
$\bar{u}^*_{Lo}$	Liquid injection velocity, ft/sec (see Equation 75)
U	Z dependent factor in axial velocity perturbation

I, D, Definition of Terms (cont.)

v	Gas velocity in the radial direction nondimensionalized by C_o*
V_o	Amplitude of radial velocity oscillation
V_{oo}	Constant coefficient measuring the amplitude of radial velocity
V	Z-dependent factor in radial velocity perturbation
w	Gas velocity in the tangential direction nondimensionalized by C_o*
W_o	Amplitude of tangential velocity oscillation
W_T	Total propellant weight flow rate, lb/sec
W_{fT}	Total fuel flow rate excluding film coolant, lb/sec
W_{XT}	Total oxidizer flow rate excluding film coolant, lb/sec
W_{E_i}	Flow rate of the ith element, lb/sec
$Y_\nu(\ )$	Bessel function of the second kind of order ν
$Y'_\nu(\ )$	Derivative with respect to the argument of the Bessel function of the second kind of order ν
Y_x	Mass fraction of vaporized oxidizer
Y_1, Y_2, Y_3	Variables used in transverse analysis
Z	Length along the chamber axis nondimensionalized by the chamber length L_c* for a longitudinal mode or by the chamber radius r_c* for a transverse mode
Z_e	Axial plane at nozzle entrance
Z^*_{1i}	Axial distance from throat to first transition from curve to tangent for inner contour of an annular nozzle, feet
Z^*_{1o}	Axial distance from throat to first transition from curve to tangent for outer contour of an annular nozzle, feet
Z^*_{2i}	Axial distance from throat to second transition from tangent to curve for inner contour of an annular nozzle, feet
Z^*_{2o}	Axial distance from throat to second transition from tangent to curve for outer contour of an annular nozzle, feet
Z^*_{3i}	Axial distance from throat to transition from chamber to nozzle of the inner contour of an annular nozzle, feet
Z^*_{3o}	Axial distance from throat to transition from chamber to nozzle, feet

I, D, Definition of Terms (cont.)

α_o	Convergence angle of outer contour of an annular nozzle (see Figure 14)
α_i	Convergence angle of inner contour of an annular nozzle (see Figure 14)
α_1	Real part of derivative of ζ at throat
α_o	Real part of ζ at throat
β	Angle of orientation relative to the θ coordinate (see Figure 10)
β_1	Imaginary part of derivative of ζ at throat
β_N	Longitudinal admittance coefficient for entropy, complex, $\beta_N = \beta_{N_r} + i\,\beta_{N_i}$
β_o	Imaginary part of ζ at throat
γ	Ratio of specific heats
Γ	Z-dependent factor in the first order solution for pressure defined by Equation 13
δ_r	Displacement of gas in the radial direction due to pressure oscillations
δ_θ	Displacement of gas in the tangential direction due to pressure oscillations
δ	Term used in longitudinal analysis
δ_m	Term used in longitudinal analysis
ΔP	Injector pressure drop, lb/in.^2
ζ	Variable in Z direction, used in various integrals
$\hat{\zeta}$	Auxiliary function in nozzle analysis complex
η	Index numbering the zeros in the derivative of the Bessel function
θ	Coordinate in the tangential direction
θ_1	Term used in longitudinal analysis
θ_F	Fuel angle of injection measured from Z axis
θ_x	Oxidizer angle of injection measured from Z axis
Θ	θ dependent factor in the solution for pressure defined in Equation 13
Θ *	Complex conjugate of Θ
κ	Momentum interchange coefficient between droplets and gas $* = \frac{9}{2}\,\frac{\mu^*}{r_d^{*2}\,\rho_L{}^*}$ nondimensionalized by $\frac{C_o{}^*}{r_c{}^*}$ or $\frac{C_o{}^*}{L_c{}^*}$ for transverse or longitudinal modes

I, D, Definition of Terms (cont.)

λ	Real part of complex frequency σ
μ	Burning rate distribution function = $\frac{\sigma}{\sigma_{mean}}$
μ^*	Viscosity in lb-sec/ft^2
ν	Order of the Bessel function which indicates the number of cycles of pressure oscillation when traversing in the tangential direction at a fixed radius, axial length and time
ν_F	Fuel injection velocity, ft/sec
ν_x	Oxidizer injection velocity, ft/sec
ξ	Factor involving droplet drag, frequency and liquid inertance defined in Equation 10f
ξ_1	Distance from injector to concentrated combustion front nondimensionalized by chamber length
ξ_L	Term used in longitudinal analysis
$\xi^{(1)}$	Auxiliary function in nozzle analysis (complex)
$\xi^{(2)}$	Auxiliary function in nozzle analysis (complex)
ρ	Density of chamber gases nondimensionalized by the stagnation gas density ρ_o^*
ρ_L^*	Density of liquid droplets, lb-sec^2/ft^4
σ	Complex frequency nondimensionalized as the reciprocal of τ_i
σ	Injection density
σ_{mean}	Mean injection density
τ_i	Portion of time lag which is insensitive to pressure oscillations nondimensionalized by $\frac{L_c^*}{C_o^*}$ for a longitudinal mode and $\frac{r_c^*}{C_o^*}$ for a transverse mode
τ	Mean sensitive time lag, nondimensional as τ_i
τ_T	Total time delay between injection of propellant and burning of that same propellant, nondimensional as τ_i
τ^*	Sensitive time lag, milliseconds
ϕ	Velocity potential in the exhaust nozzle
ϕ_P	Function relating input pressure to the combustion to the output (burning rate)

I, D, Definition of Terms (cont.)

ϕ_R	Function relating input radial velocity to the combustion to the output (burning rate)
ϕ_T	Function relating input tangential velocity to the combustion to the output (burning rate)
ϕ_1	Term used in longitudinal analysis
χ	ωt
ψ	Stream function in the exhaust nozzle
Ψ	r dependent factor in the solution for pressure defined by Equation 13 c
ω	Imaginary part of complex Laplace variable σ For a transverse mode in chamber $\omega = \frac{2 f^* r_c^*}{C_o^* 12}$ For a longitudinal mode in chamber $\omega = \frac{2 f^* L_c^*}{C_o^* 12}$ For the nozzle $\omega = \frac{2 f^* R^*_{AT_o}}{C^*_{th} 12}$
ω_f	Frequency of input signal to a nonlinear element
$\hat{\omega}$	Nondimensional frequency used in nozzle analysis (see Equation 44)
Ω	$(s_{\nu\eta}^2 - \omega^2)^{1/2}$
A	Nozzle admittance coefficient (see Equation 38) (complex number $A = A_r + i A_i$)
B	Nozzle admittance coefficient (see Equation 38) (complex number $B = B_r + i B_i$)
C	Nozzle admittance coefficient (see Equation 38) (complex number $C = C_r + i C_i$)
E	Combined nozzle admittance coefficient defined by Equation 49 (complex number $E = E_r + i E_i$)
P	Pressure sensitive combustion response defined as $n(1-e^{-\sigma\tau})$
R	Radial velocity sensitive combustion response defined by $l_r(1-e^{-\sigma\tau})$
R_δ	Radial displacement sensitive combustion response defined by $m_r(1-e^{-\sigma\tau})$
T_δ	Tangential displacement sensitive combustion response defined by $m_\theta(1-e^{-\sigma\tau})$

I, D, Definition of Terms (cont.)

τ	Tangential velo-ity sensitive combustion response defined by $l_\theta(1-e^{-\sigma T})$
$\bar{}$	A bar over a variable indicates it is a mean quantity and does not depend on time
'	A prime on a variable indicates it is a perturbation quantity
+	A (+) when used as a subscript, means before the combustion front
-	A (-) when used as a subscript, means after the combustion front

I, Problem (cont.)

E. SPECIAL OPTIONS

The stability solution of any given liquid rocket engine must consider the three main portions of that engine: (1) the injector, (2) the combustion chamber, and (3) the exhaust nozzle. It is clear that these considerations serve as the basis for building the computer program.

The complete program consists of nine separate subprograms which may be used independently or in various combinations depending upon the data available and the information required by the designer. Figure 16 shows the interrelationships of the programs and the optional routes for proceeding with the calculation. The individual programs are:

A. Distributed Combustion Analysis for Transverse Modes (not combined with longitudinal)

B. Concentrated Combustion Analysis Valid for any Mode

C. Nozzle Admittance. Here the resistive effects of the nozzle are calculated for both longitudinal and transverse modes.

D. Expansion of Results from Programs A and B by interpolation and add nonuniformity and velocity effects.

E. Injector Nonuniformity Coefficients. Calculates modulating coefficients for the effects resulting from uneven injection distribution and nonlinear combustion response.

F. Stability Zone Mapping (n, τ)

G. Not in use

H. Not in use

I. Nonlinear Combustion Response. Input to Program E.

J. Injected Mass Distribution Effects

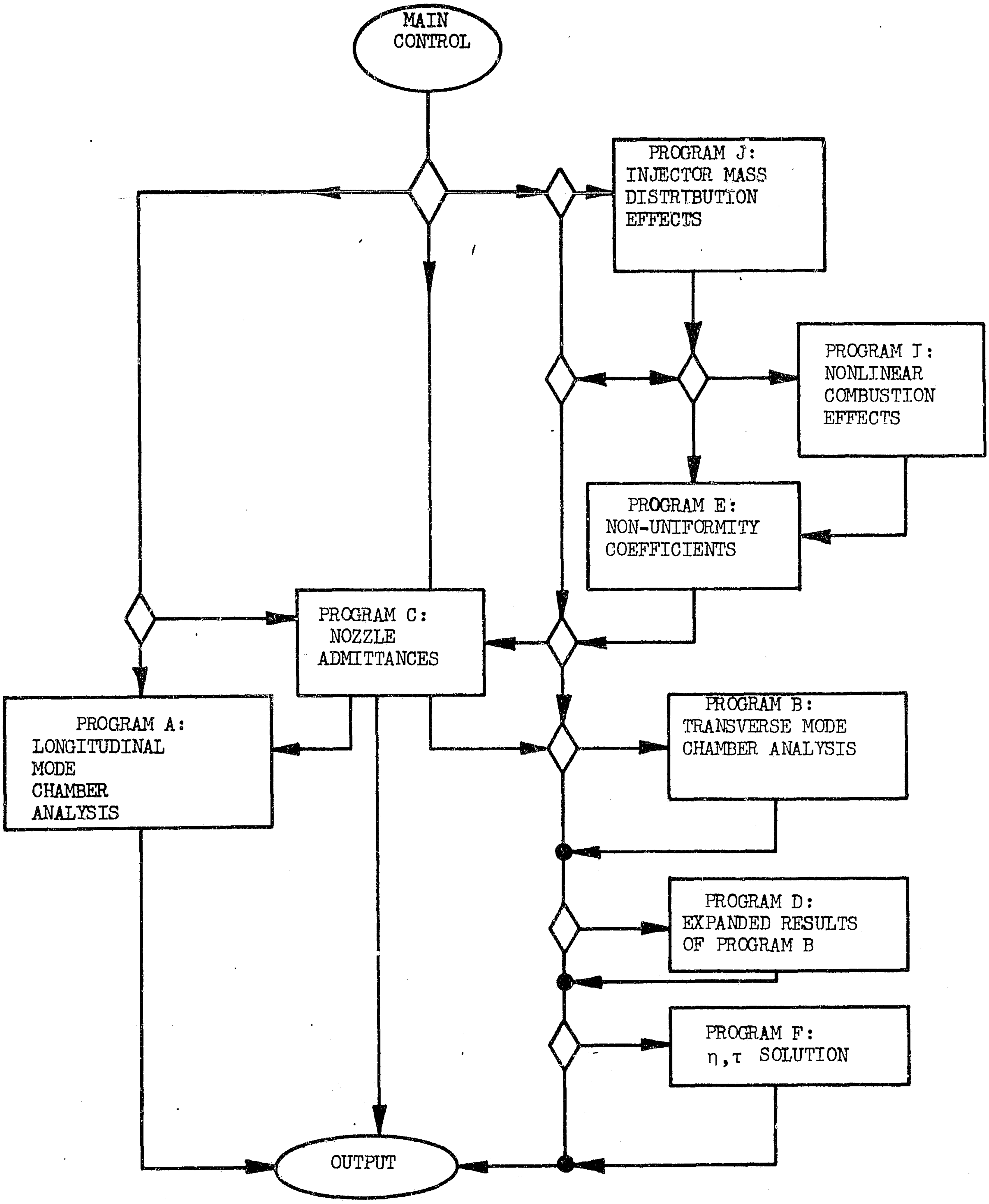


Figure 16 -- General Flow Diagram of the Computer Program

I, E, Special Options (cont.)

To assure that the proper subprograms are requested it is necessary to introduce a regulatory device that monitors the course of the entire program as it proceeds from one portion of the engine to another. This regulatory device is called MAIN CONTROL and serves to direct the flow of the solution in a consistent manner. Figure 16 illustrates the basic framework of the computer program. The diamond-shaped symbols can be considered as switches that are turned on if the particular item is to be analyzed by the program or turned off if the item is not needed. It is the function of MAIN CONTROL to turn these switches on or off according to the dictates of the problem.

The versatility of the program is illustrated in Figure 16 by the various flow-paths that can be constructed. For example, if the required injector and nozzle data were known beforehand, one could determine the transverse stability of the engine by merely running the combustion chamber program. If, on the other hand, all the data were unknown, proper use of MAIN CONTROL would initiate the injector, nozzle, and chamber programs. Therefore, the program is as suitable for parametric studies as it is for analysis of specific designs.

The injector requires programs J, I, and E to evaluate the injector parameters that affect stability. The chamber analysis requires A or B. Program C is run if the nozzle admittance is needed.

I, E, Special Options (cont.)

Following is an outline which will help determine which programs are to be run when evaluating an engine design.

I. Stability evaluation of an engine.

- A. Is the mode a pure transverse mode?
  Yes - either program A or B can be used.
  No - program B only must be used.

- B. Is the Mach number greater than about .3?
  Yes - program B must be used.
  No - programs A or B can be used.

- C. Is injection distribution uniform?
  Yes - A or B is all that is necessary.
  No - run (A or B) and D.

  - 1. Are distribution coefficients known?
    - a. Yes - run (A or B) and D.
    - b. No - run (A or B) and D and E.

      - (1) Is injection distribution known?
        - (a) Yes - run (A or B) and D and E.
        - (b) No - run (A or B) and D and E and J.

      - (2) Are non-linear effects desired?
        - (a) Yes - run (A or B) and D and E and I.
        - (b) No - run (A or B) and D and E.

- D. Are nozzle admittance coefficients known?
  - A. Yes - run A or B
  - B. No - run (A or B) and C

II. Evaluation of Nozzle Admittance Only.
Run program C alone.

There are many more possible combinations of programs that can be run but this should serve as a useful guide.

I, Problem (cont.)

F. NUMERICAL METHODS OF SOLUTION*

As seen in Section C, the nozzle admittance coefficients are defined in terms of a system of non-linear differential equations which must be integrated from the throat to the nozzle entrance. These equations are integrated using Adams-Bashforth integration. Adams-Bashforth integration, as developed at Aerojet, uses several subroutines: ADSET, ADINT, ADCOR, ADRES and ADPAR. A description of Adams-Bashforth integration, which is reproduced from an Aerojet manual, follows:

PURPOSE

To integrate a system of N simultaneous first order differential equations.

RESTRICTIONS

N 200

METHOD

Adam's method of integration is used. Accurate computation of fundtional values between integration steps is possible using partial step integration. The set of differential equations must be written as a system of first order equations in the form:

$$\frac{dy_1}{dx} = f_1\,(x, y_1, y_2, \ldots, y_n)$$

$$\frac{dy_n}{dx} = f_n\,(x, y_1, y_2, \ldots, y_n)$$

---

*Nomenclature in this section does not correspond to the rest of the report.

I, F, Numerical Methods of Solution (cont.)

USAGE

This subroutine has five separate entries:

1. The initial setup of the integration which specified number of equations and storage locations is done at the beginning of the program with the statement:

CALL ADSET (N, F, D, P, T, X, H, E)

with

N = number of first order differential equations to be integrated.

F = location of integrated y_1, y_2, y_n.

D = location of derivatives $\frac{dy_1}{dx}, \frac{dy_2}{dx}, \frac{dy_n}{dx}$

P = location of interpolated or partial step

$$y_1^p, y_2^p, \ldots, y_n^p.$$

T = temporary storage having 8N cells

X = independent variable

H = integration step (initially set at approximate value)

E = location of desired accuracy, E (1) = accuracy of y_1 ...

E (n) = accuracy of y_n. If E (I) = 0 then program sets

E(I) = 10^{-8}

I, F, Numerical Methods of Solution (cont.)

2. To integrate forward a single integration step use:

```
            CALL ADINT
     100    D(1) = ...
            D(2) = ...
              .
              .
              .
            D(N) = ...
            CALL ADCOR (& 100)
```

The ADINT entry will predict values of $y_1, \ldots, y_n$ at x+h and the ADCOR corrects these values. After the CALL ADCOR statement one has available the values of y_1, y_2, y_n at the new value of x.

3. To obtain functional values $y_1, \ldots y_n$ at a discontinuity or print point x_p use the statement:

```
            CALL ADPAR (XP)
```

This will compute y_1P, .. y_nP and put these values in the P(N) array. This will not affect the values in F, D, T.

4. To restart the integration such as at a discontinuity:

```
            CALL ADRES
```

This will zero out the difference table, compute new values for desired accuracy (e.g., if numbers in E have been changed) and set a flag to restart the integration.

The following two equations are the predicter and corrector equations written in terms of the backward difference operator $\nabla_n = y_n(x_k) - y_n(x_{k-1})$:

$$y_n^P(x_{k+1}) = y_n(x_k) + H\left(1 + \frac{1}{2}\nabla + \frac{5}{12}\nabla^2 + \frac{3}{8}\nabla^3 + \frac{251}{720}\nabla^4\right)\frac{dy_n(x_k)}{dx}$$

I, F, Numerical Methods of Solution (cont.)

$$y_n^{\,c}\,(x_{k+1}) = y_n\,(x_k) + H\left[\frac{251}{720}\,\frac{dy_n\,(x_{k+1})}{dx} + \left(\frac{469}{720} + \frac{109}{720}\,\nabla + \frac{49}{720}\,\nabla^2 + \frac{32}{720}\,\nabla^3\right)\frac{dy_n\,(x_k)}{dx}\right]$$

The predictor and corrector are in addition modified to account for the previous error. This effectively increases the order of the predictor and corrector. In these equations the integration error is controlled by changing H, the integration step size. The table of backward differences is modified whenever the integration step is changed. This table of backward differences is also used to interpolate for intermediate values of the function by using a 4th order interpolating polynomial.

Also in the nozzle admittance calculation a table of velocity potential is calculated by integrating the steady state velocity profile (Equation 43). Simpson's one third rule is used to evaluate this integral. This method approximates the curve to be integrated by a parabola.

In calculating the sensitive time lag τ an inverse tangent function of a fraction is required (Eq. 42). The inverse tangent in the computer library picks an angle between plus ninety degrees and minus ninety degrees. The correct quadrant of τ is determined by the signs of the numerator and denominator of the fraction. Subroutine QUAD calculates the correct quadrant.

I, F, Numerical Methods of Solution (cont.)

Boole's Integration

The Boole's integration subroutine for evaluating a definite integral in program A uses two entries: INTGR, INTGS. This subroutine uses Boole's formula and Simpson's rule to find the integral of a function between known limits. Boole's formula is used to extrapolate two Simpson's rule values with H = ΔX and 2 ΔX to obtain an answer for the integral which is more accurate than either of the two alone.

$$\int_a^b y\,dx = I_H = \frac{H}{3}\left[y(x_o) + 4\,y(x_1) + 2\,y(x_2) + 4\,y(x_3) + \cdots\cdots \cdots y(x_k)\right]$$

$$I_{Boole} = I_{H=\Delta X} + \frac{I_{H=\Delta X} - I_{H=2\Delta X}}{15}$$

The subroutines are used as follows:

```
      CALL INTGR (A,B,X,N)

100 CONTINUE

      F =

      CALL INTGS ($ 100,F,Y,E,M)
```

A = lower limit of independent variable
B = upper limit of independent variable
X = independent variable
N = initial number of increments
F = integrand
Y = integral
E = allowable relative error

M = final number of increments used

I, F, Numerical Methods of Solution (cont.)

Bessel Functions

The BESJ subroutine computes the J Bessel function for a given argument and integer order by using the recurrence relationship:

$$F_{n+1}(x) + F_{n-1}(x) = (\frac{2n}{x}) F_n(x)$$

The desired Bessel function is:

$$J_n(x) = \frac{F_n(x)}{\alpha}$$

where

$$\alpha = F_0(x) + 2 \sum_{M-1}^{M-2} F_{2m}(x)$$

M is initialized at M_0.

M_0 is the greater of M_A and M_B where:

$$M_A = \begin{cases} [X+6] \text{ for } X < 5 \\ [1.4X + 60/X] \text{ for } X \geq 5 \end{cases}$$

$$M_B = [n+x/4+2]$$

F_{M-2}, F_{M-3}, ..., F_2, F_1, F_0 is evaluated using the recurrence relationship above with $F_M = 0$ and $F_{M-1} = 10^{-30}$.

I, F, Numerical Methods of Solution (cont.)

α and $J_n(x)$ are then computed.

The computation is repeated for M+3.

The values of $J_n(x)$ for M and M+3 are compared.

If $\left| J_n(x)_M - J_n(x)_{M+3} \right| \leq \delta \left| J_n(x)_{M+3} \right|$

this value is accepted as $J_n(x)$; if not, the computation is repeated by adding 3 to M and using this as a new value for M. If M reaches M_{MAX} before the desired accuracy is obtained, execution is terminated. M_{MAX} is defined as:

$$M_{MAX} = \begin{cases} [20 + 10x - \frac{x^2}{3}] & \text{for } x \leq 15 \\ [90 + x/2] & \text{for } x > 15 \end{cases}$$

The BESY subroutine computes the Y Bessel function for a given argument x and order n. The recurrence relation:

$$Y_{n+1}(x) = (\frac{2n}{x}) \cdot Y_n(x) - Y_{n-1}(x)$$

is used for this evaluation.

I, F, Numerical Methods of Solution (cont.)

For x > 4

$$Y_0(x) = \left(\frac{2}{\pi x}\right)^{1/2} \left(P_0(x)\sin\left(x - \frac{\pi}{4}\right)\right.$$

$$\left. + Q_0(x)\cos\left(x - \frac{\pi}{4}\right)\right)$$

$$Y_1(x) = \left(\frac{2}{\pi x}\right)^{1/2} \left(-P_1(x)\cos\left(x - \frac{\pi}{4}\right)\right.$$

$$\left. + Q_1(x)\sin\ x - \frac{\pi}{4}\right)$$

$P_0(x)$, $Q_0(x)$, $P_1(x)$, and $Q_1(x)$ are:

$$\begin{aligned}\frac{1}{\sqrt{2\pi}} P_0\left(\frac{4}{t}\right) = {} & 0.3989422793 - 0.0017530620t^2 \\ & +0.0001734300t^4 - 0.0000487613t^6 \\ & +0.0000173565t^8 - 0.0000037043t^{10}\end{aligned}$$

$$\begin{aligned}\frac{1}{t\sqrt{2\pi}} Q_0\left(\frac{4}{t}\right) = {} & -0.124669441 + 0.0004564324t^2 \\ & -0.0000869791t^4 + 0.0000342468t^6 \\ & -0.0000142078t^8 + 0.0000032312t^{10}\end{aligned}$$

$$\begin{aligned}\frac{1}{\sqrt{2\pi}} P_1\left(\frac{4}{t}\right) = {} & 0.3989422819 + 0.0029218256t^2 \\ & -0.0002232030t^4 + 0.0000580759t^6 \\ & -0.0000200920t^8 + 0.0000042414t^{10}\end{aligned}$$

$$\frac{1}{t\sqrt{2\ \pi}} \quad Q_1\left(\frac{4}{t}\right) = \begin{array}{ll} 0.0374008364 & - 0.0006390400t^2 \\ +0.0001064741t^4 & - 0.0000398708t^6 \\ +0.0000162200t & - 0.0000036594t^{10} \end{array}$$

where $t = \frac{4}{x}$

For $x \leq 4$

$$Y_0(x) = \frac{2}{\pi} \sum_{m=0}^{15} (-1)^m \left(\frac{x}{2}\right)^{2m} \frac{1}{(m!)^2}$$

$$\left[\log \frac{x}{2} + \gamma - H_m\right]$$

where

$$H_m = \begin{cases} \sum_{r=1}^{m} \frac{1}{r} & \text{for } m \geq 1 \\ 0 & \text{for } m = 0 \end{cases}$$

and γ = Euler's constant = 0.5772156649

$$Y_1(x) = -\frac{2}{\pi\ x} + \frac{2}{\pi} \sum_{m=1}^{16} (-1)^{M+1} \left(\frac{x}{2}\right)^{2m-1}$$

$$\frac{1}{m!\ (m-1)!} \cdot \left[\log \frac{x}{2} + \gamma - H_m + \frac{1}{2m}\right]$$

I, Problem (cont.)

G. TECHNICAL REFERENCES

1. Crocco, L., "Aspects of Combustion Instability in Liquid Propellant Rockets," J. *Amer. Rocket Soc.*, Part I: vol. 21, November 1951, Part II: vol. 22, January 1952.

2. Reardon, F. H., Mc Bride, J. M., and Smith, A. J., Jr., *Effect of Injection Distribution on Combustion Stability*, AIAA Paper No. 65-613, AIAA Propulsion Joint Specialist Conference, Colorado Springs, Colorado, June 1965.

3. Gunder, D. F., and Friant, D. R., "Stability of Flow in a Rocket Motor," *J. Appl. Mech.*, vol. 17, p. 327, 1950.

4. Yachter, M., "Discussion of the Paper by Gunder and Friant," *J. Appl. Mech.*, vol. 18, p. 114, 1951.

5. Summerfield, M., "A Theory of Unstable Combustion in Liquid Propellant Rocket Systems," *J. Amer. Rocket Soc.*, vol. 21, p. 108, 1951.

6. Crocco, L., and Cheng, S. I., *Theory of Combustion Instability in Liquid Propellant Rocket Motors*, AGARDograph No. 8, Butterworths Scientific Pub., Ltd, London, 1956.

7. Scala, S. M., *Transverse Wave and Entropy Wave Combustion Instability in Liquid Propellant Rockets*, Princeton University Aeronautical Engineering Report No. 380, April 1957.

8. Reardon, F. H., *An Investigation of Transverse Mode Combustion Instability in Liquid Propellant Rocket Motors*, Princeton University Aeronautical Engineering Report No. 550, June 1961.

9. Bridge, J. F., and Angrist, S. W., "An Extended Table of Roots of $J_n'(x)\ Y_n'(\beta x) - J_n'(\beta x)\ Y_n'(x) = 0$", Mathematics of Computation, vol. 16, no. 78, p. 168, 1962.

10. Wieber, P. R., and Mickelsen, W., *Effect of Transverse Acoustic Oscillations on the Vaporization of a Liquid-Fuel Droplet*, NASA TN D-287, May 1960.

11. Crocco, L., and Sirignano, W. A., *Behavior of Supercritical Nozzles Under Three Dimensional Oscillatory Conditions*, AGARDograph 117, 1967.

12. Culick, F. E. C., *Stability of High Frequency Pressure Oscillations in Gas and Liquid Rocket Combustion Chambers*, M. I. T. Aerophysics Laboratory Report No. 480, June 1961.

I, G, Technical References (cont.)

13. Faddeev, D.K., and Faddeeva, V.N., Computational Methods of Linear Algebra, W.H. Freeman and Co., (1963).

14. Ralston and Wilf, Numerical Methods for Digital Computers, Wiley (1960).

15. Forsythe, G.E., J. SIAM, 5:74 (1957).

16. MATH-PACK, Program Abstracts UNIVAC 1108, Sperry-Rand Corp., 1967.

I, Problem (cont.)

H. RELATED PROJECTS

This program and its predecessors have been used to analyze many engines and proposed engines. The parts of the program concerned with velocity effects and non-linear response have not been used since not enough is known about either of these effects to define the parameters involved. These parts of the program are intended to be research tools.

One of the major programs on which this program has been used is the Gemini Stability Improvement Program (AF 04(695)-517), during which the effect of nonuniform injection on combustion stability was demonstrated. A series of contracts (NAS8-4008, NAS8-11741, NAS8-20672) which were concerned with the stability characteristics of hydrogen - oxygen at high chamber pressures (500 to 2500 psi) used this program to help correlate stability data. The program was also used on Apollo (NAAI-M6J7XAA-400000A), M-1 (NAS3-2555) and Transtage (F04695-68-C-0297).

II. PROGRAMMING

A. PROGRAM SUBROUTINES

Header	Subroutine	Description	Page
MAIN		Main Program	118
SUB01	BLOCK DATA		118
SUB02	OUTAS8(A,K)		118
SUB03	INAS58(A)		118
SUB04	INT4(X,Y,XI,YO)	Interpolation	118
SUB05	INTD4(X,Y,XI,YO,DY)		120
SUB06	PAGE(LINES)	Calls new page	121
SUB07	CHAMBER		121
SUB08	GENMEG(W)		121
SUB09	TRANS(BIN,XOUT,CB,KER)	Concentrated Combustion (Program B)	121
SUB10	DDD(DIN,DOUT,CD,NER,ERR)	Program D	122
SUB11	Deleted		-
SUB12	Deleted		-
SUB13	Deleted		-
SUB14	FFF(FIN,FOUT,CF,NER,W)	n, solution (Program F)	122
SUB15	QUAD(A,B,ANGLE)		123
SUB16	CORE(X,N,CODE)		123
SUB17	INJCTR		124
SUB18	JJJ	Program J	124
SUB19	INJDIS	Injector Distribution (Program E & I)	124
SUB20	TBLCAL	Annular Chamber	124
SUB21	Deleted		-
SUB22	AS138(C,ITITLE, IERR)	Simplified Input Procedure	124
SUB23	Deleted		-
SUB24	INTGR(A,B,X,N)	Boole's Integration	125
SUB25	DISTR	Distributed Combustion (Program A)	125
SUB26	ADSET(N,F,D,FP,T,X,HA,E)	Adam's Integration	125

II, A, Program Subroutines (cont.)

Header	Subroutine	Description	Page
SUB27	CCC	Nozzle Admittance	126
SUB28	BTLDA	Term in Nozzle Admittance	126
SUB29	ORTHLS	Curve Fit	127
SUB30	COEFS	Curve Fit	135
CGJR	CGJR	Solves Simultaneous Equations	141
BESJ	BESJ(X,N,BJ,D,IER)	Calculates Function	147
BESSEL	BESSEL(J,Y,V,X,K)	Calculates Function	148
BESY	BESY(X,N,BY,IER)	Calculates Function	149

II, A, Program Subroutines (cont.)

1. MAIN

   a. Purpose:

      Overall controlling subroutine

   b. User Procedure:

      EXECUTE MAIN

2. SUB01

   a. Purpose:

      Storage subroutine

   b. User Procedure:

      CALL BLOCK DATA

3. SUB02 - no longer used

4. SUB03 - no longer used

5. SUB04

   a. Purpose

      Table interpolation in a table given by a set of x's and a set of y's.

   b. User procedure

      (1) Entry

      CALL INT4(X,Y,XI,YO)

      where:

	DESCRIPTION	TYPE
X	is the first location of the table of X values	real; input
Y	is the first location of the table of Y values	real; input
XI	is the input X	real; input
YO	is the interpolated Y output	real; input

The end of the table is sensed when both x = 0 and y = 0. Thus in setting up the tables of x and y the last entry should be followed by a zero.

(2) Restrictions

There must be at least 3 points in the table. The last point in the table should be followed by a 0,0 point.

(3) Mathematical Method

The interpolation is done using an interpolating function giving a continuous function and a continuous derivative. Consider an example of six points. For a general interior point, such as between 3,4, a cubic interpolating polynomial is used passing through 3 and 4 and having the average of the slopes 2-3 and 3-4 at the point 3 and the average of the slopes 3-4 and 4-5 at the point 4. For the end regions, a quadratic interpolating polynomial is used passing through the two outermost points and having the average slope at the interior of the two.

The interior regions are controlled by the following equations.

$$D = x_3 - x_2$$

$$A1 = x - x_2$$

$$A2 = x - x_3$$

$$M_{12} = (Y_2 - Y_1)/(X_2 - X_1)$$

$$M_{23} = (Y_3 - Y_2)/X_3 - X_2)$$

$$M_{34} = (Y_4 - Y_3)/(X_4 - X_3)$$

$$Y = \frac{(A1)(A2)^2}{2D^2} (M_{12} - M_{23}) + \frac{(A2)(A1)^2}{2D^2} (M_{34} - M_{23}) - \frac{A_2 Y_2}{D} + \frac{A_1 Y_3}{D}$$

Likewise the end regions are controlled by

$$D = X_2 - X_1$$

$$A_1 = X - X_1$$

$$A_2 = X - X_2$$

$$Y = \frac{(A1)(A2)}{2D}\left(\frac{Y_3 - Y_2}{X_3 - X_2} - \frac{Y_2 - Y_1}{X_2 - X_1}\right) - \frac{A2}{D} Y_1 + \frac{A1}{D} Y_2$$

6. SUB05

a. Purpose

This subroutine interpolates in a table given by a set of x's and a set of y's. Both the function YO and the derivative DY are obtained at the point XI.

b. User procedure

(1) Entry

CALL INT4D(X,Y,XI,YO,DY)

where:

	DESCRIPTION	TYPE
X	is first location of x table	real; input
Y	is first location of y table	real; input
XI	is input x	real; input
YO	is interpolated y	real; input
DY	is derivative at interpolated point	real; input

(2) Restrictions

A minimum of 3 points on the curve is required. Last point in table should be followed by a 0,0 point.

(3) Mathematical Method

See writeup for SUB04.

II, A, Program Subroutines (cont.)

7. SUB06

   a. Purpose

      Calls new page

   b. User procedure

      CALL PAGE (LINES)

   where:

	DESCRIPTION	TYPE
LINES	Page indexing variable	input

8. SUB07

   a. Purpose

      Subroutine used to control logical flow of program

   b. User procedure

      CALL CHAMBER

9. SUB08

   a. Purpose

      To generate frequencies used in program

   b. User procedure

      CALL GENMEG(W)

   where:

	DESCRIPTION	TYPE
W	Numerical value associated with specific acoustic mode under consideration	Floating point; input

10. SUB09

   a. Purpose

      Calculates characteristic function for chamber with concentrated combustion.

b. User procedure

```
CALL TRANS(BIN,XOUT,CB,KER)
```

where:

	DESCRIPTION	TYPE
BIN	Input variables (see program listing)	floating point; input
XOUT	Chamber admittance	floating point; output
CB	Used to control amount of output	floating point; input
KER	Number of frequencies	floating point; input

11. SUB10

a. Purpose:

Includes injection distribution and velocity effects in chamber admittance calculations.

b. User procedure

```
CALL DDD(DIN,DOUT,CD,NER,ERR)
```

where:

	DESCRIPTION	TYPE
DIN	See BIN SUB09	
DOUT	See XOUT SUB09	
CD	See CB SUB09	
NER	See KER SUB09	
ERR	error message	output

12. SUB14

a. Purpose

Calculates n and τ values

b. User procedure

```
CALL FFF(FIN,FOUT,CF,NER,W)
```

where:

	DESCRIPTION	TYPE
FIN	See BIN SUB09	
FOUT	See XOUT SUB09	
CF	See CB SUB09	
NER	See KER SUB09	
W	No longer used	

13. SUB15

a. Purpose

Determines quadrant of τ

b. User procedure

```
CALL QUAD(A,B,ANGLE)
```

where:

	DESCRIPTION	TYPE
A	Chamber admittance-real part	floating point; input
B	Chamber admittance-imaginary part	floating point; input
ANGLE	Value of τ	floating point; output

14. SUB16

a. Purpose

Input data dump

b. User procedure

```
CALL CORE(X,N,CODE)
```

where:

	DESCRIPTION	TYPE
X	Value to be dumped	floating point; output
N	Number of dumps	integer; input
CODE	Code to determine if dump is performed	floating point; input

II, A, Program Subroutines (cont.)

15. SUB17

a. Purpose

Subroutine used to control logic of injector calculations

b. User procedure

CALL INJCTR

16. SUB18

a. Purpose

Calculates injection parameters

b. User procedure

CALL JJJ

17. SUB19

a. Purpose

Injection distribution calculations

b. User procedure

CALL INJDIS

18. SUB20

a. Purpose

Calculates table of velocity potential vs square of velocity used in nozzle admittance calculations.

b. User procedure

CALL TBLCAL

19. SUB22

a. Purpose

To Process input

b. User procedure

CALL AS138(C,ITITLE,IERR)

where:

	DESCRIPTION	TYPE
C	Input array	Real; output
ITITLE	Job title	output
IERR	Error indicator	integer; output

20. SUB24

a. Purpose

Performs Booles integration

b. User procedure

CALL INTGR(A,B,X,N)

See Section I,F for description of terms and methods of use

21. SUB25

a. Purpose

Calculates chamber admittance for transverse modes with distributed combustion

b. User procedure

CALL DISTR

22. SUB26

a. Purpose

Performs Adams Integration

b. User procedure

CALL ADSET(N,F,D,FP,T,X,HA,E)

See Section I,F for description of terms and methods of use

II, A, Program Subroutines (cont.)

23. SUB27

a. Purpose

Calculates Nozzle Admittance

b. User procedure

CALL CCC

24. SUB28

a. Purpose

Calculates term for nozzle admittance

b. User procedure

CALL BTLDA

25. SUB29 (Ref 16)

a. Purpose

This subroutine finds the least squares polynomial which best approximates a weighted set of data points using orthogonal polynomials.

b. User Procedure

(1) Entry

```
CALL ORTHLS(X,Y,W,N,L,J,C,ALPHA,BETA,K,T1,T2,T3,IND1)
```

where:

	DESCRIPTION	TYPE
X	is the array of N data points for the independent variable.	floating-point array; input and output
Y	is the array of N data points for the dependent variable.	floating-point array; input and output
W	is the array of N weights for the data points.	floating-point array; input
N	is the number of elements in the X,Y, and W arrays. Also, N is the number of elements in the T1, T2 and T3 arrays.	FORTRAN integer; input
L	is the weight switch: = 0 if weights are all set equal to one; = 1 if weights are input in the W array.	FORTRAN integer; input
J	is the number of low-order coefficients to be set equal to zero	FORTRAN integer; input
C	is the array of K + 1 computed polynomial coefficients.	floating-point array; output

	DESCRIPTION	TYPE
ALPHA	is the computed α array.	floating-point array; output
BETA	is the computed β array.	floating-point array; output
K	is the maximum degree of the polynomial to be fitted. Also, K + 1 is the number of elements in the C array and K is the number of elements in the ALPHA and BETA arrays.	FORTRAN integer; input
T1	is an array of N elements used for temporary storage by ORTHLS. Its contents upon return from ORTHLS are of no significance to the user.	floating-point array
T2	is an array of N elements used for temporary storage by ORTHLS. The contents upon return from ORTHLS are of no significance to the user.	floating-point array
T3	is an array of N elements for temporary storage by ORTHLS. The contents upon return from ORTHLS are of no significance to the user.	floating-point array
IND1	is the error indicator: = -1, when $J > K$. = +1, when $J \leq K$.	FORTRAN integer: output

(2) Restrictions

(1) $J \leq K$

(2) $N > K$, since a minimum of N + 1 data points are required to fit a polynomial of degree K.

(3) Special Considerations

(a) If L = 1 in the calling sequence (i.e., non-unit weights), the input data points, X and Y, are altered when ORTHLS is executed.

(b) Since there are K + 1 coefficients for a Kth degree polynomial, the C array must be dimensioned for a minimum of K + 1 elements in the calling program. The computed coefficients are atored as:

$$
\begin{array}{ll}
c_0 & \ldots\ C(1) \\
c_1 & \ldots\ C(2) \\
\vdots & \quad\ \vdots \\
c_k & \ldots\ C(K + 1)
\end{array}
$$

(4) Error Returns

If, upon entering ORTHLS, J > K:

(a) the error indicator IND1 is set equal to -1;

(b) all the elements in the C, ALPHA and BETA arrays are set equal to zero.

(c) control returns to the calling program through the normal exit.

c. Supporting Information

(1) Mathematical Method

Let

$$y = \sum_{j=0}^{k} C_j P_j(X) \tag{55}$$

where $P_j(x)$ is a polynomial of degree j. To obtain the best fit in the weighted least-squares sense to the set of data points (X_i,Y_i), $i = 1,2,...N$ with positive non-zero weights (W_i), $i = 1,2,...N$, then

$$\sum_{i=1}^{N} W_i^2 \left[Y_i - \sum_{j=0}^{K} C_j P_j(X_i) \right]^2 = \text{minimum} \tag{56}$$

By letting

$$r_{jm} = \sum_{i=1}^{N} W_i^2 P_j(X_i) P_m(X_i) \tag{57}$$

$$S_j = \sum_{i=1}^{N} W_i^2 Y_i P_j(X_i) \quad j = 0,1,2...K \tag{58}$$

A set of normal equations of the following form is obtained.

$$S_j = \sum_{m=0}^{K} C_m r_{jm} \qquad j = 0,1,2\ldots K \tag{59}$$

Equation (59) consists of a set of K + 1 simultaneous equations with K + 1 unknowns. However, suppose that the polynomials $P_j(X)$ can be chosen to be orthogonal in the summation; i.e.,

$$r_{jm} = \begin{cases} 0 & \text{for } j \neq m \\ \sum_{i=1}^{N} W_i^2 P_j^2(X_i) & \text{for } j = m \end{cases} \tag{60}$$

Under the condition of orthogonality, the unknown coefficients C_j can now be evaluated by

$$C_j = \frac{S_j}{r_{jj}} \qquad j = 0,1,2\ldots K \tag{61}$$

Consequently, the use of orthogonal polynomials makes the least-squares analysis appear as if each parameter were the only one used. Also, the coefficients C_j are independent of each other; as K is increased, the previously calculated C_j do not change. However, the problem of generating the orthogonal polynomials must be solved.

The polynomials $P_j(X)$ are defined recursively as

$$\left.\begin{aligned} &P_{-1}(X) = 0 \\ &P_0(X) = 1 \\ &P_1(X) = (X - \alpha_1)P_0(X) - \beta_0 P_{-1}(X) \\ &\cdot \\ &\cdot \\ &\cdot \\ &P_{J+1}(X) = (X - \alpha_{j+1})P_j(X) - \beta_j P_{j-1}(X) \end{aligned}\right\} \qquad (62)$$

with $\beta_0 = 0$. The α and β must be chosen in such a way that the polynomials $P_0(X)$, $P_1(X)$, ...$P_j(X)$ are all orthogonal to each other in the sense of Equation (60).

The (J + 1)th polynomial is defined as

$$P_{J+1}(X_i) = (X_i - \alpha_{j+1})P(X_i) - \beta_j P_{j-1}(X_i) \qquad (63)$$

Multiplying Equation (63) through by $W_i^2 P_j(X_i)$ and summing over all i yields

$$\begin{aligned} \sum_{i=1}^{N} W_i^2 P_{j+1}(X_i)P_j(X_i) &= \sum_{i=1}^{N} W_i^2 (X_i \cdot \alpha_{j+1})P_j^2(X_i) + \\ &- \sum_{i=1}^{N} W_i^2 \beta_j P_{j-1}(X_i)P_j(X_i) \end{aligned} \qquad (64)$$

The term on the left-hand side of Equation (64), is zero since $j + 1 \neq j$ and Equation (60) requires the summation to vanish for $j \neq m$. Likewise, the second term on the right-hand side vanishes leaving

$$\sum_{i=1}^{N} W_i^2(X_i - \alpha_{j+1})P_j^2(X_i) = 0 \qquad (65)$$

Since α_{j+1} does not depend on i, Equation (65) can be rearranged to yield

$$\alpha_{j+1} = \frac{\sum_{i=1}^{N} W_i^2 X_i P_j^2(X_i)}{\sum_{i=1}^{N} W_i^2 P_j^2(X_i)} \qquad j = 0,1,\ldots,K \qquad (66)$$

By multiplying Equation (63) through by $W_i^2 P_{j-1}(X_i)$ and again using Equation (60) it can be shown that

$$\beta_j = \frac{\sum_{i=1}^{N} W_i^2 X_i P_{j-1}(X_i) P_j(X_i)}{\sum_{i=1}^{N} W_i^2 P_{j-1}^2(X_i)} \qquad j = 0,1,2\ldots K - 1 \qquad (67)$$

Multiplying Equation (63) through by $W_i^2 P_{j+1}(X_i)$ and summing over all i, yields the identity

$$\sum_{i=1}^{N} W_i^2 P_{j+1}^2(X_i) \equiv \sum_{i=1}^{N} W_i^2 X_i P_j(X_i) P_{j+1}(X_i) \qquad (68)$$

Thus, substituting Equation (68) into Equation (67) β_j may be computed by an alternative formula,

$$\beta_j = \frac{\sum_{i=1}^{N} W_i^2 P_j^2(X_i)}{\sum_{i=1}^{N} W_i^2 P_{j-1}^2(X_i)} \qquad j = 0,1,2\ldots K - 1 \qquad (69)$$

II, A, Program Subroutines (cont.)

(2) Programming Method

The subroutine ORTHLS has the option of placing certain constraints on the form of the fitted polynomial, that is, if J is a non-zero positive integer, then the fitted polynomial has the form:

$$Y = \sum_{j=0}^{K-J} C_j \overline{P}_j(X) \tag{70}$$

where the $\overline{P}_j(X)$ are defined recursively as

$$\left.\begin{aligned}
&\overline{P}_{-1}(X) = 0\\
&\overline{P}_0(X) = X^J\\
&\overline{P}_1(X) = (X - \alpha_1)\overline{P}_0(X) - \beta_0\overline{P}_{-1}(X)\\
&\cdot\\
&\cdot\\
&\cdot\\
&\overline{P}_{j+1}(X) = (X - \alpha_{j+1})\ \overline{P}_j(X) - \beta_j\overline{P}_{j-1}(X)
\end{aligned}\right\} \tag{71}$$

It can be shown that the set $\overline{P}_0, \overline{P}_1, \ldots \overline{P}_{j+1}$, are orthogonal by requiring that α and β satisfy the same relations (Equations (66) and (67)) as before. These constraints allow the first J coefficients in Equation (70) to be set equal to zero thus forcing the fitted polynomial and its first J - 1 derivatives to be zero at X = 0.

The subprogram ORTHLS uses Equations (61), (66), and (69) to compute the orthogonal coefficients C_j defined by Equation (55) and the parameters α_j and β_j, respectively.

Further information can be found in Reference 15.

(3) Storage

ORTHLS	285
NEXP5$	33
Total Storage:	318 positions.

(4) Nomenclature

B	is the recursive β used in computing the polynomials.
I	is the index used for summing over all the data points.
II	is the index used to determine the successful completion of the calculation.
KJ1	is the value of K - J + 1.
R	is the parameter defined by Equation (57).
RO	is the previous value of R.
S	is the parameter defined by Equation (58).
SUM	is the initial value of RO.
SUMXPS	is the numeration of Equation (66).
TEMP	is the variable used for computing the (j + 1)th polynomial from the jth and (j - 1)th polynomials.

All other variables are defined in paragraph II.A.25.b.(1).

26. SUB30 (Ref 16)

a. Purpose

This subroutine computes the coefficients of the least-squares polynomial which best approximates the weighted set of data points.

Given the coefficients, c_j, and the orthogonality parameters of the polynomials $P_j(x)$ for the equation

$$y = \sum_{j=0}^{K-J} C_j P_j(x)$$

COEFS computes the coefficients a_j for the equation

$$y = \sum_{j=J}^{K} a_j x^j$$

b. User Procedure

(1) Entry

CALL COEFS(J,C,ALPHA,BETA,KC,A,T1,T2,T3,IND2)

where:

	DESCRIPTION	TYPE
J	is the number of low-order coefficients to be set equal to zero.	FORTRAN integer; input
C	is the array of KC + 1 polynomial coefficients.	floating-point array; input
ALPHA	is the α array of KC elements.	floating-point array; input
BETA	is the β array of KC elements.	floating-point array; input
KC	is the degree of the polynomial for which the coefficients are to be found. Also, KC + 1 is the number of elements in the arrays C, A, T1, T2, and T3. KC is also the number of elements in the ALPHA and BETA arrays.	FORTRAN integer; input
A	is the computed coefficient array with KC + 1 elements.	floating-point array; output
T1	is an array of KC + 1 elements used for temporary storage by COEFS. The contents upon return from COEFS are of no significance to the user.	floating-point array

	DESCRIPTION	TYPE
T2	is an array of KC + 1 elements used for temporary storage by COEFS. The contents array upon return from COEFS are of no significance to the user.	floating-point array
T3	is an array of KC + 1 elements used for temporary storage by COEFS. The contents array upon return from COEFS are of no significance to the user.	floating-point array
IND2	is an error indicator. = -2, when $J > KC$ = $+2$, when $J \le KC$	FORTRAN integer; output

(2) Restrictions

(a) $J \le KC$.

(b) $KC \le K$, where K is the maximum degree of the polynomial used by subprogram ORTHLS.

(3) Special Considerations

(a) The value of J used by COEFS must be the same as that used by subprogram ORTHLS in computing the coefficients c_j, and the parameters α_j and β_j.

(b) The C, A, T1, T2, T3 arrays must be dimensioned for a minimum of KC + 1 elements in the calling program. The input coefficients should be stored as:

c_0 ... C(1)

c_1 ... C(2)

c_{KC} ... C(KC + 1)

The computed coefficients are stored as:

a_0 ... A(1)

a_1 ... A(2)

a_{KC} ... A(KC + 1)

(4) Other Subprograms Required

The subprogram COEFS requires that the subprogram ORTHLS be executed first. ORTHLS computes the coefficients, c_j, and the parameters α_j and β_j used by COEFS in computing the coefficients a_j.

(5) Error Returns

If, upon entering COEFS, J > KC:

The error indicator IND2 is set equal to -2; all elements in the A array are set equal to zero; control returns to the calling program through the normal exit.

c. Supporting Information

(1) Mathematical Method

Let

$$y = \sum_{j=0}^{KC-J} C_j P_j(x) \tag{72}$$

with the polynomials defined recursively as

$$\left.\begin{aligned}
&P_{-1}(x) = 0\\
&P_0(x) = x^J\\
&P_1(x) = (x - \alpha_1)P_0(x) - \beta_0 P_{-1}(x)\\
&\quad\vdots\\
&P_{j+1}(x) = (x - \alpha_{j+1})P_j(x) - \beta_j P_{j-1}(x)\\
&\text{with } \beta_0 = 0.
\end{aligned}\right\} \qquad (73)$$

Given the coefficients C_j, and the parameters J, α_j, β_j and KC, COEFS computes the coefficients, a_j, of the equation

$$y = \sum_{j=J}^{KC} a_j x^j \qquad (74)$$

(2) Programing Method

See Mathematical Method.

(3) Storage

COEFS: 198 positions.

II, A, Program Subroutines (cont.)

(4) Nomenclature

B	is the recursive β value used in computing the coefficients.
II	is the index used to determine the successful completion of the calculations.
KCJ1	is the value of KC - J + 1.
N1	is the index used in arranging the coefficients properly if J > 0.
N2	is the index used in arranging the coefficients properly if J > 0.
NN	is the index used in accumulating the coefficients.

All other variables are defined in paragraph II.A.26.b.(1).

II, A, Program Subroutines (cont.)

27. CGJR - (Ref 16)

a. Purpose

This subroutine solves simultaneous equations, computes a determinant, or inverts a matrix or any combination of the three above by using a Gauss-Jordan elimination technique with column pivoting.

b. User Procedure

(1) Entry

```
CALL CGJR(A,NC,NR,N,MC,$k,JC,V)
```

where:

	DESCRIPTION	TYPE
A	is the matrix whose inverse or determinant is to be determined. If simultaneous equations are solved, the last MC - N columns of the matrix are the constant vectors of the equations to be solved. On output, if the inverse is computed, it is stored in the first N columns of A. If simultaneous equations are solved, the last MC - N columns contain the solution vectors.	complex array; input and output
NC	is the maximum number of columns of the array A.	FORTRAN integer; input
NR	is the maximum number of rows of the array A.	FORTRAN integer; input
N	is the number of rows of the array A.	FORTRAN integer; input
MC	is the number of columns of the array A. This entry is a dummy argument, if sumultaneous equations are not to be solved.	FORTRAN integer; input

II, A, Program Subroutines (cont.)

	DESCRIPTION	TYPE
k	is a statement number in the calling program to which control is returned if an overflow is detected. It must be preceded by $ in the calling sequence.	input
JC	is a one-dimensional permutation array of N elements used for permuting the rows and columns of A if an inverse is computed. If an inverse is not computed this argument must have at least one cell for the error return identification. On output, the first element of the array is N if control is returned normally. If an overflow is detected, the first element is the negative of the last correctly completed row of the reduction. If matrix singularity is detected, the entry contains the value of the last row before the singularity was detected.	FORTRAN integer array; input and output
V	on input REAL (V) is the option indicator, its values are set as follows:	complex variable; input and output

	REAL(V)						
Operation	1.	2.	3.	4.	5.	6.	7.
Compute Determinant	no	yes	yes	no	no	yes	yes
Invert Matrix	yes	no	yes	no	yes	no	yes
Solve Equations	no	no	no	yes	yes	yes	yes

On normal return from the program V contains the value of the natural logarithm of the determinant. If an error return is made and the determinant to be computed is set to (0,0), or if an overflow return was made, V contains the last correct partially-computed value of the determinant.

(2) Restrictions

None

c. Special Considerations

(1) If the matrix is singular or ill-conditioned, roundoff error may cause large discrepancies in the results.

(2) In the case of a singular matrix, return may not be made through the singularity exit because of roundoff error.

(3) See 7.1.2.3. Ref 16 for notes on usage of the row-dimension arguments N and NR.

d. Error Returns

(1) If a singularity is detected, the first element of the array JC is set to the number of the last correctly completed row; and V is set to (0.,0.), if the determinant was to be computed. Control is then returned to the calling program at the statement number specified.

(2) If an overflow is detected, JC(1) is set to the negative of the last correctly completed row of the reduction. Control is then returned to the calling program at the statement number specified.

e. Mathematical Method

For any matrix A, if a matrix B exists such that $BA = AB = I$, where I is the unit matrix, then $B = A^{-1}$.

If $AX = C$ where A is n by n, X is n by p, and C is n by p, then the solution to these sets of simultaneous equations is $X = A^{-1}C$.

II, A, Program Subroutines (cont.)

The determinant of A is defined by the following equation

$$A = \Sigma \; (-1)^{f(j_1,\ldots,j_N)} \prod_{i=1}^{N} a_{ij_i}$$

where

a_{i_j} is the (i,j)th element of the matrix A; $f(j_1,\ldots j_N)$ is the number of transpositions required to transform $(1,\ldots,N)$ to $(j_1\ldots\ldots,j_N)$; and the summation is over all permutations $(j_1,\ldots,j_N)$ of the integers $(1,\ldots,N)$.

The solution to all of these problems is found by using a Gauss-Jordan elimination scheme with maximal pivoting by columns. More information can be found in References 13 and 14.

f. Programming Method

(1) For each diagonal, the program searches for a pivotal element for each column below the diagonal by finding the element of maximum norm in the remaining rows of the column. The norm is defined as Norm $(Z) = |X| + |Y|$, where a complex number is represented as $Z = X + iY$.

(2) This row is interchanged with the row of the diagonal.

(3) Each of the elements of the pivotal row is divided by the pivot except the pivot which is replaced by its reciprocal.

II, A, Program Subroutines (cont.)

(4) All other rows of the array are changed by the formula

$$a_{ij} = a_{ij} - a_{ik}a_{kj}$$

where

a_{kk} is the pivotal element. If $i = k$, a_{ij} is replaced by 0.

(5) When this process has been completed for each diagonal of the array, the columns of the matrix are repermuted to give the inverse in the first N columns of the array A.

(6) If the determinant is to be found, each permutation of rows and columns changes the value of its sign. The complex natural logarithm of the absolute value of the diagonal element is summed after step 2.

(7) Only the computations necessary for the options specified are carried out.

g. Storage

GJR 557 positions, not including the Library Subroutines OVERFL, CLOG, NERR$2, and NCDP$.

h. Nomenclature

ANQRM	is used for storing the complex norm of elements of the matrix A.
IBIT	is the number of sign changes in the calculation of the determinant.
IFL	is the overflow test indicator.

II, A, Program Subroutines (cont.)

IW	is the FORTRAN integer value of the option indicator.
KD	is the option key for determinant evaluation.
KI	is the option key for matrix inversion.
L	is the column control for solution of simultaneous equations.
M	is the column control for matrix inversion.
MU	is a dummy variable.
S	is the sign control for determinant evaluation.
X	is a dummy variable.
XC	is a dummy complex variable used in the permutation of rows of the matrix A.
Z	is a complex variable used to store the value of CLOG(A(I,I)).

All other variables are either simple loop indices or as defined earlier in this section.

II, A, Program Subroutines (cont.)

28. BESJ

a. Purpose

To calculate the Bessel function of the first kind

b. User Procedure

CALL BESJ (X,N,BJ,D,IER)

where:

	DESCRIPTION	TYPE
X	argument	floating point; input
N	order	integer; input
BJ	value of Bessel function	floating point; output
D	error criteria	floating point; input
ER	error code	integer; output

See Section I,F for further information.

II, A, Program Subroutines (cont.)

29. BESSEL

a. Purpose

Calculates Bessel functions

b. User Procedure

```
CALL BESSEL (J,Y,V,X,K)
```

where :

	DESCRIPTION	TYPE
J	Bessel function of 1st kind	floating point; output
Y	Bessel function of 2nd kind	floating point; output
V	Order	integer; input
X	argument	floating point; input
K	error code	integer; output

See Section I.F. for further information

II, A, Program Subroutines (cont.)

30. BESY

a. Purpose

Calculates Bessel function of the second kind

b. User Procedure

CALL BESY (X,N,BY,IER)

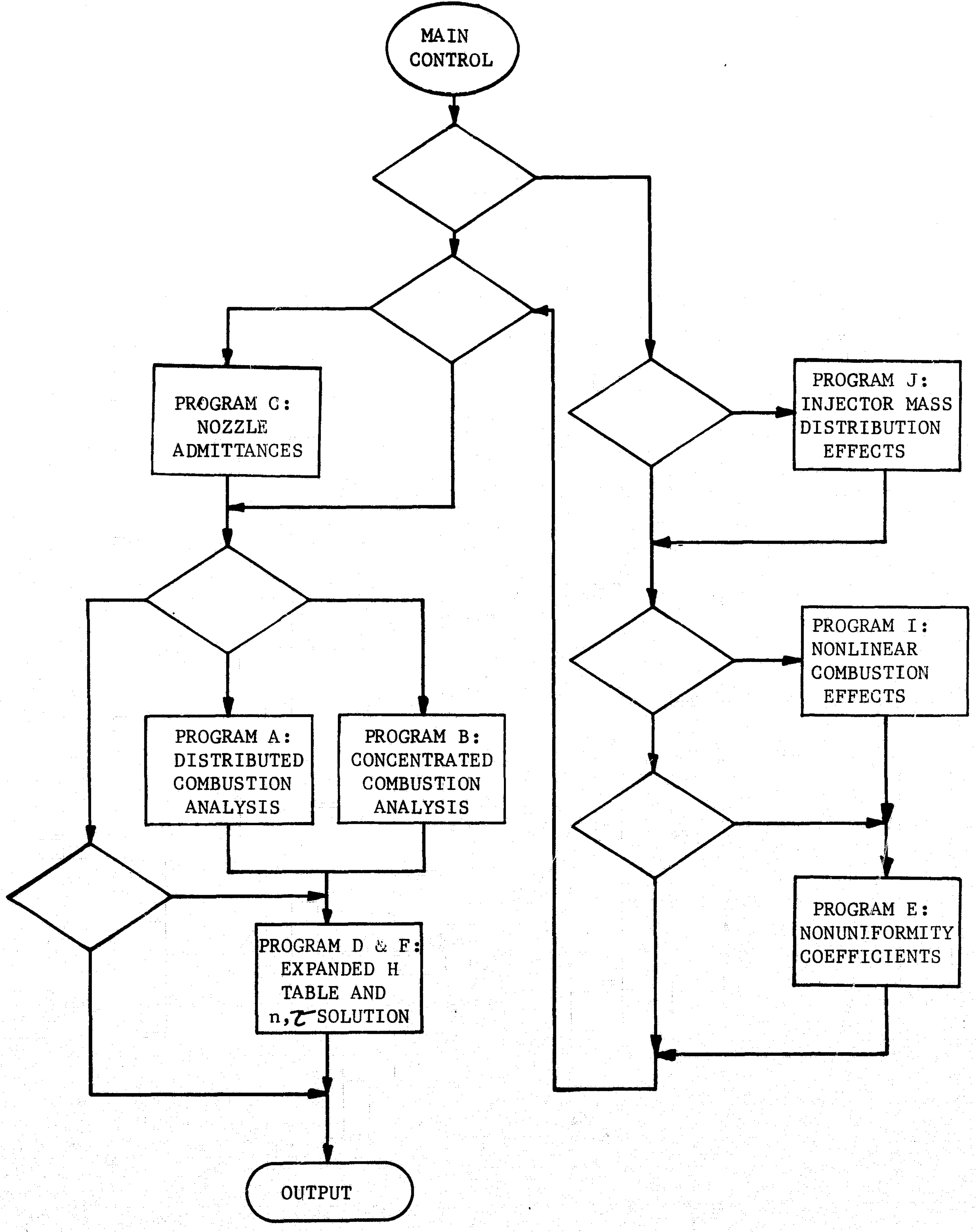
where :

	DESCRIPTION	TYPE
X	argument	floating point; input
N	order	integer; input
BY	value of Bessel function	floating point; output
IER	error code	integer; output

See Section I.F. for further information

II, Programing (cont.)

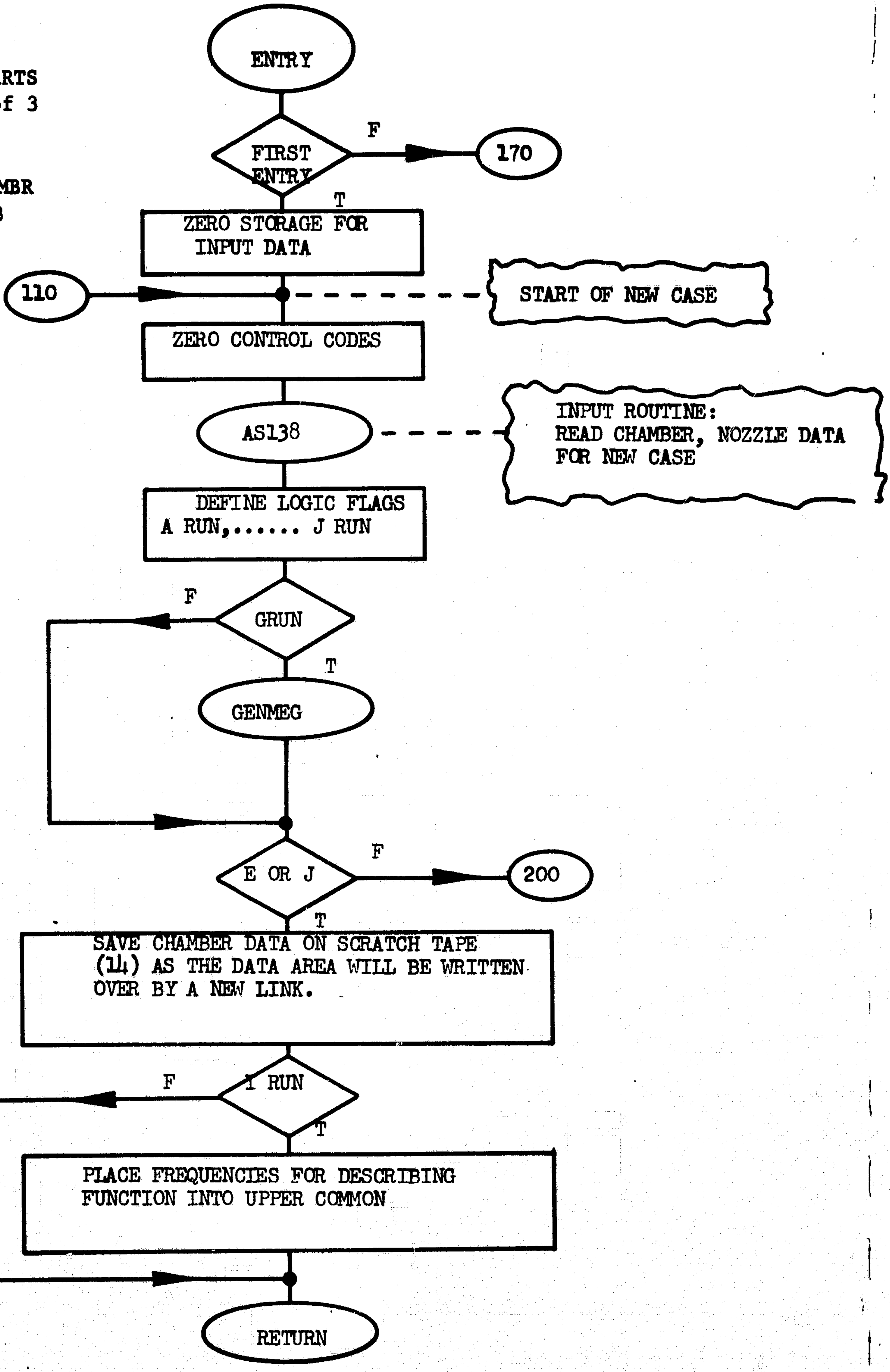
B. BLOCK DIAGRAM



II, Programming

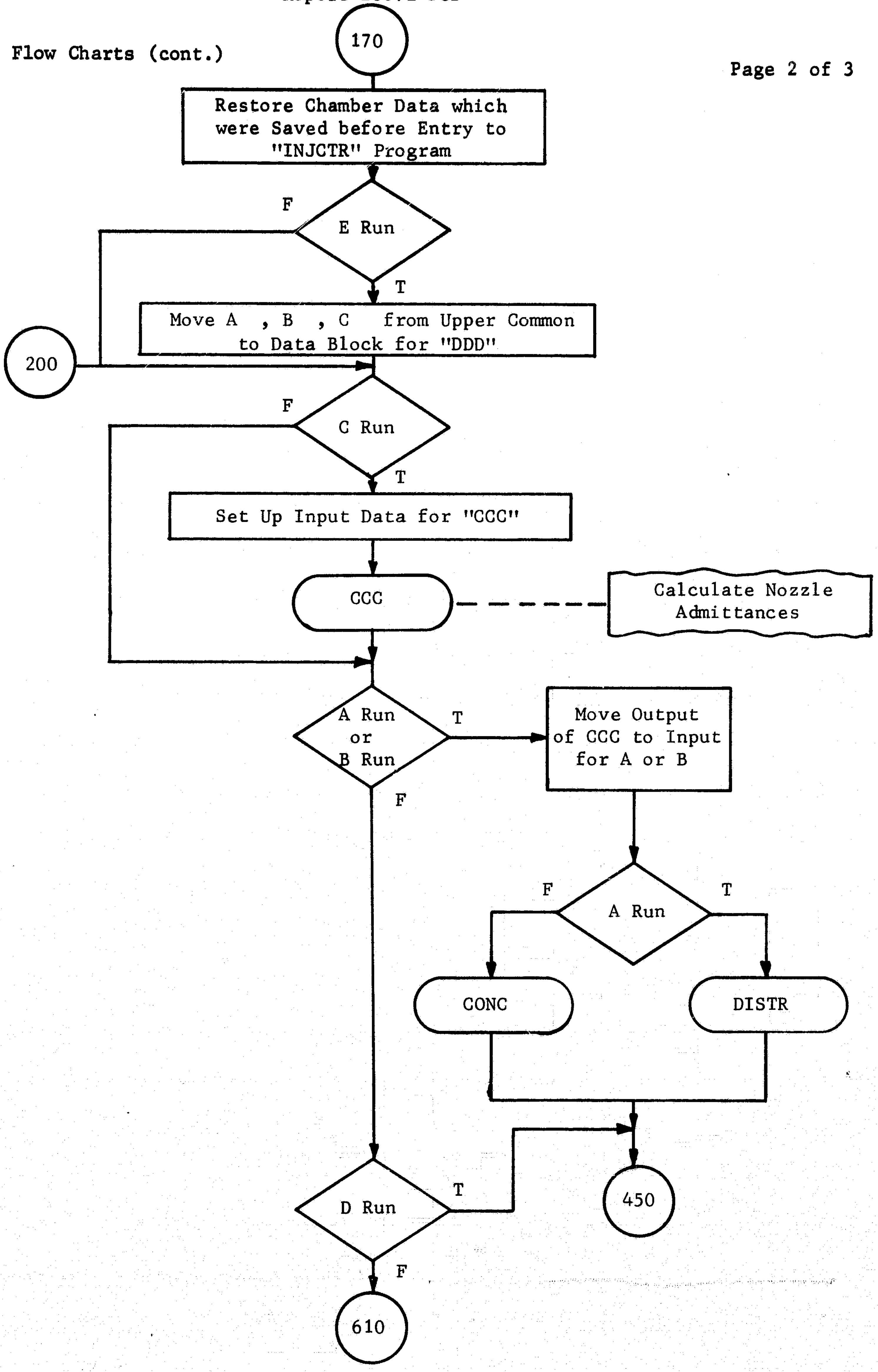
C. FLOW CHARTS
Page 1 of 3

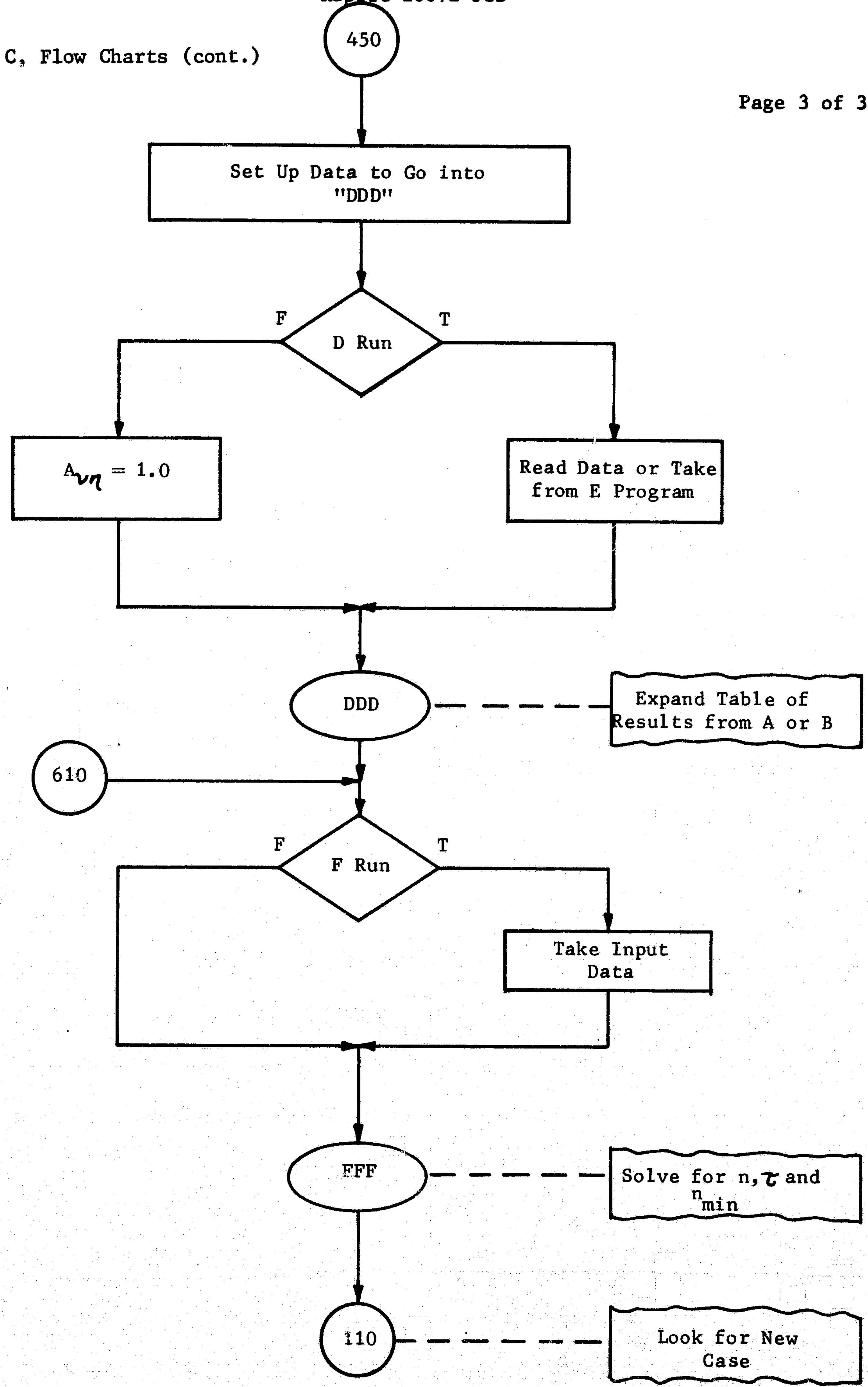
ENTRY = CHAMBR
Page 1 of 3



II, C, Flow Charts (cont.)

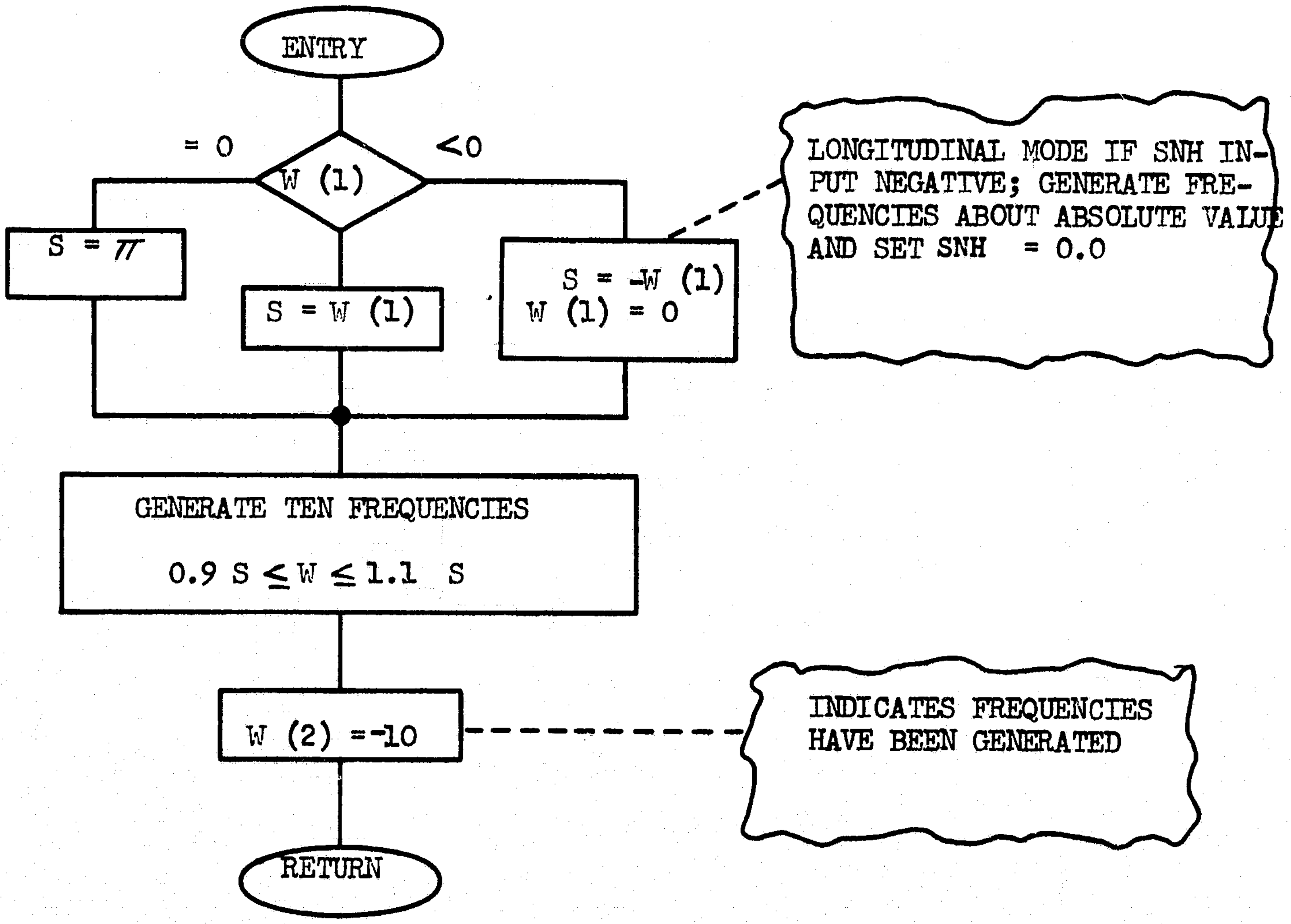
Page 2 of 3




450
Set Up Data to Go into "DDD"
F
D Run
T
$A_{\nu\eta} = 1.0$
Read Data or Take from E Program
DDD
Expand Table of Results from A or B
610
F
F Run
T
Take Input Data
FFF
Solve for n, τ and n_{min}
110
Look for New Case

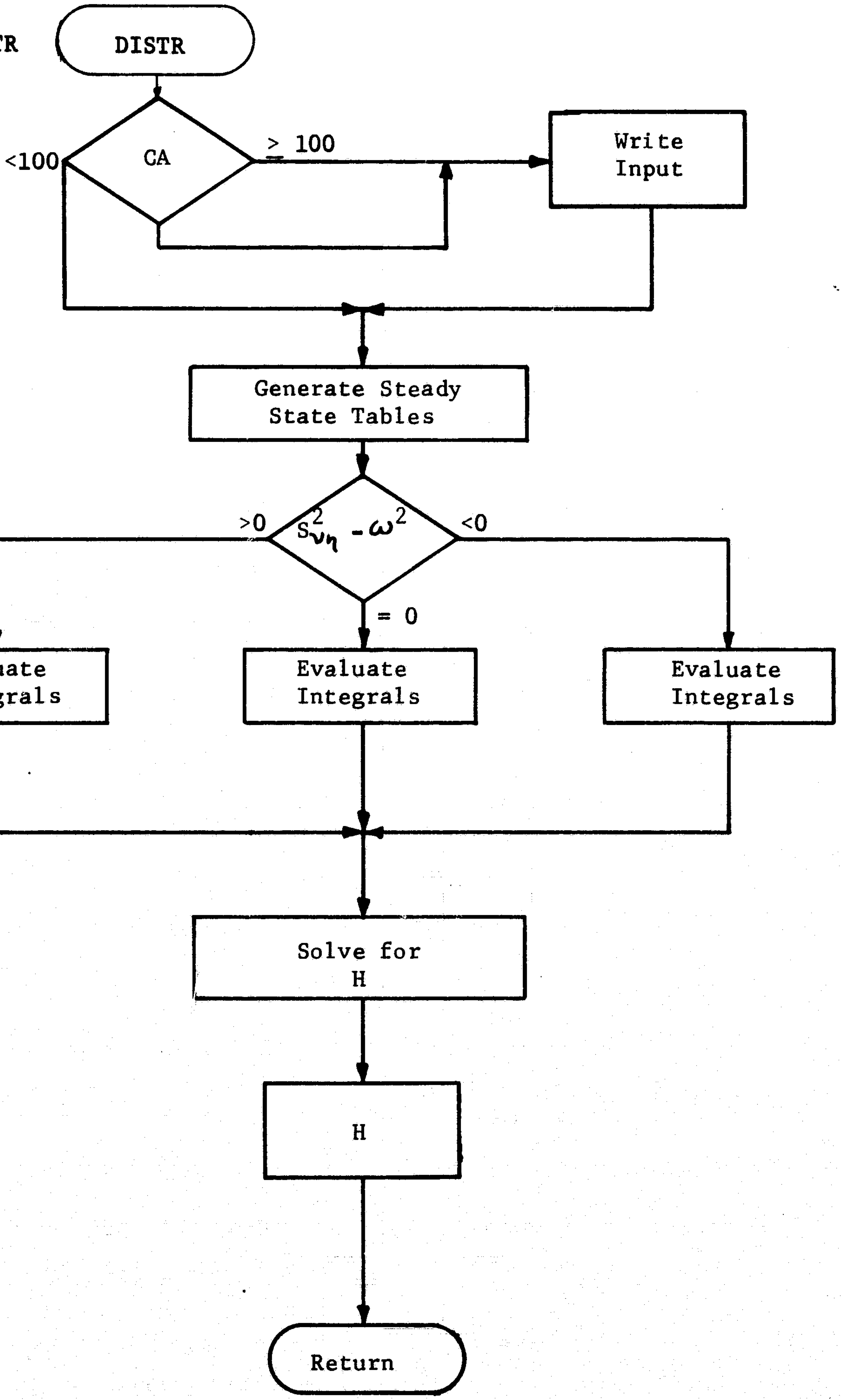
II, C, Flow Charts (cont.)

ENTRY = GEMMEG



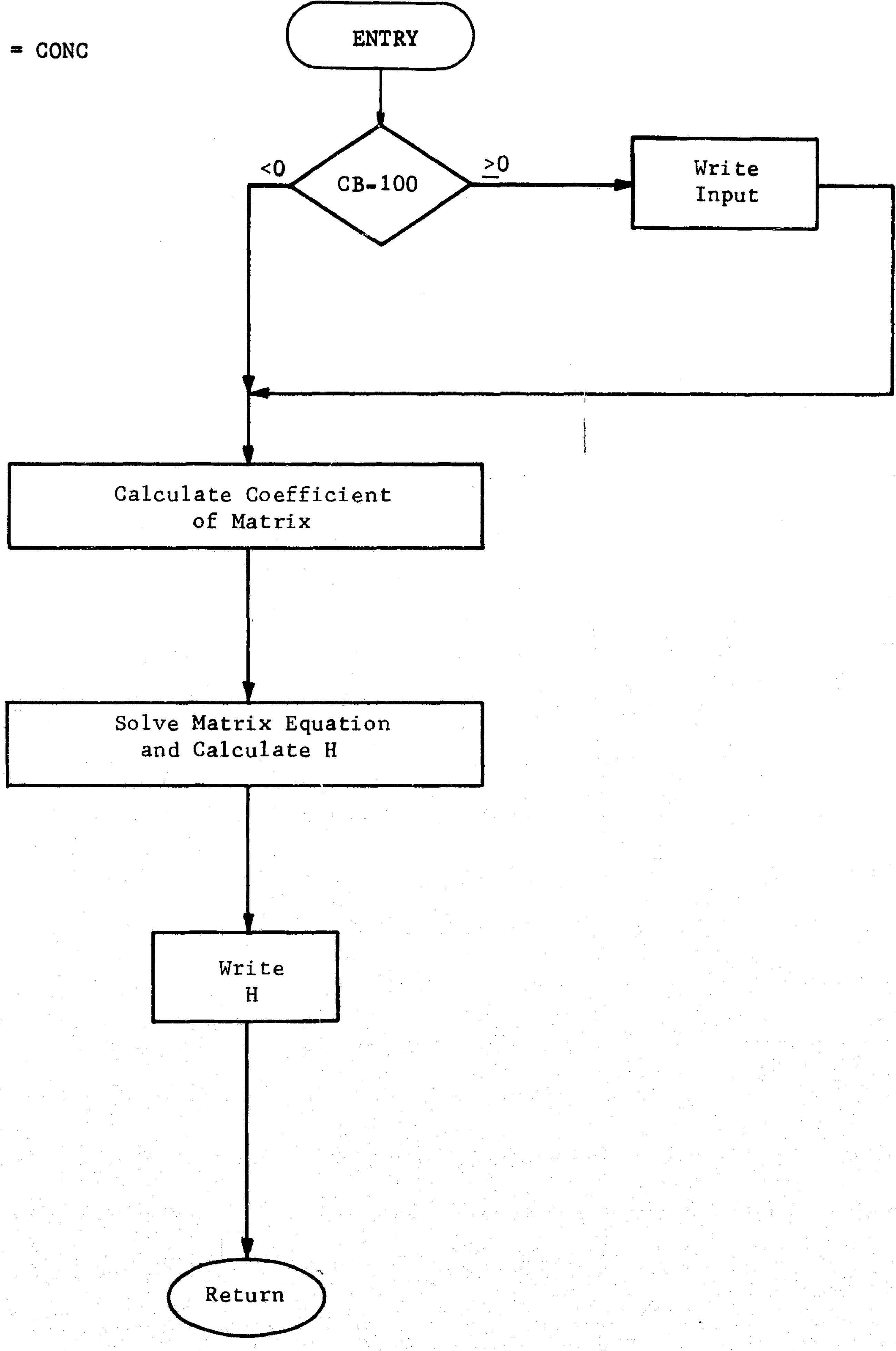
II, C, Flow Charts (cont.)

ENTRY = DISTR

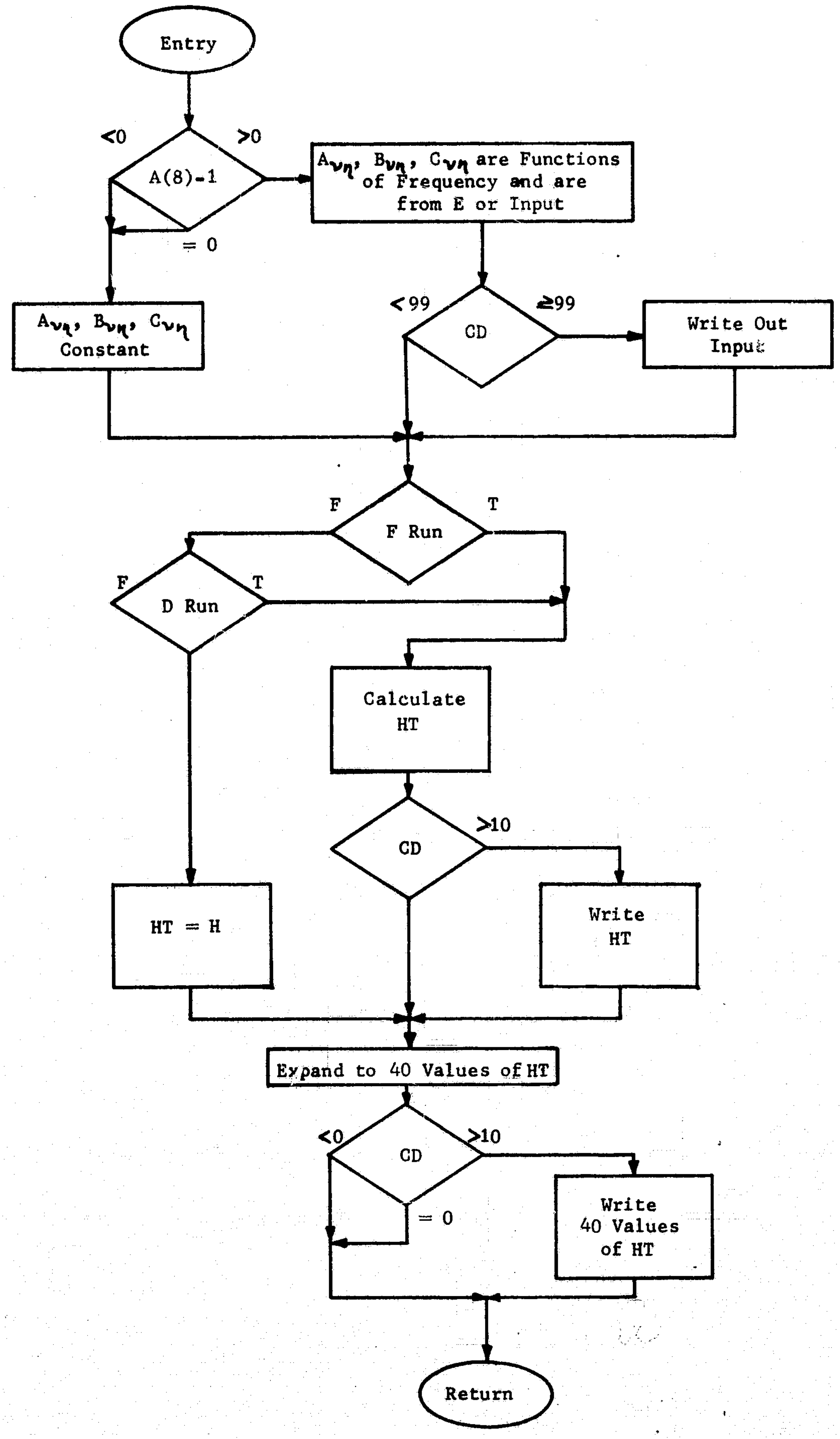


II, C, Flow Charts (cont.)

ENTRY = CONC

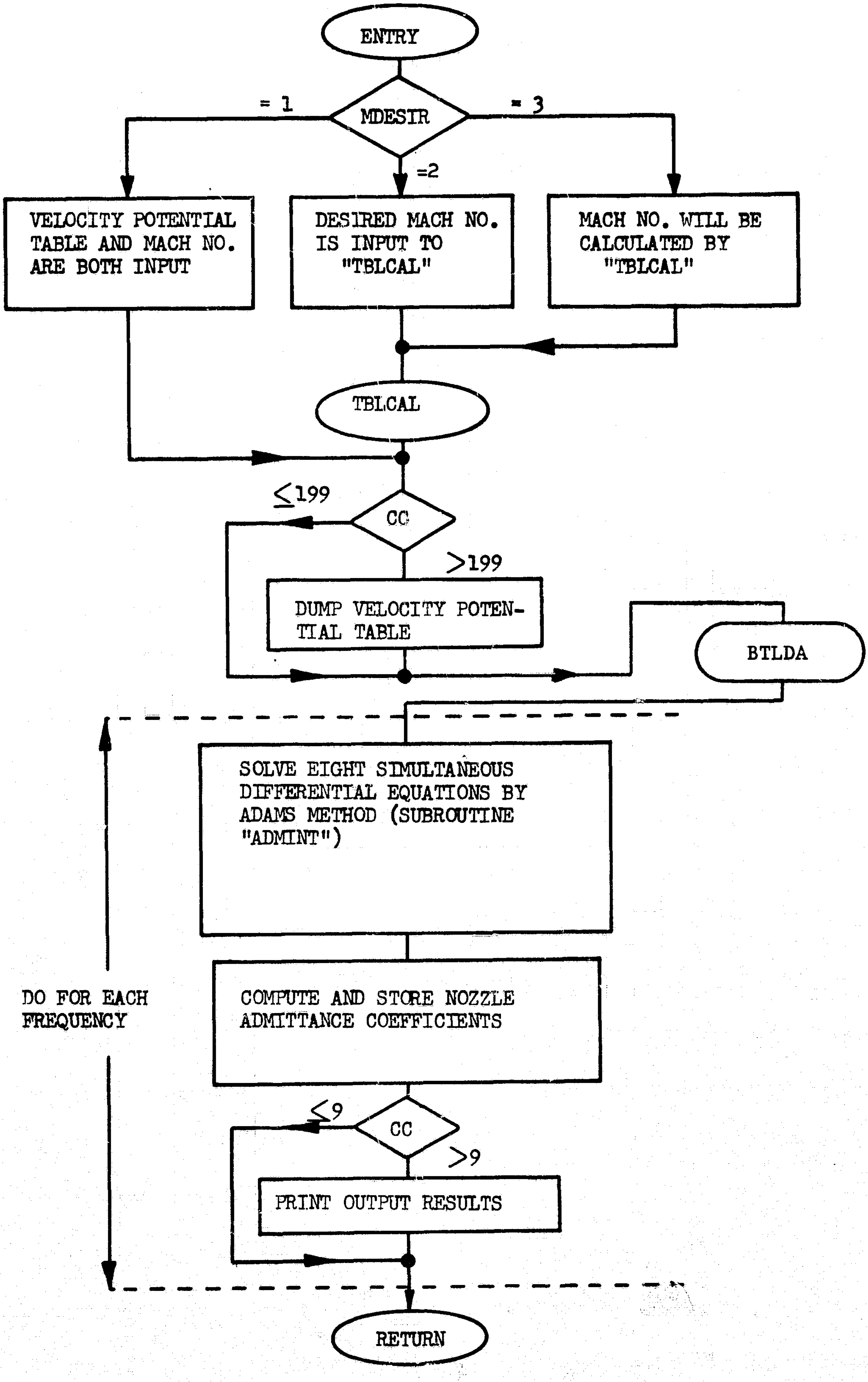


ENTRY = DDD


Entry
<0
>0
A(8)-1
= 0
$A_{\nu\eta}$, $B_{\nu\eta}$, $C_{\nu\eta}$ are Functions of Frequency and are from E or Input
$A_{\nu\eta}$, $B_{\nu\eta}$, $C_{\nu\eta}$ Constant
<99
≥99
CD
Write Out Input
F
T
F Run
F
T
D Run
Calculate HT
CD
>10
HT = H
Write HT
Expand to 40 Values of HT
<0
>10
CD
= 0
Write 40 Values of HT
Return

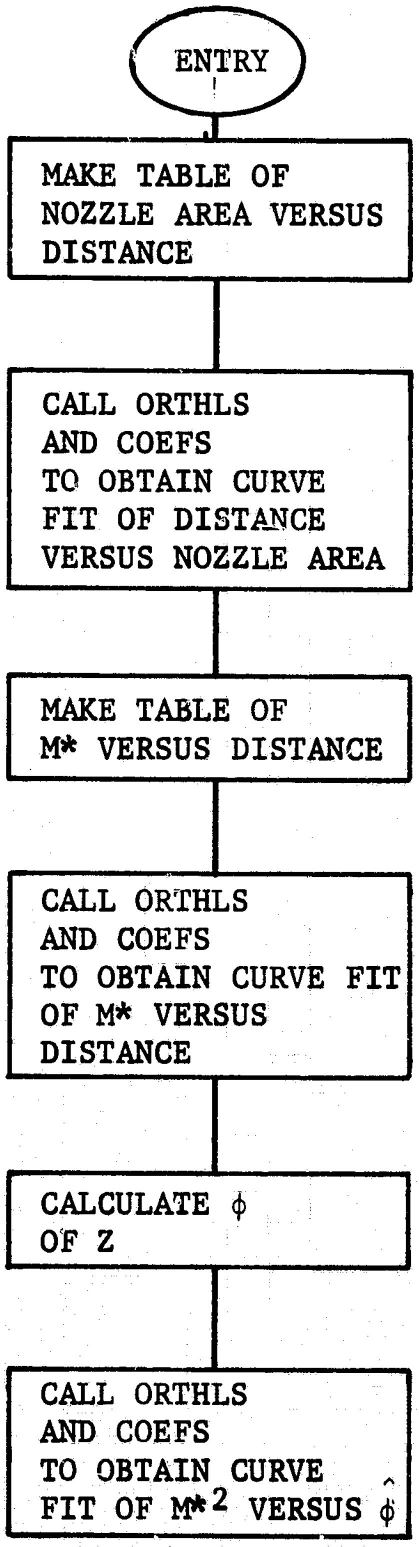
II, C, Flow Charts (cont.)

ENTRY = CCC

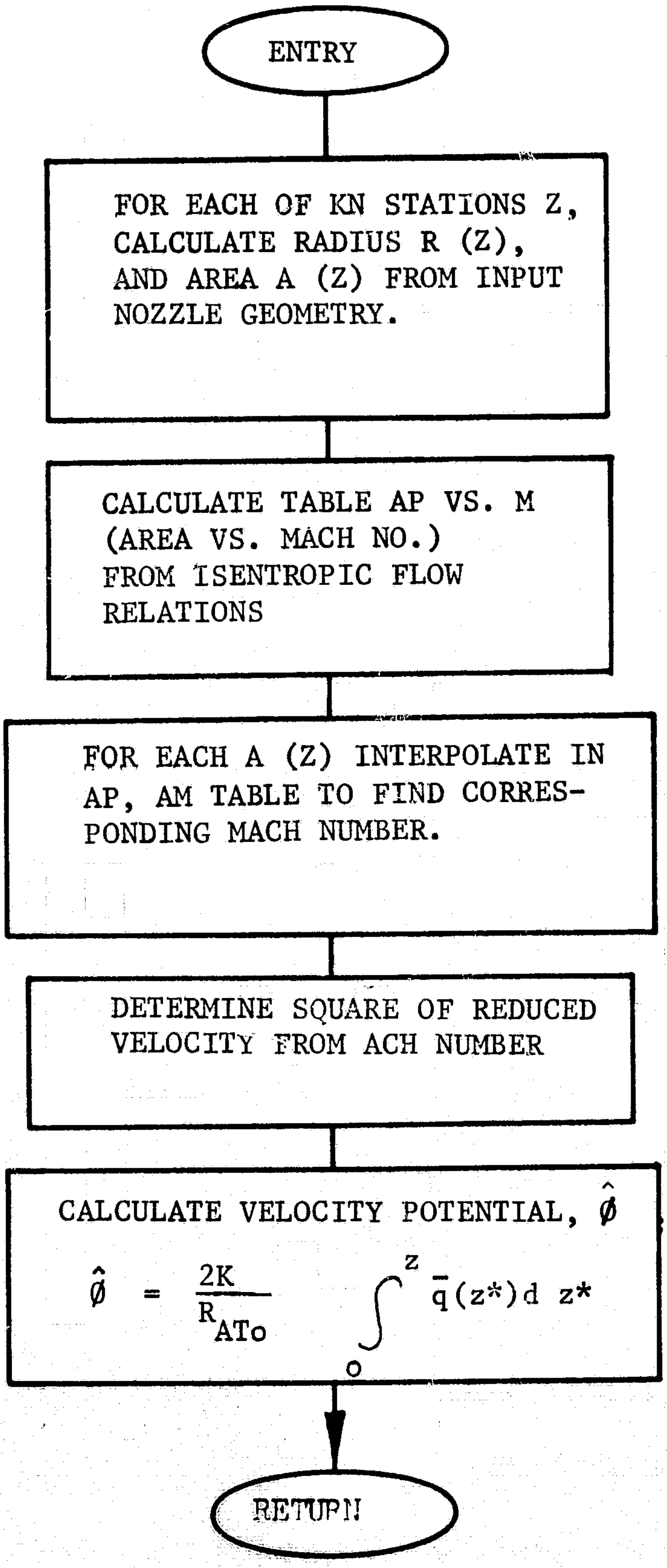


II, C, Flow Charts (cont.)

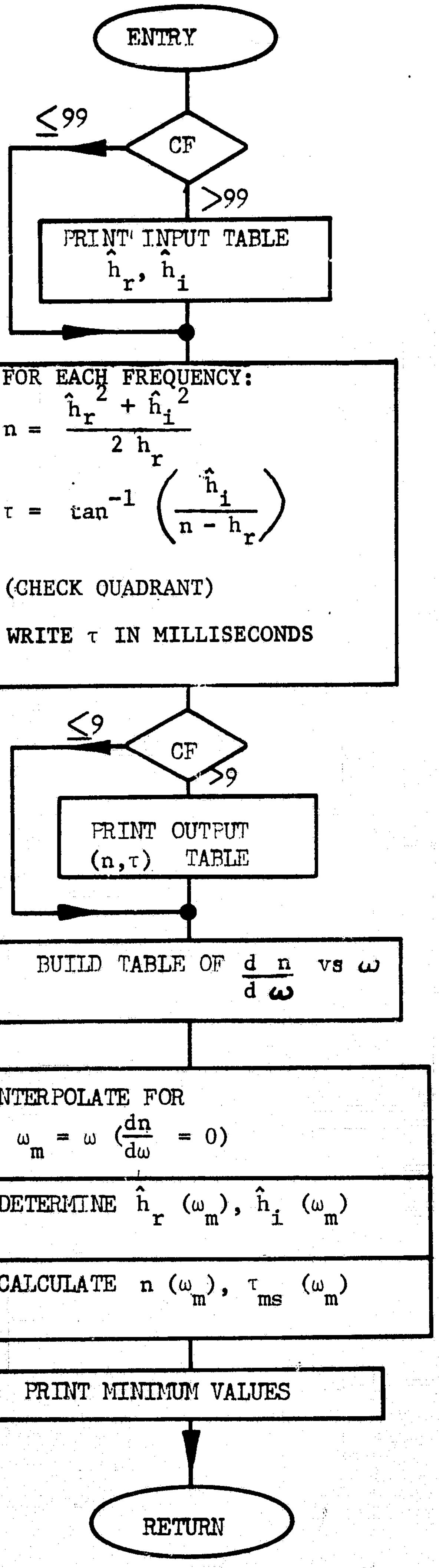
BTLDA



ENTRY = TBLCAL

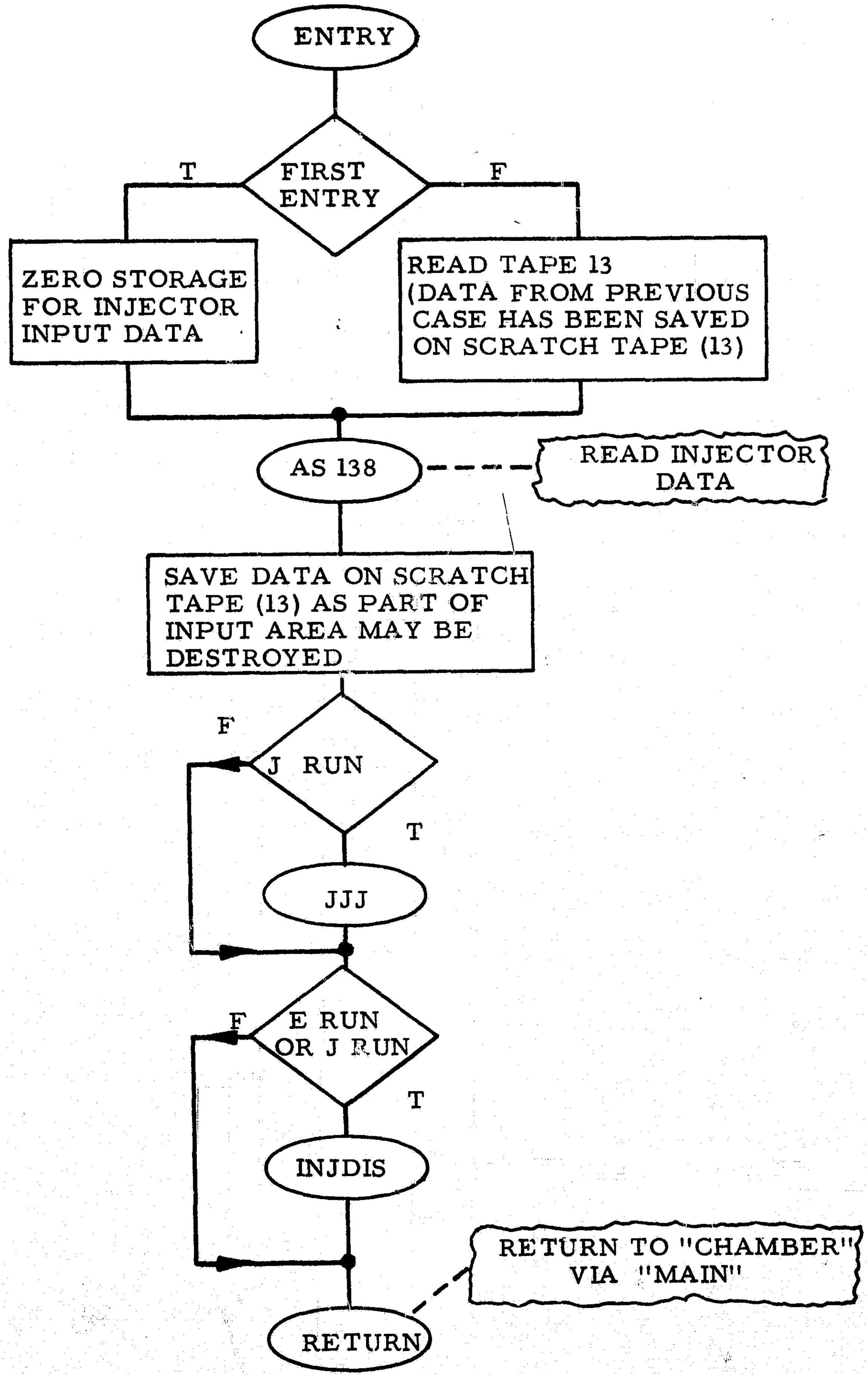


ENTRY = FFF

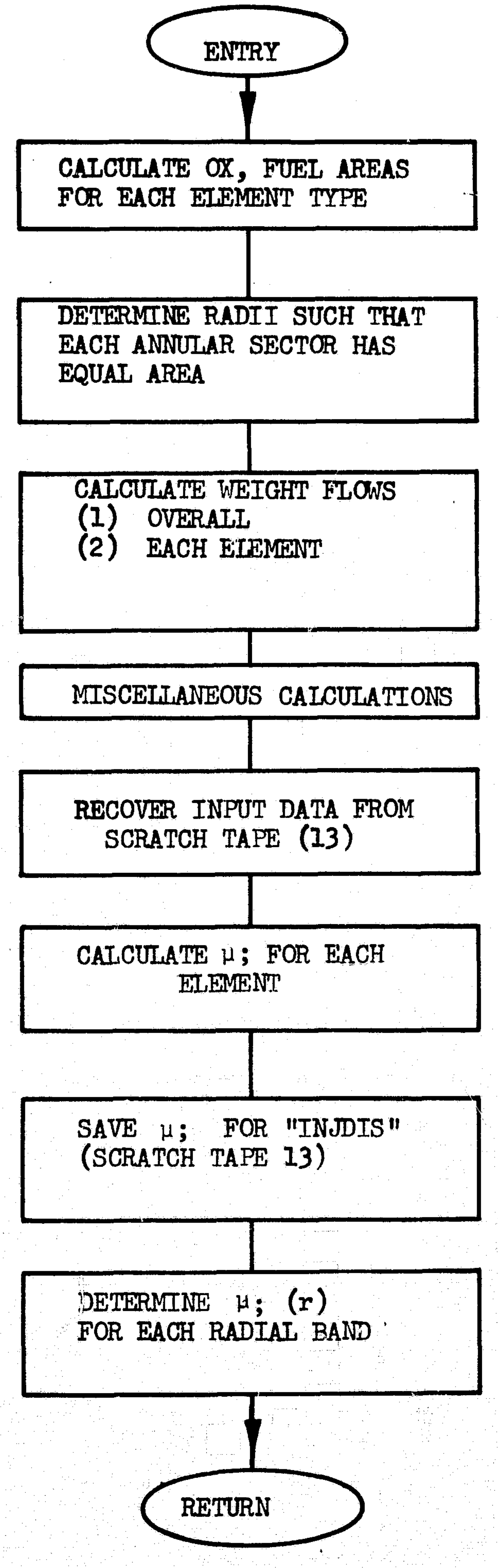

ENTRY
CF
≤99
>99
PRINT INPUT TABLE
$\hat{h}_r, \hat{h}_i$
FOR EACH FREQUENCY:
$n = \frac{\hat{h}_r^2 + \hat{h}_i^2}{2 h_r}$
$\tau = \tan^{-1}\left(\frac{\hat{h}_i}{n - h_r}\right)$
(CHECK QUADRANT)
WRITE τ IN MILLISECONDS
CF
≤9
>9
PRINT OUTPUT (n,τ) TABLE
BUILD TABLE OF $\frac{dn}{d\omega}$ vs ω
INTERPOLATE FOR
$\omega_m = \omega\ (\frac{dn}{d\omega} = 0)$
DETERMINE $\hat{h}_r\ (\omega_m), \hat{h}_i\ (\omega_m)$
CALCULATE $n\ (\omega_m), \tau_{ms}\ (\omega_m)$
PRINT MINIMUM VALUES
RETURN

II, C, Flow Charts (cont.)

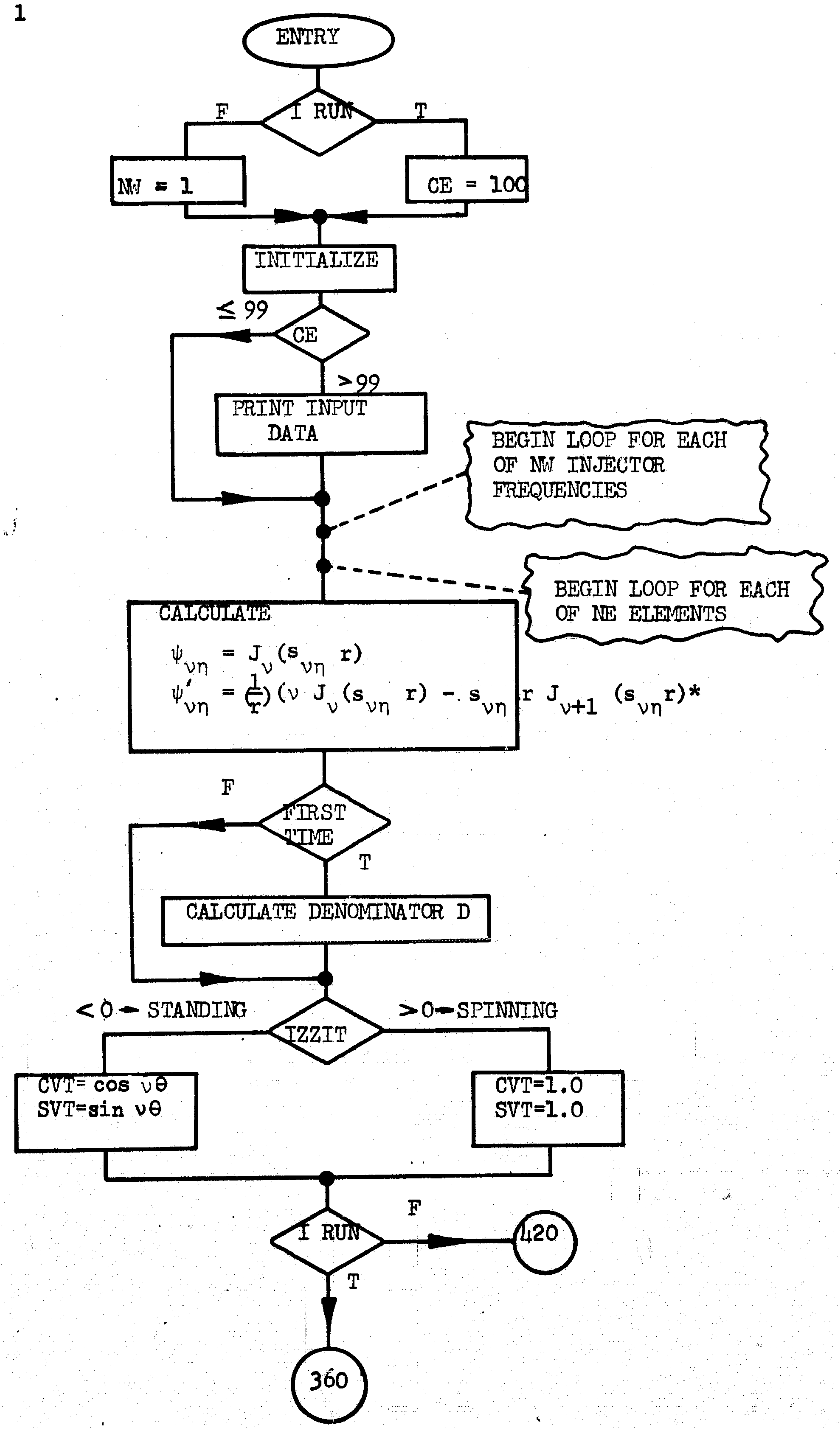
ENTRY = INJCTR



ENTRY = JJJ

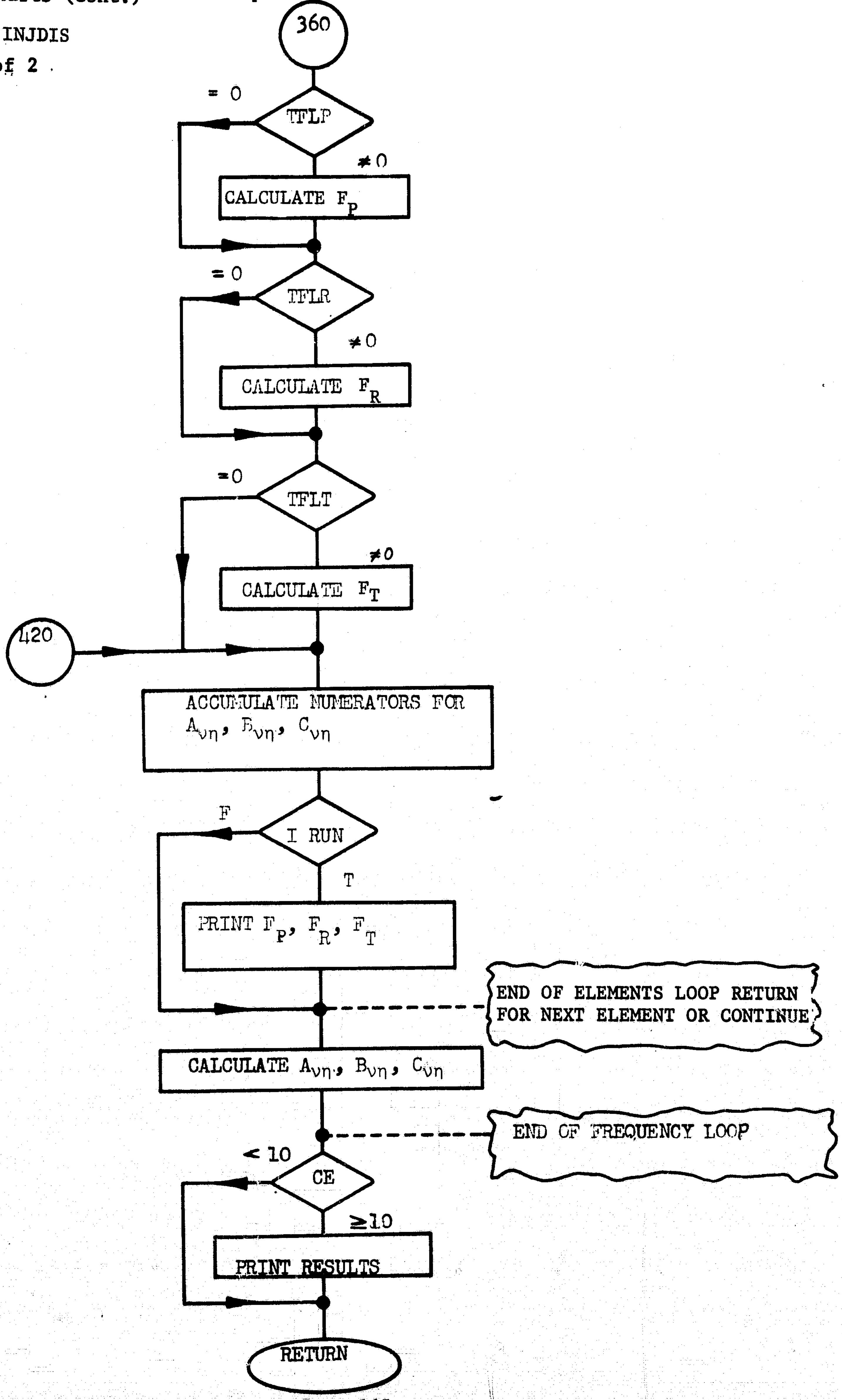

ENTRY
CALCULATE OX, FUEL AREAS FOR EACH ELEMENT TYPE
DETERMINE RADII SUCH THAT EACH ANNULAR SECTOR HAS EQUAL AREA
CALCULATE WEIGHT FLOWS
(1) OVERALL
(2) EACH ELEMENT
MISCELLANEOUS CALCULATIONS
RECOVER INPUT DATA FROM SCRATCH TAPE (13)
CALCULATE μ; FOR EACH ELEMENT
SAVE μ; FOR "INJDIS" (SCRATCH TAPE 13)
DETERMINE μ; (r) FOR EACH RADIAL BAND
RETURN

ENTRY = INJDIS
Page 1 of 1



ENTRY = INJDIS

Page 2 of 2



II, Programming (cont.)

D. PROGRAM LISTING

```
@ ELT MAIN,1,690702, 33249

      COMMON  /PROLOG/  LOGIK(50), SL1, SL2, EORJ
      LOGICAL  LOGIK,  SL1,  SL2,  EORJ
C
   10 CALL  CHAMBR
C
C         RETURNS FROM MAJOR PROGRAM IF AND ONLY IF AN INJECTOR PROGRAM
C          IS TO BE CALLED.,. INJCTR DETERMINES WHICH, WRITES SCRATCH
C          TAPE, AND CALLS PROPER ROUTINE.
C
      CALL  INJCTR
C
      GO TO 10
         END
```

II, D, Program Listing (cont.)

```
@ ELT SUB01,1,690702, 33250

      BLOCK DATA                                                        BLOK  30
C                                                                       BLOK  40
      COMMON  /PROLOG/  LOGIK(50), SL1, SL2, EORJ                       BLOK  50
      LOGICAL  LOGIK,  SL1,  SL2,  EORJ                                 BLOK  60
C                                                                       BLOK  70
      DATA  SL1, SL2  /  .FALSE., .FALSE.  /                            BLOK  80
      END                                                               BLOK  90
```

```
@ ELT SUB02,1,690702, 33250

      SUBROUTINE OUTAS8(A,K)
      DIMENSION A(12)
      WRITE(6,10)(A(I),I=1,K)
   10 FORMAT(1X,12A6)
      RETURN
      END
```

II, D, Program Listing (cont.)

```
@ ELT SUB03,1,690702, 33251

              SUBROUTINE INAS58(A)                                 IN    10
              DIMENSION A(12)                                      IN    20
              READ(5,10)A                                          IN    30
           10 FORMAT(12A6)                                         IN    40
              RETURN                                               IN    50
              END                                                  IN    60
```

II, D, Program Listing (cont.)

```
@ ELT SUB04,1,690717, 38451

      SUBROUTINE INT4(X,Y,XI,YO)                                        INT4  10
      DIMENSION X(9),Y(9),XC(4),YC(4)                                   INT4  20
      EQUIVALENCE (XC(1),X1),(XC(2),X2),(XC(3),X3),(XC(4),X4),(YC(1),Y1)INT4  30
     1,(YC(2),Y2),(YC(3),Y3),(YC(4),Y4)                                 INT4  40
10    NA=1
      J=2                                                               INT4  60
      B=XI                                                              INT4  70
20    IF(X(J).NE.0..OR.Y(J).NE.0.) GO TO (90,160),NA
      IF(J.GT.2) GO TO 70
      YE=0.
      GO TO 180                                                             *NEW
70    NB=1                                                                  **-1
      J=J-1                                                             INT4 150
   80 X1=X(J)                                                           INT4 160
      X2=X(J-1)                                                         INT4 170
      X3=X(J-2)                                                         INT4 180
      Y1=Y(J)                                                           INT4 190
      Y2=Y(J-1)                                                         INT4 200
      Y3=Y(J-2)                                                         INT4 210
      GO TO (150,170),NB
   90 IF(X(J)-B)120,100,100                                             INT4 230
100   IF(J.LE.2) GO TO 130
110   NA=2
  120 J=J+1                                                             INT4 260
      GO TO 20                                                          INT4 270
  130 DO 140 J=1,3                                                      INT4 280
      XC(J)=X(J)                                                        INT4 290
  140 YC(J)=Y(J)                                                        INT4 300
  150 D=X2-X1                                                           INT4 310
      A1=B-X1                                                           INT4 320
      A2=B-X2                                                           INT4 330
      YE=A1*A2/2.0/D*((Y3-Y2)/(X3-X2)-(Y2-Y1)/D)-A2/D*Y1+A1/D*Y2        INT4 340
      GO TO 180                                                         INT4 350
160   NB=2
      GO TO 80                                                          INT4 370
  170 X4=X(J-3)                                                         INT4 380
      Y4=Y(J-3)                                                         INT4 390
      D=X3-X2                                                           INT4 400
      A1=B-X2                                                           INT4 410
      A2=B-X3                                                           INT4 420
      XM12=(Y2-Y1)/(X2-X1)                                              INT4 430
      XM23=(Y3-Y2)/D                                                    INT4 440
      XM34=(Y4-Y3)/(X4-X3)                                              INT4 450
      YE=A1*A2**2/2.0/D**2*(XM12-XM23)+A2*A1**2/2.0/D**2*(XM34-XM23)-A2*INT4 460
     1Y2/D+A1*Y3/D                                                      INT4 470
  180 YO=YE                                                             INT4 480
      RETURN                                                            INT4 490
      END                                                               INT4 500
```

II, D, Program Listing (cont.)

```
@ ELT SUB05,1,690717, 38453

      SUBROUTINE INT4D(X,Y,XI,YO,DY)                                    INT4D 10
      DIMENSION X(9),Y(9),XC(4),YC(4)                                   INT4D 20
      EQUIVALENCE (XC(1),X1),(XC(2),X2),(XC(3),X3),(XC(4),X4),(YC(1),Y1)INT4D 30
     1,(YC(2),Y2),(YC(3),Y3),(YC(4),Y4)                                 INT4D 40
10    NA=1
      J=2                                                               INT4D 60
      B=XI                                                              INT4D 70
20    IF(X(J).NE.0..OR.Y(J).NE.0.) GO TO (90,160),NA
      IF(J.GT.2) GO TO 70
      YE=0.
      GO TO 180                                                             *NEW
70    NB=1                                                                  **-1
      J=J-1                                                             INT4D150
   80 X1=X(J)                                                           INT4D160
      X2=X(J-1)                                                         INT4D170
      X3=X(J-2)                                                         INT4D180
      Y1=Y(J)                                                           INT4D190
      Y2=Y(J-1)                                                         INT4D200
      Y3=Y(J-2)                                                         INT4D210
      GO TO (150,170),NB
   90 IF(X(J)-B)120,100,100                                             INT4D230
100   IF(J.LE.2) GO TO 130
110   NA=2
  120 J=J+1                                                             INT4D260
      GO TO 20                                                          INT4D270
  130 DO 140 J=1,3                                                      INT4D280
      XC(J)=X(J)                                                        INT4D290
  140 YC(J)=Y(J)                                                        INT4D300
  150 D=X2-X1                                                           INT4D310
      A1=B-X1                                                           INT4D320
      A2=B-X2                                                           INT4D330
      XM23=(Y3-Y2)/(X3-X2)                                              INT4D340
      XM12=(Y2-Y1)/(X2-X1)                                              INT4D350
      XM2B=(XM23-XM12)/2.0/D                                            INT4D360
      YO=A1*A2*XM2B-A2*Y1/D+A1*Y2/D                                     INT4D370
      DY=XM2B*(A1+A2)+XM12                                              INT4D380
      GO TO 180                                                         INT4D390
160   NB=2
      GO TO 80                                                          INT4D410
  170 X4=X(J-3)                                                         INT4D420
      Y4=Y(J-3)                                                         INT4D430
      D=X3-X2                                                           INT4D440
      A1=B-X2                                                           INT4D450
      A2=B-X3                                                           INT4D460
      XM12=(Y2-Y1)/(X2-X1)                                              INT4D470
      XM23=(Y3-Y2)/D                                                    INT4D480
      XM34=(Y4-Y3)/(X4-X3)                                              INT4D490
      AM2=A2*(XM12-XM23)                                                INT4D500
      AM1=A1*(XM34-XM23)                                                INT4D510
      YO=(A1*A2/2.0/D*(AM2+AM1)-A2*Y2+A1*Y3)/D                          INT4D520
      DY=(AM2*(2.0*A1+A2)+AM1*(2.0*A2+A1))/2.0/D**2+XM23                INT4D530
  180 RETURN                                                            INT4D540
      END                                                               INT4D550
```

II, D, Program Listing (cont.)

```
@ ELT SUB06,1,690807, 35835

            SUBROUTINE PAGE(LINES)                                          PAGE  10
      C                                                                     PAGE  20
      C     HEAD MOVED TO /PROLOG/ AND PAGE MODIFIED TO PRINT HEAD 25JUL 67 PAGE  30
      C                                                                     PAGE  40
            COMMON  /PROLOG/  LOGIK(38), HEAD(12), SL1, SL2, EORJ           PAGE  50
            DIMENSION  TODAY (2)                                            PAGE  60
            DATA  KPG / 0 /                                                 PAGE  70
      C                                                                     PAGE  80
            IF(LINES-60)20,10,10                                            PAGE  90
         10 L=2                                                             PAGE 100
            GO TO 60                                                        PAGE 110
         20 K=L+LINES                                                       PAGE 120
            IF(K-60)30,30,50                                                PAGE 130
         30 L=K                                                             PAGE 140
         40 RETURN                                                          PAGE 150
      C                                                                     PAGE 160
         50 L=LINES+2                                                       PAGE 170
         60 IF  (KPG.EQ.0) GO TO 90                                             *NEW
         70 KPG = KPG + 1                                                   PAGE 190**-1
            WRITE (6,80)  TODAY, HEAD, KPG                                  PAGE 200
         80 FORMAT ( 1H1 3X 6HDATE   2A6, 12X 12A6, 11X 5HPAGE  I5 )        PAGE 210
            GO TO 40                                                        PAGE 220
      90    TODAY(1)=6H                                                         *NEW
            TODAY(2)=6H                                                         *NEW
            CALL DATE  (12,TODAY)                                               *NEW
            GO TO 70                                                            *NEW
            END                                                             PAGE 230
```

```
@ ELT SUB07,1,700722, 38212

      SUBROUTINE  CHAMBR
C
C           20 SEP 67  MODIFIED FOR TABULAR INJECTOR COEFFICIENTS
C
      LOGICAL  LOGIK, ARUN,BRUN,CRUN,DRUN,ERUN,FRUN,GRUN,HRUN,IRUN,JRUN
      LOGICAL  SL1, SL2, EORJ
      REAL  MACH
      COMMON  / /
     1   GAM, NWI, WIT(30), AVN(30), BVN(30), CVNR(30), CVNI(30), EE,DSC
      COMMON  /PROLOG/  LOGIK(38), HEAD(12), SL1, SL2, EORJ
      COMMON/ABCDF/ DINP,STOW
      DIMENSION  EXTRA(100),                       WC(75)
      DIMENSION DINP(4300),A(1),B(1),C(1),D(1)
      DIMENSION  X(133),Y(133),Q(134),STOW(222),STODAT(4607)
      DIMENSION ZZ(205)
      DIMENSION G(1),DISTL(20),DISTM(20)
      DIMENSION  AMIT(90)
      EQUIVALENCE ( EXTRA, DINP, STODAT )
      EQUIVALENCE
     1   (LOGIK(1) , ARUN),    (LOGIK(2) , BRUN),    (LOGIK(3) , CRUN),
     2   (LOGIK(4) , DRUN),    (LOGIK(5) , ERUN),    (LOGIK(6) , FRUN),
     3   (LOGIK(7) , GRUN),    (LOGIK(8) , HRUN),    (LOGIK(9) , IRUN),
     4   (LOGIK(10), JRUN)
      EQUIVALENCE (EXTRA(1),CA),(EXTRA(2),CB),(EXTRA(3),CC),(EXTRA(4),
     1CD),(EXTRA(5),CE),(EXTRA(6),CF),(EXTRA(7),CG),(EXTRA(8),CH)
      EQUIVALENCE (EXTRA(9),CI)
      EQUIVALENCE (DINP(1),A),(DINP(3001),B),(DINP(3401),D),
     1(DINP(3501),G),(DINP(3801),C),(EXTRA(21),WC) , (B(4),UE)
      EQUIVALENCE  (DINP(601),AMIT),(DINP(2813),Q),(DINP(3501),X),
     1     (DINP(3651),Y)
      EQUIVALENCE
     1  (STOW(1),ZZ), (STOW(213),YH)
     2   ,  (STOW(214),J), (STOW(215),NE), (STOW(216),YL), (STOW(217),K1)
     3   ,  (STOW(218),N), (STOW(219),KER), (STOW(220),XH)
     4   ,  (STOW(221),KQUAD), (STOW(222),XL)
     5    ,   (EXTRA(51), DISTL), (EXTRA(71), DISTM )
     6    ,  (EXTRA(10),CJ), (EXTRA(21),SNH ), (EXTRA(12), MACH )
     7    ,   (DINP(107),U1BAR)
C
C**************,*******************************************************
C
   10 FORMAT(1H0,60H      A     B     C     D     E     F     G     H
     1 I     J   // 3X,(10F6.0))
   20 FORMAT(///,9X,108H************************ THE FOLLOWING MAIN CONT
     1ROL DATA WILL BE USED IN THIS CASE ************************   ,///,
     245X,33HRATIO OF SPECIFIC HEAT (GAMMA) = ,F7.4,//,45X,22HDESIRED MA
     3CH NUMBER = ,E12.5,26H (=0 IF BEING CALCULATED) //45X        17HCH
     5AMBER RADIUS = ,F7.3, 9H (INCHES),//,45X,17HCHAMBER LENGTH = ,F7.3
     6, 9H (INCHES),//,45X, 17HSPEED OF SOUND = ,F10.3,  9H (FT/SEC),//,
     745X, 27HCHAMBER MODE DESCRIPTION = ,F8.5, 28H (=0 FOR LONGITUDINAL
     8 MODES),/,)
   30 FORMAT(///,33X,48H********** CHAMBER FREQUENCIES (WC) **********
     1    ,//,)
   40 FORMAT(5F20.5,/ )
   50 FORMAT(30X, 64H***** MACH DISTRIBUTION IN CHAMBER AS A FUNCTION OF
```

II, D, Program Listing (cont.)

```
     1 LENGTH *****,//,11X,7HCHAMBER,15X,4HMACH,14X,7HCHAMBER,15X,4HMACHCHAM 560
     2,14X,7HCHAMBER,15X,4HMACH,/,12X,6HLENGTH,11X,12HDISTRIBUTION,11X, CHAM 570
     36HLENGTH,11X,12HDISTRIBUTION,11X,6HLENGTH,11X,12HDISTRIBUTION,/,) CHAM 580
   60 FORMAT(6(10X,F10.5))                                              CHAM 590
   65 FORMAT ( 11X,42HDISTANCE FROM INJECTOR TO COMBUSTION FRONT ,3X,
     1F10.5,6HINCHES)
C*************,********************************************************* CHAM 600
C*************,********************************************************* CHAM 610
C                                                                       CHAM 620
      IF ( SL2 )                GO TO 170                               CHAM 630
C                                                                       CHAM 640
      SL2 = .TRUE.                                                      CHAM 650
      DO 70 I = 1, 4300                                                 CHAM 660
   70    DINP (I) = 0.0                                                 CHAM 670
      GO TO 110                                                         CHAM 680
C                                                                       CHAM 690
C*********************************************************************** CHAM 700
C                                                                       CHAM 710
   80 WRITE (6,100)    NE                                               CHAM 720
   90 CALL EXIT                                                         CHAM 730
  100 FORMAT (1H010X17HINPUT ERROR, NE = I3, 19H, HENCE TERMINATION )   CHAM 740
C                                                                       CHAM 750
C*********************************************************************** CHAM 760
C*********************************************************************** CHAM 770
C                                                                       CHAM 780
  110 KER = 0                                                           CHAM 790
      CALL  DVCHK  (KCHK)                                               CHAM 800
C                                                                       CHAM 810
      DO 120 I = 1, 10                                                  CHAM 820
  120    DINP(I) = 0.0                                                  CHAM 830
      CALL  AS130  ( DINP(1), HEAD(1), NE )                             CHAM 840
      IF ( NE .NE. 1 )    GO TO 80                                      CHAM 850
      ARUN = CA .NE. 0.0                                                CHAM 860
      BRUN = CB .NE. 0.0                                                CHAM 870
      CRUN = CC .NE. 0.0                                                CHAM 880
      DRUN = CD .NE. 0.0                                                CHAM 890
      ERUN = CE .NE. 0.0                                                CHAM 900
      FRUN = CF .NE. 0.0                                                CHAM 910
      IRUN = CI .NE. 0.0                                                CHAM 920
      JRUN = CJ .NE. 0.0                                                CHAM 930
     0GRUN =         DINP(22) .LE. 0.0                                  CHAM 940
     1            .AND. ( ARUN .OR. BRUN .OR. CRUN .OR. IRUN )          CHAM 950
C                                                                       CHAM 960
C     PRINT NEW MAIN CONTROL DATA                                       CHAM 970
C*********************************************************************** CHAM 980
      CALL  PAGE  ( 60 )                                                CHAM 990
      WRITE (6,10)  (DINP(I), I=1,10)                                   CHAM1000
      WRITE (6,20)  DINP(11), MACH, DINP(14), DINP(15), DINP(16), SNH   CHAM1010
C*********************************************************************** CHAM1020
C     ARE FREQUENCIES TO BE CALCULATED... IF SO, CALCULATE AND PRINT.   CHAM1030
C*********************************************************************** CHAM1040
      IF ( .NOT. GRUN )        GO TO 130                                CHAM1050
      CALL  GENMEG  ( WC(1) )                                           CHAM1060
      IF ( CRUN )       CC = CC + 11.0                                  CHAM1080
  130 JOMEGA=ABS(EXTRA(22))+22.0001                                     CHAM1090
      WRITE (6,30)                                                      CHAM1100
      WRITE (6,40)(EXTRA(I),I=23,JOMEGA)                                CHAM1110
      IF((.NOT. ARUN) .AND. (.NOT. BRUN)) GO TO 132                            *NEW
      IF ( BRUN ) GO TO 131
```

II, D, Program Listing (cont.)

```
      WRITE (6,50)                                                      CHAM1120
      WRITE (6,60)( DISTL(I), DISTM(I), DISTL(I+7),         DISTM(I+7),CHAM1130
     1, DISTL(I+14), DISTM(I+14), I=1,6 ), DISTL(7),         DISTM(7), CHAM1140
     2DISTL(14), DISTM(14)                                              CHAM1150
      GO TO 132
  131 WRITE(6,65) DINP(52)
C*********************************************************************** CHAM1160
C                                                                       CHAM1170
  132 CONTINUE
      EORJ = ERUN .OR. JRUN .OR. IRUN                                   CHAM1180
      IF ( CRUN .OR. EORJ )   CALL  PAGE  ( 60 )                        CHAM1190
  140 IF ( .NOT. EORJ )        GO TO 200                                CHAM1200
      WRITE (14)       STODAT                                           CHAM1210
      BACKSPACE  14                                                     CHAM1220
      EE  = CE                                                          CHAM1230
      DSC = CI                                                          CHAM1240
      GAM = DINP(11)                                                    CHAM1250
      IF ( .NOT. IRUN )       GO TO 160                                 CHAM1260
      NWI = ( JOMEGA-22 )/2 +1                                          CHAM1270
      DO 150 I = 1, NWI                                                 CHAM1280
  150    WIT(I) = DINP( 2*I+21 )                                        CHAM1290
      WIT(NWI) = DINP( JOMEGA )                                         CHAM1300
C                                                                       CHAM1310
  160 RETURN                                                            CHAM1320
C     ******                                                            CHAM1330
C           PROGRAM RETURNS IF AND ONLY IF INJECTOR PROGRAM IS TO BE RUN CHAM1340
C                                                                       CHAM1350
C*********************************************************************** CHAM1360
C*********************************************************************** CHAM1370
C                                                                       CHAM1380
C                                                                       CHAM1390
  170 READ (14)  STODAT                                                 CHAM1400
      BACKSPACE  14                                                     CHAM1410
      IF ( .NOT. ERUN )       GO TO 200                                 CHAM1420
C                                                                       CHAM1430
      DINP(3408) = NWI                                                  CHAM1440
      DO 180 I = 1, NWI                                                 CHAM1450
         DINP(I+4522) = WIT(I)                                          CHAM1460
         DINP(I+4539) = AVN (I)                                         CHAM1470
         DINP(I+4556) = BVN (I)                                         CHAM1480
         DINP(I+4573) = CVNR(I)                                         CHAM1490
  180    DINP(I+4590) = CVNI(I)                                         CHAM1500
      DO 190 I = NWI, 84, 17                                            CHAM1510
  190    DINP(I+4523) = 0.0                                             CHAM1520
C                                                                       CHAM1530
C*********************************************************************** CHAM1540
C                                                                       CHAM1550
  200 WC(2) = ABS( WC(2) )                                              CHAM1560
C*********************************************************************** CHAM1570
C     FIX NUMBER OF FREQUENCIES AND TEST DESIRE FOR INTERNAL C FLAG     CHAM1580
C*********************************************************************** CHAM1590
  210 NOMEG=WC(2)+.0001                                                 CHAM1600
      IF(EXTRA(12))230,220,230                                          CHAM1610
  220 DINP(3802)=3.0                                                    CHAM1620
      GO TO 240                                                         CHAM1630
  230 U1BAR = MACH                                                      CHAM1640
      UE = MACH                                                         CHAM1650
  240 IF ( .NOT. CRUN )            GO TO 390                            CHAM1660
C*********************************************************************** CHAM1670
```

```
C     SET UP DATA FOR PROGRAM  C.                                       CHAM1680
C*********************************************************************** CHAM1690
  250 DINP(3801)=EXTRA(11)                                              CHAM1700
      NMHAF = NOMEG/2 + 1                                               CHAM1710
      ZZ(1)=EXTRA(14)                                                   CHAM1720
      ZZ(2)=EXTRA(15)                                                   CHAM1730
      ZZ(3)=EXTRA(16)                                                   CHAM1740
       DINP(3805) = EXTRA(14)                                           CHAM1750
      DINP(3803)=NMHAF                                                  CHAM1760
      IF ( DINP(3809) .EQ. 0.0 )         DINP(3809) = 101.0             CHAM1770
      K = 0                                                             CHAM1780
      RATO=DINP(3804)
      RACO=DINP(3805)
      RCCO=DINP(3806)
      RCTO=DINP(3807)
      RATI=DINP(91)
      RCTI=DINP(92)
      RACI=DINP( 93)
      RCCI=DINP(94)
      IF(RCTI.EQ.0.) RCTI=1.
      GRAD=RATO*SQRT(2./(1.+EXTRA(11))*(RATO/RCTO-RATI/RCTI)/(RATO*
     1RATO-RATI*RATI))
      ZAVE=SQRT((1.+EXTRA(11))/2.)*RATO/GRAD
      ZZ(4)=ZAVE
      IF(EXTRA(21))270,260,270                                          CHAM1800
  260 SNOZ = 0.0                                                        CHAM1810
C*********************************************************************** CHAM1820
C     USING HALF OF CHAMBER FREQUENCIES, CALCULATE NOZZLE FREQUENCIES   CHAM1830
C       FOR USE IN PROG C.                                              CHAM1840
C*********************************************************************** CHAM1850
      RORL = EXTRA(15)                                                  CHAM1860
      GO TO 280                                                         CHAM1870
  270 RORL = EXTRA(14)                                                  CHAM1880
      SNOZ = SNH/GRAD                                                   CHAM1900
  280 DO 290 I=1,NMHAF                                                  CHAM1910
      KN=(2*I)+1                                                        CHAM1920
      NN=4210+K                                                         CHAM1930
      DINP(NN) = ZAVE*WC(KN)/RORL                                       CHAM1940
      DINP(NN+1) = SNOZ                                                 CHAM1950
      DINP(NN+2) = MACH                                                 CHAM1960
      K=K+3                                                             CHAM1970
  290 CONTINUE                                                          CHAM1980
      DINP(NN)=ZAVE*(WC(NOMEG+2)/EXTRA(14))                             CHAM1990
  300 CALL CCC(C(1),ZZ(1),WC(1),CC,KER)                                 CHAM2000
C     SET UP NOZZ ADM FOR A,B                                           CHAM2010
      IF(KER) 310,330,310                                               CHAM2020
  310 WRITE (6,320)                                                     CHAM2030
  320 FORMAT (1H0 30X39H ERROR PROGRAM C,  ALL CASES TERMINATED )       CHAM2040
      CALL CORE(C(1),420,CC)                                            CHAM2050
      GO TO 90                                                          CHAM2060
  330 GO TO 360
  360 CONTINUE
C*********************************************************************** CHAM2270
C     MOVE OUTPUT FROM PROG C INTO INPUT BLOCK FOR PROG B.              CHAM2280
C*********************************************************************** CHAM2290
  370 DO 380 I=18,200                                                   CHAM2300
  380 B(I)=ZZ(I)                                                        CHAM2310
      B(4)=ZZ(205)                                                      CHAM2320
C                                                                       CHAM2330
```

II, D, Program Listing (cont.)

```
  390 CONTINUE                                                          CHAM2340
      CONTINUE
C     NOZZLE ADMITTANCE IS INPUT TO PROGS A, B.                         CHAM2360
C********************************************************************** CHAM2370
C                                                                       CHAM2380
C********************************************************************** CHAM2490
C     SET UP MAIN CONTROL DATA FOR USE IN B OR A
C********************************************************************** CHAM2510
  450 IF ((.NOT. BRUN) .AND. (.NOT. ARUN)) GO TO 530
      B(1)=WC(1)                                                        CHAM2530
      B(2)=EXTRA(15)/EXTRA(14)                                          CHAM2540
      B(3)=EXTRA(11)                                                    CHAM2550
      B(5)=EXTRA(16)                                                    CHAM2560
      B(6) = DINP(17)                                                   CHAM2570
      B(7) = DINP(18)                                                   CHAM2580
      B(8) = DINP(19)                                                   CHAM2590
      B(19)=WC(2)                                                       CHAM2600
      DO 460 I=1,NOMEG                                                  CHAM2610
      B(I+169)=WC(I+2)                                                  CHAM2620
  460 CONTINUE                                                          CHAM2630
      B(200) = DISTL(1) / DINP(14)                                      CHAM2640
      B(220) = DISTM(1)                                                 CHAM2650
      B(17) = 1.0                                                       CHAM2660
      DO 480 I = 2, 20                                                  CHAM2670
      IF(DISTM(I))490,490,470                                           CHAM2680
  470 B(I+199)=DISTL(I)/EXTRA(14)                                       CHAM2690
      B(I+219)=DISTM(I)                                                 CHAM2700
      B(17)=B(17)+1.0                                                   CHAM2710
  480 CONTINUE                                                          CHAM2720
  490 CONTINUE                                                          CHAM2730
      IF( .NOT. ARUN) GO TO 492
      CALL DISTR(B,X,CA,KER)
      GO TO 499
  492 CALL CONC (B,X,CB,KER)
  499 DO 500 I=9,100
  500    D(I) = X(I)                                                    CHAM2760
      IF(KER)530,510,530                                                CHAM2770
  510 WRITE (6,520)                                                     CHAM2780
  520 FORMAT (1H0,40X,21H ERROR PROGRAM A OR B   )
      CALL CORE(B(1),400,CB)                                            CHAM2800
      GO TO 110                                                         CHAM2810
C********************************************************************** CHAM2820
  530 CONTINUE
      IF((.NOT. ARUN) .AND. (.NOT. BRUN) .AND. (.NOT. DRUN)) GO TO 610
      IF(.NOT. DRUN ) GO TO 575
  540 DO 550 I=1,8                                                      CHAM2840
      X(I)=D(I)                                                         CHAM2850
  550 CONTINUE                                                          CHAM2860
  560 X(5)=EXTRA(11)                                                    CHAM2870
      DO 570 I = 9, 100                                                 CHAM2880
  570    X(I) = D(I)                                                    CHAM2890
  575 CALL DDD ( X(1),Y(1), CD,KER, ERR )
      IF ( ERR )                   110,580,580                          CHAM2910
  580 IF ( KER )                   610,590,610                          CHAM2920
  590 WRITE (6,600)                                                     CHAM2930
  600 FORMAT (1H0,40X,18H ERROR PROGRAM D   )                           CHAM2940
      CALL CORE(X(1),100,CD)                                            CHAM2950
      GO TO 110                                                         CHAM2960
  610 CONTINUE                                                          CHAM2970
```

II, D, Program Listing (cont.)

```
  620 IF (( .NOT. ARUN) .AND.(.NOT. BRUN) .AND.(.NOT. FRUN)) GO TO 670
C********************************************************************** CHAM2990
C      SET UP DATA FOR PROG F    (N, TAU )                               CHAM3000
C********************************************************************** CHAM3010
  630 Y(1)=EXTRA(14)                                                      CHAM3020
      IF(SNH) 632,631,632
  631 Y(1)=EXTRA(15)
  632 CONTINUE
      Q(101) = EXTRA(16)/Y(1) * 12.0/6.2831853                            CHAM3030
  640 Y(2)=EXTRA(16)                                                      CHAM3040
      CALL  FFF  ( Y(1), Q(1), CF, KER, WC(1) )                           CHAM3050
      IF ( KER )                    670,650,670                           CHAM3060
  650 WRITE (6,660)                                                       CHAM3070
  660 FORMAT (1H0,40X,18H ERROR PROGRAM F   )                             CHAM3080
      CALL CORE(Y(1),100,CF)                                              CHAM3090
  670 GO TO 110                                                           CHAM3100
C                                                                         CHAM3110
C                                                                         CHAM3120
C********************************************************************** CHAM3130
C********************************************************************** CHAM3140
C********************************************************************** CHAM3150
        END                                                               CHAM3160
```

II, D, Program Listing (cont.)

```
@ ELT SUB08,1,690702, 33259

      SUBROUTINE GENMEG(W)
C
C********************************************************************
      DIMENSION W(1)
C
C   LONGITUDINAL WILL HAVE NEG SNH WHICH WILL BE SET TO ZERO BEFOR RETUR
C  IF SNH 0 LONGITIDUNIAL AND GENERATE + OR - 10 PERCENT OF PIE
C********************************************************************
      IF(W(1))40,50,10
C********************************************************************
C   FOR TRANSVERSE GENERATE 10 VALUES AROUND SNH 9 PERCENT BELOW AND
C   11 PERCENT ABOVE
C********************************************************************
   10 SNH=W(1)
   20 DELMEG=.1*SNH
      W(3)=SNH-DELMEG
      DELMEG=DELMEG/5.0
      W(3)=W(3)+DELMEG
      DO 30 I=4,12
      W(I)=W(I-1)+DELMEG
   30 CONTINUE
C********************************************************************
C  NEGITIVE 10 INDICATES TO PROGRAM THAT FREQUENCES ARE GENERATED INTERN
C********************************************************************
      W(2)=-10.0
      RETURN
C********************************************************************
C   IF NEG GENERATE + OR - 10 PERCENT OF POSITIVE INITIAL GUESS
C********************************************************************
   40 SNH=-W(1)
      W(1)=0.0
      GO TO 20
   50 SNH=3.141592
      GO TO 20
      END
```

II, D, Program Listing (cont.)

```
@ ELT SUB09,1,700717, 49719

               SUBROUTINE  CONC (BIN, XOUT, CB, KER )
         C                                                                HYMN  30
         C     CALCULATES COMBUSTION PARAMETERS  HR, HI                   HYMN  40
         C     THIS PROGRAM WAS MODIFIED MAR,1970 BY R. WAUGH AND D. BENNETT TO
         C     INCORPORATE A HIGH MACH NUMBER ANALYSIS REPORTED IN AGC REPORT
         C     NUMBER 20672-09 DATED 13 JANUARY, 1970
         C                                                                HYMN 100
         C*************,*******************************************************  HYMN 110
         C                                                                HYMN 120
               LOGICAL  LOGIK, HRUN, CKOUT, SIMPL, KNOT, LIMIT, TABLR     HYMN 140
         C                                                                HYMN 150
               COMMON  /PROLOG/  LOGIK(38), HEAD(12), SL1, SL2, EORJ      HYMN 160
               COMMON  /ABCDF / DIN (4300)                                HYMN 170
              1   ,SPACE(222)
               DIMENSION    B(240),THR (30),THI (30)
               DIMENSION
              1   BIN  ( 240 ) ,  XOUT ( 100 ) ,  WC    ( 28 )       ,    HYMN 320
              2   WET  (  30 ) ,  ERT  (  30 ) ,  EIT   ( 30 )       ,    HYMN 330
              3   CRT  (  30 ) ,  CIT  ( 30 )
               EQUIVALENCE                                                HYMN 360
              1   ( HRUN , LOGIK( 8)),( CKOUT, LOGIK(13)),( SIMPL, LOGIK(14)),  HYMN 370
              2   ( KNOT , LOGIK(15)),( LIMIT, LOGIK(16)),( TABLR, LOGIK(17))   HYMN 380
               EQUIVALENCE                                                HYMN 390
              1   ( B(1)    ,  SNH    ),(DIN(14) ,  RCH    ),(B(18)    ,  XNE    ),
              3                         ( B(19)  ,  XNW    ),
              4   ( B(4) , UE     ),( B(170) , WC     ),( B(20)    , WET    ),   HYMN 430
              5   ( B(5)    , SOUND  ),( B(50)  ,  ERT    ),
              6   ( B(6)    ,  ULM   ),( B(80)  , EIT     ),
              7   ( B(7) , XK      ),( B(110) , CRT     ),                  HYMN 460
              8   ( B(8) , XCMPL ),( B(140) , CIT     )                     HYMN 470
               EQUIVALENCE                                                HYMN 480
              1   ( DIN(11), GAMMA  ),( DIN(14), RCSTAR ),( DIN(16), CBAROS ),
              2   ( DIN(17), VBARLS ),( DIN(21), SNUETA ),( DIN(51), PBAROS ),
              3   ( DIN(52), XSTAR  ),( DIN(53), VBARM  ),( DIN(54), WBARS  )
               COMPLEX   A,  EOMEGA,COMEGA,Y,H,K1,K2,S,DUMV,V
               DIMENSION A(6,7) ,JC(6)
               DATA A /42*(0.,0.)/
               DATA  PBARP,RHOBRP,HBARP,HBARL/4*1.0/
         C*************,*******************************************************  HYMN 550
         C                                                                HYMN 560
            10 FORMAT ( / 65H TRANSVERSE STABILITY PROGRAM... CALCULATES  HR, HI HYMN 570
              1                   )
            20 FORMAT ( // 11H INPUT DATA// 9X3HSNH12X2HZE10X5HGAMMA9X2HUE9X   HYMN 590
              1   14HSOUND (FT/SEC) 2X13H ULM (FT/SEC)// F15.8,F14.8, F12.5, F14.
              2,, F16.2, 2X F15.6 //)
            30 FORMAT ( // 25H NOZZLE ADMITTANCES INPUT//  15X9HOMEGA(CH) 12X    HYMN 630
              1   3HERT 17X3HEIT 17X3HCRT 17X3HCIT // )                       HYMN 640
            40 FORMAT ( ////22H CALCULATED RESULTS    // 17X5HOMEGA 12X6HH REAL
              114X6HH IMAG)
            50 FORMAT ( 9XF15.7,2E20.8 )                                      HYMN 680
            60 FORMAT ( 9XF15.7,4E20.8 )                                      HYMN 690
           110 FORMAT ( 6H ERROR 6X6E18.8 )                                   HYMN 740
           160 FORMAT ( 7E18.8 )                                              HYMN 830
         C                                                                HYMN 840
         C*************,*******************************************************  HYMN 850
```

II, D, Program Listing (cont.)

```
C********************************************************************** HYMN 860
      DO 170 I = 1, 240                                                 HYMN 910
  170    B(I) = BIN(I)                                                  HYMN 920
  180 CONTINUE
  190 CALL  PAGE  (  70 )                                               HYMN 940
      KOUT = 2                                                          HYMN 950
      NET = XNE                                                         HYMN 960
      WRITE (6,10)                                                      HYMN 970
      IF ( CB-100.0 )          220,200,200                              HYMN 980
  200 WRITE(6,20)(B(I),I=1,6)                                                   *NEW
      WRITE (6,30)                                                      HYMN1000**-1
      WRITE (6,60)  ( WET(I), ERT(I),EIT(I), CRT(I),CIT(I), I=1,NET )   HYMN1010
      CKOUT = ( CB .GT. 200.0 )                                         HYMN1020
C                                                                       HYMN2340
C********************************************************************** HYMN2350
C                                                                       HYMN2360
 220  GAM=GAMMA-1.
      VBARM=UE/SQRT(1.+GAM/2.*UE*UE)
      VBARM2=VBARM*VBARM
      VBARL=VBARLS/CBAROS
      VBARL2=VBARL*VBARL
      RHBROS=PBAROS*GAMMA/(CBAROS*CBAROS)
      RHOBRM=WBARS/(RHBROS*101.065*RCSTAR*RCSTAR*CBAROS*VBARM)
      PBARM=1.-GAMMA*(RHOBRM*VBARM2-RHOBRM*VBARM*VBARL)
      HBARM=1.+VBARL2/2.-VBARM2/2.
      IF(SNUETA) 222,222,221
  221 X=XSTAR/RCSTAR
      GO TO 223
  222 X=XSTAR/DIN(15)
  223 TBARM=HBARM
      NW=XNW+.1
      DO 820 IW=1,NW
      DO 85 LLL=1,42
  85  A(LLL)=(0.,0.)
      W=WC(IW)
      A(1,7)=(1.,0)
      A(3,7)=HBARL+GAM*VBARL2/2.
      A(2,7)=VBARL
      A(1,2)=-VBARM
      A(1,3)=-RHOBRM
      A(2,1)=-1./GAMMA
      A(2,2)=-VBARM2
      A(2,3)=-2.*RHOBRM*VBARM
      A(2,4)=1./GAMMA
      A(3,2)=-GAM*VBARM2*VBARM/2.-HBARM*VBARM
      A(3,3)=-GAM*3.*RHOBRM*VBARM2/2.-HBARM*RHOBRM
      A(3,5)=-RHOBRM*VBARM
      A(4,1)=(1.,0.)
      A(4,2)=(-1.,0.)
      A(4,5)=(-1.,0.)
      A(5,5)=(-1.,0.)
      A(5,1)=GAM/(GAMMA*RHOBRM)
      A(5,6)=TBARM
      A(6,3)=GAMMA
C                                                                       HYMN2420
         CALL  INT4  ( WET, ERT, W, ER )                                HYMN2430
         CALL  INT4  ( WET, EIT, W, EI )                                HYMN2440
C                                                                       HYMN4240
         CALL  INT4  ( WET, CRT, W, CR )                                HYMN4250
```

II, D, Program Listing (cont.)

```
            CALL  INT4  ( WET, CIT, W, CI )                             HYMN4260
C                                                                       HYMN4270
      EOMEGA=CMPLX(ER ,EI )
      COMEGA=CMPLX(CR ,CI )
      S=CMPLX(0.,W)
      DUMV=CSQRT(S*S+SNUETA*SNUETA)
      Y=-CTANH(DUMV*X)*DUMV/(S*GAMMA)
      A(1,4)=Y
      A(3,4)=Y*RHOBRP*HBARP
      A(6,1)=EOMEGA                                                         *NEW
      IF(SNUETA)120,120,121                                                 **-1
  120 A(6,1)=EOMEGA*GAM                                                     *NEW
  121 CONTINUE                                                              **-1
      A(6,6)=GAMMA*COMEGA
      V=(5.,0.)
      CALL CGJR (A,7,6,6,7,$123,JC,V)
      GO TO 124
 123  WRITE (6,125)   A
 125  FORMAT (/1X'OVERFLOW',/(6E20.6/))
 124  K1=-A(4,7)
      K2=-A(1,7)
      H=2.*PBARM/(RHOBRM*VBARM*(K1+K2))
      HR=REAL(H)
      HI=AIMAG(H)
      I=IW
C                                                                       HYMN7080
      THR (IW)=HR
      THI (IW)=HI
  820    CONTINUE                                                       HYMN7090
C                                END  OF  W-LOOP                        HYMN7100
C*********************************************************************** HYMN7110
C                                                                       HYMN7120
      KER = NW                                                          HYMN7130
      XOUT(9)  =  XNW                                                   HYMN7140
C                                                                       HYMN7150
C                                                                       HYMN7170
      DO 830 IW = 1, NW                                                 HYMN7180
         XOUT(IW+ 9)  =  WC (IW)                                        HYMN7190
         XOUT(IW+39)  =  THR(IW)                                        HYMN7200
         XOUT(IW+69)  =  THI(IW)                                        HYMN7210
  830    CONTINUE                                                       HYMN7220
  840 IF ( CKOUT )      CALL PAGE   ( 70 )                              HYMN7240
      WRITE (6,40)                                                      HYMN7250
      WRITE (6,50)    ( WC(I), THR(I), THI(I),   I = 1, NW   )          HYMN7260
C                                                                       HYMN7280
C                                                                       HYMN7490
C                                                                       HYMN7500
C                                                                       HYMN7510
C     THIS SUBROUTINE HAS CALCULATED HR AND HI. NO INJECTION DISTRIBUTIO
C     N OR VELOCITY EFFECTS HAVE BEEN INCLUDED. THESE EFFECTS CAN BE
C     ADDED IN SUBROUTINE DDD WHICH MUST BE RUN TO GET AN N, TAU PLOT
C                                                                       HYMN7570
C                                                                       HYMN7580
C                                                                       HYMN7590
  890 RETURN                                                            HYMN7600
            END                                                         HYMN7610
```

II, D, Program Listing (cont.)

```
@ ELT SUB10,1,700708, 29819

      SUBROUTINE DDD(DIN,DOUT,CD,NER,ERR)                               HTNT  10
C                                                                       HTNT  20
C     PROGRAM  D  COMPUTE HTR,HTI + INTERPOLATE 40 POINTS               HTNT  30
C           20 SEP 67  MODIFIED FOR TABULAR INJECTOR COEFFICIENTS       HTNT  40
C                                                                       HTNT  50
      LOGICAL LOGIK,TABLR,DRUN,ERUN                                         *NEW
      COMMON /PROLOG/ LOGIK(50)                                         HTNT  70**-1
      COMMON  /ABCDF /  EXTRA(100),  ABLOK(600), A   , B , SPACE(3556), HTNT  80
     1         WIT(17), AVN(17), BVN(17),CVNR(17), CVNI(17)             HTNT  90
      DIMENSION DIN(1),DOUT(1),A(133),B(133),OMEGA(1),HR(1),HI(1)       HTNT 100
     1,HTR(30),HTI(30),HTRINT(1),HTIINT(1), OMGA(1)                     HTNT 110
     2,DOM(30)                                                          HTNT 120
      EQUIVALENCE (LOGIK(4), DRUN),(LOGIK(5), ERUN),(LOGIK(17),TABLR)       *NEW
      EQUIVALENCE (A(1),ANH),(A(2),BNH),(A(3),CNHRE),(A(4),CNHIM),      HTNT 140**-1
     1(A(5),GAMMA),(A(6),XLRN),(A(7),XLON),(A(9),XNW),(A(10),OMEGA),    HTNT 150
     2(A(40),HR),(A(70),HI),(B(9),ONW),(B(10),OMGA),(B(51),HTRINT),     HTNT 160
     3(B(92),HTIINT) ,  ( OMEGA, DOM )                                  HTNT 170
C                                                                       HTNT 180
   10 FORMAT ( 21H0 INPUT TO PROGRAM  D // 10X8H GAMMA = F8.4,5X6HLR/N =HTNT 190
     1  F12.8, 5X6HLT/N = F12.8 )                                       HTNT 200
   20 FORMAT ( 16X3HANH12X3HBNH11X6HCNH RE 9X6HCNH IM // 9X 4F15.7 // ) HTNT 210
   30 FORMAT ( // 38H INJECTOR DISTRIBUTION COEFFICIENTS... //          HTNT 220
     117X5HOMEGA 11X3HANH 12X3HBNH 11X6HCNH RE 9X6HCNH IM // (9X5F15.7))HTNT 230
   40 FORMAT (1H0, 19X,8HOMEGA(C),9X3H HR,13X3H HI   )                  HTNT 240
   50 FORMAT (1H0,19X, 8HOMEGA(C),9X3HHTR 13X3HHTI )                    HTNT 250
   60 FORMAT(1H ,10X,3F16.6)                                            HTNT 260
   70 FORMAT (1H0 ,20H PROGRAM  D  OUTPUT  // 19X,8HOMEGA(C) 9X         HTNT 270
     16H   HTR ,10X,6H   HTI  // )                                      HTNT 280
   80 FORMAT (91H0ALL VALUES OF HTR ARE NEGATIVE- ( I.E. OUT OF RANGE OFHTNT 290
     1INTEREST- WILL PROCEED TO NEXT CASE))                             HTNT 300
   90 FORMAT(//,30X,69H   FOLLOWING WILL BE INTERPOLATION WITHIN HTR HTIHTNT 310
     1 TABLE GIVEN ABOVE   )                                            HTNT 320
  100 FORMAT (19X,8H OMEGA   ,9X6HHTRINT,10X,6HHTIINT   )               HTNT 330
  110 FORMAT (11X,F10.5,10X,F10.5,10X,F10.5)                            HTNT 340
C                                                                       HTNT 350
      ERR=0.0                                                           HTNT 360
      C=0.0                                                             HTNT 370
      CALL DVCHK (K000FX)                                               HTNT 380
  120 DO 130 I=1,133                                                    HTNT 390
      A(I)= DIN(I)                                                      HTNT 400
  130 B(I)=0.0                                                          HTNT 410
      NER=IFIX(XNW)                                                     HTNT 420
      IF(NER)140,140,160                                                HTNT 430
  140 WRITE (6,150)NER                                                  HTNT 440
  150 FORMAT (1H0,10X,31H NUMBER OF OMEGAS IN ERROR  =  ,3X,I4   )      HTNT 450
      GO TO 510                                                         HTNT 460
  160 IF(NER-29)170,170,140                                             HTNT 470
  170 ONW = 40.0                                                        HTNT 480
      CONTINUE                                                              *NEW
      NWI = A(8) + 0.01                                                 HTNT 500**-1
      IF ( NWI .GT. 2 )       GO TO 180                                 HTNT 510
      ASSIGN 280 TO  NT                                                 HTNT 520
      TABLR = .FALSE.                                                   HTNT 530
      ANH = AVN (1)                                                     HTNT 540
      BNH = BVN (1)                                                     HTNT 550
```

```
      CNHRE = CVNR(1)                                                   HTNT 560
      CNHIM = CVNI(1)                                                   HTNT 570
      GO TO 190                                                         HTNT 580
  180 ASSIGN 270 TO  NT                                                 HTNT 590
      TABLR = .TRUE.                                                    HTNT 600
  190 IF(CD-99.0)260,260,200                                            HTNT 610
  200 CALL  PAGE  ( 60 )                                                HTNT 620
      WRITE (6,10)  GAMMA, XLRN, XLON                                   HTNT 630
      CONTINUE                                                              *NEW
      IF ( TABLR )              GO TO 210                               HTNT 650**-1
      WRITE (6,20)  ANH, BNH, CNHRE, CNHIM                              HTNT 660
      GO TO 220                                                         HTNT 670
  210 WRITE (6,30)   ( WIT(I), AVN(I), BVN(I), CVNR(I), CVNI(I),        HTNT 680
     1                  I = 1, NWI )                                    HTNT 690
  220 WRITE (6,40)                                                      HTNT 700
      GO TO 240                                                         HTNT 710
  240 DO 250 I = 1,NER                                                  HTNT 730**-1
  250 WRITE (6,60)OMEGA(I),HR(I),HI(I)                                  HTNT 740
C                                                                       HTNT 750
  260 IF((.NOT. DRUN) .AND. (.NOT. ERUN)) GO TO 350                         *NEW
      DO 340 I=1,NER                                                    HTNT 770**-1
      W = OMEGA(I)                                                      HTNT 780
      GO TO NT,(270,280)                                                HTNT 790
  270 CALL  INT4  ( WIT, AVN , W, ANH   )                               HTNT 800
      CALL  INT4  ( WIT, BVN , W, BNH   )                               HTNT 810
      CALL  INT4  ( WIT, CVNR, W, CNHRE )                               HTNT 820
      CALL  INT4  ( WIT, CVNI, W, CNHIM )                               HTNT 830
  280 DEN = GAMMA*W                                                     HTNT 840
      X = ANH - CNHIM/DEN*XLON                                          HTNT 850
      Y = BNH / DEN * XLRN + CNHRE * XLON / DEN                         HTNT 860
      SQD = X * X + Y * Y                                               HTNT 870
      HTR(I) = (X * HR(I) + Y * HI(I) ) /SQD                            HTNT 880
      HTI(I) = (X * HI(I) - Y * HR(I) ) / SQD                           HTNT 890
      CALL DVCHK (K000FX)                                               HTNT 900
       GO TO (310,290),K000FX                                           HTNT 910
  290 IF (CD-10.0) 340,300,300                                          HTNT 920
  300 IF (I-1) 330,320,330                                              HTNT 930
  310 C=10.0                                                            HTNT 940
  320 LIN=NER+6                                                         HTNT 950
      CALL PAGE (LIN)                                                   HTNT 960
      WRITE (6,70)                                                      HTNT 970
  330 WRITE (6,60)DOM(I),HTR(I),HTI(I)                                  HTNT 980
  340 CONTINUE                                                          HTNT 990
      GO TO 370                                                         HTNT1000
C                                                                       HTNT1010
  350 DO 360 I = 1,NER                                                  HTNT1020
      HTI(I) = HI(I)                                                    HTNT1030
  360 HTR(I) = HR(I)                                                    HTNT1040
  370 JOMEG1 = 0                                                        HTNT1050
      JOMEG2=0                                                          HTNT1060
      IF ( CD .LE. 99.0 )      CALL   PAGE   (60)                       HTNT1065
      DO 390 I =1,NER                                                   HTNT1070
      IF (HTR(I)) 390,390,380                                           HTNT1080
  380 JOMEG1 = I                                                        HTNT1090
      GO TO 410                                                         HTNT1100
  390 CONTINUE                                                          HTNT1110
      IF (JOMEG1)400,400,410                                            HTNT1120
  400 WRITE (6,80)                                                      HTNT1130
      ERR= -1.0                                                         HTNT1140
```

II, D, Program Listing (cont.)

```
      GO TO 520                                                         HTNT1150
  410 DO 430 J =I,NER                                                   HTNT1160
      IF (HTR(J)) 420,430,430                                           HTNT1170
  420 JOMEG2 = J                                                        HTNT1180
      GO TO 440                                                         HTNT1190
  430 CONTINUE                                                          HTNT1200
      JOMEG2 =NER                                                       HTNT1210
  440 DLTMEG = (DOM(JOMEG2) -DOM(JOMEG1))/ 39.0                         HTNT1220
      L=NER+1                                                           HTNT1230
      HTI(L)=0.0                                                        HTNT1240
      HTR(L)=0.0                                                        HTNT1250
      DOM(L)=0.0                                                        HTNT1260
      OMGA(1) =DOM(JOMEG1)                                              HTNT1270
      DO 490 I=1,40                                                     HTNT1280
      CALL INT4 (DOM(1),HTR(1),OMGA(I),HTRINT(I) )                      HTNT1290
      CALL INT4 (DOM(1),HTI(1),OMGA(I),HTIINT(I) )                      HTNT1300
      IF (CD-10,0)480,450,450                                           HTNT1310
  450 IF ( I-1 )   470,460,470                                          HTNT1320
  460 CALL PAGE (60)                                                        *NEW
      WRITE (6,90)                                                      HTNT1340**-1
      WRITE (6,100)                                                     HTNT1350
  470 WRITE (6,60) OMGA(I),HTRINT(I),HTIINT(I)                          HTNT1360
  480 OMGA(I+1) = OMGA(I) + DLTMEG                                      HTNT1370
  490 CONTINUE                                                          HTNT1380
      DO 500 I=1,133                                                    HTNT1390
  500 DOUT (I) = B(I)                                                   HTNT1400
      IF(C) 510,520,510                                                 HTNT1410
  510 NER=0                                                             HTNT1420
  520 RETURN                                                            HTNT1430
      END
```

II, D, Program Listing (cont.)

```
@ELT SUB14,1,700708, 29821

      SUBROUTINE  FFF ( FIN, FOUT, CF, NER, W )                         NTAU  10
C*********************************************************************** NTAU  20
      DIMENSION FIN(1),FOUT(1),A(133),B(133),OMGA(1),XN(1),TAU(1),OMX(1)NTAU  30
     1,HTRINT(1),HTIINT(1)                                              NTAU  40
      DIMENSION XNNEW(41),FCCPS(41),W (1)                               NTAU  50
      COMMON  /ABCDF /  EXTRA(100),  ABLOK(600), A    , B,  XNNEW, FCCPS NTAU  60
      EQUIVALENCE (A(9),ONW),(A(10),OMGA),(A(51),HTRINT),(A(92),HTIINT),NTAU  70
     1(B(9),XNW),(B(10),OMX),(B(51),TAU),(B(92),XN)                     NTAU  80
C                                                                       NTAU  90
C                                                                       NTAU 100
      CONST=FOUT(101)                                                   NTAU 110
      XNMIN=1000.0                                                      NTAU 120
      CALL PAGE(70)                                                     NTAU 130
      CALL DVCHK (K000FX)                                               NTAU 140
   10 DO 20 I=1,133                                                     NTAU 150
      A(I) =FIN(I)                                                      NTAU 160
   20 B(I)=0.0                                                          NTAU 170
      NER=IFIX(ONW)                                                     NTAU 180
      XNW=ONW                                                           NTAU 190
      IF(CF-99.0)70,70,30                                               NTAU 200
   30 CALL PAGE (44)                                                    NTAU 210
      WRITE (6,40)                                                      NTAU 220
   40 FORMAT (1H0,50H PROGRAM   F   INPUT   SOLVE FOR N(W) AND TAU(W)   NTAU 230
     1//19X,8H(OMEGA)D,9X,6HHTRINT,10X,6HHTIINT)                        NTAU 240
      DO 50 I =1,40                                                     NTAU 250
   50 WRITE (6,60)OMGA(I),HTRINT(I), HTIINT(I)                          NTAU 260
   60 FORMAT(1H ,10X,3F16.6)                                            NTAU 270
   70 CONTINUE                                                          NTAU 280
      DO 120 I =1,40                                                    NTAU 290
      XN(I)=(HTRINT(I)*HTRINT(I) + HTIINT(I)*HTIINT(I))/(2.0*HTRINT(I)) NTAU 300
      DNOM = XN(I) - HTRINT(I)                                          NTAU 310
      CALL QUAD (DNOM,HTIINT(I),TAU(I))                                 NTAU 320
      TAU(I)=(TAU(I)*A(1)*83.333333)/(A(2)*OMGA(I))                     NTAU 330
C A(1)=L,A(2)=CO                                                        NTAU 340
      OMX(I)=OMGA(I)                                                    NTAU 350
      CALL DVCHK (K000FX)                                               NTAU 360
       GO TO (80,90),K000FX                                             NTAU 370
   80 NER=0                                                             NTAU 380
      GO TO 220                                                         NTAU 390
   90 IF(XN(I))120,120,100                                              NTAU 400
  100 IF(XN(I)-XNMIN) 110,120,120                                       NTAU 410
  110 XNMIN=XN(I)                                                       NTAU 420
      IMIN=I                                                            NTAU 430
  120 CONTINUE                                                          NTAU 440
      DO 130 I=1,100                                                    NTAU 450
  130 FOUT(I) = B(I)                                                    NTAU 460
      CONTINUE                                                                  *NEW
  140 CALL PAGE (45)                                                    NTAU 480**-1
      WRITE (6,150)                                                     NTAU 490
  150 FORMAT(1H0,20H PROGRAM   F   OUTPUT   //21X,7HFC(CPS),13X,8H(OMGA)D,NTAU 500
     113X,7HTAU(MS),16X,1HN   )                                         NTAU 510
      DO 160 I=1,40                                                     NTAU 520
      FCCPS(I)=CONST*OMX(I)                                             NTAU 530
  160 WRITE (6,170)FCCPS(I),OMX(I),TAU(I),XN(I)                         NTAU 540
  170 FORMAT(21X,F7.1 ,11X,F10.5,10X,F10.5,10X,F10.5)                   NTAU 550
```

II, D, Program Listing (cont.)

```
180 HTIINT(41) = 0.0                                                    NTAU 560
    HTRINT(41)=0.0                                                      NTAU 570
    XN(41)=0.0                                                          NTAU 580
    OMX(41)=0.0                                                         NTAU 590
    FCCPS(41)=0.0                                                       NTAU 600
190 FORMAT(36X,5HNMIN=,F10.5,/36X,8HTAU(MS)=,F10.5,/36X,9H(OMEGA)D=,F1NTAU 610
   10.5,/36X,8HFC(CPS)=,F10.1,)                                         NTAU 620
    DO 200 I =1,40                                                      NTAU 630
200  CALL INT4D(OMX(1),XN(1),OMX(I),SAVNOT,XNNEW(I))                    NTAU 640
    XNNEW(41)=0.0                                                       NTAU 650
     CALL INT4(XNNEW(1),OMX(1),0.0,DOMMIN)                              NTAU 660
    FCMIN=CONST*DOMMIN                                                  NTAU 670
    CALL INT4(FCCPS(1),HTRINT(1),FCMIN,HTR1)                            NTAU 680
    CALL INT4(FCCPS(1),HTIINT(1),FCMIN,HTI1)                            NTAU 690
    HTRMIN=(HTR1*HTR1+HTI1*HTI1)/(2.0*HTR1)                             NTAU 700
    DNOM=HTRMIN-HTR1                                                    NTAU 710
    CALL QUAD(DNOM,HTI1,TAUMIN)                                         NTAU 720
    TAUMIN=(TAUMIN*A(1)*83.333333)/(A(2)*DOMMIN)                        NTAU 730
    CALL PAGE (8)                                                       NTAU 740
    WRITE (6,210)                                                       NTAU 750
210 FORMAT(//21X,58H THE FOLLOWING ARE VALUES INTERPOLATED AT SLOPE OFNTAU 760
   1 N=0.0                              ,/)                             NTAU 770
    WRITE (6,190)HTRMIN,TAUMIN,DOMMIN,FCMIN                             NTAU 780
220 RETURN                                                              NTAU 790
    END                                                                 NTAU 800
```

II, D, Program Listing (cont.)

```
@ ELT SUB15,1,690702, 39149

      SUBROUTINE QUAD (A,B,ANGLE)                                       QUAD  10
      IF(B) 10,50,80                                                    QUAD  20
   10 IF(A) 20,30,40                                                    QUAD  30
   20 ROTATE = 3.1415927                                                QUAD  40
      GO TO 110                                                         QUAD  50
   30 ANGLE = 4.7123890                                                 QUAD  60
      GO TO 120                                                         QUAD  70
   40 ROTATE = 6.2831853                                                QUAD  80
      GO TO 110                                                         QUAD  90
   50 IF (A) 60,70,70                                                   QUAD 100
   60 ANGLE = 3.1415927                                                 QUAD 110
      GO TO 120                                                         QUAD 120
   70 ANGLE = 0.0                                                       QUAD 130
      GO TO 120                                                         QUAD 140
   80 IF(A) 20,90,100                                                   QUAD 150
   90 ANGLE = 1.5707963                                                 QUAD 160
      GO TO 120                                                         QUAD 170
  100 ROTATE = 0.0                                                      QUAD 180
  110 ANGLE = ATAN(B/A) + ROTATE                                        QUAD 190
  120 RETURN                                                            QUAD 200
      END                                                               QUAD 210
```

II, D, Program Listing (cont.)

```
@ ELT SUB16,1,690702, 39150

              SUBROUTINE CORE(X,N,CODE)                                      KORE  10
              DIMENSION X(1)                                                 KORE  20
              IF(CODE-500.0)40,10,10                                         KORE  30
           10 CODE=100.0                                                     KORE  40
              CALL PAGE(70)                                                  KORE  50
              WRITE (6,20)                                                   KORE  60
           20 FORMAT(10X,37HINPUT DATA DUMP FOR PROGRAM FAILIER       //)    KORE  70
              WRITE (6,30)(X(I),I=1,N)                                       KORE  80
           30 FORMAT(5X,10(F10.4,2X))                                        KORE  90
           40 RETURN                                                         KORE 100
              END                                                            KORE 110
```

II, D, Program Listing (cont.)

```
@ ELT SUB17,1,690702, 39151

      SUBROUTINE  INJCTR
C
      COMMON  /PROLOG/  LOGIK(38), HEAD(12), SL1, SL2, EORJ
      COMMON  /JECTOR/  EJDATA( 9600)
C
      EQUIVALENCE  (LOGIK(5),ERUN) , (LOGIK(10),JRUN ) , (LOGIK(9),IRUN)
      LOGICAL  LOGIK, ERUN, JRUN, SL1, SL2, EORJ, IRUN
      DIMENSION  HEAD1(12)
C
      IF ( SL1 )               GO TO 20
      DO 10 I = 1, 9600
   10   EJDATA(I) = 0.0
      SL1 = .TRUE.
      GO TO 30
C
   20 READ (13)  EJDATA
      BACKSPACE  13
   30 CALL  AS138  ( EJDATA, HEAD1, NE )
      IF ( NE .NE. 1 )          CALL  EXIT
      WRITE (13)  EJDATA
      BACKSPACE  13
      IF ( .NOT. JRUN )        GO TO 40
      CALL  JJJ
   40 IF ( ERUN .OR. IRUN )    CALL  INJDIS
C
   50 RETURN
            END
```

II, D, Program Listing (cont.)

```
@ ELT SUB18,1,690708, 48741

      SUBROUTINE JJJ                                                    JECT  10
C                                                                       JECT  20
C    ************  DECK MODIFIED 20 AUG 67     *********************    JECT  30
C                                                                       JECT  40
      DIMENSION  THET1(1450), R1(1450), TMU(1000)                       JECT  50
      DIMENSION            NELE(1000),X(1000),Y(1000),NTYPEE(1000),R(22)JECT  60
     1,THETA(182),AS(20),WTE(1000),AAX(1000),AAF(1000),NBAND(1000),XMRE(JECT  70
     21000),XMUTOT(20),AXTY(200),AFTY(200),XX(1000),YY(1000)            JECT  80
      COMMON  /JECTOR/  DATA ( 9600)                                    JECT  90
C                                                                       JECT 130
      EQUIVALENCE (DATA(3),XM), (DATA(4),XN)                            JECT 140
      EQUIVALENCE (DATA(1021),THET1),(DATA(2471),R1),(DATA(3921),TMU)   JECT 150
      EQUIVALENCE (NTYPEE,DATA(2001)),(AXTY,DATA(1001)),(AFTY,DATA(1201)JECT 160
     1), (DATA(1921),XX),(DATA(2921),YY),(DATA(4921),XMUTOT)           JECT 170
C                                                                       JECT 180
      NERROR=0                                                          JECT 200
      SECT=DATA(5)                                                      JECT 210
      WT=DATA(9573)                                                     JECT 220
      RINJ=DATA(9574)                                                   JECT 230
      XMR=DATA(9575)                                                    JECT 240
      FFC=DATA(9576)                                                    JECT 250
      DFFC=DATA(9577)                                                   JECT 260
      NT=DATA(9570)+.0001                                               JECT 270
      POLAR=DATA(9572)                                                  JECT 280
      NE=DATA(9571)+.0001                                               JECT 290
      ROX=DATA(9579)                                                    JECT 300
      ROF=DATA(9580)                                                    JECT 310
      EMUMAX=5.0                                                        JECT 320
      PXADJ=1.0                                                         JECT 330
      PFADJ=1.0                                                         JECT 340
      CDX=DATA(9585)                                                    JECT 350
      CDF=DATA(9586)                                                    JECT 360
      XFC=DATA(9591)                                                    JECT 370
      DXFC=DATA(9592)                                                   JECT 380
      PFFC = DATA(9589)/100.0                                           JECT 390
      PXFC = DATA(9590)/100.0                                           JECT 400
      IF (XM .EQ. 0.0 ) XM = 20.0                                       JECT 410
      IF ( XN .EQ. 0.0 ) XN = 180.0                                     JECT 420
   10 K=319                                                             JECT 430
      THFNL=6.2831853/SECT                                              JECT 440
      IF(DATA(322))20,30,30                                             JECT 450
   20 ROTATE=3.1415926/SECT                                             JECT 460
      GO TO 40                                                          JECT 470
   30 ROTATE=0.0                                                        JECT 480
   40 IF(POLAR)50,180,50                                                JECT 490
   50 IF(SECT-1.0)80,80,60                                              JECT 500
   60 DO 70 I=1,NE                                                      JECT 510
      KK=K+1                                                            JECT 520
      NELE(I)=DATA(KK)+.0001                                            JECT 530
      X(I)=DATA(KK+1)                                                   JECT 540
      Y(I)=DATA(KK+2)                                                   JECT 550
      NTYPEE(I)=DATA(KK+3)+.0001                                        JECT 560
      SAVE=SQRT(X(I)*X(I)+Y(I)*Y(I))                                    JECT 570
      Y(I)=(ATAN(Y(I)/X(I)))+ROTATE                                     JECT 580
      X(I)=SAVE                                                         JECT 590
```

II, D, Program Listing (cont.)

```
      K=K+4
   70 CONTINUE
      GO TO 220
   80 DO 170 I=1,NE
      KK=K+1
      NELE(I)=DATA(KK)+.0001
      X(I)=DATA(KK+1)
      Y(I)=DATA(KK+2)
      NTYPEE(I)=DATA(KK+3)+.0001
C     THE FOLLOWING SEGMENT OF CODING REPLACES SUBROUTINE MOVXY...
         IF ( Y(I) )             90,130,140
   90    IF ( X(I) )          100,110,120
  100    ROTATE = 3.1415927
         GO TO 160
  110    X(I) = 1.0E-15
  120    ROTATE = 6.2831853
         GO TO 160
  130    ROTATE = 0.0
         GO TO 160
  140    IF ( X(I) )          100,150,130
  150    X(I) = 1.0E-15
         ROTATE = 0.0
  160 SAVE = SQRT ( X(I)*X(I)+Y(I)*Y(I) )
      SAVE=SQRT(X(I)*X(I)+Y(I)*Y(I))
      Y(I)=(ATAN(Y(I)/X(I)))+ROTATE
      X(I)=SAVE
      K=K+4
  170 CONTINUE
      GO TO 220
  180 IF(SECT-1.0)190,190,200
  190 ROTATE=0.0
  200 DO 210 I=1,NE
      KK=K+1
      NELE(I)=DATA(KK)+.0001
      X(I)=DATA(KK+1)
      Y(I)=(DATA(KK+2)*.0174532)+ROTATE
      NTYPEE(I)=DATA(KK+3)+.0001
      K=K+4
  210 CONTINUE
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C     CALCULATING AREAS OF ELEMENTS
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
  220 NN=4944
      DO 280 I=1,NT
      KK=NN+1
      NST=DATA(KK)+.0001
      KK1=KK+1
      JJ=KK1
      NX=DATA(KK1)+.0001
      AXTY(NST)=0.0
      IF(NX)230,250,230
  230 DO 240 JX=1,NX
      JJ=KK1+JX
      AXTY(NST)=AXTY(NST)+(.7853891*(DATA(JJ)*DATA(JJ)))
  240 CONTINUE
  250 NN1=JJ+1
      NN=NN1
      NF=DATA(NN1)+.0001
      AFTY(NST)=0.0
```

II, D, Program Listing (cont.)

```
      IF(NF)260,280,260
  260 DO 270 JF=1,NF
      NN=NN1+JF
      AFTY(NST)=AFTY(NST)+(.7853981*(DATA(NN)*DATA(NN)))
  270 CONTINUE
  280 CONTINUE
      AXTOT=0.0
      AFTOT=0.0
      DO 290 I=1,NE
      NN=NTYPEE(I)
      AAX(I)=AXTY(NN)
      AAF(I)=AFTY(NN)
      AXTOT=AXTOT+AAX(I)
      AFTOT=AFTOT+AAF(I)
  290 CONTINUE
      AXTOT=AXTOT*SECT
      AFTOT=AFTOT*SECT
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C     CALCULATING RADII AND ANGLE BOUNDRIES
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
  300 NPTS=180.0/SECT
      TPS=NPTS
      DELTH=THFNL/TPS
      AREA = 3.141592 * RINJ**2
      ASECT = AREA/(XM*XN)
      R(1) = 0.0
      ANP = ASECT * 180.0 / 3.141592
      DO 310 I = 2,21
      R(I) = SQRT ( ANP + R(I-1)**2)
  310 CONTINUE
      R(22) = 0.0
      XNUM=0.0
      NPTS1=NPTS+1
      DO 320 J=2,NPTS1
      XNUM=XNUM+1.0
      THETA(J)=DELTH*XNUM
  320 CONTINUE
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C     CALCULATING WEIGHT FLOW
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      WFT1=WT/(XMR+1.0)
      WXT1=WT-WFT1
      WFT2=WFT1
      WXT2=WXT1
      IF(PFFC)340,340,330
  330 WFT2=WFT1*(1.0-PFFC)
  340 IF(PXFC)360,360,350
  350 WXT2=WXT1*(1.0-PXFC)
  360 SIGMEN=WT/(3.141592*RINJ*RINJ)
  370 AXTMAX=AXTOT
      AXTMIN=AXTOT
      AFTMAX=AFTOT
      AFTMIN=AFTOT
  380 AFFC=0.0
      AXFC=0.0
      IF(DFFC)400,400,390
  390 AFFC=FFC*.785398*DFFC*DFFC
      AFTMIN=AFTMIN-AFFC
      AFTMAX=AFTMAX-AFFC
```

II, D, Program Listing (cont.)

```
 400 IF(DXFC)420,420,410                                                JECT1780
 410 AXFC=XFC*.785398*DXFC*DXFC                                         JECT1790
     AXTMIN=AXTMIN-AXFC                                                 JECT1800
     AXTMAX=AXTMAX-AXFC                                                 JECT1810
 420 AXNOM=(AXTMAX+AXTMIN)/2.0                                          JECT1820
     AFNOM=(AFTMAX+AFTMIN)/2.0                                          JECT1830
     IF(DFFC+DXFC)440,440,430                                           JECT1840
 430 WXT2=(WXT2*AXNOM)/(AXNOM+AXFC)                                     JECT1850
     WFT2=(WFT2*AFNOM)/(AFNOM+AFFC)                                     JECT1860
 440 XMR1=WXT2/WFT2                                                     JECT1870
 450 ETOF=0.0                                                           JECT1880
     ETFF=0.0                                                           JECT1890
     QX=WXT2/AXTOT                                                      JECT1900
     QF=WFT2/AFTOT                                                      JECT1910
     CALL DVCHK (K000FX)                                                JECT1920
 460 DO 480 I=1,NE                                                      JECT1940
     NBAND(I)=0                                                         JECT1950
     WFE1=AAF(I)*QF                                                     JECT1960
     WXE1=AAX(I)*QX                                                     JECT1970
     WTE(I)=WFE1+WXE1                                                   JECT1980
     XMRE(I)=WXE1/WFE1                                                  JECT1990
     ETOF=ETOF+WXE1                                                     JECT2000
     ETFF=ETFF+WFE1                                                     JECT2010
     CALL DVCHK (K000FX)                                                JECT2020
      GO TO (470,480),K000FX                                            JECT2030
 470 NBAND(I)=-1                                                        JECT2040
 480 CONTINUE                                                           JECT2050
     ETOF=SECT*ETOF                                                     JECT2060
     ETFF=SECT*ETFF                                                     JECT2070
     WMRELM=ETOF/ETFF                                                   JECT2080
     AXXTT=AXTOT+AXFC                                                   JECT2090
     AFFTT=AFTOT+AFFC                                                   JECT2100
      WFT2 = ETFF                                                       JECT2110
      WXT2 = ETOF                                                       JECT2120
     WFT1=WFT2/(1.0-PFFC)                                               JECT2130
     WXT1=WXT2/(1.0-PXFC)                                               JECT2140
      WFT1 = WFT1*AFFTT/AFTOT                                           JECT2150
      WXT1 = WXT1*AXXTT/AXTOT                                           JECT2160
     XMRINJ=WXT1/WFT1                                                   JECT2170
     D= ((WXT1/(CDX*AXXTT))**2)*2.2360248                               JECT2180
     WTOT=WXT1+WFT1                                                     JECT2190
     DPX1=D/ROX                                                         JECT2200
     D= ((WFT1/(CDF*AFFTT))**2)*2.2360248                               JECT2210
     DPF1=D/ROF                                                         JECT2220
      TOFCF = WXT1*(1.0 - PXFC)                                         JECT2230
      TOFCF = (TOFCF*(AXXTT - AXFC))/AXXTT                              JECT2240
      TOFCF = WXT1 - TOFCF                                              JECT2250
      TFFC = WFT1*(1.0 - PFFC)                                          JECT2260
      TFFC = (TFFC*(AFFTT - AFFC))/AFFTT                                JECT2270
      TFFC = WFT1 - TFFC                                                JECT2280
     VINJX=SQRT((9273.6*DPX1 /ROX))                                     JECT2290
     VINJF=SQRT((9273.6*DPF1 /ROF))                                     JECT2300
     XXY=TPS*SIGMEN*SECT                                                JECT2310
 490 WRITE (6,820)                                                      JECT2320
     GO TO 500                                                          JECT2330
 500 WRITE (6,760)AXTOT,AFTOT,AXFC,AFFC,AXXTT,AFFTT,DPX1,DPF1           JECT2340
     WRITE (6,770)WTOT,WMRELM,XMRINJ,ETOF,ETFF,TOFCF,TFFC, WXT1, WFT1 ,JECT2350
    1VINJX,VINJF                                                        JECT2360
     PFFC=PFFC*100.0                                                    JECT2370
```

II, D, Program Listing (cont.)

```
      PXFC=PXFC*100.0                                                   JECT2380
  510 WRITE (6,830)WT,NT,NE,CDX                                         JECT2390
      WRITE (6,780)CDF,PFFC,PXFC,DFFC,DXFC,FFC,XFC,ROX,ROF              JECT2400
      CALL PAGE(70)                                                     JECT2410
      WRITE (6,810)                                                     JECT2420
      NPGE=49                                                           JECT2430
      DO 540 I=1,NE                                                     JECT2440
      IF(NPGE-I)520,520,530                                             JECT2450
  520 CALL PAGE(70)                                                     JECT2460
      NPGE=NPGE+48                                                      JECT2470
      WRITE (6,810)                                                     JECT2480
  530 DEGREE=Y(I)/.0174532                                              JECT2490
      B=X(I)*SIN(Y(I))                                                  JECT2500
      A=X(I)*COS(Y(I))                                                  JECT2510
      WRITE (6,800)NELE(I),NTYPEE(I),X(I),DEGREE,A,B                    JECT2520
  540 CONTINUE                                                          JECT2530
      CALL PAGE(70)                                                     JECT2540
      NPGE=46                                                           JECT2550
      JCONT=0                                                           JECT2560
      WRITE (6,840)                                                     JECT2570
      NN=4944                                                           JECT2580
      DO 670 I=1,NT                                                     JECT2590
      KK=NN+1                                                           JECT2600
      NST=DATA(KK)+.0001                                                JECT2610
      KK1=KK+1                                                          JECT2620
      JJ=KK1                                                            JECT2630
      NX=DATA(KK1)+.0001                                                JECT2640
      JCONT=JCONT+NX+1                                                  JECT2650
      IF(NPGE-JCONT)550,550,560                                         JECT2660
  550 CALL PAGE(70)                                                     JECT2670
      WRITE (6,840)                                                     JECT2680
      JCONT=0                                                           JECT2690
  560 AXMAX=0.0                                                         JECT2700
      AFMAX=0.0                                                         JECT2710
      WFE1=0.0                                                          JECT2720
      WXE1=0.0                                                          JECT2730
      WRITE (6,850)NST,NX                                               JECT2740
      IF(NX)570,590,570                                                 JECT2750
  570 DO 580 JX=1,NX                                                    JECT2760
      JJ=KK1+JX                                                         JECT2770
      A=.7853891*DATA(JJ)*DATA(JJ)                                      JECT2780
      B=A*QX                                                            JECT2790
      WRITE (6,860)DATA(JJ),A,B                                         JECT2800
      AXMAX=AXMAX+A                                                     JECT2810
      WXE1=WXE1+B                                                       JECT2820
  580 CONTINUE                                                          JECT2830
  590 NN1=JJ+1                                                          JECT2840
      NN=NN1                                                            JECT2850
      NF=DATA(NN1)+.0001                                                JECT2860
      JCONT=JCONT+NF+2                                                  JECT2870
      IF(NPGE-JCONT)600,600,610                                         JECT2880
  600 CALL PAGE (70)                                                    JECT2890
      WRITE (6,840)                                                     JECT2900
      JCONT=0                                                           JECT2910
  610 WRITE (6,870)NF                                                   JECT2920
      IF(NF)620,640,620                                                 JECT2930
  620 DO 630 JF=1,NF                                                    JECT2940
      NN=NN1+JF                                                         JECT2950
      A=.7853891*DATA(NN)*DATA(NN)                                      JECT2960
```

II, D, Program Listing (cont.)

```
      B=A*QF                                                            JECT2970
      WRITE (6,880)DATA(NN),A,B                                         JECT2980
      AFMAX=AFMAX+A                                                     JECT2990
      WFE1=WFE1+B                                                       JECT3000
  630 CONTINUE                                                          JECT3010
  640 WTOT1=WXE1+WFE1                                                   JECT3020
      IF(WFE1)660,650,660                                               JECT3030
  650 WRITE (6,900)AXMAX,AFMAX,WTOT1                                    JECT3040
      GO TO 670                                                         JECT3050
  660 XMR1=WXE1/WFE1                                                    JECT3060
      WRITE (6,890)AXMAX,AFMAX,WTOT1,XMR1                               JECT3070
  670 CONTINUE                                                          JECT3080
  680 READ  (13)  DATA                                                  JECT3090
      BACKSPACE  13                                                     JECT3100
      LINKNT = 0                                                        JECT3110
      AJS = XM*XN/WT                                                    JECT3120
      DO 700 J = 1,NE                                                   JECT3130
      TMU(J) = WTE(J)*AJS                                               JECT3140
      IF (LINKNT .GT. 0) GO TO 690                                      JECT3150
      CALL PAGE(70)                                                     JECT3160
      WRITE (6,910)                                                     JECT3170
      WRITE (6,930)                                                     JECT3180
      LINKNT = 50                                                       JECT3190
  690 WRITE(6,920) J, X(J), Y(J), TMU(J)                                JECT3200
      LINKNT = LINKNT - 1                                               JECT3210
  700 CONTINUE                                                          JECT3220
      DO   705  I = 1, NE                                               JECT3222
         XX(I) = X(I)                                                   JECT3224
  705    YY(I) = Y(I)                                                   JECT3225
      WRITE  (13)       DATA                                            JECT3230
      BACKSPACE  13                                                     JECT3240
      XMUMAX=0.0                                                        JECT3250
      AJS = AJS*SECT/XN                                                 JECT3260
      DO 730 J=1,20                                                     JECT3270
      TOTW=0.0                                                          JECT3280
      XXX = XXY * ASECT                                                 JECT3290
      DO 710 I=1,NE                                                     JECT3310**-1
      IF (X(I) .LE. R(J+1) .AND. X(I) .GT. R(J)) TOTW = TOTW + WTE(I)   JECT3320
  710 CONTINUE                                                          JECT3330
      XMUTOT(J) = TOTW*AJS                                              JECT3340
      IF(XMUTOT(J)-XMUMAX)730,730,720                                   JECT3350
  720 XMUMAX=XMUTOT(J)                                                  JECT3360
  730 CONTINUE                                                          JECT3370
      CALL PAGE(70)                                                     JECT3380
      WRITE (6,790)                                                     JECT3390
      DO 740 J=1,20                                                     JECT3400
      XXMAX=XMUTOT(J)/XMUMAX                                            JECT3410
      WRITE (6,750)R(J+1),XMUTOT(J),XXMAX                               JECT3420
  740 CONTINUE                                                          JECT3430
      RETURN                                                            JECT3440
  750 FORMAT(32X,F6.3,14X,F7.3,13X,F6.4)                                JECT3450
  760 FORMAT(//,5X,28HA....PROPELLENT ORFICE AREAS,//,9X,34HELEMENT TOTAJECT3460
     1L OXIDIZER AREA      =,F11.8,8H SQ. IN.,15X,30HELEMENT TOTAL FUEL JECT3470
     2AREA      =,F11.8,8H SQ. IN.,/,9X,34HTOTAL OXIDIZER FILM COOLING AJECT3480
     3REA =,F11.8,8H SQ. IN.,15X,30HTOTAL FUEL FILM COOLING AREA =,F11.8JECT3490
     4,8H SQ. IN.,/,9X,34HINJECTOR TOTAL OXIDIZER AREA      =,F11.8,8H SQJECT3500
     5. IN.,15X,30HINJECTOR TOTAL FUEL AREA     =,F11.8,8H SQ. IN.,//,5XJECT3510
     6,54HB....INJECTOR PRESSURE DROPS FOR ABOVE INJECTOR DESIGN,//,9X, JECT3520
     724HOXIDIZER PRESSURE DROP =,F6.1,3HPSI,35X,20HFUEL PRESSURE DROP =JECT3530
```

II, D, Program Listing (cont.)

```
    8,F6.1,4H PSI,///,5X,71HC....PROPELLANT FLOWS AND INJECTOR VELOCITIJECT3540
    9ES FOR ABOVE INJECTOR DESIGN,/)                                    JECT3550
770 FORMAT(/,48X,19HTOTAL WEIGHT FLOW =,F6.1,7H LB/SEC,/,42X,39HAVERAGJECT3560
    1E MIXTURE RATIO OF THE ELEMENTS =,F7.3,/,41X,40HOVERALL MIXTURE RAJECT3570
    2TIO FOR THE INJECTOR =,F7.3,//,9X,34HELEMENT TOTAL OXIDIXER FLOW  JECT3580
    3    =,F6.1,7H LB/SEC,21X,26HELEMENT TOTAL FUEL FLOW  =,F6.1,7H LB/JECT3590
    4SEC,/,9X,34HTOTAL OXIDIZER FILM COOLING FLOW =,F6.1,7H LB/SEC,21X,JECT3600
    526HTOTAL FUEL FILM COOLING  =,F6.1,7H LB/SEC,/,9X,34HINJECTOR TOTAJECT3610
    6L OXIDIZER FLOW     =,F6.1,7H LB/SEC,21X,25HINJECTOR TOTAL FUEL FLJECT3620
    7OW=,F7.1,7H LB/SEC,/,9X,37HOXIDIZER OVERALL INJECTION VELOCITY =,FJECT3630
    86.1,7H FT/SEC,18X,33HFUEL OVERALL INJECTION VELOCITY =,F6.1,7H FT/JECT3640
    9SEC,//)                                                            JECT3650
780 FORMAT(9X,23HFUEL LOSS COEFFICIENT =,F5.3,/,9X,27HPERCENT FUEL FILJECT3660
    1M COOLING =,F5.1,/,9X,31HPERCENT OXIDIZER FILM COOLING =,F5.1,/,9XJECT3670
    2,38HDIAMETER OF FUEL FILM COOLING ORFICE =,F7.5,75H IN. (NOTE..THIJECT3680
    3S MIGHT BE AN EQUIVALENT  DIAMETER FOR MULTIPLE-ROW COOLING),/,9X,JECT3690
    442HDIAMETER OF OXIDIZER FILM COOLING ORFICE =,F7.5,21H IN. (SEE ABJECT3700
    5OVE NOTE),/,9X,51HNUMBER OF FUEL FILM COOLING ORIFICES PER INJECTOJECT3710
    6R =,F5.0,/,9X,54HNUMBER OF OXIDIZER FILM COOLING ORFICES PER INJECJECT3720
    7TOR =,F5.0,/,9X,18HOXIDIZER DENSITY =,F7.2,4H PCF,/,9X,14HFUEL DENJECT3730
    8SITY =,F6.1,4H PCF,///)                                            JECT3740
790 FORMAT(////,3X,114HSECTION 4..RESULTANT FLOW DISTRIBUTION (MU) AS JECT3750
    1A FUNCTION OF ONLY THE RADIUS, I.E.. AVERAGE OF MU IN A RADIAL BANJECT3760
    2D,     ///,32X,6HRADIUS,15X,5HMU(R),11X,13HMU(R)/MU(MAX),/,34X,3HINJECT3770
    3.,//)                                                              JECT3780
800 FORMAT(13X,I3,18X,I2,4X,4(10X,F10.5))                              JECT3790
810 FORMAT(//,2X, 49HSECTION 2.... ELEMENT LOCATION AND INJECTION TYPEJECT3800
    1,///,11X, 7HELEMENT,15X, 4HTYPE,17X,1HR,17X, 5HTHETA,17X,1HX,19X,1JECT3810
    2HY,/,13X, 3HNO.,18X, 3HNO.,14X, 8H(INCHES),12X, 8H(DEGREE),12X, 8HJECT3820
    3(INCHES),12X,8H(INCHES),//)                                        JECT3830
820 FORMAT(///,3X, 74HSECTION 1....MISCELLANEOUS INFORMATION FOR INJECJECT3840
    1TOR DESIGNED BY PROJECTS               ,/)                         JECT3850
830 FORMAT(//,5X, 46HD......INPUT INFORMATION USED IN COMPUTATIONS  ,/JECT3860
    1/,9X, 23HTOTAL PROPELLENT FLOW =,F6.1,7H LB/SEC,/,9X, 59HTOTAL NUMJECT3870
    2BER OF ELEMENT TYPES (SYMMETRICAL SECTION ONLY) = ,I3,/,9X, 54HTOTJECT3880
    3AL NUMBER OF ELEMENTS (SYMMETRICAL SECTION ONLY) = ,I4,/,9X, 27HOXJECT3890
    4IDIZER LOSS COEFFICIENT =,F5.3,)                                   JECT3900
840 FORMAT(//,2X, 78HSECTION 3....TYPE DESCRIPTION,ORIFICE AREA, PROPEJECT3910
    1LLANT FLOW, AND MIXTURE RATIO,//,3X, 84HTYPE   *-------- OXIDIZER JECT3920
    2ORIFICE DATA -----**----------- FUEL ORIFICE DATA --------*,//,93X,JECT3930
    3 5HTOTAL,5X, 5HTOTAL,5X, 5HTOTAL,4X, 7HMIXTURE,/,12X, 6HNUMBER,34XJECT3940
    4, 6HNUMBER,33X, 8HOXIDIZER,4X, 4HFUEL,3X,10HPROPELLENT,3X, 5HRATIOJECT3950
    5,/,14X,2HOF,5X, 8HDIAMETER,4X, 4HAREA,6X, 4HFLOW,7X,2HOF,5X, 8HDIAJECT3960
    6METER,4X, 4HAREA,6X, 4HFLOW,6X, 4HAREA,6X, 4HAREA,4X, 9HFLOW RATE,JECT3970
    7/,11X, 7HORIFICE,6X,3HIN.,5X, 7HSQ. IN.,4X, 6HLB/SEC,3X, 8HORIFICEJECT3980
    8S,5X, 3HIN.,4X, 7HSQ. IN.,4X, 6HLB/SEC,3X, 7HSQ. IN.,3X, 7HSQ. IN.JECT3990
    9,4X, 6HLB/SEC,//,)                                                 JECT4000
850 FORMAT(3X,I3,7X,I2)                                                JECT4010
860 FORMAT(22X,F6.4,3X,F8.6,2X,F8.6)                                   JECT4020
870 FORMAT(54X,I2)                                                     JECT4030
880 FORMAT(62X,F6.4,3X,F8.6,2X,F8.6)                                   JECT4040
890 FORMAT(91X,F8.5,2X,F8.5,2X,F8.4,3X,F7.4)                           JECT4050
900 FORMAT(91X,F8.5,2X,F8.5,2X,F8.3,2X,8HINFINITY)                     JECT4060
910 FORMAT (///10X15HELEMENT RESULTS )                                 JECT4070
920 FORMAT (32XI5,11XF10.3,10XF10.4,11XF10.4 )                         JECT4080
930 FORMAT (//31X7HELEMENT14X6HRADIUS15X5HANGLE11X12HDISTRIBUTION/90X JECT4090
    111HCOEFFICIENT/34X3HNO.35X7HRADIANS15X2HMU// )                    JECT4100
                     END                                                JECT4110
```

```
@ ELT SUB19,1,691029, 58811

      SUBROUTINE  INJDIS                                                INJD  10
C                                                                       INJD  20
C  ************  DECK MODIFIED 20 AUG 67     ********************       INJD  30
C                                                                       INJD  40
      REAL IPP,IPR,IPT,INTEG,IPX                                        INJD  50
      INTEGER DSCRB, TIME                                               INJD  60
      LOGICAL  LOGIK,  SL1,  SL2,  EORJ, IRUN                           INJD  70
      COMMON  /PROLOG/  LOGIK(50), SL1, SL2, EORJ                       INJD  80
      COMMON  /JECTOR/  DATA ( 9600)                                    INJD  90
      COMMON   GAMMA, NW,  WC,  AVN,  BVN,  CVNR,  CVNI,  CE,  CI       INJD 110
      DIMENSION                                                         INJD 120
     1    U      (1000),  AVN    (  30),  BVN    (  30),  CVNR  (  30), INJD 130
     2    CVNI  (  30),  RR    (1000),  THATA (1000),  FIRST (  30),    INJD 140
     3    SECOND(  30),  Z     (   2),                 WC    (  30),    INJD 150
     4     IPP(50), OPP(50), IPR(50), OPR(50), IPT(50), OPT(50),        INJD 160
     5    X     (1000),  Y     (1000)                                   INJD 170
      EQUIVALENCE  ( IRUN, LOGIK(9) ),                                  INJD 180
     1   (E1     ,DATA(1)),                      (XM    ,DATA(3)),      INJD 190
     2   (XN     ,DATA(4)),(V     ,DATA(6)),(SVN   ,DATA(8)),           INJD 200
     3   (P00   ,DATA(13)),                        (Z1 ,DATA(15)),      INJD 210
     4   (RR, X, DATA(1921) ),(THATA, Y, DATA(2921)),                   INJD 220
     5   (U  ,DATA(3921)),                                              INJD 230
     6                                          (IPP   ,DATA(20)),      INJD 240
     7   (OPP   ,DATA(70)),(IPR ,DATA(120)), (OPR ,DATA(170)),          INJD 250
     8   (IPT ,DATA(220)),(OPT ,DATA(270)),(TFLP,DATA(9596)),           INJD 260
     9   (TFLR,DATA(9597)),(TFLT,DATA(9598))                            INJD 270
C                                                                       INJD 280
   10 FORMAT (//12X30HRESULTS OF DESCRIBING FUNCTION // 23X10HELEMENT   INJD 290
     1 6HRADIUS4X5HANGLE3X10HFRACTIONAL /13X5HOMEGA7X3HNO.7X3H   7X3HRADINJD 300
     2 4X9HFLOW-RATE10X2HFP12X2HFR12X2HFT // )                          INJD 310
   20 FORMAT (10XF9.4,I9,F11.3,F10.4,F11.5,2X3F14.6 )                   INJD 320
   30 FORMAT (//  12X41HRESULTS OF INJECTION DISTRIBUTION EFFECTS //    INJD 330
     143X5HOMEGA6X3HAVN8X3HBVN7X3HCVN7X3HCVN / 48X3(6X4HREAL),6X4HIMAG/ INJD 340
     2/ )                                                               INJD 350
   40 FORMAT (39X5F10.4 )                                               INJD 360
   50 FORMAT ( 44X5HALL  4F10.4 )                                       INJD 370
   60 FORMAT ( // 5X39HINPUT TO INJECTION DISTRIBUTION PROGRAM//10X9HCONINJD 380
     1STANTS//14X18HNUMBER OF OMEGAS = I3//14X20HNUMBER OF ELEMENTS =I5,INJD 390
     1 13H FOR EACH OF I4, 20H SYMMETRIC SECTIONS.                      INJD 391
     2//14X22HRADIAL DIVISIONS(XM) = F5.0//14X24HANGULAR DIVISIONS (XN) INJD 400
     3=F5.0//14X27HACOUSTIC MODE NUMBER(SVN) =F7.4//14X30HORDER OF BESSEINJD 410
     4L FUNCTIONS(V) = F3.0//14X17HINJECTOR RADIUS =F8.3,5H, IN.//14X32HINJD 420
     5RATIO OF SPECIFIC HEATS(GAMMA) =F7.4//14X39HMAXIMUM PRESSURE AMPLIINJD 430
     6TUDE RATIO(P00) =F7.3///14X39HTRANSFER FUNCTIONS FOR LINEAR OPERATINJD 440
     7ION//20X16HPRESSURE(TFLP) =F7.3//20X23HRADIAL VELOCITY(TFLR) =F7.3INJD 450
     8//20X27HTANGENTIAL VELOCITY(TFLT) =F7.3 )                         INJD 460
   70 FORMAT (// 10X17HINPUT FREQUENCIES )                              INJD 470
   80 FORMAT (/ 7X 5F20.4 )                                             INJD 480
   90 FORMAT (///10X19HELEMENT INFORMATION )                            INJD 490
  100 FORMAT (//31X7HELEMENT14X6HRADIUS15X5HANGLE11X12HDISTRIBUTION/90X INJD 500
     111HCOEFFICIENT/34X3HNO.17X3HIN.15X7HRADIANS15X2HMU// )            INJD 510
  110 FORMAT (32X15,11XF10.3,10XF10.4,11XF10.4 )                        INJD 520
  120 FORMAT (1H1//26H TABULAR NONLINEAR EFFECTS//11X8HPRESSURE11X10HCOMINJD 530
     1BUSTION12X6HRADIAL12X10HCOMBUSTION10X10HTANGENTIAL10X10HCOMBUSTIONINJD 540
     2/51X8HVELOCITY32X8HVELOCITY/ 9X 3(12HPERTURBATION13X4HGAIN12X) )  INJD 550
```

II, D, Program Listing (cont.)

```
  130 FORMAT ( 6(F18.3,2X) )
  140 FORMAT (//33X65H*************  THESE VALUES PERTAIN TO A STANDING MO
     1DE  ********** )
  150 FORMAT (//33X65H*************  THESE VALUES PERTAIN TO A SPINNING MO
     1DE  ********** )
  160 FORMAT (23X42HTHE VALUE OF IZZIT IS ZERO AND NOT ALLOWED)
  170 FORMAT(23X45HVALUE OF BESSEL ARGUMENT TOO HIGH OR LOW Z = F10.4)
C
  180 DSCRB = 1
      IF ( .NOT. IRUN )     GO TO 190
      CE = 100.0
      DSCRB = 2
      GO TO 200
  190 NW = 1
      WC(1) = 1.0
  200 TIME = 1
      RINJ = DATA(9574)
      NE = DATA(9571)  +0.0001
      NS = DATA(5) + 0.0001
      NUMBR = DATA(9599)  + 0.0001
      IZZIT = DATA(2)
      K = V +0.0001
      KK = K + 1
      L=K
      M=L+1
      N=L+2
C     ONLY THE REAL PART IS USED.
      IF ( E1 .LE. 0.0 )     E1 = 0.0001
      IF ( NUMBR .LE. 0 )     NUMBR = 10
      IF ( CE .LE. 99.0 )     GO TO 260
      CALL  PAGE  ( 70 )
      WRITE (6,60) NW,NE,NS,XM,XN,SVN,V,RINJ,GAMMA,POO,TFLP,TFLR,TFLT
      GO TO (220,210), DSCRB
  210 WRITE (6,70)
      WRITE (6,80) (WC(IW), IW = 1,NW)
  220 LINKNT = 0
      DO 240 J = 1,NE
      IF (LINKNT .GT. 0) GO TO 230
      CALL  PAGE  ( 70 )
      WRITE (6,90)
      WRITE (6,100)
      LINKNT = 50
  230 WRITE (6,110) J, RR(J), THATA(J), U(J)
      LINKNT = LINKNT - 1
  240 CONTINUE
      GO TO (260,250), DSCRB
  250 WRITE (6,120)
      WRITE (6,130) (IPP(I),OPP(I),IPR(I),OPR(I),IPT(I),OPT(I),I=1,50)
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
C          RUNNING DESCRIBING FUNCTION IMPLIES THAT THE EXPANSION
C   COEFFICIENTS AVN, BVN, AND CVN ARE FUNCTIONS OF THE FREQUENCY,OMEGA.
C   THEREFORE, AN OMEGA LOOP MUST BE ESTABLISHED. OTHERWISE, THIS LOOP
C  IS GONE THROUGH ONLY ONCE.
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
  260 DO 470 IW = 1, NW
      WCR = WC(IW)
      VOO = POO/(GAMMA*WCR)
      SUMP = 0.0
      SUMR = 0.0
```

II, D, Program Listing (cont.)

```
      SUMT = 0.0                                                        INJD1130
      LINKNT = 0                                                        INJD1140
      DO 450 J = 1, NE                                                  INJD1150
      R = RR(J)/RINJ                                                    INJD1160
      ETA = U(J)/(XM*XN)                                                INJD1170
C*************,**********************************************************
C     THE ANSWERS WILL BE
C  STORED  IN THE FOLLOWING FASHION,  J(V-1) = FIRST(L),  J(V) =       INJD1230
C  FIRST(M),  AND J(V+1) = FIRST(N).                                    INJD1240
C*************,*********************************************************INJD1250
      KERR = 0                                                          INJD1260
      Z(1) = SVN*R                                                      INJD1270
      Z(2) = 0.0                                                        INJD1280
      CALL  BESSEL  ( FIRST(1), SECOND(1), KK, Z(1), KERR )             INJD1290
      IF ( KERR )      270,270,550                                      INJD1300
  270 SIVN = FIRST(M)                                                   INJD1310
      DSIVN = ( V*FIRST(M) - Z(1)*FIRST(N) )/R                          INJD1320
      Z(1) = SVN                                                        INJD1330
      Z(2) = 0.0                                                        INJD1340
      CALL  BESSEL  ( FIRST(1), SECOND(1), KK, Z(1), KERR )             INJD1350
      IF  ( KERR )      280,280,550                                     INJD1360
  280 GO TO (290,320),  TIME                                            INJD1370
  290 IF ( K )     300,300,310                                          INJD1380
  300 D = ( FIRST(M)*FIRST(M) + FIRST(L)*FIRST(L) ) / 2.0               INJD1390
      GO TO 320                                                         INJD1400
  310 D = ( FIRST(M)*FIRST(M) - FIRST(L)*FIRST(N))*3.14159/(2.0*DATA(5))INJD1410
  320 IF ( IZZIT )      330,560,340                                     INJD1420
CXXXXXXXXXXXXX,XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXINJD1430
C  IZZIT IS NEGATIVE FOR STANDING MODES AND POSITIVE FOR SPINNING MODES INJD1440
CXXXXXXXXXXXXX,XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXINJD1450
  330 VT = V*THATA(J)                                                   INJD1460
      CVT = COS(VT)                                                     INJD1470
      SVT = SIN(VT)                                                     INJD1480
      GO TO 350                                                         INJD1490
  340 CVT = 1.0                                                         INJD1500
      SVT = 1.0                                                         INJD1510
  350 FP = 1.0                                                          INJD1520
      FR = 1.0                                                          INJD1530
      FT = 1.0                                                          INJD1540
      GO TO (420,360), DSCRB                                            INJD1550
  360 PO = ABS(POO*SIVN*CVT)                                            INJD1560
      VO = ABS(VOO*DSIVN*CVT)                                           INJD1570
      WO = ABS(VOO*SIVN*V*SVT/R)                                        INJD1580
       A = -3.14159                                                     INJD1590
      B = -A                                                            INJD1600
      IF ( TFLP )      370,380,370                                      INJD1610
  370 SAVEP = 1.0/(3.14159*PO*TFLP)                                     INJD1620
      CALL INTGR(A,B,PSI,NUMBR)                                         INJD1630
 371  CONTINUE
      CPSI = COS(PSI)                                                   INJD1640
      IPX = PO*CPSI                                                     INJD1650
      CALL INT4(IPP(1), OPP(1), IPX, OPX)                               INJD1660
      F = OPX*CPSI                                                      INJD1670
      CALL INTGS($371,F,INTEG,E1,MM)
      FP = SAVEP*INTEG                                                  INJD1690
  380 IF ( TFLR )      390,400,390                                      INJD1700
  390 SAVER = 1.0/(3.14159*VO*TFLR)                                     INJD1710
      CALL INTGR(A,B,PSI,NUMBR)                                         INJD1720
 391  CONTINUE
```

II, D, Program Listing (cont.)

```
          CPSI = COS(PSI)                                               INJD1730
          IPX = VO*CPSI                                                 INJD1740
          CALL INT4 (IPR(1), OPR(1), IPX, OPX)                          INJD1750
          F = OPX*CPSI                                                  INJD1760
          CALL INTGS($391,F,INTEG,E1,MM)
          FR = SAVER*INTEG                                              INJD1780
      400 IF ( TFLT )     410,420,410                                   INJD1790
      410 SAVET = 1.0/(3.14159*WO*TFLT)                                     *NEW
           CALL INTGR(A,B,PSI,NUMBR)                                    INJD1810**-1
    411   CONTINUE
          CPSI = COS(PSI)                                               INJD1820
          IPX = WO*CPSI                                                 INJD1830
          CALL INT4 (IPT(1), OPT(1), IPX, OPX)                          INJD1840
          F = OPX*CPSI                                                  INJD1850
          CALL INTGS($411,F,INTEG,E1,MM)
          FT = SAVET*INTEG                                              INJD1870
      420 TERMP = ETA*FP*SIVN**2*CVT**2                                 INJD1880
          SUMP = SUMP + TERMP                                           INJD1890
          TERMR = ETA*FR*SIVN*DSIVN*CVT**2                              INJD1900
          SUMR = SUMR + TERMR                                           INJD1910
          TERMT = -ETA*FT*SIVN**2*CVT*V*SVT/R                           INJD1920
          SUMT = SUMT + TERMT                                           INJD1930
          TIME = 2                                                      INJD1940
          GO TO (450,430),  DSCRB                                       INJD1950
      430 IF ( LINKNT .GT. 0 )      GO TO 440                           INJD1960
          CALL  PAGE  ( 70 )                                            INJD1970
          WRITE (6,10)                                                  INJD1980
          LINKNT = 50                                                   INJD1990
      440 WRITE (6,20)  WCR, J, R, THATA(J), ETA, FP, FR, FT            INJD2000
          LINKNT = LINKNT - 1                                           INJD2010
      450 CONTINUE                                                      INJD2020
          DD = D                                                        INJD2030
          IF (IZZIT .GT. 0) DD = D/2.0                                  INJD2040
          AVN(IW) = SUMP/DD                                             INJD2050
          BVN(IW) = SUMR/DD                                             INJD2060
          IF (IZZIT .GT. 0)  GO TO 460                                  INJD2070
          CVNR(IW) = SUMT/DD                                            INJD2080
          CVNI(IW) = 0.0                                                INJD2090
          GO TO 470                                                     INJD2100
      460 CVNR(IW) = 0.0                                                INJD2110
          CVNI(IW) = SUMT/DD                                            INJD2120
      470 CONTINUE                                                      INJD2130
          IF (CE .LT. 10.0 ) GO TO 530                                  INJD2140
          CALL  PAGE  ( 70 )                                            INJD2150
          WRITE (6,30)                                                  INJD2160
          GO TO (480,490),  DSCRB                                       INJD2170
      480 WRITE (6,50)  AVN(1 ), BVN(1 ), CVNR(1 ), CVNI(1 )            INJD2180
          GO TO 500                                                     INJD2190
      490 WRITE (6,40)  ( WC(I), AVN(I), BVN(I), CVNR(I), CVNI(I),  I=1,NW )INJD2200
      500 IF ( IZZIT ) 510,560,520                                      INJD2210
      510 WRITE(6,140)                                                  INJD2220
          GO TO 530                                                     INJD2230
      520 WRITE (6,150)                                                 INJD2240
      530 CONTINUE                                                      INJD2250
      540 RETURN                                                        INJD2260
      550 WRITE (6,170) Z(1)                                            INJD2270
          GO TO 540                                                     INJD2280
      560 WRITE (6,160)                                                 INJD2290
          GO TO 540                                                     INJD2300
```

```
000233                END                                                        INJD2310
```

II, D, Program Listing (cont.)

```
000233              END                                                          INJD2310
```

II, D, Program Listing (cont.)

```
W ELT SUB20,1,700707, 29972

      SUBROUTINE TBLCAL
C TBLCAL SUBROUTINE CALCULATES XX VS U2TBL FROM NOZZLE GEOMETRY      *ANNU0030
C SIMPSONS RULE IS USED WITH KN INPUT ODD AND CHANGED TO KN/2+1 IN MAIN*ANNU0040
C  DECK ANNULR DERIVED FROM DECK VELPOT... TREATS ANNULAR NOZZLES      ANNU0050
C                                                                      ANNU0060
      LOGICAL  LOGIK,  SL1,  SL2,  EORJ                                ANNU0070
      COMMON  /PROLOG/  LOGIK(50), SL1, SL2, EORJ                      ANNU0080
C                                                                      ANNU0090
      COMMON  /ABCDF/  EXTRA(100),  ABLOK(600)                         ANNU0100
     1    ,  XX    , U2TBL  , DESIRE , RAT    , RAC    , RCC           ANNU0110
     2    ,  RCT   , HANG   , G      , KN     , Y      , YP            ANNU0120
     3    ,  YOUT  , TEMP   , E      , XTABLE , YTABLE , A             ANNU0130
     4    ,  R     , AM     , AP     , AMP    , ZZ     , AMM           ANNU0140
     5    ,  AMP2  , CALFA  , CTALFA , DELAM  , DELTZ  , FKN           ANNU0150
     6    ,  G1    , G2     , G3     , G4     , JFLAG1 , KNM1          ANNU0160
     7    ,  K     , NN     , PI     , PROD   , RSTA1  , RSTA2         ANNU0170
     8    ,  SALFA , T1     , T2     , T3     , XINT   , XK            ANNU0180
     9    ,  ZZ1   , ZZ2    , ZZ3    , A1     , ABC    , ABD           ANNU0190
      COMMON  /ABCDF/                                                  ANNU0200
     1       AI1   , ALPHAI , ALPHAR , AR1    , B101   , B102          ANNU0210
     2    ,  B10   , B1     , B2     , B3     , B4     , B5            ANNU0220
     3    ,  B6    , B7     , B8     , B91    , B92    , B9            ANNU0230
     4    ,  BI1   , BR1    , C2     , C3     , CHII   , CHIR          ANNU0240
     5    ,  CI1   , CR1    , C      , D10    , D11    , D1            ANNU0250
     6    ,  D2    , D3     , D4     , D5     , D6     , D7            ANNU0260
     7    ,  D8    , D9     , DC2    , D      , DU2    , EI            ANNU0270
     8    ,  ER    , F3I    , F3R    , FI     , FR     , H1            ANNU0280
     9    ,  H     , I      , IWO    , IW     , IWW    , J             ANNU0290
      COMMON  /ABCDF/                                                  ANNU0300
     1       MDESIR , NK    , NP     , S2     , S      , TT            ANNU0310
     2    ,  U2    , U      , W2     , W      , XI0I   , XI0R          ANNU0320
     3    ,  XI2I  , XI2R   , XI     , XJI    , XJR    , XMNEW         ANNU0330
     4    ,  XMOLD , XNEW   , XOLD   , XPT    , X      , ZI            ANNU0340
     5    ,  ZR                                                        ANNU0350
C                                                                      ANNU0360
      DIMENSION XX(200),U2TBL(200),XTABLE(200),YTABLE(200),ZZ(200)     ANNU0370
      DIMENSION Y(8),YP(8),YOUT(8),TEMP(16,8),E(8)                         *NEW
      DIMENSION A(200),R(200),AM(200),AP(200),AMP(200)                 ANNU0390**-1
C                                                                      ANNU0400
      EQUIVALENCE (XK,GRAD),(EXTRA(91),RATI),                          ANNU0410
     1   ( EXTRA(93),RACI),(EXTRA(94),RCCI),(EXTRA(95),HANGI)          ANNU0420
C                                                                      ANNU0430
      FKN = KN                                                         ANNU0440
      KNM1 = KN - 1                                                    ANNU0450
      DELAM = 1.0/(FKN+1.0)                                            ANNU0460
      PI = 3.1415927                                                   ANNU0470
C                                                                      ANNU0480
      DO 10 J = 1,200                                                  ANNU0490
      ZZ(J) = 0.0                                                      ANNU0500
      A(J) = 0.0                                                       ANNU0510
      R(J) = 0.0                                                       ANNU0520
      AMP(J) = 0.0                                                     ANNU0530
      AM(J) = 0.0                                                      ANNU0540
      XX(J) = 0.0                                                      ANNU0550
      U2TBL(J) = 0.0                                                   ANNU0560
```

II, D, Program Listing (cont.)

```
   10 AP(J) = 0.0                                                       ANNU0570
C                                                                       ANNU0580
      R(1)= RAT                                                         ANNU0590
      RI = RATI                                                         ANNU0600
      RCTI = EXTRA(92)                                                  ANNU0610
      A(1) = PI*(R(1)*R(1)-RI*RI)                                       ANNU0620
      U2TBL(1)=1.0                                                      ANNU0630
      R(KN) = RAC                                                       ANNU0640
      ALFA = HANG*.01745329                                             ANNU0650
      CALFA=COS(ALFA)                                                   ANNU0660
      SALFA=SIN(ALFA)                                                   ANNU0670
      CTALFA=CALFA/SALFA                                                ANNU0680
      RSTA1=RAT+RCT*(1.0-CALFA)                                         ANNU0690
      RSTA2=RAC-RCC*(1.0-CALFA)                                         ANNU0700
      ZZ1=RCT*SALFA                                                     ANNU0710
      ZZ2=ZZ1+CTALFA*(RSTA2-RSTA1)                                      ANNU0720
      ZZ3=ZZ2+RCC*SALFA                                                 ANNU0730
C                                                                       ANNU0740
      JFLAG1=1                                                          ANNU0750
      JFLAG2 = 1                                                        ANNU0760
      IF ( RACI .EQ. 0.0 .AND. RATI .EQ. 0.0 )  JFLAG2 = 4              ANNU0770
      ALFI = HANGI*.01745329                                            ANNU0780
      CALFI = COS(ALFI)                                                 ANNU0790
      SALFI = SIN(ALFI)                                                 ANNU0800
      CTALFI = CALFI/SALFI                                              ANNU0810
      RI1 = RATI + RCTI*(1.0-CALFI)                                     ANNU0820
      RI2 = RACI - RCCI*(1.0-CALFI)                                     ANNU0830
      ZI1 = RCTI*SALFI                                                  ANNU0840
      ZI2 = ZI1 + CTALFI*(RI2-RI1)                                      ANNU0850
      ZI3 = ZI2 + RCCI*SALFI                                            ANNU0860
      Z4 = ZZ3                                                          ANNU0870
      IF ( ZI3 .GT. ZZ3 )    Z4 = ZI3                                   ANNU0880
      DELTZ = Z4 / (FKN-1.0)                                            ANNU0890
C                                                                       ANNU0900
      DO 80 I = 2,KNM1                                                  ANNU0910
      ZZ(I) = ZZ(I-1) + DELTZ                                           ANNU0920
      Z = ZZ(I)                                                         ANNU0930
      GO TO  ( 20, 40, 60, 65 ),  JFLAG1                                ANNU0940
   20 R(I)=RAT+RCT-SQRT(RCT**2-ZZ(I)**2)                                ANNU0950
      IF(R(I)-RSTA1)70,70,30                                            ANNU0960
   30 JFLAG1=2                                                          ANNU0970
   40 R(I)=RSTA1+(RSTA2-RSTA1)*(ZZ(I)-ZZ1)/(ZZ2-ZZ1)                    ANNU0980
      IF ( Z-ZZ2 )             70,  70,  50                             ANNU0990
   50 JFLAG1 = 3                                                        ANNU1000
   60 IF ( Z-ZZ3 )             62,  65,  65                             ANNU1010
   62 R(I) = RAC - RCC + SQRT ( RCC*RCC - (ZZ3-Z)**2 )                  ANNU1020
      GO TO 70
   65 R(I) = RAC                                                        ANNU1030
      JFLAG1 = 4                                                        ANNU1040
   70 GO TO  ( 72, 74, 76, 77 ),  JFLAG2                                ANNU1050
   72 TX = SQRT ( RCTI*RCTI - Z*Z )                                     ANNU1060
      IF ( RCTI .LT. 0.0 )   TX = -TX                                   ANNU1070
      RI = RATI + RCTI - TX                                             ANNU1080
      IF ( Z-ZI1 )             78,  78,  73                             ANNU1090
   73 JFLAG2 = 2                                                        ANNU1100
   74 RI = RI1 + (RI2-RI1)*(ZZ(I)-ZI1)/(ZI2-ZI1)                        ANNU1110
      IF ( Z-ZI2 )             78,  78,  75                             ANNU1120
   75 JFLAG2 = 3                                                        ANNU1130
   76 TX = SQRT ( RCCI*RCCI - ( ZI3-Z )**2 )                            ANNU1140
```

II, D, Program Listing (cont.)

```
      IF ( RCCI .LT. 0.0 )   TX = -TX
      RI = RACI - RCCI + TX
      IF ( Z-ZI3 )           78,  77,  77
   77 RI = RACI
      JFLAG2 = 4
   78 A(I) = PI*( R(I)*R(I) - RI*RI )
   80 CONTINUE
C
      ZZ(KN) = ZZ(KNM1) + DELTZ
      A(KN) = PI*( RAC*RAC - RACI*RACI )
      AMM = 1.0+ DELAM
      G1=2.0/(G +1.0)
      G2 =(G - 1.0)/2.0
      G3 = (G + 1.0)/(2.0*G - 2.0)
      G4=1.0/G1
C
      DO 90 J = 1,KN
      AMM = AMM - DELAM
      AM(J) = AMM
      AP(J)=(A(1)/AMM)*(G1*(1.0+G2*AMM**2))**G3
   90 CONTINUE
C
      DO 100 K = 2,KN
      CALL INT4(AP(1),AM(1),A(K),AMP(K))
      AMP2=AMP(K)**2
      U2TBL(K)=(G4*AMP2)/(1.0+G2*AMP2)
  100 CONTINUE
C
      DESIRE = AMP(KN)
      XINT = 0.0
      ZZ(1)=0.0
      NN = KN - 2
      K = 1
      IF ( RCTI .EQ. 0.0 )   RCTI = 1.0
      GRAD = SQRT ( G1*(RAT/RCT - RATI/RCTI )/( RAT*RAT - RATI*RATI ) )
     1*RAT
      PROD = 2.0*XK*DELTZ/3.0
      DO 110 J = 1,NN,2
      T1 = SQRT(U2TBL(J))
      T2 = SQRT(U2TBL(J+1))
      T3 = SQRT(U2TBL(J+2))
      XINT = XINT + PROD*(T1+4.0*T2+T3)
      K = K + 1
      XX(K) = -XINT/RAT
      ZZ(K)=ZZ(K-1)+2.*DELTZ
      U2TBL(K-1) = U2TBL(J)
  110 CONTINUE
      U2TBL(K) = U2TBL(KN)
C**********
      RETURN
      END
```

II, D, Program Listing (cont.)

```
@ ELT SUB22,1,690715, 33520

      SUBROUTINE AS138 (C,ITITLE,IERR)
C     THIS ROUTINE PROVIDES A HIGHLY SIMPLIFIED INPUT PROCEDURE WHICH ISCTP  20
C     NON-CARD-COLUMN-ORIENTED.  DEVELOPED AT THE AIR FORCE ROCKET      CTP  30
C     PROPULSION LABORATORY BY RPMMD/CAPT V.L.OLIVIER/553-2819          CTP  40
C     LAST REVISED APRIL 1968                                           CTP  50
C     THE CONTROL CHARACTERS ARE DEFINED AS FOLLOWS**                   CTP  60
C          L ....SPECIFIES A LOCATION NUMBER WILL FOLLOW                CTP  70
C          + OR - AND/OR , ARE USED TO SEPARATE DATA ENTRIES            CTP  80
C          E ....SPECIFIES A POWER OF TEN WILL FOLLOW                   CTP  90
C          T ....TERMINATES THE DATA READ IN AND IS FOLLOWED BY BCD     CTP 100
C                INFORMATION THAT CAN BE PRINTED                        CTP 110
C     BLANKS AND ZEROES PRECEDING A NUMBER ARE IGNORED.                 CTP 120
C                                                                       CTP 130
C     EXAMPLES**              1         2         3         4           CTP 140
C          COLUMN 1234567890123456789012345678901234567890123456789....CTP 150
C                      L  0002300,24+-1,++E12,E-2  ,10.001   L12   + ++-1CTP 160
C                    LU+     1       ,1,      100.1E20+L50   ,,       ,  CTP 170
C                 T  (T IS THE ONLY CONTROL THAT MUST BE IN COLUMN 1)   CTP 180
C                                                                       CTP 190
C     THE RESULTS ARE STORED IN THE C-ARRAY AND ITITLE AS FOLLOWS**     CTP 200
C          C(2301)=24.                                                  CTP 210
C          C(2302)=0.                                                   CTP 220
C          C(2303)=-1.                                                  CTP 230
C          C(2304)=0.                                                   CTP 240
C          C(2305)=1.E+12                                               CTP 250
C          C(2306)=1.E-02                                               CTP 260
C          C(2307)=10.001                                               CTP 270
C          C(13)=0.                                                     CTP 280
C          C(14)=0.                                                     CTP 290
C          C(15)=0.                                                     CTP 300
C          C(16)=-1.                                                    CTP 310
C          C(1)=1.                                                      CTP 320
C          C(2)=1.                                                      CTP 330
C          C(3)=100.1E+20                                               CTP 340
C          C(51)=0.                                                     CTP 350
C          C(52)=0.                                                     CTP 360
C          C(53)=0.                                                     CTP 370
C          ITITLE=T  (T IS THE ONLY CONTROL THAT MUST BE IN COLUMN 1)   CTP 380
C                                                                       CTP 390
      DIMENSION C(1),DCARD(72),SYMB(17),XNUM(16)
      DIMENSION IDCARD(72), ITITLE(12), MULT1(6), MASK1(6)              CTP1400
      EQUIVALENCE (IDCARD(1),DCARD(1))                                  CTP1410
      DATA MULT1/01,0100,010000,01000000,0100000000,010000000000/       CTP1420
      DATA MASK1/0,04000000000,040000000,0400000,04000,040/             CTP1430
      DATA MASK/0770000000000/,BLNK/6H 00000/                           CTP1440
      LOGICAL NUMBER,DECMAL,EXPON,LOCATE                                CTP 410
      DATA BLANK,TCARD,SYMB/1H ,1HT,1H1,1H2,1H3,1H4,1H5,1H6,1H7,1H8,1H9,CTP 420
     11H0,1H+,1H-,1H,,1H.,1HE,1HL,1H8/                                       *NEW
      IERR=1                                                            CTP 440
      NUMBER=.TRUE.                                                     CTP 450
      DECMAL=.FALSE.                                                    CTP 460
      EXPON=.FALSE.                                                     CTP 470
      LOCATE=.FALSE.                                                    CTP 480
      XSIGN=0.0                                                         CTP 490
      LOCN=0                                                            CTP 500
```

```
C                                                                       CTP 510
   10 READ (5,240) DCARD                                                CTP 520
      IF (DCARD(1).EQ.TCARD) GO TO 200                                  CTP 530
      DO 190 I=1,73                                                     CTP 540
      IF (I.EQ.73) GO TO 80                                             CTP 550
      IF (DCARD(I).EQ.BLANK) GO TO 190                                  CTP 560
      DO 20 J=1,17
      IF (DCARD(I).EQ.SYMB(J)) GO TO (40,40,40,40,40,40,40,40,40,30,100,CTP 580 **-1
     160,80,50,70,90,100),J
   20 CONTINUE                                                          CTP 600
      GO TO 230                                                         CTP 610
C                                                                       CTP 620
   30 J=0                                                               CTP 630
      IF (IDIG.EQ.0) GO TO 190                                          CTP 640
C                                                                       CTP 650
   40 IDIG=IDIG+1                                                       CTP 660
      IF (.NOT.DECMAL) NDEC=IDIG-1                                      CTP 670
      XNUM(IDIG)=J                                                      CTP 680
      GO TO 190                                                         CTP 690
C                                                                       CTP 700
   50 IF (DECMAL) GO TO 230                                             CTP 710
      DECMAL=.TRUE.                                                     CTP 720
      IF (IDIG.GT.0) GO TO 190                                          CTP 730
      IDIG=1                                                            CTP 740
      XNUM(1)=0.0                                                       CTP 750
      GO TO 190                                                         CTP 760
C                                                                       CTP 770
   60 KSIGN=-1                                                          CTP 780
      GO TO 100                                                         CTP 790
C                                                                       CTP 800
   70 IF (EXPON) GO TO 230                                              CTP 810
      EXPON=.TRUE.                                                      CTP 820
      NUMBER=.FALSE.                                                    CTP 830
C                                                                       CTP 840
   80 KSIGN=0                                                           CTP 850
      GO TO 110                                                         CTP 860
C                                                                       CTP 870
   90 IF (LOCATE) GO TO 230                                             CTP 880
      LOCATE=.TRUE.                                                     CTP 890
      KSIGN=1                                                           CTP 900
      IF (XSIGN.EQ.0.0.AND.IDIG.EQ.0) GO TO 180                         CTP 910
      NUMBER=.FALSE.                                                    CTP 920
      GO TO 110                                                         CTP 930
C                                                                       CTP 940
  100 IF (XSIGN.EQ.0.0.AND.IDIG.EQ.0) GO TO 180                         CTP 950
C                                                                       CTP 960
  110 X=0.0                                                             CTP 970
      IF (IDIG.EQ.0) GO TO 130                                          CTP 980
      DO 120 K=1,IDIG                                                   CTP 990
      X=X+10.**NDEC*XNUM(K)                                             CTP1000
  120 NDEC=NDEC-1                                                       CTP1010
C                                                                       CTP1020
      IDIG=0                                                            CTP1030
      NDEC=0                                                            CTP1040
      DECMAL=.FALSE.                                                    CTP1050
C                                                                       CTP1060
      IF (XSIGN.LT.0.0) X=-X                                            CTP1070
  130 IF (.NOT.NUMBER) GO TO 140                                        CTP1080
      IF (LOCATE) GO TO 160                                             CTP1090
```

II, D, Program Listing (cont.)

```
      IF (EXPON) GO TO 170                                              CTP1100
      GO TO 150                                                         CTP1110
  140 NUMBER=.TRUE.                                                     CTP1120
      ESIGN=XSIGN                                                       CTP1130
      IF (ESIGN.EQ.0.0) ESIGN=1.0                                       CTP1140
C                                                                       CTP1150
C                                                                       CTP1160
  150 LOCN=LOCN+1                                                       CTP1170
      C(LOCN)=X                                                         CTP1180
      GO TO 180                                                         CTP1190
C                                                                       CTP1200
  160 LOCN=ABS(X)                                                       CTP1210
      LOCATE=.FALSE.                                                    CTP1220
      GO TO 180                                                         CTP1230
C                                                                       CTP1240
  170 EXPON=.FALSE.                                                     CTP1250
      IF (C(LOCN).EQ.0.0) C(LOCN)=ESIGN                                 CTP1260
      C(LOCN)=C(LOCN)*10.**IFIX(X)                                      CTP1270
C                                                                       CTP1280
  180 XSIGN=KSIGN                                                       CTP1290
      KSIGN=1                                                           CTP1300
C                                                                       CTP1310
  190 CONTINUE                                                          CTP1320
C                                                                       CTP1330
      GO TO 10                                                          CTP1340
C                                                                       CTP1350
C                                                                       CTP1360
C     THE FOLLOWING LOGIC IS TO PACK THE 'DCARD' DATA INTO 'ITITLE'     CTP1370
C     FOR A 'FORMAT(1X,12A6)' PRINTOUT.                                 CTP1380
C                                                                       CTP1390
  200 DCARD(1)=BLNK                                                     CTP1450
      L=0                                                               CTP1460
      DO 220 I=1,72,6                                                   CTP1470
      L=L+1                                                             CTP1480
      KK=I-1                                                            CTP1490
      DO 210 J=1,6                                                      CTP1500
      K=KK+J                                                            CTP1510
      DCARD(K)=AND(IDCARD(K),MASK)                                      CTP1520
      IF (J.EQ.1) GO TO 210                                             CTP1530
      IDCARD(K)=IDCARD(K)/MULT1(J)                                      CTP1540
      IF (ISIGN(1,IDCARD(K)).LT.0) IDCARD(K)=IABS(IDCARD(K))+MASK1(J)   CTP1550
  210 CONTINUE                                                          CTP1560
  220 ITITLE(L)=IDCARD(I)+ISIGN((IDCARD(I+1)+IDCARD(I+2)+IDCARD(I+3)+IDCCTP1570
     1ARD(I+4)+IDCARD(I+5)),IDCARD(I))                                  CTP1580
      RETURN                                                            CTP1590
C                                                                       CTP1600
  230 IERR=2                                                            CTP1610
      WRITE (6,250) DCARD                                               CTP1620
      GO TO 10                                                          CTP1630
C                                                                       CTP1640
C                                                                       CTP1650
  240 FORMAT (72A1)                                                     CTP1660
  250 FORMAT (1X,72A1)                                                  CTP1670
      END                                                               CTP1680-
```

II, D, Program Listing (cont.)

```
@ ELT SU324,1,690708, 37499

          SUBROUTINE  INTGR (A,B,X,N)
          X=A
          I=1
          NL=128*N
          RETURN
          ENTRY INTGS ($,F,Y,E,M)
          GO TO (1,2,3),I
C         FIRST TIME
  1       X=B
          F0 = F
          I=2
          RETURN1
C
  2       FEND = F+F0
          N= ((N+3)/4)*4
  20      DX = (B-A) / N
          F1=0.
          F2=0.
          F3=0.
          F4=0.
          K=1
          I=3
          X=A+ DX
          RETURN1
C
  3       KF = MOD(K - 1,4)+1
          GO TO (10,11,12,13),KF
  10      F1=F1 +F
          GO TO 14
  11      F2=F2 +F
          GO TO 14
  12      F3=F3 +F
          GO TO 14
  13      F4=F4 +F
  14      X=X+DX
          K=K+1
          IF(K.NE.N ) RETURN1
          SUM1 = DX/3.*(FEND+ 2.*(F2+F4) + 4.*(F1+F3))
          SUM2 = 2*DX/3.*(FEND+ 2.*F4+4.*F2)
          ER=  (SUM1-SUM2) /15.
          IF(ABS(ER/SUM1).LT.E) GO TO 15
          IF(N.GT.NL) GO TO 15
          N=2*N
          GO TO 20
  15      Y=SUM1 + ER
          M=N
          RETURN
          END
```

```
W ELT SUB25,1,700717, 49712

      SUBROUTINE DISTR (BIN, XOUT, CA, KER )
C                                                                       HYMN  30
C     CALCULATES COMBUSTION PARAMETERS  HR, HI                          HYMN  40
C        PROGRAM BY  LW VERNON  FROM  11MAY67 ANALYSIS OF  AJ SMITH, JR HYMN  50
C           INCORPORATES CORRECTIONS TO ANALYSIS THRU  1 JUN 67         HYMN  60
C           BOOLE INTEGRATION AS OF  12 JUNE 67                         HYMN  70
C           RHO, RHOL CONVENTIONS CORRECTED  22JUN 67                   HYMN  80
C           20 SEP 67  MODIFIED FOR TABULAR INJECTOR COEFFICIENTS       HYMN  90
C                                                                       HYMN 100
C********************************************************************** HYMN 110
C                                                                       HYMN 120
      LOGICAL  LOGIK, HRUN, CKOUT, SIMPL, KNOT, LIMIT, TABLR            HYMN 140
      COMPLEX E,SIGMA,Q1SI,Q1CI,Q2SI,Q2CI,H,OMEGC,SWC,CSQRT                 *NEW
C                                                                       HYMN 150
      COMMON  /PROLOG/  LOGIK(38), HEAD(12), SL1, SL2, EORJ             HYMN 160
      COMMON  /ABCDF / DIN (4300)                                       HYMN 170
      DIMENSION          B     ( 240 ) ,  ZZ    ( 102 ) ,
     1   U     ( 102 ) ,  DU    (102 ) ,  UL    ( 102 ) ,
     2   RHO   ( 102 ) ,  RHOL  (102 ) ,  ZING (    6 ) ,
     3   QBAR  ( 102 ) ,  SIMP (    6 ) ,  ZIP3 ( 102 ) ,
     4   ERRZ  (    6 ) ,  ZIP5 ( 102 ) ,  ZF    (    6 ) ,
     5   THR   (   30 ) ,  THI   (   30 ) ,  BIN   ( 240 ) ,
     6   XOUT  ( 100 ) ,  WC    (   28 ) ,  WET   (   30 ) ,
     7   ERT   (   30 ) ,  EIT   (   30 ) ,  ZDIST(   20 ) ,
     8   DISTM(   20 )
      EQUIVALENCE                                                       HYMN 360
     1   ( HRUN , LOGIK( 8)),( CKOUT, LOGIK(13)),( SIMPL, LOGIK(14)),   HYMN 370
     2   ( KNOT , LOGIK(15)),( LIMIT, LOGIK(16)),( TABLR, LOGIK(17))    HYMN 380
      EQUIVALENCE                                                       HYMN 390
     1   ( B(1) , SNH   ),( B( 10) , ZINC  ),( DIN( 14) , RCH   ),      HYMN 400
     2   ( B(2) , ZE    ),( B( 18) , XNE   ),( UL(1)    , ULO  ),       HYMN 410
     3   ( B(3) , GAM   ),( B( 19) , XNW   ),( RHOL(1)  , RHOLO ),      HYMN 420
     4   ( B(4) , UE    ),( B(170) , WC    ),( B(20)    , WET   ),      HYMN 430
     5   ( B(5) , SOUND ),( B(200) , ZDIST ),( B(50)    , ERT   ),      HYMN 440
     6   ( B(6) , ULM   ),( B(220) , DISTM ),( B(80)    , EIT   ),      HYMN 450
     7   ( B(7) , XK    ),( B(8) , XCMPL )
C********************************************************************** HYMN 550
C                                                                       HYMN 560
      NAMELIST /TABS/SWC,SNH,W,OMEGC,E,SIGMA,Q1SI,Q1CI,Q2SI,Q2CI,H          *NEW
   10 FORMAT ( / 65H TRANSVERSE STABILITY PROGRAM... CALCULATES  HR, HI HYMN 570
     1                    )
   20 FORMAT ( // 11H INPUT DATA// 9X3HSNH12X2HZE10X5HGAMMA9X2HUE9X     HYMN 590
     1   14HSOUND (FT/SEC) 2X13H ULM (FT/SEC) 5X 9HXK (DRAG)4X11HXCOMPL HYMN 600
     2(IN)  //   F15.8, F14.8, F12.5, F14.8, F16.2, 2X 3F15.6   //      HYMN 610
     3 5X10HINCREMENTS/ 5XF8.1//)
   30 FORMAT ( // 25H NOZZLE ADMITTANCES INPUT//  15X9HOMEGA(CH) 12X    HYMN 630
     1   3HERT 17X3HEIT//)
   40 FORMAT ( ////22H CALCULATED RESULTS... 10X 29HFIRST-ORDER SOLN ( UHYMN 650
     1NIF INJ )// 17X5HOMEGA 12X6HH REAL 14X6HH IMAG )
   50 FORMAT ( 9XF15.7,2E20.8 )                                         HYMN 680
   60 FORMAT ( 9XF15.7,2E20.8)
  100 FORMAT ( 12H0INTEGRALS   6E18.8 )                                 HYMN 730
  110 FORMAT ( 6H ERROR 6X6E18.8 )                                      HYMN 740
C                                                                       HYMN 840
C********************************************************************** HYMN 850
```

II, D, Program Listing (cont.)

```
C***********************************************************************HYMN 860
      CALL   DVCHK   ( KCHK )                                           HYMN 880
      NWI = DIN(3408) + 0.01                                            HYMN 890
      TABLR = NWI .GE. 3                                                HYMN 900
      DO 170 I = 1, 240                                                 HYMN 910
 170     B(I) = BIN(I)                                                  HYMN 920
 180  CONTINUE
 190  CALL   PAGE   (  70 )                                             HYMN 940
      KOUT = 2                                                          HYMN 950
      NET = XNE                                                         HYMN 960
      WRITE (6,10)                                                      HYMN 970
      IF ( CA-100.0)       220,200,200
 200  WRITE (6,20)   ( B(I), I=1,8 ),ZINC
      WRITE (6,30)                                                      HYMN1000
      WRITE (6,60)   ( WET(I), ERT(I),EIT(I),I=1,NET )
      CKOUT = ( CA .GT. 200.0 )
C                                                                       HYMN1100
C     GENERATE NORMALIZED DISTRIBUTION AND OTHER TABULAR FUNCTIONS      HYMN1110
C                                                                       HYMN1120
 220  ZC  = XCMPL / RCH                                                 HYMN1130
      ZC = ZE                                                           HYMN1140
      IDZ = ZINC + 0.001                                                HYMN1150
      CALL   INT4   ( ZDIST, DISTM, ZC, UENRM )                         HYMN1160
      SCALE = UE / UENRM                                                HYMN1170
      SIMPL = IDZ .LE. 0                                                HYMN1180
      IF ( SIMPL )             GO TO 230                                HYMN1190
      INT = 2                                                           HYMN1200
      IDZ = IDZ/4 * 4                                                   HYMN1210
C           MUST BE POSITIVE MULTIPLE OF FOUR,  LESS THAN  101          HYMN1220
      GO TO 240                                                         HYMN1230
C                                                                       HYMN1240
 230  INT = 1                                                           HYMN1250
      IDZ = -IDZ/2 * 2                                                  HYMN1260
C                                                                       HYMN1270
 240  IF ( ( IDZ .EQ. 0 ) .OR. ( IDZ .GT. 100 ) )  IDZ = 80            HYMN1280
C                                                                       HYMN1290
C         IDZ  IS NUMBER OF Z-INCREMENTS.                               HYMN1300
C                                                                       HYMN1310
      IDZP = IDZ + 1                                                    HYMN1320
      DZ  = ZC / FLOAT(IDZ)                                             HYMN1330
C                                                                       HYMN1340
      ZZ(1) = 0.0                                                       HYMN1350
      U(1)  = 1.0E-10                                                   HYMN1360
      DU(1) = 0.0                                                       HYMN1370
      RHO(1)= 1.0                                                       HYMN1380
      ULO   = ULM/SOUND                                                 HYMN1390
      GF1 = -1.0 / (GAM-1.0)                                            HYMN1400
      GF2 = (GAM-1.0) / 2.0                                             HYMN1410
      RHOZE = ( 1.0 + GF2*UE*UE ) **GF1                                 HYMN1420
      RHOLO = RHOZE * UE / ULO                                          HYMN1430
      QBAR(1) = 0.0                                                     HYMN1440
      ZIP3(1) = 0.0                                                     HYMN1450
      ZIP5(1) = RHOLO                                                   HYMN1460
C                  ABOVE ARE FIRST TABULAR ENTRIES.                     HYMN1470
C                                                                       HYMN1480
      Z = 0.0                                                           HYMN1490
C                                                                       HYMN1500
      DO 250 IZ = 2, IDZP                                               HYMN1510
         Z = Z + DZ                                                     HYMN1520
```

II, D, Program Listing (cont.)

```
         CALL  INT4D  ( ZDIST, DISTM, Z, U(IZ), DU(IZ) )                HYMN1530
         ZZ(IZ) = Z                                                     HYMN1540
         U (IZ) =  U(IZ)*SCALE                                          HYMN1550
         DU(IZ) = DU(IZ)*SCALE                                          HYMN1560
         UL(IZ) = UL(IZ-1) + XK*DZ*( U(IZ-1)-UL(IZ-1) )/UL(IZ-1)        HYMN1570
         TEMP = 1.0 + GAM*( U(IZ)-UL(IZ) )*U(IZ)                        HYMN1580
         RHO(IZ) = ( 1.0 + GF2*U(IZ)*U(IZ) ) **GF1                      HYMN1590
         RHOL(IZ) = ( RHOZE*UE - RHO(IZ)*U(IZ) ) / UL(IZ)               HYMN1600
     0   QBAR(IZ) = ( ( 1.0 - GAM*U(IZ)*U(IZ) )*RHO(IZ)*DU(IZ)          HYMN1610
     1              - GAM*U(IZ)*RHOL(IZ)*XK*( U(IZ)-UL(IZ) ) ) / TEMP   HYMN1620
         ZIP3(IZ) = DU(IZ) + DU(IZ)                                     HYMN1630
         ZIP5(IZ) = RHOL(IZ) / RHO(IZ)                                  HYMN1640
 250     CONTINUE                                                       HYMN1650
      ZZ(IDZP) = ZC                                                     HYMN1660
      U (IDZP) = UE                                                     HYMN1670
      DU(IDZP) = 0.0                                                    HYMN1680
      RHO(IDZP) = RHOZE                                                 HYMN1690
      RHOL(IDZP) = 0.0                                                  HYMN1700
      QBAR(IDZP) = 0.0                                                  HYMN1710
      ZIP3(IDZP) = 0.0                                                  HYMN1720
      ZIP5(IDZP) = 0.0                                                  HYMN1730
C                                                                       HYMN1740
C     QBAR, ZIP3, ZIP5  AS TABULATED ABOVE ARE FREQ-INDEPENDENT PARTS   HYMN1750
C        OF INTEGRANDS.                                                 HYMN1760
C                                                                       HYMN1770
      NW  = XNW                                                         HYMN2290
C                                                                       HYMN2300
C********************************************************************** HYMN2310
C                                                                       HYMN2320
C     CALCULATIONS FOR EACH FREQUENCY, W                                HYMN2330
C                                                                       HYMN2340
C********************************************************************** HYMN2350
C                                                                       HYMN2360
      DO 820 IW = 1, NW                                                 HYMN2370
         W = WC(IW)                                                     HYMN2380
         OMEG2 = SNH*SNH - W*W                                          HYMN2390
         OMEG = SQRT(ABS(OMEG2))                                        HYMN2400
         LIMIT = .FALSE.                                                HYMN2410
C                                                                       HYMN2420
         CALL  INT4  ( WET, ERT, W, ER )                                HYMN2430
         CALL  INT4  ( WET, EIT, W, EI )                                HYMN2440
         IF ( OMEG2 )          280,290,300                              HYMN2450
 280     IM = 1                                                         HYMN2460
         GO TO 310                                                      HYMN2470
C                                                                       HYMN2480
 290     IM = 2                                                         HYMN2490
        OMEG = 0.5E-10                                                  HYMN2500
        OMEG2 = 0.25E-20                                                HYMN2510
      GO TO 310                                                              *NEW
C                                                                       HYMN2520
 300     IM = 3                                                         HYMN2530
C                                                                       HYMN2540
C    EVALUATION OF SIX INTEGRALS  BY  BOOLE  FORMULA  ( OR SIMPSON )    HYMN2550
C     BOOLE INTEGRATION IF  ZINC INPUT POSITIVE, ELSE SIMPSON RULE      HYMN2560
C        ( BOOLE IS SIMPSON RULE WITH ERROR FORMULA )                   HYMN2570
C                                                                       HYMN2580
C        IM  IS  ( 1, 2, 3 )  AS  OMEG2  IS  ( -, 0, + ),               HYMN2590
C        THEN USE  ( CIRCULAR, LIMIT, HYPERBOLIC )  FUNCTIONS           HYMN2600
C                                                                       HYMN2610
```

II, D, Program Listing (cont.)

```
  310 DO 320 I = 1, 6                                                   HYMN2620
  320    ZING(I) = 0.0                                                  HYMN2630
      NOD = 3                                                           HYMN2640
      DO 430 IZ = 1, IDZP, INT                                          HYMN2650
         Z = ZZ(IZ)                                                     HYMN2660
         GO TO (350,340,330),  NOD                                      HYMN2670
C                                                                       HYMN2680
  330    WT = 1.0                                                       HYMN2690
         NOD = 1                                                        HYMN2700
         GO  TO 360                                                     HYMN2710
C                                                                       HYMN2720
  340    WT = 2.0                                                       HYMN2730
         IF ( IZ .EQ. IDZP )   GO TO 330                                HYMN2740
         NOD = 1                                                        HYMN2750
         GO TO 360                                                      HYMN2760
C                                                                       HYMN2770
  350    WT = 4.0                                                       HYMN2780
         NOD = 2                                                        HYMN2790
C                                                                       HYMN2800
  360    PSI = OMEG * (ZE-Z)                                            HYMN2810
         GO TO (390,370,380),  IM                                       HYMN2820
C                                                                       HYMN2830
  370    ZEZ = ZE - Z                                                   HYMN2840
         ZF(1) = QBAR(IZ) * ZEZ                                         HYMN2850
         ZF(2) = QBAR(IZ)                                               HYMN2860
         ZF(3) = ZIP3(IZ) * ZEZ                                         HYMN2870
         ZF(4) = ZIP3(IZ)                                               HYMN2880
         ZF(5) = ZIP5(IZ) * ZEZ                                         HYMN2890
         ZF(6) = ZIP5(IZ)                                               HYMN2900
         GO TO 410                                                      HYMN2910
C                                                                       HYMN2920
  380    CF = COSH(PSI)                                                 HYMN2930
         SF = SINH(PSI)                                                 HYMN2940
         GO TO 400                                                      HYMN2950
C                                                                       HYMN2960
  390    CF = COS (PSI)                                                 HYMN2970
         SF = SIN (PSI)                                                 HYMN2980
C                                                                       HYMN2990
  400    ZF(1) = QBAR(IZ) * SF                                          HYMN3000
         ZF(2) = QBAR(IZ) * CF                                          HYMN3010
         ZF(3) = ZIP3(IZ) * SF                                          HYMN3020
         ZF(4) = ZIP3(IZ) * CF                                          HYMN3030
         ZF(5) = ZIP5(IZ) * SF                                          HYMN3040
         ZF(6) = ZIP5(IZ) * CF                                          HYMN3050
C     ZF ARE COMPLETE INTEGRANDS                                        HYMN3060
C                                                                       HYMN3070
  410    DO 420 I = 1, 6                                                HYMN3080
            ZING(I) = ZING(I) + WT*ZF(I)                                HYMN3090
  420       CONTINUE                                                    HYMN3100
  430    CONTINUE                                                       HYMN3110
C                                                                       HYMN3120
      GO TO (460,440),  INT                                             HYMN3130
C                                                                       HYMN3140
  440 INT = 1                                                           HYMN3150
      DO 450 I = 1, 6                                                   HYMN3160
  450    SIMP(I) = ZING(I)*DZ/1.5                                       HYMN3170
C              SIMP IS SIMPSON INTEGRAL FOR IDZ/2 INCREMENTS            HYMN3180
      GO TO 310                                                         HYMN3190
C                                                                       HYMN3200
```

II, D, Program Listing (cont.)

```
  460    DO 470 I = 1, 6                                                HYMN3210
  470       ZING(I) = ZING(I)*DZ/3.0                                    HYMN3220
C                ZING IS SIMPSON INTEGRAL FOR  IDZ  INCREMENTS          HYMN3230
C                                                                       HYMN3240
      INT = 2                                                           HYMN3260**-1
         DO 480 I = 1, 6                                                HYMN3270
            ERRZ(I) = ( ZING(I)-SIMP(I) ) / 15.0                        HYMN3280
            IF ( ABS(ERRZ(I)/ZING(I) ) .GT. 0.05 )  CKOUT = .TRUE.      HYMN3290
C     ZING, SIMP ARE SIMPSON INTEGRALS WITH  IDZ, IDZ/2  INCREMENTS     HYMN3300
            ZING(I) = ZING(I) + ERRZ(I)                                 HYMN3310
C     CORRECTED  ZING  IS NOW  BOOLE INTEGRAL                           HYMN3320
  480       CONTINUE                                                    HYMN3330
C                                                                       HYMN3340
C        ABOVE  ZING(I)  ARE THE REQUIRED INTEGRALS                     HYMN3350
C                                                                       HYMN3360
      SWC=SNH*SNH-W*W                                                           *NEW
      OMEGC=CSQRT(SWC)                                                          *NEW
      E = CMPLX(ER,EI)                                                          *NEW
      SIGMA=CMPLX(0.,W)                                                         *NEW
      Q1SI=SIGMA*(ZING(3)+(GAM-1.)*ZING(1)+XK*SIGMA/(XK+SIGMA)*ZING(5))         *NEW
      Q1CI=SIGMA*(ZING(4)+(GAM-1.)*ZING(2)+XK*SIGMA/(XK+SIGMA)*ZING(6))         *NEW
      Q1CI=-Q1CI                                                                *NEW
      Q1SI=-Q1SI                                                                *NEW
      Q2SI=SIGMA*GAM/2.*ZING(3)                                                 *NEW
      Q2CI=SIGMA*GAM/2.*ZING(4)                                                 *NEW
      H=(OMEGC*OMEG*CSINH(OMEGC*ZE)-(0.,1.)*OMEG*W  *E*CCOSH(OMEGC*ZE)          *NEW
     1-OMEG*Q1CI+(0.,1.)*W  *E*Q1SI)/(OMEG*Q2CI-(0.,1.)*W  *E*Q2SI)             *NEW
      GO TO (510,500,510),IM                                                    *NEW
  500 H=(-(0.,1.)*W  *E-Q1CI+(0.,1.)*W  *E*Q1SI)/(Q2CI-(0.,1.)*W                *NEW
     1*E*Q2SI)                                                                  *NEW
  510 CONTINUE                                                                  *NEW
      HR=REAL(H)                                                                *NEW
      HI=AIMAG(H)                                                               *NEW
      WRITE(6,TABS)                                                             *NEW
C                                                                       HYMN3880*-50
         THR(IW) = HR                                                   HYMN3890
         THI(IW) = HI                                                   HYMN3900
C                                                                       HYMN3910
  820    CONTINUE                                                       HYMN7090
C                         END  OF  W-LOOP                               HYMN7100
C********************************************************************** HYMN7110
C                                                                       HYMN7120
      KER = NW                                                          HYMN7130
      XOUT(9) = XNW                                                     HYMN7140
C                                                                       HYMN7150
C                                                                       HYMN7170
      DO 830 IW = 1, NW                                                 HYMN7180
         XOUT(IW+ 9) = WC (IW)                                          HYMN7190
         XOUT(IW+39) = THR(IW)                                          HYMN7200
         XOUT(IW+69) = THI(IW)                                          HYMN7210
  830    CONTINUE                                                       HYMN7220
      GO TO 840
  840 IF ( CKOUT )     CALL PAGE  ( 70 )                                HYMN7240
      WRITE (6,40)                                                      HYMN7250
      WRITE (6,50)  ( WC(I), THR(I), THI(I),  I = 1, NW  )              HYMN7260
  850 CONTINUE
C     HR, HI ARE FINAL RESULTS OF THIS SUBROUTINE. THEY DO NOT INCLUDE
C     ANY INJECTION DISTRIBUTION EFFECTS.  TO GET N AND TAU GO TO DDD
C     THEN FFF.
```

II, D, Program Listing (cont.)

```
C                                                                       HYMN7490
C                                                                       HYMN7500
C                                                                       HYMN7510
C     N.B.   HTR, HTI  ARE FINAL RESULTS OF THIS SUBROUTINE WHEN PLACED  HYMN7520
C            IN XOUT.   HTR, HTI INCLUDE INJECTOR EFFECTS, WHILE  HR, HI HYMN7530
C            DO NOT.  HENCE IF SUBROUTINE  DDD  IS ENTERED WITH HTR, HTI,HYMN7540
C            DDD  WILL EXPAND TABLE (BY INTERPOLATION), PRINT,          HYMN7550
C            AND RETURN.                                                HYMN7560
C                                                                       HYMN7570
C                                                                       HYMN7580
C                                                                       HYMN7590
  890 RETURN                                                            HYMN7600
      END                                                               HYMN7610
```

II, D, Program Listing (cont.)

```
@ ELT SUB26,1,700610, 37330

      SUBROUTINE ADSET(N,F,D,FP,T,X,HA,E)
      IMPLICIT REAL*8(A-H,O-Z)
      REAL F,D,FP,X,HA,E,TP
      DIMENSION F(1),D(1),FP(1),T(8,8),HA(1),E(1),AC(6,3),AD(10),AH(10) 00000030
     1,ACC(6),ER(60)
      EQUIVALENCE (ACC(1),AC(13))                                       00000060
      DATA AC/ 0.D0,1.D0,0.5D0,0.4166666667D0,0.375D0,2*0.3486111111D0, 00000070
     1  0.6513888889D0,0.1513888889D0,0.0680555556D0,0.0263888889D0,   00000080
     2  0.D0,0.D0,0.D0,0.D0,0.D0,0.D0,0.D0/                             00000090
      DATA AD  /2.0D0,-1.0D0,0.D0,0.D0,4.0D0,-4.0D0,1.0D0,8.0D0,-12.0D0,00000100
     1   16.0D0/                                                        00000110
      DATA AH /0.5D0,0.125D0,0.0625D0,0.0390625D0,0.25D0,0.125D0,       00000120
     1   0.078125D0,0.125D0,0.09375D0,0.0625D0/                         00000130
      DATA CON1,CON2/  0.9462151D0,0.0537848D0/                         00000140
C                                                                       00000150
C**** THIS SUBROUTINE WILL INTEGRATE N DIFFERINTIAL EQUATIONS.           00000160
C    WHERE N MAY BE NO LARGER THAN THE SUBSCRIPT ON ER                  00000170
C**** THE CALLING SEQUENCE IS AS FOLLOWS.                               00000180
C****    N IS THE NUMBER OF EQUATIONS                                   00000190
C****    F IS THE ARRAY OF FUNCTIONS                                    00000200
C****    D IS THE ARRAY OF DERIVATIVES OF THE FUNCTIONS F               00000210
C****    FP IS THE ARRAY OF THE PARTIAL STEP VALUES OF F AT X = TP      00000220
C****    T IS AN ARRAY OF 8*N WORDS                                     00000230
C****    HA IS EITHER AN ARRAY OR A SINGLE WORD                         00000240
C****       IN EITHER CASE HA(1) IS THE INITIAL STEP SIZE GUESS         00000250
C****       IF THE ARRAY FEATURE IS USED THE FOLLOWING CONDITIONS HOLD. 00000260
C****       IF HA(2) IS EQUAL TO 1111 THEN STEP SIZE IS LIMITED         00000270
C****       HA(3) IS THE LOWEST VALE OF THE STEP SIZE ALLOWED           00000280
C****       HA(4) IS THE LARGEST VALUE OF THE STEP SIZE ALLOWED         00000290
C****       IF HA(2) IS EQUAL TO 2222 BOTH THE LIMIT AND THE CORRECTED  00000300
C****          DERIVATIVES ARE USED                                     00000310
C****       IF HA(2) IS EQUAL TO 3333 ONLY THE CORRECTED DERIVATIVES ARE00000320
C****       IF HA(2) IS NOT DEFINED THEN NEITHER ARE USED               00000330
      KKF=0                                                             00000340
C**** SET FOR NO RECAL OF DERIVATIVES                                   00000350
      CALL ELT1
      H=HA(1)                                                           00000360
      HMIN=0.D0                                                         00000370
      HMAX=1.D30                                                        00000380
      IF(HA(1).LT.0.D0)  HMIN=-HMAX                                     00000390
      IF(HA(1).LT.0.D0)  HMAX=0.D0                                      00000400
C**** SET MAX-MIN STEP SIZE                                             00000410
      IF(HA(2).EQ.1111.D0) GO TO 400                                    00000420
      IF(HA(2).EQ.2222.D0) GO TO 401                                    00000430
      IF(HA(2).EQ.3333.D0) KKF=1                                        00000440
  403 GO TO 4                                                           00000450
  400 HMIN=HA(3)                                                        00000460
      HMAX=HA(4)                                                        00000470
      GO TO 403                                                         00000480
  401 KKF=1                                                             00000490
      GO TO 400                                                         00000500
C                                                                       00000510
C                                                                       00000520
      ENTRY ADINT                                                       00000530
      IENT=1                                                            00000540
```

II, D, Program Listing (cont.)

```
      IDF=0
      IF(INT.EQ.0) GO TO 20
   21 KF=1
      X=X+H
      GO TO  100
20    H=HA(1)
506   DO 504 I=1,N
      IF(E(I).LT.1.D-9) E(I)=1.D-9
504   CONTINUE
  520 IF(IENT.NE.0) RETURN
      RETURN1
C
      ENTRY ADCOR(*)
      IENT=0
2     IF(INT.EQ.0) GO TO 23
22    IF(KKF.EQ.2) GO TO 34
      DO 41 I=1,N
      T(8,I)= F(I)
   41 T(1,I)=D(I)
      KF=2
      GO TO 100
C
   23 INT=1
      DO 40 I=1,N
      T(7,I) =F(I)
   40 T(2,I) =D(I)
      GO TO 21
C
C     TEST ERROR TERM HERE.
   42 CONTINUE
      IDF=2
      DO  16 I=1,N
      BOT=F(I)
      IF(BOT.EQ.0.D0) BOT=1.D0
      ERT=T(8,I)-F(I)
      IF(NS.EQ.6) ERT=ERT+CON1*ER(I)
      ER(I)=ERT
      ERR=DABS(ER(I)/BOT)
      IF(NS.EQ.6)  F(I) =F(I) + CON2*ER(I)
      IF(ERR.LT.E(I)) GO TO 18
   19 IDF =1
      GO TO 500
18    IF(128.D0*ERR.LT.E(I)) GO TO 16
33    IDF=0
16    CONTINUE
  340 IF(KKF.NE.0) GO TO 404
C     NEW DIFFERENCES
34    IF(NS.GT.5) NS=5
   12 DO 17  I=1,N
      T(7,I) = F(I)
      TE = D(I)
      DO 15 K=2,NS
      TF = TE -T(K,I)
       T(K,I)=TE
   15 TE =TF
17    T(NS+1,I)=TE
      IF(KKF.NE.0) KKF=1
      NS=NS+1
      IF(IDF.EQ.2) GO TO 25
```

```
501    RETURN
C
404   KKF=2
      RETURN1
C
C     HALF
500   IF(H.LE.HMIN) GO TO 340
502   INT =INT +1
      IF(INT.GE.10) GO TO 340
      X=X-H
  24  H=H/2.D0
      HA(1)=H
      GO TO 28
C     DOUBLE
25    INT=1
      IF(2.D0*H.GE.HMAX) RETURN
510   H=H+H
      HA(1)=H
28    NN=NS-2
      IF(NN.EQ.0) GO TO 260
      DO 26 I=1,N
      ER(I)=ER(I)*64.D0**(2*IDF-3)
      KKK=1
      DO 26 L=1,NN
      TE=0.D0
      DO 27 K=L,NN
      GO TO (44,45),IDF
  44  TF=AH(KKK)
      GO TO 29
  45  TF =AD(KKK)
29    TE = TE +TF*T(K+2,I)
  27  KKK=KKK+1
  26  T(L+2,I) =TE
260   IF(IDF-2)21,501,21
C
      ENTRY ADPAR(TP)
   3  KF = 3
      P= (TP-X    )/H
      P2 =P*P
      P3 =P*P2
      P4 = P*P3
      P5 = P*P4
      ACC(2)=P
      ACC(3)=P2/2.D0
      ACC(4) =(2.D0*P3+3.D0*P2)/12.D0
      ACC(5) =(P4+4.D0*(P3+P2))/24.D0
      ACC(6) =(6.D0*P5 +45.D0*P4+110.D0*P3 +90.D0*P2)/720.D0
      GO TO 100
C
43    DO 46 I=1,N
46    FP(I) = F(I)
      RETURN
C
      ENTRY ADRES
   4  INT=0
      H=H/32.D0
      HA(1)=H
       NS=2
      RETURN
```

II, D, Program Listing (cont.)

```
C
C
 100  DO 101 I=1,N
      YT=0.D0
      DO 102 K=1,NS
 102  YT = YT + AC(K,KF)* T(K,I)
      F (I) = YT *H  +  T(7,I)
      IF(KF.EQ.1.AND.NS.EQ.6) F(I)=F(I)-CON1*ER(I)
  101 CONTINUE
      GO TO (520,42,43),KF
C
C
      END
```

II, D, Program Listing (cont.)

```
00101                                                                                   *NEW
00101         SUBROUTINE CCC(DIN,DOUT,WC,CODE,NER)                                NOZM  10
00101   C
00101   C***********************************************************************NOZM  15
00101   C***** THIS PROGRAM WAS WRITTEN FROM A REPORT ON NOZZLE ADMITTANCE******NOZM  20
00101   C***** THEORY FROM PRINCETON UNIVERSITY.  THE ANALYSIS WAS DONE BY******NOZM  25
00101   C***** CARL LUNDELIUS AND THE PROGRAMMING BY JERRY HOWARD.JOB 8052******NOZM  30
00101   C*************
00101   C     NOZMIT MODIFIED 25 JUL 67  TO SUPPLY CR1,CI1 IN PLACE OF BR1,BI1.
00101   C     NOZMIT MODIFIED BY WAUGH AND DUDLEY MARCH 70  TO EXTEND NOZZLE
00101   C     AS A CYLINDER STN INCHES
00101   C***********************************************************************NOZM  35
00101   C                                                                         NOZM  40
00103         LOGICAL  LOGIK,  SL1,  SL2,  EORJ                                   NOZM 185
00104         COMPLEX  GH,SC,XJK,F3,X1,X2,ZETA
00105         COMPLEX T1X,OQ,F3P,X1P,X2P,ZETAP,DZETP,DF3P,DX1P,DX2P
00106         COMPLEX ASCRP,BSCRP,CSCRP,ESCRP
00107         COMMON  /ABCDF/  EXTRA(100),  ABLOK(600)                            NOZM  45
00107        1     ,  XX      , U2TBL   , DESIRE , RAT     , RAC     , RCC        NOZM  50
00107        2     ,  RCT     , ALFA    , G      , KN      , Y       , YP         NOZM  55
00107        3     ,  YOUT    , TEMP    , E      , XTABLE  , YTABLE  , A          NOZM  60
00107        4     ,  R       , AM      , AP     , AMP     , ZZ      , AMM        NOZM  65
00107        5     ,  AMP2    , CALFA   , CTALFA , DELAM   , DELTZ   , FKN        NOZM  70
00107        6     ,  G1      , G2      , G3     , G4      , JFLAG1  , KNM1       NOZM  75
00107        7     ,  K       , NN      , PI     , PROD    , RSTA1   , RSTA2      NOZM  80
00107        8     ,  SALFA   , T1      , T2     , T3      , XINT    , XK         NOZM  85
00107        9     ,  ZZ1     , ZZ2     , ZZ3    , A1      , ABC     , ABD        NOZM  90
00110         COMMON  /ABCDF/                                                     NOZM  95
00110        1     ,  AI1     , ALPHAI  , ALPHAR , AR1     , B101    , B102       NOZM 100
00110        2     ,  B10     , B1      , B2     , B3      , B4      , B5         NOZM 105
00110        3     ,  B6      , B7      , B8     , B91     , B92     , B9         NOZM 110
00110        4     ,  BI1     , BR1     , C2     , C3      , CHII    , CHIR       NOZM 115
00110        5     ,  CI1     , CR1     , C      , D10     , D11     , D1         NOZM 120
00110        6     ,  D2      , D3      , D4     , D5      , D6      , D7         NOZM 125
00110        7     ,  D8      , D9      , DC2    , D       , DU2     , EI         NOZM 130
00110        8     ,  ER      , F3I     , F3R    , FI      , FR      , H1         NOZM 135
00110        9     ,  H       , I       , IWO    , IW      , IWW     , J          NOZM 140
00111         COMMON  /ABCDF/                                                     NOZM 145
00111        1        MDESIR  , NK      , NP     , S2      , S       , TT         NOZM 150
00111        2     ,  U2      , U       , W2     , W       , XI0I    , XI0R       NOZM 155
00111        3     ,  XI2I    , XI2R    , XI     , XJI     , XJR     , XMNEW      NOZM 160
00111        4     ,  XMOLD   , XNEW    , XOLD   , XPT     , X       , ZI         NOZM 165
00111        5     ,  ZR                                                          NOZM 170
00112         COMMON  /PROLOG/  LOGIK(50), SL1, SL2, EORJ                         NOZM 180
00112   C                                                                         NOZM 235
00113         DIMENSION XX(200),U2TBL(200),XTABLE(200),YTABLE(200),ZZ(200)        NOZM 195
00114         DIMENSION Y(8),YP(8),YOUT(8),TEMP(16,8),E(8)
00115         DIMENSION A(200),R(200),AM(200),AP(200),AMP(200)                    NOZM 205
00116         DIMENSION   DIN(1), DOUT(1),WC(1)                                   NOZM 210
00117         EQUIVALENCE  ( VELPOT, EXTRA(101) ), ( STN, EXTRA(102) )
00120         EQUIVALENCE  ( GRAD , XK  )
00120   C                                                                         NOZM 320
00120   C**********************************************************************
00120   C                                                                         NOZM 330
00120   C                                                                         NOZM 335
```

II, D, Program Listing (cont.)

```
00121     56*           NER  =  0                                                      NOZM 340
00121     57*     C ***********                                                        NOZM 345
00121     58*     C READ INPUT*                                                        NOZM 350
00121     59*     C ***********                                                        NOZM 355
00122     60*           G  =  DIN(1)                                                   NOZM 360
00123     61*           NOMEG=WC(2)+2.0001                                             NOZM 365
00124     62*           MDESIR  =  DIN(2)  +  .0001                                    NOZM 370
00125     63*           GEO=.5*(G+1.0)*SQRT(DIN(4)*DIN(7))                             NOZM 375
00126     64*           IF  (CODE)   10,30,50                                          NOZM 380
00131     65*        10 NER =  1                                                       NOZM 385
00132     66*        30 RETURN                                                         NOZM 390
00133     67*        50 IF(WC(1))70,90,70                                              NOZM 395
00136     68*        70 WCONST=DOUT(3)/(6.2831853*DOUT(1))                             NOZM 400
00137     69*           GEO=GEO/DOUT(1)                                                NOZM 405
00140     70*           GO TO 110                                                      NOZM 410
00141     71*        90 WCONST=DOUT(3)/(6.2831853*DOUT(2))                             NOZM 415
00142     72*           GEO=GEO/DOUT(2)                                                NOZM 420
00143     73*       110 IF(CODE-99.0)190,190,130                                       NOZM 425
00143     74*     C ****************                                                   NOZM 430
00143     75*     C PRINT OUT INPUT*                                                   NOZM 435
00143     76*     C ****************                                                   NOZM 440
00146     77*       130 CALL PAGE(5)                                                   NOZM 445
00147     78*           WRITE (6,150)                                                  NOZM 450
00151     79*       150 FORMAT (1H0,75H PROGRAM  C  INPUT - CALCULATES NOZZLE ADMITTANCE  NOZM 455
00151     80*          1COEFFICIENTS USING 8052      )                                 NOZM 460
00152     81*           WRITE (6,170)G,  MDESIR                                        NOZM 465
00156     82*       170 FORMAT (1H0,5X,3HG =,F9.3,11H , MDESIR =,I2  )                 NOZM 470
00157     83*       190 CONTINUE                                                       NOZM 475
00157     84*     C ******************************************************************* NOZM 480
00157     85*     C MDESIR=1,INPUT TABLE,INPUT DESIRE  MDESIR=2,CALCULATE TABLE,INPUT* NOZM 485
00157     86*     C DESIRE  MDESIR=3,CALCULATE TABLE,CALCULATE DESIRE (AT LAST POINT)* NOZM 490
00157     87*     C ******************************************************************* NOZM 495
00160     88*           GO TO (290,210,210),MDESIR                                     NOZM 500
00161     89*       210 RAT   =  DIN(4)                                                NOZM 505
00162     90*           RAC   =  DIN(5)                                                NOZM 510
00163     91*           RCC   =  DIN(6)                                                NOZM 515
00164     92*           RCT   =  DIN(7)                                                NOZM 520
00165     93*           ALFA  =  DIN(8)                                                NOZM 525
00166     94*           KN    =  DIN(9)  +  .0001                                      NOZM 530
00167     95*           IF   (CODE  -  99.0)    270,270,230                            NOZM 535
00172     96*       230 CALL  PAGE (1)                                                 NOZM 540
00173     97*           WRITE (6,250) RAT,RAC,RCC,RCT,ALFA,KN,STN
00204     98*       250 FORMAT  (1H ,5X5HRAT =,F7.3,6H,RAC =,F7.3,6H,RCC =,F7.3,6H,RCT =, NOZM 550
00204     99*          1   F7.3, 7H,ALFA = F7.3,5H,KN = I4,6H,STN = F7.3    )
00205    100*       270 CONTINUE                                                       NOZM 560
00206    101*           CALL TBLCAL                                                    NOZM 565
00207    102*           KN = KN/2 + 1                                                  NOZM 570
00210   *DIAGNOSTIC*    THE TEST FOR EQUALITY BETWEEN NON-INTEGERS MAY NOT BE MEANINGFUL.
00210    103*           IF ( STN .EQ. 0.0 )    GO TO  370
00212    104*           H= SQRT ( U2TBL(KN))/RAT*GRAD*2.0
00213    105*           KN1 = KN+1
00214    106*           KN3 = KN+3
00215    107*           DO  280  NK = KN1,  KN3
00220    108*           XX(NK) = XX(NK-1) - H*0.01
00221    109*           ZZ(NK) = ZZ(NK-1)+.01
00222    110*       280 U2TBL(NK) = U2TBL(KN)
00224    111*           XX(KN3) = XX(KN) - H*STN
00225    112*           ZZ(KN3) = ZZ(KN)+STN
00226    113*           KN = KN3
00227    114*           GO TO 370                                                      NOZM 575
```

II, D, Program Listing (cont.)

```
00227   C ***************************************************        NOZM 580
00227   C READ VELOCITY POTENTIAL TABLE, FIRST POINT IS (0,1)*        NOZM 585
00227   C ***************************************************        NOZM 590
00230     290 J = 10                                                  NOZM 595
00231         DO 330 I = 1,200                                        NOZM 600
00234         XX(I) =  DIN(J)                                         NOZM 610
00235         U2TBL(I)  =  DIN(J+1)                                   NOZM 615
00236         J = J+2                                                 NOZM 620
00237         IF (I)  330,330,310                                     NOZM 625
00242     310 IF ( XX(I) )   330,350,350                              NOZM 630
00245     330 CONTINUE                                                NOZM 635
00245   C ************************************************            NOZM 640
00245   C KN = COUNT OF TOTAL NO. OF POINTS IN THE TABLE.*            NOZM 645
00245   C ************************************************            NOZM 650
00247     350 KN = I - 1                                              NOZM 655
00250     370 CONTINUE
00251         FORLT=U2TBL(2)
00252         IF(CODE-199.0)470,470,390
00255     390 CALL PAGE(70)                                           NOZM 665
00256         WRITE (6,430)                                           NOZM 670
00260         KKKN = KN/2
00261         DO 410 I=1,KKKN                                         NOZM 680
00264         KKK=KKKN+I                                              NOZM 685
00265         VOL1=SQRT(U2TBL(I))                                     NOZM 690
00266         VOL2=SQRT(U2TBL(KKK))                                   NOZM 695
00267         WRITE (6,450) I,ZZ(I),XX(I),VOL1,KKK,ZZ(KKK),XX(KKK),VOL2
00301     410 CONTINUE                                                NOZM 705
00303     430 FORMAT(//,40X, 25HVELOCITY POTENTIAL TABLE ,//,35X, 46HZ = AXIAL PNOZM 710
00303        1OSITION IN NOZZLE (INCHES)           ,/,35X,39HX = VELOCITY POTENNOZM 715
00303        2TIAL (NONDIMENSIONAL) ,/,35X, 69HV = AXIAL VELOCITY (UNDIMENSIONALNOZM 720
00303        3IZED BY SPEED OF SOUND AT THROAT)                       NOZM 725
00303        4 ,//,5X,5HPOINT,11X,1HZ,16X,1HX,15X,1HV,16X,5HPOINT,11X,1HZ,16X,1HNOZM 730
00303        5X,16X,1HV,//,)                                          NOZM 735
00304     450 FORMAT(6X,I3,6X,3(E12.5,5X),6X,I3,6X,2(E12.5,5X),E12.5,)  NOZM 740
00305     470 CONTINUE                                                NOZM 745
00305   C ***************************************************         NOZM 750
00305   C REVERSE ORDER OF TABLE VALUES AND SET UP FOR INT4D*         NOZM 755
00306     490 NK = KN+1                                               NOZM 760
00307         XTABLE(NK) = 0.0                                        NOZM 765
00310         YTABLE(NK) = 0.0                                        NOZM 770
00311         DO 510 I = 1,KN                                         NOZM 775
00314         NK = NK - 1                                             NOZM 780
00315         XTABLE(I) = XX(NK)                                      NOZM 785
00316         YTABLE(I) = U2TBL(NK)                                   NOZM 790
00317     510 CONTINUE                                                NOZM 795
00317   C                                                             NOZM 800
00321         VELPOT = XX(KN)
00321   C ***********************                                     NOZM 805
00321   C READ ONE CASE AT A TIME*                                    NOZM 810
00321   C ***********************                                     NOZM 815
00322         IWW  = 410                                              NOZM 820
00323         IWO  =  2                                               NOZM 825
00324         NP   = DIN(3) + .0001                                   NOZM 830
00325         IF   (CODE   -  99.0)  570,570,530                      NOZM 835
00330     530 NPR  =  NP + 3                                          NOZM 840
00331         NPT  =  3 * NP  + 409                                   NOZM 845
00332         CALL  PAGE (NPR)                                        NOZM 850
00333         WRITE (6,550)( DIN(I), I =  410, NPT  )                 NOZM 855
00341     550 FORMAT(1H0,39X, 6X,4H  WN,12X,6H(SNH)N,10X,4H DES //    NOZM 860
00341        1              (40X,3E16.6 ))                            NOZM 865
00342     570 CONTINUE                                                NOZM 870
00343     590 IF  (CODE -  9.0)   670,670,610                         NOZM 875
```

II, D, Program Listing (cont.)

```
00343   C ********************************                                      NOZM 880
00343   C PRINT HEADER AND OUTPUT SYMBOLS*                                      NOZM 885
00343   C ********************************                                      NOZM 890
00346     610 CALL PAGE(70)                                                     NOZM 895
00347         KPAGE=48                                                          NOZM 900
00350         KCOUNT=0                                                          NOZM 905
00351         WRITE (6,630)                                                     NOZM 910
00353     630 FORMAT (1H0, 20H PROGRAM  C  OUTPUT   )                           NOZM 915
00354         WRITE (6,650)                                                     NOZM 920
00356     650 FORMAT(//4X,6H(SNH)C,5X2HWC,6X,7HMACH NO,9X,2HAR,14X,2HAI,14X,2HBRNOZM 925
00356        1,14X,2HBI,14X,2HCR,14X,2HCI/4X,6H(SNH)N,5X2HWN,9X,1HG,6X,13H-AR/(MNOZM 930
00356        2ACH NO),3X,13H-AI/(MACH NO),9X,2HT1,14X,2HT2,8X,13H-CR/(MACH NO),3NOZM 935
00356        3X,13H-CI/(MACH NO)//)                                             NOZM 940
00357     670 CONTINUE                                                          NOZM 945
00360         DO 1310 IW =  1,NP                                                NOZM 950
00363         W  =   DIN(IWW)                                                   NOZM 955
00364         S  =   DIN(IWW+1)                                                 NOZM 960
00365         GO  TO (690,690,710), MDESIR                                      NOZM 965
00366     690 DESIRE   =  DIN(IWW+2)                                            NOZM 970
00367     710 IWW =  IWW + 3                                                    NOZM 975
00367   C                                                                       NOZM 980
00370         MX=1
00371         XPR=0.0
00372         XMOLD=0.0                                                         NOZM 985
00373         XMNEW=0.0                                                         NOZM 990
00374         XOLD=0.0                                                          NOZM 995
00375         XPT = 0.0                                                         NOZM1000
00376         DXI = -.001
00377         X = 0.0
00400         A1 = .5*(G+1.0)
00401         XNEW=0.0                                                          NOZM1010
00402         XI = -.001                                                        NOZM1015
00403         H=XI                                                              NOZM1020
00404         W2=W*W                                                            NOZM1025
00405         S2=S*S                                                            NOZM1030
00406         G2=G*G                                                            NOZM1035
00406   C ***********************************                                   NOZM1040
00406   C     INITIALIZE 8 EQUATIONS AT X=0.*                                   NOZM1045
00406   C ***********************************                                   NOZM1050
00407         DO 730 I = 1,8                                                    NOZM1055
00412     730 E(I) = 1.E-05                                                     NOZM1060
00414         I = 0                                                             NOZM1065
00415         SC = CMPLX(0.0,W)
00416         CALL BTLDA (EXTRA(91),EXTRA(92),RAT,RCT,G,BT)
00417         ZETA = ((W2-S2)/4.0-(G-1.)/4.0*SC)/(A1+SC)
00420         ZETAP = ((SC+A1*(A1+2.*BT))*ZETA+A1*ZETA**2+S2*(1.-G)/8.
00420        1+SC*(G-1.)/4.*(A1+2.*BT))/(-2.*A1-SC)
00421         X1 = 0.0
00422         X1P = S2/4.0/A1/(1.0+SC/A1)
00423         X2 = 0.0
00424         X2P = 1.0/(2.0*A1+2.0*SC)
00425         Y(1) = 0.0
00426         Y(2) = 0.0
00427         Y(3) = REAL (X1)
00430         Y(4) = AIMAG (X1)
00431         Y(5) = REAL (X2)
00432         Y(6) = AIMAG (X2)
00433         Y(7) = REAL(ZETA)
00434         Y(8) = AIMAG(ZETA)
```

II, D, Program Listing (cont.)

```
    YP(1) = .5
    YP(2) = 0.0
    YP(3) = REAL (X1P)
    YP(4) = AIMAG (X1P)
    YP(5) = RE AL (X2P)
    YP(6) = AIMAG(X2P)
    YP(7) = REAL(ZETAP)
    YP(8) = AIMAG(ZETAP)
    X = DXI
    CALL ADSET(8,Y,YP,YOUT,TEMP,X,H,E)
    Y(1)=Y(1)+DXI*YP(1)
    Y(2)=Y(2)+DXI*YP(2)
    Y(3)=Y(3)+DXI*YP(3)
    Y(4)=Y(4)+DXI*YP(4)
    Y(5)=Y(5)+DXI*YP(5)
    Y(6)=Y(6)+DXI*YP(6)
    Y(7)=Y(7)+DXI*YP(7)
    Y(8)=Y(8)+DXI*YP(8)
750 CALL ADINT
C ****************************************                    NOZM1245
C INTERPOLATING ON VELOCITY POTENTIAL TABLE*                  NOZM1250
C ****************************************                    NOZM1255
 770 CALL INT4D(XTABLE(1),YTABLE(1),X,U2,DU2)                 NOZM1260
    IF (U2-FORLT) 772,772,771
 771 U2 = 1.+X+BT*X**2
    DU2 = 1.+2.*BT*X
 772 C2 = .5*(G+1.0-U2*(G-1.0))
    C=SQRT(C2)                                                NOZM1270
    U=SQRT(U2)                                                NOZM1275
    DC2=-.5*(G-1.0)*DU2                                       NOZM1280
C ****************************************************       NOZM1285
C SINGULARITY AT THROAT-FAKE EQUATIONS 7 AND 8 UNTIL PAST*    NOZM1290
C ****************************************************       NOZM1295
    C2P = C2**(1.0/(G-1.0))
    SG = A1*U2/C2*DU2
    SH = W*U2
    XJ = .25*(W2-U*C2P**G*S2)
    XK1=-(G-1.)/4.*U2/C2*DU2*W
    GH = CMPLX(SG,SH)
    OQ = SC/2./U2
    B = U2*(C2-U2)
    XJK=CMPLX(XJ,XK1)
    F3P= CMPLX(Y(1),Y(2))
    X1P= CMPLX(Y(3),Y(4))
    X2P= CMPLX(Y(5),Y(6))
    ZETAP= CMPLX(Y(7),Y(8))
    DZETP = (GH*ZETAP-XJK)/B-ZETAP**2
    T1X =ZETAP-SC*(.5/U2+1./A1/(1.0-U2))
    DX1P = -T1X*X1P+S2/A1/4.0/U*C2P
    DF3P = OQ*F3P+.5*DU2
    DX2P = -T1X*X2P+1.0/A1*((DF3P-F3P/C2*DC2)/C2+S2*C2P*U/2.0
   1/SC*((1.0-U2)/2.0/U2+F3P/U2))
    YP(1) = REAL(DF3P)
    YP(2) = AIMAG(DF3P)
    YP(3) = REAL(DX1P)
    YP(4) = AIMAG(DX1P)
    YP(5) = REAL(DX2P)
    YP(6) = AIMAG(DX2P)
    YP(7) = REAL(DZETP)
    YP(8) = AIMAG(DZETP)
 930 FORMAT (3X,4E16.5/3X,4E16.5/3X,4E16.5/3X,4E16.5/3X,2E16.5/)
```

II, D, Program Listing (cont.)

```
00524   C                                                                    NOZM1445
00525         CALL ADCOR($770)
00525   C                                                                    NOZM1455
00525   C*****   X  AND  VELPOT  BOTH NEGATIVE
00525   C*****   TERMINATE AT FINAL VELOCITY POTENTIAL
00526         IF ( X - VELPOT )       990,  990,  750
00531   990   CALL ADPAR(VELPOT)
00531   C                                                                    NOZM1630
00531   C                                                                    NOZM1635
00531   C ******************************************************             NOZM1640
00531   C COMPUTE ADMITTANCE COEFFICIENTS AND PRINT FINAL RESULTS*            NOZM1645
00531   C ******************************************************             NOZM1650
00532    1010 CALL  INT4D  ( XTABLE, YTABLE, VELPOT, U2, DU2 )
00533    1030 C2=.5*(G+1.0-U2*(G-1.0))                                      NOZM1660
00534         C=SQRT(C2)                                                     NOZM1665
00535         U=SQRT(U2)                                                     NOZM1670
00536         C2P=C2**(1./(G-1.))
00536   C                                                                    NOZM1675
00537         F3R = YOUT(1)/C2
00540         F3I = YOUT(2)/C2
00541         X1R = YOUT(3)/(1.-U2)
00542         X1I = YOUT(4)/(1.-U2)
00543         X2R = YOUT(5)/(1.-U2)
00544         X2I = YOUT(6)/(1.-U2)
00545         ZETAR = YOUT(7)
00546         ZETAI = YOUT(8)
00547         F3 = CMPLX(F3R,F3I)
00550         X1 = CMPLX(X1R,X1I)
00551         X2 = CMPLX(X2R,X2I)
00552         ZETA = CMPLX(ZETAR,ZETAI)
00553         ASCRP=1./G*A1**(A1/(G-1.))*U/C2P*(C2*X1-ZETA)/(U2*(C2*X1-ZETA)
00553        1-SC/2.)
00554         BSCRP = SC*GRAD*SQRT(U/C2P)*C2*X1/(U2*(C2*X1-ZETA)-SC/2.)
00555         CSCRP=1./SQRT(A1)*U*C2*(.5*(1.-U2)*X1-.5*SC*X2+F3*ZETA)/(U2*(C2*
00555        1X1-ZETA)-.5*SC)
00556         WCP=GRAD*RAC/RAT/A1**.5*W
00557         ESCRP=G*ASCRP+(0.,1.)*BSCRP/WCP
00560         AR1 = REAL(ASCRP)
00561         AI1 = AIMAG(ASCRP)
00562         BR1 = REAL(BSCRP)
00563         BI1 = AIMAG(BSCRP)
00564         CR1 = REAL(CSCRP)
00565         CI1 = AIMAG(CSCRP)
00566         T1 = REAL(ESCRP)
00567         T2=AIMAG(ESCRP)
00567   C                                                                    NOZM1825
00570    1110 ALPHAR = -AR1/DESIRE                                           NOZM1830
00571         ALPHAI = -AI1/DESIRE                                           NOZM1835
00572         CHIR = -CR1/DESIRE                                             NOZM1840
00573         CHII = -CI1/DESIRE                                             NOZM1845
00573   C                                                                    NOZM1880
00573   C ***************************                                        NOZM1885
00573   C  CASE IS COMPLETED                                                 NOZM1890
00573   C ***************************                                        NOZM1895
```

II, D, Program Listing (cont.)

```
00573  C                                                                      NOZM1900
00573  C     FOR S=0 USE COEFFICIENTS A AND C IN CONC
00573  C
00573  C     FOR S=N USE COEFFICIENTS T AND C IN CONC OR DISTR
00573  C
00574        IF(S) 1150,1130,1150
00577   1130 DOUT(IW+19)=W/GEO
00600        DOUT(IW+49)=AR1
00601        DOUT(IW+79)=AI1
00602        DOUT(IW+109)=CR1
00603        DOUT(IW+139)=CI1
00604        GO TO 1170                                                       NOZM1950
00605   1150 CONTINUE                                                         NOZM1955
00606        DOUT(IW+19)=W/GEO                                                NOZM1960
00607        DOUT(IW+49)=T1                                                   NOZM1965
00610        DOUT(IW+79)=T2                                                   NOZM1970
00611        DOUT(IW+109) = CR1
00612        DOUT(IW+139) = CI1
00613   1170 IWO =IWO+2                                                       NOZM1985
00614        IF  ( CODE  -  9.0)    1310,1310,1190                            NOZM1990
00617   1190 CONTINUE                                                         NOZM1995
00620        KKKN=(2*IW)+1                                                    NOZM2000
00621        IF(NOMEG-KKKN)1210,1230,1230                                     NOZM2005
00624   1210 KKKN=KKKN-1                                                      NOZM2010
00625   1230 KCOUNT=KCOUNT+4                                                  NOZM2015
00626        IF(KCOUNT-KPAGE)1270,1270,1250                                   NOZM2020
00631   1250 CALL PAGE(70)                                                    NOZM2025
00632        WRITE (6,650)                                                    NOZM2030
00634        KCOUNT=0                                                         NOZM2035
00635        KPAGE=48                                                         NOZM2040
00635  C ********************                                                 NOZM2045
00635  C PRINT FINAL RESULTS*                                                 NOZM2050
00635  C ********************                                                 NOZM2055
00636   1270 WRITE (6,1290)WC(1),WC(KKKN),DESIRE,AR1,AI1,BR1,BI1,CR1,CI1,S,W,G,NOZM2060
00636       1ALPHAR,ALPHAI,T1,T2,CHIR,CHII,VELPOT
00663   1290 FORMAT(2(3X,F7.4,2X,F7.4,3X,F7.4,6E16.5/))                       NOZM2070
00664        FCCPS=(WCONST*WC(KKKN))*12.0                                     NOZM2075
00665        WRITE (6,1330)FCCPS                                              NOZM2080
00670   1310 CONTINUE                                                         NOZM2085
00672   1330 FORMAT(4X,8HFC(CPS)=F10.4//)                                     NOZM2090
00673        DOUT(205)=DESIRE                                                 NOZM2095
00674        IF(S) 1370,1350,1370                                             NOZM2100
00677   1350 DOUT(1)=NP                                                       NOZM2105
00700        DOUT(101)=NP                                                     NOZM2110
00701        RETURN                                                           NOZM2115
00702   1370 DOUT(18)=NP                                                      NOZM2120
00703        DOUT(NP+21)=0.0                                                  NOZM2125
00704        DOUT(NP+51)=0.0                                                  NOZM2130
00705        DOUT(NP+81)=0.0                                                  NOZM2135
00706        DOUT(NP+111)=0.0                                                 NOZM2140
00707        DOUT(NP+141)=0.0                                                 NOZM2145
00710        RETURN                                                           NOZM2150
00711        END                                                              NOZM2155
```

II, D, Program Listing (cont.)

```
@ ELT SUB28,1,700708, 29803

      SUBROUTINE BTLDA(RSATI,RSCTI,RSATO,RSCTO,G,BT)
      DIMENSION  XM(10)      ,ZS(10),XMS(10),XMS2(10),PHI(10),
     1         T1(10),T2(10),T3(10),W(10),C(10),ALPHA(10),BETA(10),A(10)
     2,B(10)
      DIMENSION E(10),ARAT(10),ZSA(10),ARA(10)
      L = 0
      J = 0
      N = 5
      KC = 2
      K = 2
      ARAT(1) = 1.
      XM(1) = 1.
      XMS(1) = 1.
      XM(2) = .99
      XMS2(1) = 1.
      ZSG=-ABS(RSCTO/2.)                                                *NEW
      DO 20 I = 2,5                                                     **-1
      XM(1+1) = XM(I)-.01
      ARAT(I) = 1./XM(I)*(2./(G+1.)*(1.+(G-1.)/2.*XM(I)*XM(I)))**((G+1.)
     1/2./(G-1.))
   20 XMS(I) = SQRT((G+1.)/2.*XM(I)**2/(1.+(G-1.)/2.*XM(I)**2))
   48 RSO = RSATO+RSCTO-SQRT(ABS(RSCTO*RSCTO-ZSG*ZSG))
      RSI = RSATI + RSCT1-SQRT(ABS(RSCTI*RSCTI-ZSG*ZSG))
      IF (RSCTI) 47,49,46
   49 IF (RSATI) 51,51,46
  51  RSI = 0.
      GO TO 46
   47 RSI = RSATI + RSCTI+SQRT(ABS(RSCTI*RSCTI-ZSG*ZSG))
   46 ARATI = (RSO*RSO-RSI*RSI)/(RSATO*RSATO-RSATI*RSATI)
      IF(ARATI-ARAT(5)) 50,60,52
   52 IF(M .EQ. 2) GO TO 60
      ZSG=ZSG/2.
      M = 1
      GO TO 48
  50  ZSG=ZSG*2.
      IF(M .EQ. 1) GO TO 60
      M = 2
      GO TO 48
  60  ZSA(2) = ZSG/5.
      DZ = ZSA(2)
      ZSA(1) = 0.
      ARA(1) = 1.
      DO 80 I = 2,5
      RSO = RSATO +RSCTO-SQRT(ABS(RSCTO*RSCTO-ZSA(I)*ZSA(I)))
      RSI = RSATI+RSCTI-SQRT(ABS(RSCTI*RSCTI-ZSA(I)*ZSA(I)))
      IF(RSCTI) 82,83 ,81
   83 IF (RSATI)  84,84,81
   84 RSI = 0.
      GO TO 81
   82 RSI = RSATI+RSCTI-SQRT(ABS(RSCTI*RSCTI-ZSA(I)*ZSA(I)))
   81 ARA(I)=(RSO*RSO-RSI*RSI)/(RSATO*RSATO-RSATI*RSATI)
   80 ZSA(I+1)=ZSA(I)+DZ
      CALL ORTHLS(ZSA ,ARA,W,N,L,J,C,ALPHA,BETA,K,T1,T2,T3,IND1)
      CALL COEFS(J,C,ALPHA,BETA,KC,A,T1,T2,T3,IND2)
      DO 90 I=2,5
```

II, D, Program Listing (cont.)

```
90   ZS(1) =-SQRT(-(A(1)-ARAT(I))/A(3))
     CALL ORTHLS(ZS,XMS,W,N,L,J,C,ALPHA,BETA,K,T1,T2,T3,IND1)
     CALL COEFS(J,C,ALPHA,BETA,KC,B,T1,T2,T3,IND2)
     PHI(1) = 0.
     DO 30 I = 2,5
     PHI(I) = 2.*B(2)       *(B(1)*ZS(I)+B(2)/2.*ZS(I)**2+B(3)*ZS(I)**3
    1/3.)
30   XMS2(I) = XMS(I)**2
     CALL ORTHLS(PHI,XMS2,W,N,L,J,C,ALPHA,BETA,K,T1,T2,T3,IND1)
     CALL COEFS(J,C,ALPHA,BETA,KC,E,T1,T2,T3,IND2)
     BT = E(3)
     RETURN
     END
```

```
@ ELT SUB29,1,700707, 30007

      SUBROUTINE ORTHLS (X,Y,W,N,L,J,C,ALPHA,BETA,K,T1,T2,T3,IND1)      ORTHLS
C-----------------------------------------------------------------------ORTHLS
C     THIS SUBROUTINE COMPUTES THE COEFFICIENTS OF THE POLYNOMIAL      ORTHLS
C     EQUATION OF DEGREE K AND THE ALPHA AND BETA PARAMETERS.         ORTHLS
C-----------------------------------------------------------------------ORTHLS
      DIMENSION X(N),Y(N),W(N),C(K),ALPHA(K),BETA(K),T1(N),T2(N),T3(N)  ORTHLS
C-----------------------------------------------------------------------ORTHLS
C     PROGRAM INITIALIZATION.                                           ORTHLS
C-----------------------------------------------------------------------ORTHLS
      KJ1=K-J+1                                                         ORTHLS
      IF (KJ1.LE.0) GO TO 16                                            ORTHLS
      SUM=0.0                                                           ORTHLS
      IF (L.EQ.1) GO TO 3                                               ORTHLS
      DO 2 I=1,N                                                        ORTHLS
      T3(I)=X(I)                                                        ORTHLS
      IF (J.GT.0) GO TO 1                                               ORTHLS
      SUM=SUM+1.0                                                       ORTHLS
      GO TO 2                                                           ORTHLS
    1 SUM=SUM+X(I)**(2*J)                                               ORTHLS
    2 W(I)=1.0                                                          ORTHLS
      GO TO 7                                                           ORTHLS
    3 DO 6 I=1,N                                                        ORTHLS
      T3(I)=X(I)                                                        ORTHLS
      IF (J.GT.0) GO TO 4                                               ORTHLS
      SUM=SUM+W(I)                                                      ORTHLS
      GO TO 5                                                           ORTHLS
    4 SUM=SUM+W(I)*X(I)**(2*J)                                          ORTHLS
    5 X(I)=W(I)*X(I)                                                    ORTHLS
    6 Y(I)=W(I)*Y(I)                                                    ORTHLS
    7 B=0.0                                                             ORTHLS
      RO=SUM                                                            ORTHLS
      DO 9 I=1,N                                                        ORTHLS
      IF (J.GT.0) GO TO 8                                               ORTHLS
      T2(I)=1.0                                                         ORTHLS
      GO TO 9                                                           ORTHLS
    8 T2(I)=T3(I)**J                                                    ORTHLS
    9 T1(I)=0.0                                                         ORTHLS
C-----------------------------------------------------------------------ORTHLS
C     BEGIN COMPUTATION.                                                ORTHLS
C-----------------------------------------------------------------------ORTHLS
      II=1                                                              ORTHLS
   10 S=0.0                                                             ORTHLS
      DO 11 I=1,N                                                       ORTHLS
   11 S=S+Y(I)*T2(I)                                                    ORTHLS
C-----------------------------------------------------------------------ORTHLS
C     COMPUTATION OF A COEFFICIENT IN THE POLYNOMIAL EQUATION.         ORTHLS
C-----------------------------------------------------------------------ORTHLS
      C(II)=S/RO                                                        ORTHLS
      IF (II.GE.KJ1) GO TO 15                                           ORTHLS
C-----------------------------------------------------------------------ORTHLS
C      COMPUTATION OF AN ALPHA FOR THE POLYNOMIAL EQUATION.            ORTHLS
C-----------------------------------------------------------------------ORTHLS
      SUMXPS=0.0                                                        ORTHLS
      DO 12 I=1,N                                                       ORTHLS
   12 SUMXPS=SUMXPS+X(I)*T2(I)*T2(I)                                    ORTHLS
```

II, D, Program Listing (cont.)

```
      ALPHA(II)=SUMXPS/RO                                               ORTHLS
C-----------------------------------------------------------------------ORTHLS
C     COMPUTATION OF A NEW POLYNOMIAL.                                  ORTHLS
C-----------------------------------------------------------------------ORTHLS
      DO 13 I=1,N                                                       ORTHLS
      TEMP=T2(I)                                                        ORTHLS
      T2(I)=(T3(I)-ALPHA(II))*T2(I)-B*T1(I)                             ORTHLS
   13 T1(I)=TEMP                                                        ORTHLS
C-----------------------------------------------------------------------ORTHLS
C     COMPUTATION OF A BETA FOR THE POLYNOMIAL EQUATION.                ORTHLS
C-----------------------------------------------------------------------ORTHLS
      R=0.0                                                             ORTHLS
      DO 14 I=1,N                                                       ORTHLS
   14 R=R+W(I)*T2(I)*T2(I)                                              ORTHLS
      BETA(II)=R/RO                                                      RTHLS
      RO=R                                                              ORTHLS
      B=BETA(II)                                                        ORTHLS
      II=II+1                                                           ORTHLS
      GO TO 10                                                          ORTHLS
C-----------------------------------------------------------------------ORTHLS
C     SUCCESSFUL RETURN.                                                ORTHLS
C-----------------------------------------------------------------------ORTHLS
   15 IND1=+1                                                           ORTHLS
      RETURN                                                            ORTHLS
C-----------------------------------------------------------------------ORTHLS
C     ERROR RETURN.  SET ALL C COEFFICIENTS, ALPHA AND BETA TO ZERO.    ORTHLS
C-----------------------------------------------------------------------ORTHLS
   16 DO 17 II=1,K                                                      ORTHLS
      C(II)=0.0                                                         ORTHLS
      ALPHA (II)=0.0                                                    ORTHLS
   17 BETA(II)=0.0                                                      ORTHLS
      C(K+1)=0.0                                                        ORTHLS
      IND1=-1                                                           ORTHLS
      RETURN                                                            ORTHLS
      END                                                               ORTHLS
```

II, D, Program Lising (cont.)

@ ELT SUB30,1,700/07, 30008

```
      SUBROUTINE COEFS (J,C,ALPHA,BETA,KC,A,T1,T2,T3,IND2)              COEFS
C-----------------------------------------------------------------------COEFS
C     THIS SUBROUTINE COMPUTES THE A COEFFICIENTS FOR A POLYNOMIAL      COEFS
C     OF DEGREE KC WHERE KC IS LESS THAN OR EQUAL TO K.                 COEFS
C-----------------------------------------------------------------------COEFS
      DIMENSION C(KC),ALPHA(KC),BETA(KC),A(KC),T1(KC),T2(KC),T3(KC)     COEFS
C-----------------------------------------------------------------------COEFS
C     PROGRAM INITIALIZATION.                                           COEFS
C-----------------------------------------------------------------------COEFS
      KCJ1=KC-J+1                                                       COEFS
      IF(KCJ1.LE.0) GO TO 9                                             COEFS
      B=0.0                                                             COEFS
      DO 1 NN=1,KCJ1                                                    COEFS
      A(NN)=C(NN)                                                       COEFS
      T1(NN)=0.0                                                        COEFS
      T2(NN)=0.0                                                        COEFS
    1 T3(NN)=0.0                                                        COEFS
      IF (KC.LE.J) GO TO 5                                              COEFS
      II=2                                                              COEFS
C-----------------------------------------------------------------------COEFS
C     BEGIN COMPUTATION.                                                COEFS
C-----------------------------------------------------------------------COEFS
    2 T2(II)=1.0                                                        COEFS
      DO 3 NN=2,II                                                      COEFS
      T3(NN)=T2(NN-1)-T2(NN)*ALPHA(II-1)-B*T1(NN)                       COEFS
C-----------------------------------------------------------------------COEFS
C     COMPUTATION OF AN A COEFFICIENT.                                  COEFS
C-----------------------------------------------------------------------COEFS
    3 A(NN-1)=A(NN-1)+C(II)*T3(NN)                                      COEFS
      IF (II.GE.KCJ1) GO TO 5                                           COEFS
C-----------------------------------------------------------------------COEFS
C     RESETTING THE VECTORS FOR THE NEXT COEFFICIENT.                   COEFS
C-----------------------------------------------------------------------COEFS
      DO 4 NN=1,II                                                      COEFS
      T1(NN)=T2(NN)                                                     COEFS
    4 T2(NN)=T3(NN)                                                     COEFS
      B=BETA(II-1)                                                      COEFS
      II=II+1                                                           COEFS
      GO TO 2                                                           COEFS
    5 IF (J.LE.0) GO TO 8                                               COEFS
C-----------------------------------------------------------------------COEFS
C      ARRANGE COEFFICIENTS PROPERLY IF J IS NON ZERO.                  COEFS
C-----------------------------------------------------------------------COEFS
      DO 6 NN=1,KCJ1                                                    COEFS
      N1=KCJ1-NN+1                                                      COEFS
      N2=N1+J                                                           COEFS
    6 A(N2)=A(N1)                                                       COEFS
      DO 7 NN=1,J                                                       COEFS
    7 A(NN)=0.0                                                         COEFS
C---------------------------------------------------------------------- OEFS
C     SUCCESSFUL RETURN.                                                OEFS
C---------------------------------------------------------------------- OEFS
    8 IND2=+2                                                           OEFS
      RETURN                                                            OEFS
C---------------------------------------------------------------------- OEFS
```

II, D, Program Listing (cont.)

```
C       ERROR RETURN.  SET ALL THE A COEFFICIENTS EQUAL TO ZERO.          COEFS
C-------------------------------------------------------------------------COEFS
    9 DO 10 NN=1,KC                                                       COEFS
   10 A(NN)=0.0                                                           COEFS
      A(KC+1)=0.0                                                         COEFS
      IND2=-2                                                             COEFS
      RETURN                                                              COEFS
      END                                                                 COEFS
```

II, D, Program Listing (cont.)

```
@ ELT BESJ,1,690714, 35959

      SUBROUTINE BESJ(X,N,BJ,D,IER)
      BJ=.0
      IF(N)10,20,20
   10 IER=1
      RETURN
   20 IF(X)30,30,31
   30 IER=2
      RETURN
   31 IF(X-15.)32,32,34
   32 NTEST=20.+10.*X-X** 2/3
      GO TO 36
   34 NTEST=90.+X/2.
   36 IF(N-NTEST)40,38,38
   38 IER=4
      RETURN
   40 IER=0
      N1=N+1
      BPREV=.0
C         COMPUTE STARTING VALUE OF M
      IF(X-5.)50,60,60
   50 MA=X+6.
      GO TO 70
   60 MA=1.4*X+60./X
   70 MB=N+IFIX(X)/4+2
      MZERO=MA
      IF(MA-MB)80,90,90
   80 MZERO=MB
C         SET UPPER LIMIT OF M
   90 MMAX=NTEST
  100 DO 190 M=MZERO,MMAX,3
C         SET F(M),F(M-1)
      FM1=1.0E-28
      FM=.0
      ALPHA=.0
      IF(M-(M/2)*2)120,110,120
  110 JT=-1
      GO TO 130
  120 JT=1
  130 M2=M-2
      DO 160 K=1,M2
      MK=M-K
      BMK=2.*FLOAT(MK)*FM1/X-FM
      FM=FM1
      FM1=BMK
      IF(MK-N-1)150,140,150
  140 BJ=BMK
  150 JT=-JT
      S=1+JT
  160 ALPHA=ALPHA+BMK*S
      BMK=2.*FM1/X-FM
      IF(N)180,170,180
  170 BJ=BMK
  180 ALPHA=ALPHA+BMK
      BJ=BJ/ALPHA
      IF(ABS(BJ-BPREV)-ABS(D*BJ))200,200,190
```

II, D, Program Listing (cont.)

```
190 BPREV=BJ
    IER=3
200 RETURN
    END
```

II, D, Program Listing (cont.)

```
* ELT BESY,1,690714, 35961

          SUBROUTINE BESY(X,N,BY,IER)
C            CHECK FOR ERRORS IN N AND X
          IF(N)180,10,10
       10 IER=0
          IF(X)190,190,20
       20 P1=3.141592653
C            BRANCH IF X LESS THAN OR EQUAL 4
          IF(X-4.)40,40,30
C          COMPUTE Y0 AND Y1 FOR X GREATER THAN 4
       30 T=4./X
          P0=.3989422793
          Q0=-.0124669441
          P1=.3989422819
          Q1=.0374008364
          A=T*T
          B=A
          P0=P0-.0017530620*A
          Q0=Q0+.0004564324*A
          P1=P1+.0029218256*A
          Q1=Q1-.00063904*A
          A=A*A
          P0=P0+.00017343*A
          Q0=Q0-.0000869791*A
          P1=P1-.000223203*A
          Q1=Q1+.0001064741*A
          A=A*B
          P0=P0-.0000487613*A
          Q0=Q0+.0000342468*A
          P1=P1+.0000580759*A
          Q1=Q1-.0000398708*A
          A=A*B
          P0=P0+.0000173565*A
          Q0=Q0-.0000142078*A
          P1=P1-.000020092*A
          Q1=Q1+.00001622*A
          A=A*B
          P0=P0-.0000037043*A
          Q0=Q0+.0000032312*A
          P1=P1+.0000042414*A
          Q1=Q1-.0000036594*A
          A=SQRT(2.*PI)
          B=4.*A
          P0=A*P0
          Q0=B*Q0/X
          P1=A*P1
          Q1=B*Q1/X
          A=X-PI/4.
          B=SQRT(2./(PI*X))
          Y0=B*(P0*SIN(A)+Q0*COS(A))
          Y1=B*(-P1*COS(A)+Q1*SIN(A))
          GO TO 90
C           COMPUTE Y0 AND Y1 FOR X LESS THAN OR EQUAL TO 4
       40 XX=X/2.
          X2=XX*XX
          T=ALOG(XX)+.5772156649
```

```
      SUM=0.
      TERM=T
      Y0=T
      DO 70 L=1,15
      IF(L-1)50,60,50
   50 SUM=SUM+1./FLOAT(L-1)
   60 FL=L
      TS=T-SUM
      TERM=(TERM*(-X2)/FL**2)*(1.-1./(FL*TS))
   70 Y0=Y0+TERM
      TERM = XX*(T-.5)
      SUM=0.
      Y1=TERM
      DO 80 L=2,16
      SUM=SUM+1./FLOAT(L-1)
      FL=L
      FL1=FL-1.
      TS=T-SUM
      TERM=(TERM*(-X2)/(FL1*FL))*((TS-.5/FL)/(TS+.5/FL1))
   80 Y1=Y1+TERM
      PI2=2./PI
      Y0=PI2*Y0
      Y1=-PI2/X+PI2*Y1
C        CHECK IF ONLY Y0 OR Y1 IS DESIRED
   90 IF(N-1)100,100,130
C        RETURN EITHER Y0 OR Y1 AS REQUIRED
  100 IF(N)110,120,110
  110 BY=Y1
      GO TO 170
  120 BY=Y0
      GO TO 170
C       PERFORM RECURRENCE OPERATIONS TO FIND YN(X)
  130 YA=Y0
      YB=Y1
      K=1
  140 T=FLOAT(2*K)/X
      YC=T*YB-YA
      K=K+1
      IF(K-N)150,160,150
  150 YA=YB
      YB=YC
      GO TO 140
  160 BY=YC
  170 RETURN
  180 IER=1
      RETURN
  190 IER=2
      RETURN
      END
```

II, D, Program Listing (cont.)

```
@ ELT BESSEL,1,690716, 76335

              SUBROUTINE BESSEL(J,Y,V,X,K)
              INTEGER V                                                    *NEW
              REAL J,Y                                                     **-1
              DIMENSION J(1),Y(1)
              D=1.E-6                                                      *NEW
              NV=V+1
              DO 1 I=1,NV
              CALL BESJ(X,I-1,J(I),D,IER )
              IF(IER.NE.0) GO TO 10
              CALL BESY(X,I-1,Y(I),IER)                                    *NEW
              IF(IER.NE.0) GO TO 11                                        **-1
        1     CONTINUE
              RETURN
        10    WRITE(6,100)X,V
              GO TO 12
        11    WRITE(6,101)X,V
        12    K=1
              RETURN
        100   FORMAT(10X,28HERROR IN BESJ, X AND V ARE. ,2E15.7)
        101   FORMAT(10X,28HERROR IN BESY, X AND V ARE. ,2E15.7)
              END
```

II, Programming (cont.)

E. METHODS OF VERIFICATION

For most sections of this program, the only method of verification is the "reasonableness" of the numbers output. This is a nebulous statement and the only way an engineer can know what is reasonable is from experience with the sensitive time lag theory and its application. There are some guidelines that can be given, however, that may help those new to the program.

1. If $A_{\nu\eta}$, $B_{\nu\eta}$, and $C_{\nu\eta}$ are 1.0, the n minimum will be on the order of 0.5 to 1.0. It will always be positive.

2. The frequency of the n minimum is near the acoustic mode frequency of a cylinder. This frequency is given in a formula on Figure 2.

3. The calculation of $A_{\nu\eta}$, $B_{\nu\eta}$, and $C_{\nu\eta}$ can be checked by running a case with combustion concentrated at a particular location. The answer can then be easily checked using the formulas in Section I,B,4,b (5) and Figure 8.

4. When τ at n minimum is given in seconds, it can be converted to an equivalent frequency by the relation $\tau^* = \frac{1}{2f^*}$.

5. The test cases given in this manual are a convenient reference to establish whether the program and computer are working correctly.

III. DECK SETUP

A. COMPUTER CONFIGURATION

1. Univac 1108 computer, 65K of core minimum
2. 46211_8 for code, 72245_8 for data
3. FORTRAN V
4. Executive II monitor system
5. No plot output required
6. No punch output required
7. Units 12, 13, 14 are used for temporary storage.

B. ESTIMATED RUNNING TIME

Because of the extremely large number of ways of running this program, it would be very difficult to give a formula for running time. As a guide, the sample case given in Section III,L ran in 2.30 minutes. The nozzle admittance calculation is the single most time consuming portion of the program taking about 80% of the above time.

III, Deck Setup (cont.)

C. DECK SEQUENCE

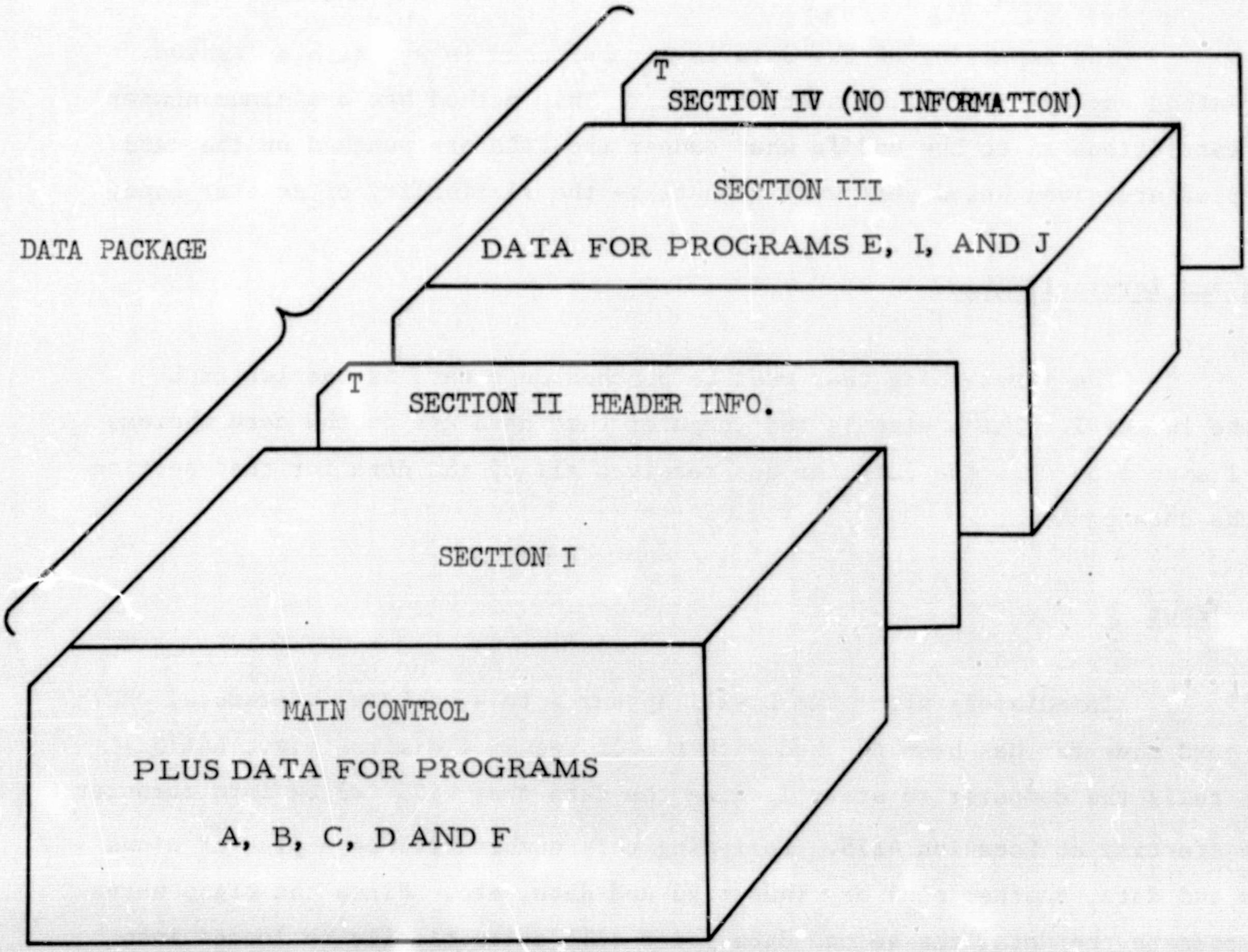
It is necessary to discuss the general organization of the data as well as how the data actually get into the computer before proceeding with any further discussion of the computer program. Hereafter, the input data deck will be referred to as a set of data packages.

The manner in which the data deck is assembled is shown in Figure 17. It can be seen that the data package (one package required per case), is divided into four sections. The first section contains the data for MAIN CONTROL and for Programs A, B, C, D and F. The second section is one card that has the letter T punched in card column one. This card serves the purpose of an end-data signal and of a header card such that all information that follows the T will be printed on the top of each output page. The third section contains the data for Programs E, I, and J. The fourth section is a T-card that is used only to signal the end of that data.

It should be noted that if programs E, I, or J are not run, Sections III and IV are not needed. Sections I and II, however, must always be in the data package.

Running Multiple Cases

The flexibility of the program is such that stacking cases (i.e., running multiple cases) is an easy task. All that is required is to add the required number of data packages to the data deck. There is no restriction as to the flow paths of individual data packages; therefore, data packages of parametric studies involving the use of one program can be intermingled with data packages that use any of the other program combinations for an analysis.



IF SECTION III IS NOT USED, SECTION IV MUST NOT BE USED.

Figure 17 -- Input Load Sequence Required by the Computer Program

III, C, Deck Sequence (cont.)

Getting the Data Onto the Card

The inputting of the data is accomplished in a flexible fashion. The method employed is called scatter load.* This method has a minimum number of restrictions as to how and in what manner the data are punched on the card. Examples are given below that will illustrate the flexibility of scatter load.

Load and Terminal Flags

The first thing that must be punched on a card is the letter L or the letter T. The L signals the computer that data are on the card whereas the T signifies that the computer has received all of the data for that section of the data package.

Data Input

Immediately after the L will appear 1 to 4 numbers; therefore, the card thus far has been punched with L followed by 4 digits, e.g., L4175. This tells the computer to start loading the data that will follow into computer core starting at Location 4175. Following this number will be a plus or minus sign and data, another plus or minus sign and data, etc. Since the signs serve to separate the data, the second data point will automatically be loaded into core location 4176. This process continues across the first 72 columns of the data card.

Consider the following example of input data

---

*Except for the input to main control.

III, C, Deck Sequence (cont.)

L0 + 10.0
L1 + 19.4
L2 + 0.0
L3 - 0.92
L8 + 6.4

This may be punched on a single card as

(Example 1) L0 + 10.0 + 19.4 + 0.0 - 0.92 L8 + 6.4
(Example 2) L0 + 10.0 + 19.4 + -0.92 + + + + + 6.4
(Example 3) L8 + 6.4 L0 + 10.0 + 19.4 + -0.92
(Example 4) L39 + 0.21 + 3.14 + 5.0 + 1.5 -35.5 L39 + 1545.

Example 1 illustrates a normal load sequence for the given input. No data are transferred into core locations L4 thru L7; therefore, any previous data stored there remains there. Example 2 illustrates how algebraic signs can be used to index the core location to the proper station. Note also the L2, which is identically zero, has no number punched on the card. This illustrates that the computer, when it sees no number following the sign, loads the value of zero into that core location. If data have been previously stored in Locations L4 thru L7 and are to be used again, the use of indexing presented in Example 2 will load zeros into those core locations; consequently, the data will be lost.

Example 3 illustrates that the L-numbers do not have to be punched in sequential order. Example 4 illustrates the method of correcting data. Specifically, L39 was initially loaded with 0.21 and later loaded with 1545.0; therefore, the former number is erased on core and the latter included in its place. The most convenient method for correcting data cards is to punch all the corrections on one or more cards and place these cards immediately before the T-card of that data section.

III, C, Deck Sequence (cont.)

Presented below are some basic rules to employ when punching data cards:

1. DO start each input card with L. The L does not have to be punched in card column 1 but it must appear before the data.

2. DO pay close attention to the sequence of the remaining data included on the card so that data are not loaded into the wrong core location.

3. DO NOT use more than 8 significant digits (including decimal point) for the input data. Exponential notion is permitted. That is, the number 0.0625 can be loaded as 6.25 E-02.

4. DO NOT start the data on one card and complete it on another.

5. DO NOT use more than 72 card columns for data input. The computer looks at 73 to 80 but does not transfer that information to core. Therefore, columns 73 and 80 can be used for the card sequence number in the data deck or identification of some significant aspect of the data so that it can be identified at some later date.

6. DO include a T-card at the end of a data section in the data package. The T <u>must</u> be punched in card column 1. <u>If the T appears anywhere else on the card, the computer will dump the entire run.</u>

7. DO include a description of the case on the T-card of Section 2. This information serves as a header for each output page. This is a convenience more than a necessity.

8. DO NOT use two or more T cards for a header. This can confuse the computer's input logic.

Thus far the storage locations for the individual data bits have not been noted. They will be provided in Section III,E. It is sufficient to state here that the input data have reserved locations in core and that these data remain in those locations unless over-written by the next set of input data. Therefore, it is not necessary to input all the data on the next run if only one parameter (e.g., the ratio of the specific heats) is to be changed. Only the data that changes from one case to the next have to be input on successive data packages (except for main control).

Operation of MAIN CONTROL

Since the computer obeys every command explicitly, it is necessary to indicate correctly to the computer the programs that are desired for a particular case. Furthermore, this must be done for every case because the computer will turn off the switch that activated the program after it is through with the program. This is done purposely to avoid using a program that is not needed for the second case.

The first 10 core locations are reserved for MAIN CONTROL in the following order:

L0: Program A, Distributed Combustion Analysis (Transverse Modes Only)

L1: Program B, Concentrated Combustion Analysis

L2: Program C, Exhaust Nozzle Admittance Coefficients for Longitudinal and Transverse Modes

III, C, Deck Sequence (cont.)

L3: Program D, Expansion of Results from Program B.
L4: Program E, Injector Nonuniformity Coefficients
L5: Program F, Final Solution of Instability Zones
L6: Program G, Not used in this package
L7: Program H, Not used in this package.
L8: Program I, Nonlinear Combustion Response Analysis
L9: Program J, Injected Mass Distribution Effects

A number greater than zero instructs the computer to execute this program at the time it is required. The program knows when to execute the program and a number in the proper place tells the computer whether or not to execute the program. Running Program A or B automatically runs D and F as well. This is done to simplify the operation of this program.

Program Options

Most programs have various options concerning the desired output to be printed. These options are exercised by the same control number that executes the program and are keyed by the magnitude of the number. These print options are presented in the discussion of the individual programs. However, a few examples are presented here to illustrate this point:

EXAMPLE 1: EXECUTE PROGRAM C - PRINT NO OUTPUT

LO + + + 9.0 + + + + + + (0<L2≤9.0)

EXAMPLE 2: EXECUTE PROGRAM C - PRINT OUTPUT

LO + + + 99.0 + + + + + + + (10<L2 ≤ 99.0)

EXAMPLE 3: EXECUTE PROGRAM C - PRINT INPUT AND OUTPUT

L0 + + + 199.0 + + + + + + + + (100 $<$ L2 $\leq$ 199.0)

D. INPUT DATA

1. Program A: Distributed Combustion Stability Analysis

a. Input Requirements: The following data go into Section I of data package.

Core Location	Mnemonic	Description
L0	--	To execute Program A L0 must be a positive number. The magnitude of the number determines the print options as shown in paragraph III,D,1,b.
L10	γ	Ratio of the specific heats.
L11	M_e	Steady-state Mach number at the entrance to the exhaust nozzle. This can be determined by the contraction ratio or by Program C. In the latter case, leave L11 blank.
L13	r_c^*	Chamber radius, in. also loaded in L3804, Program C.
L14	L_c^*	Length of cylindrical portion of chamber, in.
L15	c_o^*	Speed of sound in the chamber, ft/sec.
L16	$\bar{u}_{L0}^*$	Weighted liquid injection velocity, ft/sec. See Equation (75) at the end of this section.
L17	$\varkappa$	Gas/liquid momentum interchange coefficient. ($\varkappa = 0$ for no droplet momentum effects)
L20	$s_{\nu\eta}$	Mode number. These numbers are given in Section III,D,1,e, figure 9. Must not be zero for this program.
L21	$N_{\omega c}$	Number of chamber frequencies to be used. Leaving this column blank will turn on Program GENMEC which will select 10 frequencies according to the relationship given by Eq. (76).
L22 . . . . . . L49	ω	Table of nondimensional, chamber frequencies arranged in ascending order. A zero must be included at the end of the table. The maximum number of frequencies is 28 (including the zero point). This is left blank if L21 is blank.

III, D, Input Data (cont.)

Core Location	Mnemonic	Description
L50 . . . L69	Z*	Table of ascending axial postions at which the combustion distribution (i.e., the variation of the local Mach number with respect to Z) is known. Dimensioned in inches. A zero value must appear at the last point. Minimum of four real data points is required.
L70 . . L89	M(Z*)	The local steady-state Mach number that corresponds to the given L_z. A zero <u>must</u> appear as the last value.
L3017	N_ω	The number of input frequencies at which the nozzle admittance coefficients are known. This number must be greater than 2 and less than 30. If the admittance for a nozzle is unknown and Program C is run, C will provide this number.
L3019 . . . L3048	ω	The ascending table of chamber frequencies for which the real and imaginary parts of the nozzle admittance coefficient $\mathcal{E}_r$, $\mathcal{E}_i$. If Program C is run to determine the admittances, these frequencies will be automatically transferred. The last point in the table must be zero.
L3049 . . . L3078	$\mathcal{E}_r$	The real part of the admittances corresponding to the above frequencies. The last point in the table must be zero. These can be determined by running Program C.
L3079 . . . L3108	$\mathcal{E}_i$	The imaginary part of the admittance corresponding to the above frequencies. The last point in the table must be zero. These can be determined by running Program C.

b. Print Options

$0 \leq L0 \leq 99$: Execute A, print output

$100 \leq L0 \leq 100$: Execute A, print input and output (recommended)

$200 \leq L0 \leq 299$: Execute A, print input and output, print steady state tables and values of integrals used in calculations.

c. Placement of the Data into the Data Package

These data are inserted anywhere in Section I of the data package.

d. Output

Label in Output	Symbol in Analysis
ULM	$\bar{u}^*_{LO}$
XK	$\varkappa$
OMEGA (CH)	ω
ERT	$\mathcal{E}_r$
EIT	$\mathcal{E}_i$
H REAL	h_r
H IMAG	h_i

e. Tabulation of Transverse Acoustic Mode Number ($s_{\nu\eta}$)

(1) Tangential Modes

First tangential: $s_{11} = 1.8413$
Second tangential: $s_{21} = 3.0543$
Third tangential: $s_{31} = 4.2012$
Fourth tangential: $s_{41} = 5.3175$
Fifth tangential: $s_{51} = 6.4154$

(2) Radial Modes

First radial: $s_{02} = 3.8317$
Second radial: $s_{03} = 7.0156$
Third radial: $s_{04} = 10.1734$

III, D, Input Data (cont.)

2T-1R: s_{22} = 6.7060
2T-2R: s_{23} = 9.9695
2T-3R: s_{24} = 13.1705
3T-1R: s_{32} = 8.0151
3T-2R: s_{33} = 11.3459
3T-3R: s_{34} = 14.5858

f. Auxiliary Equations

(1) Axial Liquid Velocity (Weighted)

$$\bar{u}^*_{L0} = \frac{(MR)\nu_x \cos\theta_x + \nu_F \cos\theta_F}{MR+1}, \text{ ft/sec} \qquad (75)$$

MR = mixture ratio = $\dot{w}_x/\dot{w}_F$
ν_x = oxidizer injection velocity, ft/sec
ν_F = fuel injection velocity, ft/sec
$\theta_{x,F}$ = oxidizer and fuel impingement angle

(2) Selection of Frequencies by GENMEG

$$(\text{Transverse}) = (1 \pm 0.10)\, s_{\nu\eta} \qquad (76)$$

where $s_{\nu\eta}$ = transverse acoustic mode number given in section III,D,1,e.

g. Comments:

When L0 is made positive and all the other required input for A is supplied an n, τ plot for uniform injection is produced. This can be modified to include non-uniform injection or non-linear effects by making L3 or L4 positive with the corresponding data for programs D and E.

III, D, Input Data (cont.)

2. Program B: Concentrated Combustion Stability Analysis

a. Input Requirements: the following data go into Section I of the data package.

Core Location	Mnemonic	Description
L1	--	Execute Program B. Magnitude of number depends on print options given in Paragraph III,D,2,b.
L10	γ	Ratio of the specific heats.
L11	M_e	Steady-state Mach number at the entrance to the exhaust nozzle. This can be determined by the contraction ratio or by Program C. In the latter case, leave L11 blank.
L13	r_c^*	Chamber radius, in. also loaded in L3804, Program C.
L14	L_c^*	This length is used to determine the frequency of a longitudinal mode. $\frac{\omega^* L_c^*}{c_o^*} = \pi$ determining the resonant frequency. It may not correspond to any physical dimension on the chamber. Table I can be used to estimate the longitudinal mode frequency. L_c^* is used as a non-dimensionalizing factor for longitudinal modes ($s_{\nu\eta}=0$).
L15	c_o^*	Speed of sound in the chamber, ft/sec.
L16	$\bar{u}^*_{LO}$	Weighted liquid injection velocity, ft/sec (see Equation (75).
L20	$s_{\nu\eta}$	Mode number. These numbers are given in section III,D,2,e,and Figure 9.
L21	$N_{\omega c}$	Number of chamber frequencies to be used. Leaving this column blank will turn on Program GENMEC which will select 10 frequencies according to the relationship given by Equation (77). GENMEG gives first mode only.
L22 . . . . . L49	ω	Table of nondimensional, chamber frequencies arragged in ascending order. A zero must be included at the end of the table. The maximum number of frequencies is 28 (including the zero point). This is left blank if L21 is blank.
L50	$\bar{P}_c^*$	Mean chamber pressure in pounds per square inch.
L51	L_{jc}^*	Length from injector to combustion front in inches.
L53	$\bar{W}^*$	Mean chamber weight flow in pounds per second.

III, D, Input Data (cont.)

Core Location	Mnemonic	Description
L3017	N_ω	The number of input frequencies at which the nozzle admittance coefficients are known. This number must be greater than 2 and less than 30. If the admittance for a nozzle is unknown and Program C is run, C will provide this number.
L3019 ⋮ L3048	ω	The values of the chamber frequencies, arranged in ascending order, for which the admittances are known. Program C will provide them if it is run. The last value must be zero.
L3049 ⋮ L3078	$\mathcal{A}_r$ or $\mathcal{E}_r$	The real part of the nozzle admittance coefficient. If $s_{\nu\eta} = 0$ use $\mathcal{A}_r$, otherwise use $\mathcal{E}_r$. These values must correspond to the frequencies given above. The last point in the table must be zero. Program C will provide these values if it is run.
L3079 ⋮ L3108	$\mathcal{A}_i$ or $\mathcal{E}_i$	Imaginary part of the nozzle admittance coefficient. If $s_{\nu\eta} = 0$ use $\mathcal{A}_i$ otherwise use $\mathcal{E}_i$. These values must correspond to the frequencies given above. The last point in the table must be zero.
L3109 ⋮ L3138	C_r	The real part of the entropy nozzle admittance coefficient. These values correspond to the frequencies given above. The last point in the table must be zero. Program C will provide these values if it is run.
L3139 ⋮ L3168	C_i	The imaginary part of the entropy nozzle admittance coefficient. These values correspond to the frequencies given above. The last point in the table must be zero. Program C will provide these values if it is run.

b. Print Options

$0 \leq L1 \leq 99$: Execute B, print output

$100 \leq L1 \leq 199$: Execute B, print input and output (recommended)

$200 \leq L1 \leq 299$: Execute B, print input and output, and matrix before inversion.

c. Placement of the Data into the Data Package

These data are inserted anywhere in Section I of the data package.

d. Output

Label in Output	Symbol in Analysis
GAMMA	γ
DESIRED MACH NUMBER	M_e
CHAMBER RADIUS	R^*_{AC}
CHAMBER LENGTH	L^*_c
SPEED OF SOUND	c^*_o
CHAMBER MODE DESCRIPTION	$s_{\nu\eta}$
CHAMBER FREQUENCIES	ω
WC	ω
LJC	L_{jc}
SNH	$s_{\nu\eta}$
ZE	L_c
UE	M_e
SOUND	c^*_o

e. Tabulation of Transverse Acoustic Mode Number ($s_{\nu\eta}$)

(1) Tangential Modes

First tangential: $s_{11} = 1.8514$
Second tangential: $s_{21} = 3.0543$
Third tangential: $s_{31} = 4.2012$
Fourth tangential: $s_{41} = 5.3175$
Fifth tangential: $s_{51} = 6.4154$

(2) Radial Modes

First radial: $s_{02} = 3.8317$
Second radial: $s_{03} = 7.0156$
Third radial: $s_{04} = 10.1734$

(3) Combined tangential-radial modes

1T-1R: $s_{12} = 5.3313$
1T-2R: $s_{13} = 8.5263$
1T-3R: $s_{14} = 11.7059$
2T-1R: $s_{22} = 6.7060$
2T-2R: $s_{23} = 9.9695$
2T-2R: $s_{24} = 13.1705$
3T-1R: $s_{32} = 8.0151$
3T-2R: $s_{33} = 11.3459$
3T-3R: $s_{34} = 14.5858$

(4) Longitudinal Mode

$s_{\nu\eta} = 0$

III, D, Input Data (cont.)

e. Auxiliary Equations

(1) Axial Liquid Velocity (Weighted)

$$\bar{u}^*_{L0} = \frac{(MR)\nu_x \cos \theta_x + \nu_F \cos \theta_F}{MR+1}, \text{ ft/sec} \qquad (75)$$

MR = mixture ratio = $\dot{w}_x/\dot{w}_F$

ν_x = oxidizer injection velocity, ft/sec

ν_F = fuel injection velocity, ft/sec

$\theta_{x,F}$ = oxidizer and fuel impingement angle

(2) Selection of Frequencies by GENMEG

$$\begin{aligned} &(\text{Longitudinal}) = (1 \pm 0.10)\ \pi \\ &(\text{Transverse}) = (1 \pm 0.10)\ s_{\nu\eta} \end{aligned} \qquad (77)$$

where

π = 3.14159 + = resonant frequency (non-dimensional) for longitudinal modes

$s_{\nu\eta}$ = transverse acoustic mode number given in the Program B writeup.

(3) Other Information

For a longitudinal mode a resonant frequency of π is true for a very short nozzle. It may be necessary to override GENMEG to find an n minimum. Also GENMEG gives only the first mode. Other modes, such as 1T-1L, can be obtained be specifying the frequencies to be run.

When L1 is made positive and all other required data for program B is supplied an n, τ plot for uniform injection is produced. This can be modified to include non-uniform injection effects or non-linear effects by making L3 or L4 positive with the corresponding data for programs D and E.

L101 in the nozzle input is generally required with Program B.

III, D, Input Data (cont.)

(4) Estimate of Chamber Weight Flow

$$\bar{w}^* = \frac{\bar{P}_c A_{throat} g}{C_{star}}$$

$$C_{star} = \frac{\bar{C}^*_o}{\gamma \left(\frac{2}{\gamma+1} \right)^{\frac{\gamma+1}{2(\gamma-1)}}}$$

III, D, Input Data (cont.)

3. Program C: Exhaust Nozzle Admittance Coefficients for Longitudinal and Transverse Modes

a. Input Requirements: The following data go into Section I of the data package.

Core Location	Mnemonic	Description
L2		Run program C if positive.
L3801	M	= 1.: the table of velocity potential values within the nozzle is input and the Mach number to the entrance of the nozzle is input. = 2.: the Mach number at the entrance to the nozzle is input but the velocity potential table must be calculated. = 3.: the Mach number and the velocity potential table must be calculated.
L3803	R^*_{ATO}	Radius of the throat, in.
L3804	R^*_{ACO}	Radius of the chamber, in., also loaded in L13, Program A.
L3805	R^*_{CCO}	Radius of chamber curvature at the nozzle entrance, in.
L3806	R^*_{CTO}	Radius of curvature at the throat, in.
L3807	α_o	Nozzle convergent half-angle, deg.
L90*	R^*_{ATi}	Radius of centerbody throat, in.
L91*	R^*_{CTi}	Radius of curvature of the centerbody throat, in.
L101		Length from end of nozzle to combustion zone when using Program B (zero for Program A).

---

*Note 1: These values may be plus or minus, see Figures 14 and 15.
Note 2: For cylindrical chambers, leave L90 through L94 blank.

Core Location	Mnemonic	Description Description
L92*	R^*_{ACi}	Radius of chamber of centerbody, in.
L93*	R^*_{CCi}	Radius of curvature of centerbody at the nozzle entrance.
L94*	α_i	Centerbody nozzle convergent half-angle, deg.
L20	$s_{\nu\eta}$	Transverse acoustic mode number is given on the previous page for cylindrical chambers. For annular chambers see Table 1, Section I,B.
L3808	KN	If L3801 $\geq$ 2, then KN is either (1) An odd integer less than 200 telling the desired size of the program generated velocity potential table, or (2) Blank and program will assume K_N = 101 (recommended)
L3809 L3811 L3813 . L4205	$\phi(Z)$	Dimensionaless velocity potential table (199 values maximum) in the odd numbered locations only. If L3801 $\geq$ 2, this is not necessary.
L3810 L3812 L3814 . L4206	$\bar{q}^2(Z)$	Squares of the reduced velocity table (199 values maximum) in the even numbered locations only. If L3801 $\geq$ 2, this not necessary.

---

*Note 1: These values may be plus or minus, see Figures 14 and 15.
Note 2: For cylindrical chambers, leave L90 through L94 blank.

When this program is run by itself the frequencies to be run must either be input or generated by GENMEG. As discussed later this admittance calculation is done for every other input or generated frequency. To run Program C by itself the above data must be supplemented as follows.

Core Location	Mnemonic	Description
L10	γ	Ratio of specific heats
L11	M_e	Steady state Mach number at the entrance to the nozzle. Not needed if L3801 is 3.

III, D, Input Data (cont.)

Core Location	Mnemonic	Description
L13	r_c^*	Chamber radius, in.
L14	L_c^*	Chamber length, in.
L15	c_o^*	Speed of sound in the chamber, ft/sec.
L21	$N_{\omega c}$	Number of chamber frequencies to be used leaving this column blank will turn on Program GENMEG. Admittances will actually be calculated for every other frequency as discussed later.
L22 • L49	ω	Table of non-dimensional chamber frequencies arranged in ascending order. Last point must be zero.

b. Print Options

$0 < L2 \leq 9$:	Execute Program C, print no output
$10 \leq L2 \leq 99$:	Execute Program C, print output
$100 \leq L2 \leq 199$:	Execute Program C, print input and output (recommended)
$200 \leq L2 \leq 299$:	Execute Program C, print input and output, and print the nondimensional velocity potential table.

c. Placement of the Data in the Data Package

This data is inserted anywhere in Section I of the data package.

d. Output

Label In Output	Symbol In Analysis
(SNH)C	$s_{\nu\eta}$
(SNH)N	$\hat{s}_{\nu\eta}$
WC	ω
WN	$\hat{\omega}$

III, D, Input Data (cont.)

Label In Output	Symbol In Analysis
MACH NO.	M_e
G	γ
AR	A_r
AI	A_i
-AR/(MACH NO.)	α_{Nr}
-AI/(MACH NO.)	α_{Ni}
BR	B_r
BI	B_i
T1	E_r
T2	E_i
CR	C_r
CI	C_i
-CR/(MACH NO.)	β_{Nr}
-CI/(MACH NO.)	β_{Ni}
FC (CPS)	f^*

e. Auxiliary Equations

(1) The Selection of the Chamber Frequencies to be Used

Because of the long execution time of this program for a given value of ω, the program starts with the first chamber frequency and uses every other frequency thereafter; that is, if n is the total number of chamber frequencies, Program C will use m number of frequencies according to the relationship

III, D, Input Data (cont.)

$$m = \frac{n}{2} + 1$$

In computer language, m and n are whole numbers.

Therefore, if n = 11

$$m = \frac{11}{2} + 1$$

$$m = 5 + 1 \text{ (where 0.5 has been truncated out)}$$

$$m = 6$$

(2) Nondimensional chamber frequency

$$\omega = \frac{(2\pi f^*)\ (\nu^*_{AC})}{c^*_o} \tag{78}$$

where

$*$ = denotes dimensional variables
subscript c denotes chamber conditions

f^* = chamber frequency, cps

c^*_o = chamber speed of sound ft/sec

ν_{AC} = chamber length L^*_c or radius r^*_c (depending if longitudinal or transverse modes respectively are desired).

(3) Nondimensional nozzle frequency

$$\hat{\omega} = \frac{\omega}{\hat{\kappa}} \tag{79}$$

where

$$\hat{\kappa} = R^*_{ATO} \left[\frac{2}{\gamma+1} \left(\frac{R^*_{ATo}}{R^*_{CTo}} - \frac{R^*_{ATi}}{R^*_{CTi}} \right) \Big/ \left({R^*_{ATo}}^2 - {R^*_{ATi}}^2 \right) \right]^{1/2}$$

(4) Nozzle transverse acoustic nozzle number

$$\hat{s}_{\nu\eta} = \frac{s_{\nu\eta}}{\hat{\kappa}} \tag{80}$$

4. Program D: Expansion of Results from Program B

a. Input Requirements

The following data go into Section I of the data package:

Core Location	Mnemonic	Description
L3		Must be greater than zero to run this program. Use only if $A_{\nu\eta}$, $B_{\nu\eta}$, $C_{\nu\eta}$ are input or calculated in Program E.
L3405	l_r/η	The ratio of the radial velocity coefficient to the pressure interaction index. Generally set equal to zero.
L3406	l_θ/η	The ratio of the tangential velocity coefficient to the pressure interaction index. Generally set equal to zero.
L3407		= 1 for linear combustion response. >1 for nonlinear combustion response. This parameter indicates the number of frequencies for which the injector nonuniformity coefficients, $A_{\nu\eta}$, $B_{\nu\eta}$, and $C_{\nu\eta}$, are known. The running of Program I will deterine this number. (Not required if Program B is run.)
L3408		The number of chamber frequencies (Not required if Program B is run.)
L3409 ⋮ L3437 ⋮	ω	The table of frequencies arranged in ascending order. The last value must be zero. (Not required if Program B is run.)
L3439 ⋮ L3467 ⋮	$h_r(\omega)$	Real part of the damping effects computed in Program A or B. The last value must be zero. (Not required if Program A or B is run.)
L3469 ⋮ L3497 ⋮	$h_i(\omega)$	Imaginary part of the damping effects computed in Program B. The last value must be zero. (Not required if Program A or B is run.)
L4522 ⋮ L4538 ⋮	ω	The values of chamber frequencies for which the nonuniformity coefficients are known. Use only if data are available.

III, D, Input Data (cont.)

Core Location	Mnemonic	Description
L4539 ⋮ L4555	$A_{\nu\eta}$	The pressure nonuniformity coefficient corresponding to the above frequencies (ω). $A_{\nu\eta} = 1.0$ for uniform injection distribution.
L4556 ⋮ L4572	$B_{\nu\eta}$	The radial velocity nonuniformity coefficient corresponding to the above frequencies ω. Generally set equal to zero.
L4573 ⋮ L4589	$C_{\nu\eta R}$	The real part of the tangential velocity nonuniformity coefficient corresponding to the above frequencies ω. Generally set equal to zero.
L4590 ⋮ L4606	$C_{\nu\eta I}$	The imaginary part of the tangential velocity nonuniformity coefficient corresponding to the above frequencies ω. Generally set equal to zero.

NOTES: a. All these data, with the exception of l_r/η and l_θ/η, can be calculated using Programs B, E, and I.

b. This program was constructed specifically to include the effects of the nonuniformity coefficients for the low Mach number cases.

b. Print Options

$0 < L3 \leq 9$: Execute D, print no output or input

$10 \leq L3 \leq 99$: Execute D, print output only

$100 \leq L3 \leq 199$: Execute D, print input and output

c. Placement of the Data into the Data Package

These data can be inserted anywhere in Section I of the data package.

d. Output

Label on Output	Symbol in Analysis
LR/N	$\frac{l_r}{n}$
LT/N	$\frac{l_\theta}{n}$
ERT	$\mathcal{E}_r$
EIT	$\mathcal{E}_i$
CRT	$\mathcal{C}_r$
CIT	$\mathcal{C}_i$
OMEGA (C)	ω
OMEGA (CH)	ω
HTR	$\tilde{h}_r$
HTI	$\tilde{h}_i$
HTRINT	$\tilde{h}_r$ (interpolated)
HTIINT	$\tilde{h}_i$ (interpolated)
OMEGA	ω

5. Program E: Injector Nonuniformity Coefficients: $A_{\nu\eta}$, $B_{\nu\eta}$, $C_{\nu\eta}$

a. Input Requirements (not programmed for annular chambers)

MAIN CONTROL requires L4 in Section I of data package. The remaining data go into Section III of the data package.

Core Location	Mnemonic	Description
L1	IZZIT	Indicates whether the mode is standing or spinning: Standing: IZZIT <0 Spinning: IZZIT >0
L2	X_M	The number of radial divisions desired on the injector face ($10 \leq X_M \leq 21$).
L3	X_N	The number of angular divisions desired around the injector face ($20 \leq X_N \leq 181$).
L4	SECTOR	The number of symmetrical sectors that the injector can be divided into, ($1 \leq \text{SECTOR} \leq 180$).
L5	ν	Order of transverse mode in tangential direction 1, 2, 3, etc. For radial modes $\nu = 0$.
L7	$s_{\nu\eta}$	Transverse acoustic mode number given by tabulation presented in Program B write-up.
L9573	R_{inj}	Injector radius, in.
L9570	NE	Number of elements per symmetrical sector. The maximum number of elements is 1000. Ignore if Program J is run.
L1920 : L2919	r	The radial position of each element within the symmetrical sector, in. Program J supplies these data when they are run.
L2920 : L3919	θ	The angular displacement (from any convenient reference line on the injector face) of each element within the symmetrical sector, radians. Program J supplies these data when they are run.
L3920 : L4919	μ	The element injection distribution given by Equation (81) at the end of this section. Usually Program J provides this information to Program E.

III, D, Input Data (cont.)

b. Print Options

$0 \leq L4 \leq 9$: Execute E, do not print input or output
$10 \leq L4 \leq 99$: Execute E, print output only
$100 \leq L4 \leq 199$: Execute E, print input and output

c. Placement of the Data into the Data Package

Main control data: Section I of data package.

All other data: Section III of data package. Section IV must be included in the data package.

d. Output

Label on Output	Symbol in Analysis
AVN	$A_{\nu\eta}$
BVN	$B_{\nu\eta}$
CVN	$C_{\nu\eta}$
MU	μ

e. Auxiliary Equations

Distribution coefficient, μ_E

$$\mu_E = \frac{(\dot{w}_T)_s E/(A_s)_E}{\dot{w}_{T/A_{inj}}} = \left(\frac{1}{x_M \cdot x_N}\right)^{-1} \frac{(W_T)_E}{\dot{w}_T} \qquad (81)$$

where

$(w_T)_E$ = total weight flow rate of element, lb/sec

w_T = total weight flow rate of injector, lb/sec

$(A_s)_E$ = surface area serviced by the element, in.^{2}

A_{inj} = total surface area of injector, in.^{2}

6. Program F: Final Solution for Instability Zones

a. Input Requirements

The input for this program comes directly from Program D and consists of the following parameters:

The frequency in the chamber, ω

The radius of the chamber R^*_{AC}, in.

Speed of sound in the chamber, c^*_o, ft/sec

Real and imaginary parts of the damping parameters $\underset{\sim}{h}$ corresponding to the above frequencies.

This program can be run by itself if the data are inserted into those locations specified by Program D.

b. Print Options

$0 < L5 \leq 9$:	Execute Program F, print no input and output
$10 \leq L5 \leq 99$:	Execute Program F, print output only
$100 \leq L5 \leq 199$:	Execute Program F, print input and output

c. Placement of the Data into the Data Package

These data are included into Section I of the data package.

d. Output

The following output results from this program:

f^* = frequency of oscillation, Hz

ω = nondimensional frequency

τ^* = sensitive time lag corresponding to the above frequency, millisec

n = pressure interaction index

At the bottom of the page, an effort is made to locate the minimum interaction index. In most cases, this information is valid. However, effects from an adjacent mode will intervene occasionally thereby invalidating this information.

e. Other Information

The program obtains its result by the solution of the following equations:

Frequency, cps

$$f^* = \frac{\omega}{2\pi} \frac{c^*_o\,12}{R^*_{AC}} \text{ for transverse modes.}$$

for a longitudinal mode replace R^*_{AC} by L^*_c.

Pressure interaction index, n

$$n = \frac{\tilde{h}_r^2 + \tilde{h}_i^2}{2\tilde{h}_r}$$

Sensitive time lag, τ, millisec

$$\tau^* = (83.33)\,\frac{R^*_{ACo}}{c^*_o 12}\,\frac{1}{f^*}\,\tan^{-1}\frac{h_i}{n-h_r}$$

for a longitudinal mode replace R^*_{ACo} by L^*_c.

7. <u>Program G: Obsolete and deleted from the listing</u>.

8. <u>Program H: Obsolete and deleted from the listing</u>.

Program H is not operational. Recommended procedure is to place a negative number in load location L7 of MAIN CONTROL.

9. <u>Program I: Nonlinear Combustion Response (Option for Program E)</u>

a. Input Requirements

MAIN CONTROL requires L8 for Section I of data package. The remaining data go into Section III of data package:

III, D, Input Data (cont.)

ALL THE DATA REQUIRED TO RUN PROGRAM E AS WELL AS:

Core Location	Mnemonic	Description
L0	E1	The permissible error. If blank, the program will assume E1 = 0.001.
L12	P_{oo}	The ratio of the maximum pressure amplitude at the injector face to the steady-state pressure value.
L19 L68	IPP	The values of the pressure perturbation associated with the pressure-dependent nonlinear element.
L69 L118	OPP	The values of the combustion perturbation corresponding to the above IPP values.
L119 L168	IPR	The values of the radial velocity perturbation associated with the velocity-dependent nonlinear element.
L169 L218	OPR	The values of the combustion perturbation corresponding to the above IPR values.
L219 L268	IPT	The values of the tangential velocity perturbation associated with the velocity dependent nonlinear element.
L269 L318	OPT	The values of the combustion perturbation corresponding to the above IPT values.
L9595	TFLP	The linear transfer function for equivalent linear operation associated with the pressure-dependent nonlinear element.
L9596	TFLR	The same as TFLP except for radial velocity-dependent nonlinear element.
L9597	TFLT	The same as TFLP except for tangential velocity-dependent nonlinear elements.
L9598	NUMBR	The number of steps to be used in the integration scheme associated with this problem. If left blank, the computer will assume 20, which is sufficient for most cases.

b. Print Options

$0 \leq L8 \leq 9$: Execute I, print no input or output data

$10 \leq L8 \leq 99$: Execute I, print output only

$100 \leq L8 \leq 199$: Execute I, print input and output

c. Placement of the Data into the Data Package

These data are included in Section III of the data package. Along with these data, Section IV must be included.

d. Output

The output of this program is as follows:

The frequency, ω

The element number, location, fractional flow rate, F_p, F_R, and F_T necessary for Program E.

This will automatically change the output from Program E as follows:

The frequency, ω

The frequency dependent expansion coefficients $A_{\nu\eta}$, $B_{\nu\eta}$, and $C_{\nu\eta}$.

10. Program J: Injected Mass Distribution Effects (Not Programed for Annular Chambers)

a. Input Requirements

MAIN CONTROL requires L9 in Section I of data package. The remaining data go in Section III of data package.

Core Location	Mnemonic	Description
L319 L323 L327 L319 + 4n	n	Element number (maximum = 1000) within symmetrical sector.
L320 L324 L328 L320 + 4n	X_n or r_n	Element location, in.
L321 L325 L329 L321 + 4n	Y_n or θ_n	Element location, in. or degrees
L322 L326 L330 L322 + 4n	T_n	Type of injection element. The details concerning this type number are given next.
L4	DX	Number of symmetrical sectors, DX = 1 for no symmetry.
L4944 to L9568	-	The data input in this section serves to define the element types and consists of a maximum of 100 variable length data sets. Each set contains:

III, D, Input Data (cont.)

a. The element type number (≤ 100)
b. Number of oxidizer orifices for that type
c. Diameters of all the oxidizer orifices
d. Number of fuel orifices for that type
e. Diameters of all the fuel orifices

EXAMPLE

let T_i = i^{th} element type
NX_i = number of oxidizer orifices for the i^{th} type element
DX_{ij} = diameter of the j^{th} oxidizer orifice in the i^{th} type element
NF_i = number of fuel orifices for the i^{th} type element
DF_{ij} = diameter of the j^{th} fuel orifice in the i^{th} type element

Then a typical input would read

$$L4944 + T_1 + NX_1 + DX_{1,1} + DX_{1,2} + DX_{1,3} + NF_1 + DF_{1,1} + DF_{1,2} + T_2 + NX_2 + DX_{2,1} + \ldots.$$

If NF_i or NX_i is zero, then do not set $DF_{i,1} = 0$ or $DX_{i,1} = 0$.

Assume $NX_1 \equiv 0.0$, then

$$L4944 + T_1 + NX_1 + NF_1 + DF_{1,1} + DF_{1,2} + DF_{1,3} + T_2 + NX_2 + DX_{2,1} + \ldots$$

III, D, Input Data (cont.)

Core Location	Mnemonic	Description
L9569	NT	Total number of injection types
L9570	NE	Number of elements per symmetrical sector
L9571	COORD	COORD = 0: elements are located using polar coordinates COORD > 0: elements are located using cartesian coordinates
L9572	WT	Total injector weight flow, lb/sec
L9573	R_{inj}	Injector radius, in.
L9574	MR	Injector mixture ratio
L9575	NFFC	Total number of fuel film cooling holes
L9576	DFFC	Diameter of fuel film cooling holes

In the event of fuel or oxidizer film cooling, either the percent film cooling or the actual orifice dimensions can be used to describe the cooling. Place a zero(s) in the one(s) not used.

L9578	ROX	Oxidizer density, lb/ft^3
L9579	ROF	Fuel density, lb/ft^3
L9584	CDX	Oxidizer orifice loss coefficient
L9585	CDF	Fuel orifice loss coefficient
L9588	PFFC	Percent fuel film cooling
L9589	PXFC	Percent oxidizer film cooling
L9590	XFC	Total number of oxidizer film cooling orifices
L9591	DFFX	Diameter of film cooling orifices

L2, L3, and L4 from Program E.

III, D, Input Data (cont.)

b. Print Options

$0 < L9 \leq 9$:	Execute J, do not print output or input data
$10 \leq L9 \leq 99$:	Execute J, print output only
$100 \leq L9 \leq 199$:	Execute J, print input and output
$L9 \leq 500$:	Execute J, print input and output, and, if error occurs in J, dump error message with input data.

c. Placement of the Data into the Data Package

Main control data: Section I of the data package.

All other data: Section III of the data package. Section IV must be included in the data package.

d. Output

The following output results as a consequence of this program:

Miscellaneous information concerning the total orifice area, circuit pressure drops, overall mixture ratio, etc.

Element number, injection type, and location.

Description of various parameters associated with the different types of injection elements.

Element number, location, and distribution coefficient.

The distribution coefficient as a function of the radius across the face of the injector.

E. RESTRICTIONS AND LIMITATIONS

1. **One limitation of the distributed combustion analysis (Program A) which cannot easily be implied from the information given in this manual is the frequency limit in transverse mode analysis. This limit is implied from the relationship**

$$\Omega \sinh \Omega Z \leq 0 (M_e)$$

This limitation means that combined transverse-longitudinal modes cannot be analyzed with this program. Long chambers (long compared to their diameter) may also produce erroneous results for the pure transverse modes. This limitation does not apply to the concentrated combustion analysis (Program B).

2. The restrictions of the annular nozzle analysis have already been discussed in Section I,B,4,b,(6) of this manual.

3. The distributed combustion (Program A) analysis is restricted to low Mach numbers, those less than approximately 0.3. It is also limited to low combustion intensity by the assumption that $\frac{d\bar{u}}{dZ}$ is the same order of magnitude as $\bar{u}$.

4. The sensitive time lag theory is developed for liquid propellant rocket engines. Before applying it to gas injection, many assumptions must be examined critically.

5. Calculations of the distribution coefficients $A_{\nu\eta}$, $B_{\nu\eta}$, and $C_{\nu\eta}$ are not applicable to annular chambers.

F. OUTPUT DEFINITION

1. Program A

Label in Output	Symbol in Analysis
GAMMA	γ
DESIRED MACH NUMBER	M_e
CHAMBER RADIUS	R^*_{AC}
CHAMBER LENGTH	L^*_c
SPEED OF SOUND	c^*_o
CHAMBER MODE DESCRIPTION	$s_{\nu\eta}$
MACH DISTRIBUTION	$M(Z)^*$
UIBAR	M_e
K	$\varkappa$
x	ξ_1
FC(CPS)	f^*
OMEGA	ω
TAU (MS)	τ^*
N	n
ULM	$\bar{u}^*_{LO}$
WC	ω

2. Program B

Label in Output	Symbol in Analysis
GAMMA	γ
DESIRED MACH NUMBER	M_e
CHAMBER RADIUS	R^*_{AC}
CHAMBER LENGTH	L^*_c
SPEED OF SOUND	c^*_o

III, F, Output Definition (cont.)

Label in Output	Symbol in Analysis
CHAMBER MODE DESCRIPTION	$s_{\nu\eta}$
CHAMBER FREQUENCIES	ω
WC	ω
MACH DISTRIBUTION	$M(Z^*)$
SNH	$s_{\nu\eta}$
ZE	L_c
UE	M_e
SOUND	c_o^*
ULM	$\bar{u}_{LO}^*$
XK	κ
OMEGA (CH)	ω
ERT	ϵ_r
EIT	ϵ_i
H REAL	h_r
H IMAG	h_i

3. Program C

Label in Output	Symbol in Analysis
(SNH)C	$s_{\nu\eta}$
(SNH)N	$s_{\nu\eta}$
WC	ω
WN	$\hat{\omega}$

III, F, Output Definition (cont.)

Label in Output	Symbol in Analysis
MACH NO.	M_e
G	γ
AR	A_r
AI	A_i
-AR/(MACH NO.)	α_r
-AI/(MACH NO.)	α_i
BR	B_r
BI	B_i
T1	E_r
T2	E_i
CR	C_r
CI	C_i
-CR/(Mach No.)	β_{nr}
-CI/(Mach No.)	β_{ni}
FC (CPS)	f^*

4. Program D

Label on Output	Symbol in Analysis
LR/N	$\frac{1r}{n}$
LT/N	$\frac{1\theta}{n}$
ERT	E_r
EIT	E_i
CRT	C_r
CIT	C_i
OMEGA (C)	ω
OMEGA (CH)	ω

III, F, Output Definition (cont.)

Label On Output	Symbol in Analysis
HTR	$\tilde{h}_i$
HTI	$\tilde{h}_r$
HTRINT	$\tilde{h}_i$ (interpolated)
HTIINT	$\tilde{h}_r$ (interpolated)
OMEGA	ω

5. Program E

Label on Output	Symbol in Analysis
AVN	$A_{\nu\eta}$
BVN	$B_{\nu\eta}$
CVN	$C_{\nu\eta}$
MU	μ

6. Program F

Label on Output	Symbol in Analysis
FC(CPS)	f^*
(OMGA)D	ω
TAU(MS)	τ^*
N	n
NMIN	n_{min}

III, F, Output Definition (cont.)

7. Program I

Label on Output	Symbol in Analysis
PRESSURE PERTURBATION	p'
COMBUSTION GAIN	ϕ_p
RADIAL VELOCITY PERTURBATION	v'
COMBUSTION GAIN	ϕ_R
TANGENTIAL VELOCITY PERTURBATION	w'
COMBUSTION GAIN	ϕ_T
OMEGA	ω
FP	F_p
FR	F_R
FT	F_T
AVN REAL	$A_{\nu\eta}$
BVN REAL	$B_{\nu\eta}$
CVN REAL	$C_{\nu\eta R}$
CVN IMAG	$C_{\nu\eta I}$

8. Program J

Label on Output	Symbol in Analysis
PRESSURE (TFLP)	TF_{LP}
RADIAL VELOCITY (TFLR)	TF_{LR}
TANGENTIAL VELOCITY (TFLT)	TF_{LT}
DISTRIBUTION COEFFICIENT MU	μ

III, Deck Setup (cont.)

G. INPUT FORM

A tabulation of the input for the illustrative problem given below is shown immediately following the discussion of the problems in Figure 19.

1. Illustrative Design Problem

a. Longitudinal Mode Analysis

The following data concerning a hypothetical engine will be used for this analysis:

Engine Geometry: See Figure 19

$\gamma = 1.218$

$C^*_o = 3800$ ft/sec

Mach Number: unknown

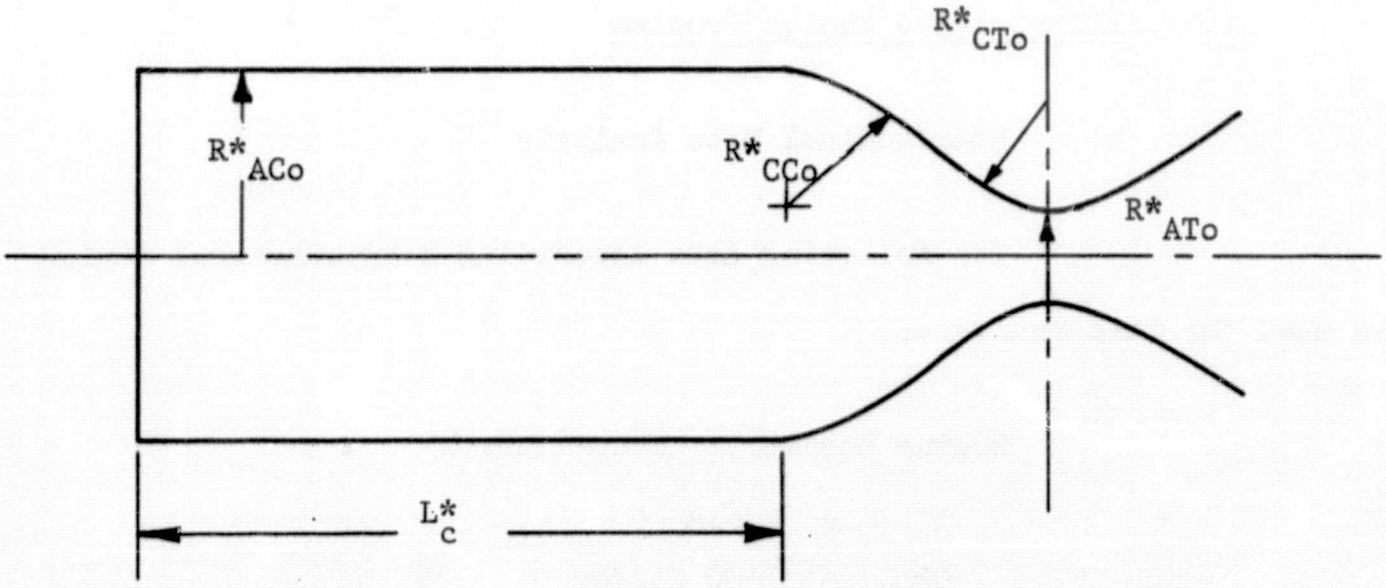
Momentum interchange coefficient = 0.0

Combustion front location: 2 inches from injector face

Combustion distribution: a linear combustion distribution with complete combustion occurring at 6.4 in.

$P_c = 600$ psia

The problem is to find the n, zone for the first longitudinal mode.



L^*_c = 14.8, in.

R^*_{ACo} = 6.4, in.

R^*_{CCo} = 3.0, in.

R^*_{CTo} = 7.16, in.

R^*_{ATo} = 3.58, in.

α = 15.0°

Figure 18 -- Definition of the Geometrical Factors used in Example Problem

III, G, Input Form (cont.)

b. Transverse Mode

For the transverse case, the problem will be to find the n,τ zones for the first tangential and the second tangential modes. The data given in the longitudinal case will be supplemented by the following information:

$$s_{11} = 1.8413 \text{ and } s_{12} = 3.0543$$

Injector
Number of radial baffle compartments = 180
Total weight flow = 150 lb/sec
Mixture ratio = 2.0
Injection occurs at three radii:

Unlike doublets at r = 2.0 in.
Triplets (X-F-X) at r = 4.0 in.
Pentads (4F-X) at r = 6.0 in.

Injector radius = chamber radius
No film cooling
Storable propellants
Fuel and oxidizer loss coefficients = 0.75
Spinning modes
Tangential mode numbers = 1 and 2
No radial or tangential velocity effects

III, G, Input Form (cont.)

c. Nonlinear Combustion Response

For this case, the first tangential mode will be examined using a deadband nonlinear element. In addition to the above injector data the following additional information will be used:

P_{oo} = 1.0
TFLP = 1.0

III, Deck Setup (cont.)

```
L1+201.+201.
L10+1.218+0
L13+6.4+14.8+3800+20.42+0.0
L20+0.0
L21+15+2.0+2.5+3.0+3.25+3.5+3.75+4.0+4.25+4.5+4.75+5.0
+5.5+6.0+6.5+7.0
L3801+3.0+0.+3.58+0.+3.0+7.16+15.0+0.
L50+600.+2.++151.
L101+12.8
TEXAMPLE PROBLEM... LONGITUDINAL MODE FOR HYPOTHETICAL ENGINE

L0+100++10+10+100L9+100
L101+0.0
L3407+1
L20+1.8411838+0.0
L50+0.0+2.5+5.0+6.3+6.35+6.38+6.39+6.40+6.41+6.42+6.43+6.5
+7.5+12.5+25.+50.+0.00000
L70+0.0+2.5+5.0+6.3+6.35+6.38+6.39+6.395+6.4+6.4+6.4+6.4
+6.4+6.4+6.4+6.4+0.00000
T EXAMPLE PROBLEM... FIRST TANGENTIAL MODE, HYPOTHETICAL ENGINE
L0+0+1+20+180+180+1++1.8411838
L4944+1+1+.0785+1+.0785
+2+2+.0785+.0785+1+.0785
+3+1+.0785+4+.0785+.0785+.0785+.0785
L319+1+2.0+1.0+1
L323+2+4.0+1+2
L327+3+6.0+1.0+3
L9569+3+3+0+150+6.4+2.0++
L9578+89.52+56.1
L9584+0.75+0.75
T

L1+100+  +10+100+10
L20+3.0542369+0.0
L50+600.+2.++151.
L101+12.8
T EXAMPLE PROBLEM... SECOND TANGENTIAL MODE, HYPOTHETICAL ENGINE
L0+0+1+20+180+180+2++3.0542369
T

L4+200
L8+50
L20+1.8411838+0.0
L3405+1.0
T 1T MODE, SHOWING EFFECTS OF DEADBAND NONLINEARITY
L0+0+1+20+180+180+1++1.8411838
L12+1.0L9598+1.0
L119-5.1-2.1-1.1-0.60-0.20-0.15-0.11-0.10-0.09-0.05
L169-5.0-2.0-1.0-0.50-0.10-0.05-0.01+0.00+0.00+0.00
L129-0.01+0.01+0.05+0.09+0.10+0.11+0.15+0.20+0.60
L179+0.00+0.00+0.00+0.00+0.00+0.01+0.05+0.10+0.50
L138+1.10+2.1+5.1+0.000000
L188+1.00+2.0+5.0+0.000000
T
```

Figure 19. Tabulation of Input for Illustrative Design Problem

III, Deck Setup (cont.)

H. SAMPLE OUTPUT

The following output is run from the sample input given in the previous section:

```
DATE      25 JUL 70              EXAMPLE PROBLEM<-/',:?4ITUDINAL MODE FOR HYPOTHETICAL ENGINE                       PAGE      1
   A      B      C      D      E      F      G      H      I      J
   0.   201.   201.    0.     0.     0.     0.     0.     0.     0.


    ************************ THE FOLLOWING MAIN CONTROL DATA WILL BE USED IN THIS CASE ************************

                              RATIO OF SPECIFIC HEAT (GAMMA) =  1.2180

                              DESIRED MACH NUMBER =     .00000    (=0 IF BEING CALCULATED)

                              CHAMBER RADIUS =   6.400 (INCHES)

                              CHAMBER LENGTH =  14.800 (INCHES)

                              SPEED OF SOUND =   3800.000 (FT/SEC)

                              CHAMBER MODE DESCRIPTION =   .00000 (=0 FOR LONGITUDINAL MODES)


                    ********** CHAMBER FREQUENCIES (WC) **********

      2.00000          2.50000          3.00000          3.25000          3.50000
      3.75000          4.00000          4.25000          4.50000          4.75000
      5.00000          5.50000          6.00000          6.50000          7.00000
     DISTANCE FROM INJECTOR TO COMBUSTION FRONT      2.00000INCHES
```

```
DATE     25 JUL 70             EXAMPLE PROBLEM<-/',:?4ITUDINAL MODE FOR HYPOTHETICAL ENGINE                       PAGE     2

PROGRAM  C  INPUT - CALCULATES NOZZLE ADMITTANCE  COEFFICIENTS USING 8052

    G =     1.218 , MDESIR = 3
    RAT =  3.580,RAC =  6.400,RCC =  3.000,RCT =  7.160,ALFA = 15.000,KN = 101,STN = 12.800
```

```
DATE      25 JUL 70            EXAMPLE PROBLEM<-/',:?4ITUDINAL MODE FOR HYPOTHETICAL ENGINE                  PAGE      3

                                    VELOCITY POTENTIAL TABLE

                               Z = AXIAL POSITION IN NOZZLE (INCHES)
                               X = VELOCITY POTENTIAL (NONDIMENSIONAL)
                               V = AXIAL VELOCITY (UNDIMENSIONALIZED BY SPEED OF SOUND AT THROAT)

POINT          Z              X              V              POINT          Z              X              V

  1         .00000         .00000        .10000+01           28        .64055+01     -.13654+01     .32922+00
  2         .23724+00     -.87012-01     .95571+00           29        .66427+01     -.13943+01     .32022+00
  3         .47448+00     -.17011+00     .91188+00           30        .68799+01     -.14224+01     .31162+00
  4         .71172+00     -.24933+00     .86862+00           31        .71172+01     -.14498+01     .30340+00
  5         .94896+00     -.32473+00     .82602+00           32        .73544+01     -.14764+01     .29553+00
  6         .11862+01     -.39637+00     .78416+00           33        .75917+01     -.15024+01     .28799+00
  7         .14234+01     -.46432+00     .74314+00           34        .78289+01     -.15277+01     .28075+00
  8         .16607+01     -.52867+00     .70303+00           35        .80661+01     -.15523+01     .27381+00
  9         .18979+01     -.58948+00     .66399+00           36        .83034+01     -.15764+01     .26714+00
 10         .21352+01     -.64698+00     .62899+00           37        .85406+01     -.15999+01     .26072+00
 11         .23724+01     -.70156+00     .59834+00           38        .87779+01     -.16228+01     .25455+00
 12         .26096+01     -.75357+00     .57103+00           39        .90151+01     -.16452+01     .24861+00
 13         .28469+01     -.80327+00     .54637+00           40        .92523+01     -.16671+01     .24289+00
 14         .30841+01     -.85088+00     .52390+00           41        .94896+01     -.16885+01     .23737+00
 15         .33214+01     -.89657+00     .50326+00           42        .97268+01     -.17093+01     .23205+00
 16         .35586+01     -.94050+00     .48420+00           43        .99641+01     -.17298+01     .22691+00
 17         .37958+01     -.98279+00     .46650+00           44        .10201+02     -.17497+01     .22195+00
 18         .40331+01     -.10236+01     .45000+00           45        .10439+02     -.17693+01     .21716+00
 19         .42703+01     -.10629+01     .43456+00           46        .10676+02     -.17884+01     .21253+00
 20         .45075+01     -.11009+01     .42006+00           47        .10913+02     -.18071+01     .20805+00
 21         .47448+01     -.11377+01     .40642+00           48        .11150+02     -.18254+01     .20377+00
 22         .49820+01     -.11733+01     .39356+00           49        .11387+02     -.18434+01     .20060+00
 23         .52193+01     -.12078+01     .38139+00           50        .11625+02     -.18612+01     .19876+00
 24         .54565+01     -.12412+01     .36987+00           51        .11862+02     -.18788+01     .19815+00
 25         .56937+01     -.12736+01     .35893+00           52        .11872+02     -.18796+01     .19815+00
 26         .59310+01     -.13051+01     .34854+00           53        .11882+02     -.18803+01     .19815+00
 27         .61682+01     -.13357+01     .33865+00           54        .24662+02     -.28302+01     .19815+00

                                   WN              (SNH)N            DES

                               .758748+00        .000000          .000000
                               .113812+01        .000000          .000000
                               .132781+01        .000000          .000000
                               .151750+01        .000000          .000000
                               .170718+01        .000000          .000000
                               .189687+01        .000000          .000000
                               .227625+01        .000000          .000000
                               .614112+01        .000000          .000000
```

```
DATE      25 JUL 70              EXAMPLE PROBLEM<-/',:?4ITUDINAL MODE FOR HYPOTHETICAL ENGINE                      PAGE      4

PROGRAM  C  OUTPUT

 (SNH)C       WC     MACH NO          AR                 AI              BR                BI                CR                CI
 (SNH)N       WN        G       -AR/(MACH NO)     -AI/(MACH NO)          T1                T2         -CR/(MACH NO)     -CI/(MACH NO)


  .0000    2.0000     .1885     -.26984+00         .27232+00        .00000            .00000          -.61422-01        -.26865-01
  .0000     .7587    1.2180      .14313+01        -.14444+01      -.32866+00         .33168+00         .32580+00         .14250+00

 -2.8302
  FC(CPS)=  980.7386


  .0000    3.0000     .1885     -.97749+00        -.28783+00        .00000            .00000           .34497-01         .57043-01
  .0000    1.1381    1.2180      .51849+01         .15267+01      -.11906+01        -.35057+00        -.18298+00        -.30257+00

 -2.8302
  FC(CPS)= 1471.1079


  .0000    3.5000     .1885     -.87266+00         .37683-01        .00000            .00000          -.35382-01         .30422-01
  .0000    1.3278    1.2180      .46288+01        -.19988+00      -.10629+01         .45898-01         .18767+00        -.16136+00

 -2.8302
  FC(CPS)= 1716.2925


  .0000    4.0000     .1885     -.82148+00        -.12272+00        .00000            .00000          -.18879-01        -.32612-01
  .0000    1.5175    1.2180      .43574+01         .65093+00      -.10006+01        -.14947+00         .10014+00         .17298+00

 -2.8302
  FC(CPS)= 1961.4772


  .0000    4.5000     .1885     -.10790+01        -.61946-01        .00000            .00000           .31844-01        -.11966-01
  .0000    1.7072    1.2180      .57234+01         .32858+00      -.13142+01        -.75451-01        -.16891+00         .63468-01

 -2.8302
  FC(CPS)= 2206.6618


  .0000    5.0000     .1885     -.92186+00         .25215+00        .00000            .00000           .84186-02         .20991-01
  .0000    1.8969    1.2180      .48898+01        -.13375+01      -.11228+01         .30712+00        -.44655-01        -.11134+00

 -2.8302
  FC(CPS)= 2451.8464


  .0000    6.0000     .1885     -.67979+00        -.75759-01        .00000            .00000           .10422-02        -.13956-01
  .0000    2.2762    1.2180      .36058+01         .40185+00      -.82799+00        -.92274-01        -.55280-02         .74026-01

 -2.8302
  FC(CPS)= 2942.2157


  .0000    7.0000     .1885     -.80277+00         .30663-02        .00000            .00000           .93632-03         .12451-02
```

```
 .0000   6.1411    1.2180        .42581+01      -.16265-01      -.97777+00       .37348-02      -.49665-02      -.66043-02
-2.6302
 FC(CPS)= 3432.5850
```

```
   DATE     25 JUL 70            EXAMPLE PROBLEM<-/',:?4ITUDINAL MODE FOR HYPOTHETICAL ENGINE                    PAGE     5
 TRANSVERSE STABILITY PROGRAM,.. CALCULATES  HR, HI

 INPUT DATA
        SNH            ZE          GAMMA        UE        SOUND (FT/SEC)   ULM (FT/SEC)

     .00000000    2.31250000    1.21800    .18852762       3800.00        20.420000


 NOZZLE ADMITTANCES INPUT
             OMEGA(CH)           ERT               EIT               CRT               CIT

             2.0000000     -.26983859+00     .27231600+00     -.61422292-01     -.26865192-01
```

```
DATE      25 JUL 70              EXAMPLE PROBLEM<-/',:?4ITUDINAL MODE FOR HYPOTHETICAL ENGINE                    PAGE      6


CALCULATED RESULTS
                    OMEGA              H REAL              H IMAG
                  2.0000000          .11152828+01        .93327805+00
                  2.5000000          .15325213+01        .16800883+01
                  3.0000000          .17270751+01        .21513920+01
                  3.2500000          .16893139+01        .21996202+01
                  3.5000000          .16139799+01        .22259782+01
                  3.7500000          .15737817+01        .24372959+01
                  4.0000000          .15769181+01        .26913456+01
                  4.2499999          .17041927+01        .29135248+01
                  4.4999999          .18208100+01        .31073267+01
                  4.7499999          .17901384+01        .32061581+01
                  5.0000000          .17180099+01        .33261167+01
                  5.4999999          .15904034+01        .39081204+01
                  6.0000000          .14842304+01        .46183871+01
                  6.4999999          .14287092+01        .53301924+01
                  7.0000000          .13770885+01        .61530102+01
```

```
DATE      25 JUL 70              EXAMPLE PROBLEM<-/',:?4ITUDINAL MODE FOR HYPOTHETICAL ENGINE                    PAGE    8
PROGRAM  F  OUTPUT

                  FC(CPS)               (OMGA)D,               TAU(MS)                N
                   980.7                2.00000                .28367              .94813
                  1043.6                2.12821                .25133             1.15447
                  1106.5                2.25641                .22662             1.35069
                  1169.3                2.38462                .20686             1.53419
                  1232.2                2.51282                .19050             1.70348
                  1295.1                2.64103                .17653             1.86052
                  1357.9                2.76923                .16442             2.00293
                  1420.8                2.89744                .15386             2.12482
                  1483.7                3.02564                .14461             2.21896
                  1546.5                3.15385                .13655             2.25803
                  1609.4                3.28205                .12894             2.28362
                  1672.3                3.41026                .12177             2.30069
                  1735.2                3.53846                .11377             2.37642
                  1798.0                3.66667                .10469             2.54619
                  1860.9                3.79487                .09652             2.74482
                  1923.8                3.92308                .08928             2.96544
                  1986.6                4.05128                .08449             3.14990
                  2049.5                4.17949                .08205             3.27774
                  2112.4                4.30769                .07984             3.39624
                  2175.2                4.43590                .07776             3.50808
                  2238.1                4.56410                .07488             3.61243
                  2301.0                4.69231                .07128             3.71256
                  2363.8                4.82051                .06752             3.83567
                  2426.7                4.94872                .06359             3.98963
                  2489.6                5.07692                .05927             4.24519
                  2552.4                5.20513                .05486             4.58604
                  2615.3                5.33333                .05062             4.99461
                  2678.2                5.46154                .04671             5.45309
                  2741.0                5.58974                .04315             5.95343
                  2803.9                5.71795                .03982             6.52140
                  2866.8                5.84615                .03680             7.14338
                  2929.6                5.97436                .03413             7.79593
                  2992.5                6.10256                .03184             8.45198
                  3055.4                6.23077                .02984             9.11516
                  3118.2                6.35897                .02803             9.81155
                  3181.1                6.48718                .02632            10.57552
                  3244.0                6.61538                .02469            11.42604
                  3306.8                6.74359                .02317            12.34961
                  3369.7                6.87179                .02174            13.35083
                  3432.6                7.00000                .02042            14.43477

                  THE FOLLOWING ARE VALUES INTERPOLATED AT SLOPE OF N=0.0

                                NMIN=      .40249
                                TAU(MS)=    .47258
                                (OMEGA)D=   1.65950
                                FC(CPS)=     813.8
```

```
DATE      25 JUL 70              EXAMPLE PROBLEM+/*'41%ST TANGENTIAL MODE!'2-%?THETICAL ENGINE                    PAGE      9

  A      B      C      D      E      F      G      H      I      J

 100.     0.    10.    10.   100.     0.     0.     0.     0.   100.


     *********************** THE FOLLOWING MAIN CONTROL DATA WILL BE USED IN THIS CASE ***********************

                                    RATIO OF SPECIFIC HEAT (GAMMA) =  1.2180

                                    DESIRED MACH NUMBER =     .00000    (=0 IF BEING CALCULATED)

                                    CHAMBER RADIUS =   6.400 (INCHES)

                                    CHAMBER LENGTH =  14.800 (INCHES)

                                    SPEED OF SOUND =   3800.000 (FT/SEC)

                                    CHAMBER MODE DESCRIPTION =  1.84118 (=0 FOR LONGITUDINAL MODES)


                         ********** CHAMBER FREQUENCIES (WC) **********

        1.69389          1.73071          1.76754          1.80436          1.84118

        1.87801          1.91483          1.95165          1.98848          2.02530
                    ***** MACH DISTRIBUTION IN CHAMBER AS A FUNCTION OF LENGTH *****

     CHAMBER            MACH               CHAMBER            MACH               CHAMBER            MACH
     LENGTH         DISTRIBUTION           LENGTH         DISTRIBUTION           LENGTH         DISTRIBUTION

        .00000           .00000            6.40000          6.39500           25.00000          6.40000
       2.50000          2.50000            6.41000          6.41000           50.00000          6.40000
       5.00000          5.00000            6.42000          6.40000             .00000           .00000
       6.30000          6.30000            6.43000          6.40000             .00000           .00000
       6.35000          6.35000            6.50000          6.40000             .00000           .00000
       6.38000          6.38000            7.50000          6.40000             .00000           .00000
       6.39000          6.39000           12.50000          6.40000
```

```
DATE      25 JUL 70               EXAMPLE PROBLEM+/W'41%ST TANGENTIAL MODE!'2-%?THETICAL ENGINE                    PAGE    10

SECTION 1....MISCELLANEOUS INFORMATION FOR INJECTOR DESIGNED BY PROJECTS


  A....PROPELLENT ORFICE AREAS

     ELEMENT TOTAL OXIDIZER AREA       = 3.48462972 SQ. IN.              ELEMENT TOTAL FUEL AREA       = 5.22700441 SQ. IN.
     TOTAL OXIDIZER FILM COOLING AREA =  .00000000 SQ. IN.              TOTAL FUEL FILM COOLING AREA =  .00000000 SQ. IN.
     INJECTOR TOTAL OXIDIZER AREA   .  = 3.48462972 SQ. IN.              INJECTOR TOTAL FUEL AREA      = 5.22700441 SQ. IN.

  B....INJECTOR PRESSURE DROPS FOR ABOVE INJECTOR DESIGN

     OXIDIZER PRESSURE DROP =  36.6PSI                                  FUEL PRESSURE DROP =   6.5 PSI


  C....PROPELLANT FLOWS AND INJECTOR VELOCITIES FOR ABOVE INJECTOR DESIGN


                                              TOTAL WEIGHT FLOW = 150.0 LB/SEC
                                        AVERAGE MIXTURE RATIO OF THE ELEMENTS =  2.000
                                       OVERALL MIXTURE RATIO FOR THE INJECTOR =  2.000

     ELEMENT TOTAL OXIDIXER FLOW       = 100.0 LB/SEC                    ELEMENT TOTAL FUEL FLOW  =  50.0 LB/SEC
     TOTAL OXIDIZER FILM COOLING FLOW =     .0 LB/SEC                    TOTAL FUEL FILM COOLING  =     .0 LB/SEC
     INJECTOR TOTAL OXIDIZER FLOW      = 100.0 LB/SEC                    INJECTOR TOTAL FUEL FLOW=  50.0 LB/SEC
     OXIDIZER OVERALL INJECTION VELOCITY =  61.5 FT/SEC                  FUEL OVERALL INJECTION VELOCITY =  32.7 FT/SEC



  D......INPUT INFORMATION USED IN COMPUTATIONS

     TOTAL PROPELLENT FLOW = 150.0 LB/SEC
     TOTAL NUMBER OF ELEMENT TYPES (SYMMETRICAL SECTION ONLY) =    3
     TOTAL NUMBER OF ELEMENTS (SYMMETRICAL SECTION ONLY) =     3
     OXIDIZER LOSS COEFFICIENT = .750
     FUEL LOSS COEFFICIENT = .750
     PERCENT FUEL FILM COOLING =    .0
     PERCENT OXIDIZER FILM COOLING =    .0
     DIAMETER OF FUEL FILM COOLING ORFICE = .00000 IN. (NOTE..THIS MIGHT BE AN EQUIVALENT  DIAMETER FOR MULTIPLE-ROW COOLING)
     DIAMETER OF OXIDIZER FILM COOLING ORFICE = .00000 IN. (SEE ABOVE NOTE)
     NUMBER OF FUEL FILM COOLING ORIFICES PER INJECTOR =   0.
     NUMBER OF OXIDIZER FILM COOLING ORFICES PER INJECTOR =   0.
     OXIDIZER DENSITY =  89.52 PCF
     FUEL DENSITY =  56.1 PCF
```

```
DATE     25 JUL 76              EXAMPLE PROBLEM+/#'41%ST TANGENTIAL MODE!'2-%?THETICAL ENGINE                  PAGE    11

SECTION 2.... ELEMENT LOCATION AND INJECTION TYPE

          ELEMENT                TYPE                R                  THETA               X                   Y
            NO.                  NO.             (INCHES)            (DEGREE)           (INCHES)            (INCHES)


             1                    1               2.00000             1.00000             1.99970              .03490
             2                    2               4.00000             1.00000             3.99939              .06981
             3                    3               6.00000             1.00000             5.99909              .10471
```

```
DATE     25 JUL 70            EXAMPLE PROBLEM+/W'41%ST TANGENTIAL MODE!'2-%?THETICAL ENGINE                 PAGE   12

SECTION 3....TYPE DESCRIPTION,ORIFICE AREA, PROPELLANT FLOW, AND MIXTURE RATIO

 TYPE   *-------- OXIDIZER ORIFICE DATA -----**---------- FUEL ORIFICE DATA --------*

                                                                                          TOTAL      TOTAL      TOTAL     MIXTURE
        NUMBER                                      NUMBER                               OXIDIZER    FUEL     PROPELLENT   RATIO
          OF     DIAMETER    AREA      FLOW           OF     DIAMETER    AREA      FLOW    AREA      AREA     FLOW RATE
       ORIFICE     IN.     SQ. IN.    LB/SEC      ORIFICES     IN.     SQ. IN.   LB/SEC  SQ. IN.   SQ. IN.    LB/SEC

  1       1
                 .0785    .004840   .138889
                                                     1
                                                              .0785   .004840   .046296
                                                                                         .00484    .00484     .1852     3.0000
  2       2
                 .0785    .004840   .138889
                 .0785    .004840   .138889
                                                     1
                                                              .0785   .004840   .046296
                                                                                         .00968    .00484     .3241     6.0001
  3       1
                 .0785    .004840   .138889
                                                     4
                                                              .0785   .004840   .046296
                                                              .0785   .004840   .046296
                                                              .0785   .004840   .046296
                                                              .0785   .004840   .046296
                                                                                         .00484    .01936     .3241      .7500
```

```
DATE      25 JUL 70             EXAMPLE PROBLEM+/*'41%ST TANGENTIAL MODE!'2-%?THETICAL ENGINE                PAGE     13

     ELEMENT RESULTS

                         ELEMENT              RADIUS               ANGLE             DISTRIBUTION
                                                                                      COEFFICIENT
                           NO.                                    RADIANS                 MU

                             1                2.000                .0175                4.4444
                             2                4.000                .0175                7.7778
                             3                6.000                .0175                7.7778
```

```
DATE      25 JUL 70              EXAMPLE PROBLEM+/W'41%ST TANGENTIAL MODE!'2-%?THETICAL ENGINE                    PAGE    14


SECTION 4..RESULTANT FLOW DISTRIBUTION (MU) AS A FUNCTION OF ONLY THE RADIUS, I.E.. AVERAGE OF MU IN A RADIAL BAND

                         RADIUS            MU(R)          MU(R)/MU(MAX)
                           IN.

                          1.431             .000             .0000
                          2.024            4.444             .5714
                          2.479             .000             .0000
                          2.862             .000             .0000
                          3.200             .000             .0000
                          3.505             .000             .0000
                          3.786             .000             .0000
                          4.048            7.778            1.0000
                          4.293             .000             .0000
                          4.525             .000             .0000
                          4.746             .000             .0000
                          4.957             .000             .0000
                          5.160             .000             .0000
                          5.355             .000             .0000
                          5.543             .000             .0000
                          5.724             .000             .0000
                          5.901             .000             .0000
                          6.072            7.778            1.0000
                          6.238             .000             .0000
                          6.400             .000             .0000
```

```
INPUT TO INJECTION DISTRIBUTION PROGRAM

     CONSTANTS

         NUMBER OF OMEGAS =  1

         NUMBER OF ELEMENTS =    3 FOR EACH OF  180 SYMMETRIC SECTIONS.

         RADIAL DIVISIONS(XM) =  20.

         ANGULAR DIVISIONS (XN) = 180.

         ACOUSTIC MODE NUMBER(SVN) = 1.8412

         ORDER OF BESSEL FUNCTIONS(V) = 1.

         INJECTOR RADIUS =   6.400, IN.

         RATIO OF SPECIFIC HEATS(GAMMA) = 1.2180

         MAXIMUM PRESSURE AMPLITUDE RATIO(POO) =   .000

         TRANSFER FUNCTIONS FOR LINEAR OPERATION

               PRESSURE(TFLP) =   .000

               RADIAL VELOCITY(TFLR) =   .000

               TANGENTIAL VELOCITY(TFLT) =   .000
```

```
DATE     25 JUL 70            EXAMPLE PROBLEM+/W'41%ST TANGENTIAL MODE!'2-%?THETICAL ENGINE               PAGE    16

     ELEMENT INFORMATION

                         ELEMENT            RADIUS            ANGLE          DISTRIBUTION
                                                                              COEFFICIENT
                           NO.               IN.             RADIANS              MU

                            1               2.000             .0175             4.4444
                            2               4.000             .0175             7.7778
                            3               6.000             .0175             7.7778
```

```
DATE      25 JUL 70              EXAMPLE PROBLEM+/W'41%ST TANGENTIAL MODE!'2-%?THETICAL ENGINE                    PAGE    17

        RESULTS OF INJECTION DISTRIBUTION EFFECTS
                                                  OMEGA       AVN        BVN        CVN        CVN
                                                              REAL       REAL       REAL       IMAG

                                                   ALL       1.2745     .8766      .0000     -1.8125

                                          ***********  THESE VALUES PERTAIN TO A SPINNING MODE  **********
```

```
 DATE      25 JUL 70            EXAMPLE PROBLEM+/W'41%ST TANGENTIAL MODE!'2-%?THETICAL ENGINE                PAGE    18

PROGRAM  C  OUTPUT

 (SNH)C      WC      MACH NO        AR               AI               BR               BI               CR               CI
 (SNH)N      WN         G      -AR/(MACH NO)    -AI/(MACH NO)         T1               T2         -CR/(MACH NO)    -CI/(MACH NO)

 1.8412    1.6939     .1885      -.49834-03       .39852+00       -.42140-01       -.40101+00        .43929-02       -.38922-03
 2.7421    1.4861    1.2180       .26433-02      -.21139+01        .23613+00        .46052+00       -.23301-01        .20645-02

-1.8788
 FC(CPS)= 1920.8346


 1.8412    1.7675     .1885       .11057-01       .33418+00       -.53721-01       -.36851+00        .38370-02       -.21846-02
 2.7421    1.5507    1.2180      -.58648-01      -.17726+01        .22195+00        .37664+00       -.20353-01        .11588-01

-1.8788
 FC(CPS)= 2004.3491


 1.8412    1.8412     .1885       .23791-01       .26500+00       -.56067-01       -.33223+00        .23592-02       -.36299-02
 2.7421    1.6153    1.2180      -.12619+00      -.14056+01        .20942+00        .29232+00       -.12514-01        .19254-01

-1.8788
 FC(CPS)= 2087.8636


 1.8412    1.9148     .1885       .35725-01       .18962+00       -.47438-01       -.29679+00        .17772-03       -.42791-02
 2.7421    1.6799    1.2180      -.18950+00      -.10058+01        .19851+00        .20618+00       -.94266-03        .22698-01

-1.8788
 FC(CPS)= 2171.3781


 1.8412    1.9885     .1885       .44991-01       .10706+00       -.28209-01       -.26732+00       -.22948-02       -.37970-02
 2.7421    1.7445    1.2180      -.23865+00      -.56789+00        .18923+00        .11622+00        .12172-01        .20140-01

-1.8788
 FC(CPS)= 2254.8926


 1.8412    2.0253     .1885       .48110-01       .62812-01       -.15334-01       -.25637+00       -.34685-02       -.30747-02
 2.7421    1.7768    1.2180      -.25519+00      -.33317+00        .18518+00        .68934-01        .18398-01        .16309-01

-1.8788
 FC(CPS)= 2296.6498
```

```
   DATE     25 JUL 70             EXAMPLE PROBLEM+/W'41%ST TANGENTIAL MODE!'2-%?THETICAL ENGINE                PAGE     19

TRANSVERSE STABILITY PROGRAM... CALCULATES  HR, HI


INPUT DATA

       SNH              ZE           GAMMA        UE         SOUND (FT/SEC)  ULM (FT/SEC)     XK (DRAG)    XCOMPL (IN)

   1.84118374    2.31250000     1.21800     .18852762        3800.00        20.420000        .000000        .000000

   INCREMENTS
        .0



NOZZLE ADMITTANCES INPUT

                 OMEGA(CH)              ERT                 EIT


                 1.6938890          .23612978+00        .46052134+00
                 1.7675364          .22195471+00        .37663660+00
                 1.8411837          .20942160+00        .29232068+00
                 1.9148310          .19851034+00        .20618085+00
                 1.9884783          .18923368+00        .11621683+00
                 2.0253020          .18518268+00        .68934425-01
$TABS
S*C       =    ( .52069750E+00,    .00000000E+00),
SNH       =      .18411837E+01,
W         =      .16938890E+01,
OMEGC     =    ( .72159372E+00,    .00000000E+00),
E         =    ( .23612978E+00,    .46052134E+00),
SIGMA     =    ( .00000000E+00,    .16938890E+01),
Q1SI      =    ( .00000000E+00,   -.12950017E+01),
Q1CI      =    ( .00000000E+00,   -.14993784E+01),
Q2SI      =    ( .00000000E+00,    .71299743E+00),
Q2CI      =    ( .00000000E+00,    .82565935E+00),
H         =    ( .17504949E+01,   -.25148608E+01),
$END
$TABS
S*C       =    ( .39459112E+00,    .00000000E+00),
SNH       =      .18411837E+01,
W         =      .17307127E+01,
OMEGC     =    ( .62816468E+00,    .00000000E+00),
E         =    ( .22883699E+00,    .41863286E+00),
SIGMA     =    ( .00000000E+00,    .17307127E+01),
Q1SI      =    ( .00000000E+00,   -.10085253E+01),
Q1CI      =    ( .00000000E+00,   -.12387506E+01),
Q2SI      =    ( .00000000E+00,    .55528316E+00),
Q2CI      =    ( .00000000E+00,    .68218827E+00),
H         =    ( .17280367E+01,   -.22175635E+01),
$END
$TABS
S*C       =    ( .26577279E+00,    .00000000E+00),
SNH       =      .18411837E+01,
W         =      .17675364E+01,
OMEGC     =    ( .51553155E+00,    .00000000E+00),
```

```
E          =   ( .22195471E+00,   .37663660E+00),
SIGMA      =   ( .00000000E+00,   .17675364E+01),
Q1SI       =   ( .00000000E+00,  -.78727534E+00),
Q1CI       =   ( .00000000E+00,  -.10742069E+01),
Q2SI       =   ( .00000000E+00,   .43349116E+00),
Q2CI       =   ( .00000000E+00,   .59163223E+00),
H          =   ( .17021257E+01,  -.17422520E+01),
$END
$TABS
SWC        =   ( .13424250E+00,   .00000000E+00),
SNH        =     .18411837E+01,
W          =     .18043600E+01,
OMEGC      =   ( .36639119E+00,   .00000000E+00),
E          =   ( .21548416E+00,   .33461957E+00),
SIGMA      =   ( .00000000E+00,   .18043600E+01),
Q1SI       =   ( .00000000E+00,  -.53023429E+00),
Q1CI       =   ( .00000000E+00,  -.91210546E+00),
Q2SI       =   ( .00000000E+00,   .29197411E+00),
Q2CI       =   ( .00000000E+00,   .50241572E+00),
H          =   ( .16637479E+01,  -.12504508E+01),
$END
$TABS
SWC        =   ( .23841858E-06,   .00000000E+00),
SNH        =     .18411837E+01,
W          =     .18411837E+01,
OMEGC      =   ( .48828125E-03,   .00000000E+00),
E          =   ( .20942160E+00,   .29232068E+00),
SIGMA      =   ( .00000000E+00,   .18411837E+01),
Q1SI       =   ( .00000000E+00,  -.66701029E-03),
Q1CI       =   ( .00000000E+00,  -.75323950E+00),
Q2SI       =   ( .00000000E+00,   .36731105E-03),
Q2CI       =   ( .00000000E+00,   .41497515E+00),
H          =   ( .16040049E+01,  -.73145898E+00),
$END
$TABS
SWC        =   (-.13695401E+00,   .00000000E+00),
SNH        =     .18411837E+01,
W          =     .18780073E+01,
OMEGC      =   ( .00000000E+00,   .37007298E+00),
E          =   ( .20376244E+00,   .24960378E+00),
SIGMA      =   ( .00000000E+00,   .18780073E+01),
Q1SI       =   ( .00000000E+00,  -.47517942E+00),
Q1CI       =   ( .00000000E+00,  -.59836961E+00),
Q2SI       =   ( .00000000E+00,   .26168950E+00),
Q2CI       =   ( .00000000E+00,   .32972904E+00),
H          =   ( .15051324E+01,  -.16932303E+00),
$END
$TABS
SWC        =   (-.27662018E+00,   .00000000E+00),
SNH        =     .18411837E+01,
W          =     .19148310E+01,
OMEGC      =   ( .00000000E+00,   .52594693E+00),
E          =   ( .19851034E+00,   .20618086E+00),
SIGMA      =   ( .00000000E+00,   .19148310E+01),
Q1SI       =   ( .00000000E+00,  -.63188915E+00),
Q1CI       =   ( .00000000E+00,  -.44821967E+00),
Q2SI       =   ( .00000000E+00,   .34801620E+00),
Q2CI       =   ( .00000000E+00,   .24707574E+00),
H          =   ( .13280004E+01,   .45938596E+00),
```

```
$END
$TABS
SWC     =     (-.41899833E+00,    .00000000E+00),
SWH     =        .18411837E+01,
W       =        .19516547E+01,
OMEGC   =     ( .00000000E+00,    .64730080E+00),
E       =     ( .19369643E+00,    .16172541E+00),
SIGMA   =     ( .00000000E+00,    .19516547E+01),
Q1SI    =     ( .00000000E+00,   -.72407361E+00),
Q1CI    =     ( .00000000E+00,   -.30347320E+00),
Q2SI    =     ( .00000000E+00,    .39881757E+00),
Q2CI    =     ( .00000000E+00,    .16739143E+00),
H       =     ( .97758906E+00,    .11801624E+01),
$END
$TABS
SWC     =     (-.56408846E+00,    .00000000E+00),
SWH     =        .18411837E+01,
W       =        .19884783E+01,
OMEGC   =     ( .00000000E+00,    .75105823E+00),
E       =     ( .18923366E+00,    .11621683E+00),
SIGMA   =     ( .00000000E+00,    .19884783E+01),
Q1SI    =     ( .00000000E+00,   -.77798069E+00),
Q1CI    =     ( .00000000E+00,   -.16477004E+00),
Q2SI    =     ( .00000000E+00,    .42854564E+00),
Q2CI    =     ( .00000000E+00,    .91028284E-01),
H       =     ( .21287662E+00,    .19509883E+01),
$END
$TABS
SWC     =     (-.71189046E+00,    .00000000E+00),
SWH     =        .18411837E+01,
W       =        .20253020E+01,
OMEGC   =     ( .00000000E+00,    .84373601E+00),
E       =     ( .18518267E+00,    .68934424E-01),
SIGMA   =     ( .00000000E+00,    .20253020E+01),
Q1SI    =     ( .00000000E+00,   -.80446833E+00),
Q1CI    =     ( .00000000E+00,   -.32703382E-01),
Q2SI    =     ( .00000000E+00,    .44317819E+00),
Q2CI    =     ( .00000000E+00,    .18312837E-01),
H       =     (-.14011466E+01,    .22003332E+01),
$END
```

```
   DATE      25 JUL 70              EXAMPLE PROBLEM+/W'41%ST TANGENTIAL MODE:'2-%?THET'CAL ENGINE                    PAGE    20


CALCULATED RESULTS...          FIRST-ORDER SOLN ( UNIF INJ )

                      OMEGA            H REAL            H IMAG
                    1.6938890       .17504949+01      -.25148608+01
                    1.7307127       .17280367+01      -.22175635+01
                    1.7675364       .17021257+01      -.17422520+01
                    1.8043600       .16637479+01      -.12504508+01
                    1.8411837       .16040049+01      -.73145898+00
                    1.8780073       .15051324+01      -.16932303+00
                    1.9148310       .13280004+01       .45938596+00
                    1.9516547       .97758906+00       .11801624+01
                    1.9884783       .21287662+00       .19509883+01
                    2.0253020      -.14011486+01       .22003332+01

 PROGRAM  D  OUTPUT

                       OMEGA(C)            HTR              HTI


                       1.693889         1.373502        -1.973252
                       1.730713         1.355881        -1.739982
                       1.767536         1.335550        -1.367035
                       1.804360         1.305438         -.981150
                       1.841184         1.258561         -.573930
                       1.878007         1.180982         -.132857
                       1.914831         1.041998          .360451
                       1.951655          .767052          .925999
                       1.988478          .167031         1.530817
                       2.025302        -1.099393         1.726462
```

```
                         FOLLOWING WILL BE INTERPOLATION WITHIN HTR HTI TABLE GIVEN ABOVE
            OMEGA               HTRINT            HTIINT
           1.693889            1.373502         -1.973252
           1.702387            1.369676         -1.931818
           1.710885            1.365706         -1.882945
           1.719382            1.361591         -1.826634
           1.727880            1.357333         -1.762885
           1.736378            1.353000         -1.690428
           1.744876            1.348704         -1.607302
           1.753373            1.344205         -1.517776
           1.761871            1.339244         -1.426523
           1.770369            1.333600         -1.337834
           1.778867            1.327555         -1.249955
           1.787364            1.321018         -1.161421
           1.795862            1.313732         -1.071923
           1.804360            1.305438          -.981151
           1.812858            1.296393          -.889327
           1.821355            1.286685          -.796572
           1.829853            1.275798          -.702425
           1.838351            1.263219          -.606423
           1.846849            1.248932          -.508461
           1.855346            1.233754          -.409131
           1.863844            1.216690          -.307847
           1.872342            1.196607          -.203932
           1.880840            1.172675          -.096822
           1.889338            1.147202           .012709
           1.897835            1.118764           .124939
           1.906333            1.084612           .240606
           1.914831            1.041999           .360448
           1.923329             .994491           .485222
           1.931826             .942842           .614384
           1.940324             .880079           .746717
           1.948822             .799229           .881007
           1.957320             .699321          1.020978
           1.965817             .590285          1.174381
           1.974315             .461141          1.326201
           1.982813             .299305          1.459906
           1.991311             .093280          1.560390
           1.999808            -.151651          1.634594
           2.008306            -.432072          1.687007
           2.016804            -.747981          1.717630
           2.025302           -1.099378          1.726462
```

```
 DATE      25 JUL 70             EXAMPLE PROBLEM+/W'41%ST TANGENTIAL MODE!'2-%?THETICAL ENGINE                 PAGE     23
PROGRAM  F  OUTPUT

                      FC(CPS)            (OMGA)D:             TAU(MS)                N
                       1920.8             1.69389              .41984             2.10419
                       1930.5             1.70239              .41631             2.04717
                       1940.1             1.71088              .41248             1.98089
                       1949.7             1.71938              .40831             1.90605
                       1959.4             1.72788              .40377             1.82347
                       1969.0             1.73638              .39875             1.73250
                       1978.7             1.74488              .39308             1.63209
                       1988.3             1.75337              .38691             1.52898
                       1997.9             1.76187              .38042             1.42937
                       2007.6             1.77037              .37384             1.33784
                       2017.2             1.77887              .36705             1.25222
                       2026.8             1.78736              .35995             1.17106
                       2036.5             1.79586              .35250             1.09418
                       2046.1             1.80436              .34463             1.02143
                       2055.7             1.81286              .33632              .95324
                       2065.4             1.82136              .32752              .88992
                       2075.0             1.82985              .31817              .83127
                       2084.7             1.83835              .30819              .77717
                       2094.3             1.84685              .29751              .72797
                       2103.9             1.85535              .28610              .68471
                       2113.6             1.66384              .27389              .64729
                       2123.2             1.87234              .26080              .61568
                       2132.8             1.88084              .24672              .59033
                       2142.5             1.88934              .23173              .57367
                       2152.1             1.89784              .21588              .56636
                       2161.7             1.90633              .19915              .56899
                       2171.4             1.91483              .18145              .58334
                       2181.0             1.92333              .16300              .61562
                       2190.7             1.93183              .14433              .67160
                       2200.3             1.94032              .12545              .75682
                       2209.9             1.94882              .10612              .88519
                       2219.6             1.95732              .08613             1.09495
                       2229.2             1.96582              .06651             1.46337
                       2238.8             1.97432              .04758             2.13759
                       2248.5             1.98281              .02863             3.71011
                       2258.1             1.99131              .00842            13.09772
                       2267.7             1.99981              .20750            -8.88517
                       2277.4             2.00831              .18451            -3.50946
                       2287.0             2.01680              .16146            -2.34613
                       2296.6             2.02530              .13912            -1.90531

                      THE FOLLOWING ARE VALUES INTERPOLATED AT SLOPE OF N=0.0

                                    NMIN=      .56602
                                    TAU(MS)=     .21213
                                    (OMEGA)D=   1.89978
                                    FC(CPS)=    2154.3
```

```
DATE     25 JUL 70            EXAMPLE PROBLEM+/W'$590ND TANGENTIAL MODE2 HYOOTHETICAL ENGINE                PAGE    24

  A     B     C     D     E     F     G     H     I     J

  0.  100.   0.   10.  100.   10.    0.    0.    0.    0.

     *********************** THE FOLLOWING MAIN CONTROL DATA WILL BE USED IN THIS CASE ***********************

                              RATIO OF SPECIFIC HEAT (GAMMA) =  1.2180

                              DESIRED MACH NUMBER =    .00000    (=0 IF BEING CALCULATED)

                              CHAMBER RADIUS =   6.400 (INCHES)

                              CHAMBER LENGTH =  14.800 (INCHES)

                              SPEED OF SOUND =   3800.000 (FT/SEC)

                              CHAMBER MODE DESCRIPTION =  3.05424 (=0 FOR LONGITUDINAL MODES)


                    ********** CHAMBER FREQUENCIES (WC) **********

        2.80990          2.87098          2.93207          2.99315          3.05424
        3.11532          3.17641          3.23749          3.29858          3.35966
      DISTANCE FROM INJECTOR TO COMBUSTION FRONT      2.00000INCHES
```

```
DATE     25 JUL 70              EXAMPLE PROBLEM+/W'$590ND TANGENTIAL MODE2 HYOOTHETICAL ENGINE                PAGE    26

 INPUT TO INJECTION DISTRIBUTION PROGRAM

      CONSTANTS

           NUMBER OF OMEGAS =  1

           NUMBER OF ELEMENTS =    3 FOR EACH OF  180 SYMMETRIC SECTIONS.

           RADIAL DIVISIONS(XM) =  20.

           ANGULAR DIVISIONS (XN) = 180.

           ACOUSTIC MODE NUMBER(SVN) = 3.0542

           ORDER OF BESSEL FUNCTIONS(V) = 2.

           INJECTOR RADIUS =   6.400, IN.

           RATIO OF SPECIFIC HEATS(GAMMA) = 1.2180

           MAXIMUM PRESSURE AMPLITUDE RATIO(POO) =   .000

           TRANSFER FUNCTIONS FOR LINEAR OPERATION

                 PRESSURE(TFLP) =   .000

                 RADIAL VELOCITY(TFLR) =   .000

                 TANGENTIAL VELOCITY(TFLT) =   .000
```

```
DATE      25 JUL 70              EXAMPLE PROBLEM+/W'$59OND TANGENTIAL MODE2 HYOOTHETICAL ENGINE              PAGE    27

     ELEMENT INFORMATION

                         ELEMENT              RADIUS              ANGLE          DISTRIBUTION
                                                                                  COEFFICIENT
                           NO.                  IN.              RADIANS              MU

                            1                  2.000              .0175             4.4444
                            2                  4.000              .0175             7.7778
                            3                  6.000              .0175             7.7778
```

```
DATE     25 JUL 70            EXAMPLE PROBLEM+/W'$590ND TANGENTIAL MODE2 HYDOTHETICAL ENGINE                PAGE    28

         RESULTS OF INJECTION DISTRIBUTION EFFECTS
                                                   OMEGA       AVN        BVN        CVN        CVN
                                                               REAL       REAL       REAL       IMAG

                                                    ALL       1.2761     1.2828      .0000    -3.2524

                                         ***********  THESE VALUES PERTAIN TO A SPINNING MODE  **********
```

```
   DATE     25 JUL 70               EXAMPLE PROBLEM+/W'$590ND TANGENTIAL MODE2 HYOOTHETICAL ENGINE                PAGE    29

TRANSVERSE STABILITY PROGRAM... CALCULATES  HR, HI

INPUT DATA
        SNH          ZE            GAMMA          UE          SOUND (FT/SEC)   ULM (FT/SEC)

   3.05423680    2.31250000    1.21800     .18852762         3800.00         20.420000


NOZZLE ADMITTANCES INPUT
                  OMEGA(CH)           ERT                  EIT                  CRT                  CIT

                  1.6938890       .23612978+00         .46052134+00         .43929329-02        -.38921888-03
                  1.7675364       .22195471+00         .37663660+00         .38370105-02        -.21846309-02
                  1.8411837       .20942160+00         .29232068+00         .23591529-02        -.36299161-02
                  1.9148310       .19851034+00         .20618085+00         .17771693-03        -.42791335-02
                  1.9884783       .18923368+00         .11621683+00        -.22948344-02        -.37970401-02
                  2.0253020       .18518268+00         .68934425-01        -.34685028-02        -.30747288-02


CALCULATED RESULTS
                    OMEGA            H REAL               H IMAG
                  2.8098978       .84986281+00         .75610643+00
                  2.8709826       .89863663+00         .10063908+01
                  2.9320673       .94631281+00         .12515626+01
                  2.9931520       .99286398+00         .14920600+01
                  3.0542367       .10382651+01         .17283221+01
                  3.1153214       .10824934+01         .19607932+01
                  3.1764061       .11255292+01         .21899250+01
                  3.2374909       .11673558+01         .24161806+01
                  3.2985756       .12079595+01         .26400370+01
                  3.3596603       .12473300+01         .28619903+01
 PROGRAM  D  OUTPUT
                    OMEGA(C)            HTR            HTI


                    2.809898          .665967         .592498
                    2.870983          .704187         .788625
                    2.932067          .741547         .980746
                    2.993152          .778025        1.169204
                    3.054237          .813602        1.354343
                    3.115321          .848260        1.536511
                    3.176406          .881984        1.716063
                    3.237491          .914760        1.893361
                    3.298576          .946578        2.068778
                    3.359660          .977429        2.242705
```

```
DATE      25 JUL 70             EXAMPLE PROBLEM+/W'$590ND TANGENTIAL MODE2 HYOOTHETICAL ENGINE                     PAGE     31

                                FOLLOWING WILL BE INTERPOLATION WITHIN HTR HTI TABLE GIVEN ABOVE
                   OMEGA             HTRINT             HTIINT
                  2.809898           .665967            .592498
                  2.823994           .674863            .638114
                  2.838091           .683714            .683516
                  2.852187           .692519            .728705
                  2.866284           .701278            .773681
                  2.880380           .709991            .818439
                  2.894477           .718659            .862976
                  2.908573           .727281            .907302
                  2.922670           .735856            .951431
                  2.936766           .744384            .995372
                  2.950863           .752866           1.039112
                  2.964959           .761300           1.082655
                  2.979055           .769687           1.126015
                  2.993152           .778025           1.169204
                  3.007248           .786316           1.212215
                  3.021345           .794558           1.255045
                  3.035441           .802753           1.297705
                  3.049538           .810898           1.340208
                  3.063634           .818994           1.382556
                  3.077731           .827042           1.424743
                  3.091827           .835040           1.466772
                  3.105924           .842969           1.508659
                  3.120020           .850888           1.550414
                  3.134117           .858737           1.592024
                  3.148213           .866536           1.633493
                  3.162310           .874285           1.674835
                  3.176406           .881984           1.716062
                  3.190502           .889632           1.757170
                  3.204599           .897230           1.798151
                  3.218695           .904777           1.839019
                  3.232792           .912273           1.879790
                  3.246888           .919717           1.920466
                  3.260985           .927111           1.961033
                  3.275081           .934454           2.001504
                  3.289178           .941745           2.041891
                  3.303274           .948985           2.082210
                  3.317371           .956173           2.122452
                  3.331467           .963310           2.162616
                  3.345564           .970395           2.202700
                  3.359660           .977429           2.242704
```

```
DATE      25 JUL 70              EXAMPLE PROBLEM+/W'$590ND TANGENTIAL MODE2 HYOOTHETICAL ENGINE                PAGE    32
PROGRAM  F  OUTPUT

                  FC(CPS)            (OMGA)D,            TAU(MS)             N
                   3186.4            2.80990             .08428            .59655
                   3202.4            2.82399             .08085            .63911
                   3218.3            2.83809             .07769            .68352
                   3234.3            2.85219             .07479            .72965
                   3250.3            2.86628             .07211            .77742
                   3266.3            2.88038             .06964            .82672
                   3282.3            2.89448             .06734            .87747
                   3298.3            2.90857             .06521            .92958
                   3314.2            2.92267             .06323            .98301
                   3330.2            2.93677             .06138           1.03768
                   3346.2            2.95086             .05964           1.09353
                   3362.2            2.96496             .05802           1.15048
                   3378.2            2.97906             .05650           1.20850
                   3394.2            2.99315             .05506           1.26754
                   3410.2            3.00725             .05371           1.32756
                   3426.1            3.02134             .05244           1.38848
                   3442.1            3.03544             .05123           1.45029
                   3458.1            3.04954             .05009           1.51296
                   3474.1            3.06363             .04900           1.57646
                   3490.1            3.07773             .04797           1.64072
                   3506.1            3.09183             .04699           1.70573
                   3522.1            3.10592             .04605           1.77148
                   3538.0            3.12002             .04516           1.83796
                   3554.0            3.13412             .04430           1.90511
                   3570.0            3.14821             .04349           1.97290
                   3586.0            3.16231             .04270           2.04135
                   3602.0            3.17641             .04195           2.11045
                   3618.0            3.19050             .04123           2.18017
                   3633.9            3.20460             .04054           2.25046
                   3649.9            3.21870             .03987           2.32135
                   3665.9            3.23279             .03923           2.39284
                   3681.9            3.24689             .03861           2.46493
                   3697.9            3.26098             .03801           2.53755
                   3713.9            3.27508             .03744           2.61073
                   3729.9            3.28918             .03688           2.68449
                   3745.8            3.30327             .03634           2.75883
                   3761.8            3.31737             .03582           2.83373
                   3777.8            3.33147             .03531           2.90917
                   3793.8            3.34556             .03482           2.98515
                   3809.8            3.35966             .03434           3.06165

                  THE FOLLOWING ARE VALUES INTERPOLATED AT SLOPE OF N=0.0

                                NMIN=     .27513
                                TAU(MS)=     .19057
                                (OMEGA)D=   2.60202
                                FC(CPS)=    2950.6
```

```
DATE      25 JUL 70             \T LODE2 RHOWING EFFECTS OF DEADBAND NONLINEARITY                          PAGE    33
  A       B       C       D       E       F       G       H       I       J
  0.      0.      0.      0.    200.      0.      0.      0.     50.      0.

     *********************** THE FOLLOWING MAIN CONTROL DATA WILL BE USED IN THIS CASE ***********************

                              RATIO OF SPECIFIC HEAT (GAMMA) =  1.2180

                              DESIRED MACH NUMBER =     .00000    (=0 IF BEING CALCULATED)

                              CHAMBER RADIUS =   6.400 (INCHES)

                              CHAMBER LENGTH =  14.800 (INCHES)

                              SPEED OF SOUND =   3800.000 (FT/SEC)

                              CHAMBER MODE DESCRIPTION =  1.84118 (=0 FOR LONGITUDINAL MODES)


                         ********** CHAMBER FREQUENCIES (WC) **********

        1.69389            1.73071            1.76754            1.80436            1.84118
        1.87801            1.91483            1.95165            1.98848            2.02530
```

```
DATE      25 JUL 70              \T LODE2 RHOWING EFFECTS OF DEADBAND NONLINEARITY                          PAGE    35

 INPUT TO INJECTION DISTRIBUTION PROGRAM
      CONSTANTS
          NUMBER OF OMEGAS =  6
          NUMBER OF ELEMENTS =    3 FOR EACH OF  180 SYMMETRIC SECTIONS.
          RADIAL DIVISIONS(XM) =  20.
          ANGULAR DIVISIONS (XN) = 180.
          ACOUSTIC MODE NUMBER(SVN) = 1.8412
          ORDER OF BESSEL FUNCTIONS(V) = 1.
          INJECTOR RADIUS =   6.400 IN.
          RATIO OF SPECIFIC HEATS(GAMMA) = 1.2180
          MAXIMUM PRESSURE AMPLITUDE RATIO(POO) =  1.000

          TRANSFER FUNCTIONS FOR LINEAR OPERATION
                 PRESSURE(TFLP) =   .000
                 RADIAL VELOCITY(TFLR) =  1.000
                 TANGENTIAL VELOCITY(TFLT) =   .000

      INPUT FREQUENCIES
                1.6939          1.7675          1.8412          1.9148          1.9885
                2.0253
```

```
ELEMENT INFORMATION

                    ELEMENT           RADIUS           ANGLE          DISTRIBUTION
                                                                       COEFFICIENT
                      NO.               IN.            RADIANS             MU

                      1               2.000            .0175            4.4444
                      2               4.000            .0175            7.7778
                      3               6.000            .0175            7.7778
```

TABULAR NONLINEAR EFFECTS

PRESSURE PERTURBATION	COMBUSTION GAIN	RADIAL VELOCITY PERTURBATION	COMBUSTION GAIN	TANGENTIAL VELOCITY PERTURBATION	COMBUSTION GAIN
.000	.000	-5.100	-5.000	.000	.000
.000	.000	-2.100	-2.000	.000	.000
.000	.000	-1.100	-1.000	.000	.000
.000	.000	-.600	-.500	.000	.000
.000	.000	-.200	-.100	.000	.000
.000	.000	-.150	-.050	.000	.000
.000	.000	-.110	-.010	.000	.000
.000	.000	-.100	.000	.000	.000
.000	.000	-.090	.000	.000	.000
.000	.000	-.050	.000	.000	.000
.000	.000	-.010	.000	.000	.000
.000	.000	.010	.000	.000	.000
.000	.000	.050	.000	.000	.000
.000	.000	.090	.000	.000	.000
.000	.000	.100	.000	.000	.000
.000	.000	.110	.010	.000	.000
.000	.000	.150	.050	.000	.000
.000	.000	.200	.100	.000	.000
.000	.000	.600	.500	.000	.000
.000	.000	1.100	1.000	.000	.000
.000	.000	2.100	2.000	.000	.000
.000	.000	5.100	5.000	.000	.000
.000	.000	.000	.000	.000	.000
.000	.000	.000	.000	.000	.000
.000	.000	.000	.000	.000	.000
.000	.000	.000	.000	.000	.000
.000	.000	.000	.000	.000	.000
.000	.000	.000	.000	.000	.000
.000	.000	.000	.000	.000	.000
.000	.000	.000	.000	.000	.000
.000	.000	.000	.000	.000	.000
.000	.000	.000	.000	.000	.000
.000	.000	.000	.000	.000	.000
.000	.000	.000	.000	.000	.000
.000	.000	.000	.000	.000	.000
.000	.000	.000	.000	.000	.000
.000	.000	.000	.000	.000	.000
.000	.000	.000	.000	.000	.000
.000	.000	.000	.000	.000	.000
.000	.000	.000	.000	.000	.000
.000	.000	.000	.000	.000	.000
.000	.000	.000	.000	.000	.000
.000	.000	.000	.000	.000	.000
.000	.000	.000	.000	.000	.000
.000	.000	.000	.000	.000	.000
.000	.000	.000	.000	.000	.000
.000	.000	.000	.000	.000	.000
.000	.000	.000	.000	.000	.000
.000	.000	.000	.000	.000	.000
.000	.000	.000	.000	.000	.000

```
DATE     25 JUL 70           \T LODE2 RHOWING EFFECTS OF DEADBAND NONLINEARITY                          PAGE    37

         RESULTS OF DESCRIBING FUNCTION

                    ELEMENT   RADIUS     ANGLE   FRACTIONAL
          OMEGA       NO,                 RAD    FLOW-RATE          FP              FR              FT

          1.6939       1        .313     .0175    .00123          1.000000        .678814        1.000000
          1.6939       2        .625     .0175    .00216          1.000000        .493693        1.000000
          1.6939       3        .938     .0175    .00216          1.000000        .000000        1.000000
```

```
RESULTS OF DESCRIBING FUNCTION

             ELEMENT   RADIUS     ANGLE   FRACTIONAL
 OMEGA         NO.                  RAD    FLOW-RATE          FP            FR            FT

 1.7675         1        .313     .0175     .00123         1.000000       .665192      1.000000
 1.7675         2        .625     .0175     .00216         1.000000       .472795      1.000000
 1.7675         3        .938     .0175     .00216         1.000000       .000000      1.000000
```

```
RESULTS OF DESCRIBING FUNCTION

            ELEMENT   RADIUS     ANGLE   FRACTIONAL
OMEGA         NO.                  RAD    FLOW-RATE          FP              FR             FT

1.8412         1        .313     .0175     .00123        1.000000         .651640       1.000000
1.8412         2        .625     .0175     .00216        1.000000         .452589       1.000000
1.8412         3        .938     .0175     .00216        1.000000         .000000       1.000000
```

```
DATE      25 JUL 70              \T LODE2 RHOWING EFFECTS OF DEADBAND NONLINEARITY                          PAGE    40

        RESULTS OF DESCRIBING FUNCTION

                   ELEMENT    RADIUS     ANGLE   FRACTIONAL
         OMEGA       NO.                   RAD    FLOW-RATE           FP               FR               FT


         1.9148       1        .313     .0175     .00123         1.000000          .637977         1.000000
         1.9148       2        .625     .0175     .00216         1.000000          .432330         1.000000
         1.9148       3        .938     .0175     .00216         1.000000          .000000         1.000000
```

```
RESULTS OF DESCRIBING FUNCTION

             ELEMENT   RADIUS    ANGLE   FRACTIONAL
OMEGA          NO.                 RAD   FLOW-RATE          FP              FR              FT

1.9885          1        .313    .0175     .00123        1.000000         .624504        1.000000
1.9885          2        .625    .0175     .00216        1.000000         .412330        1.000000
1.9885          3        .938    .0175     .00216        1.000000         .000000        1.000000
```

```
RESULTS OF DESCRIBING FUNCTION

             ELEMENT   RADIUS    ANGLE   FRACTIONAL
 OMEGA         NO.                 RAD    FLOW-RATE          FP             FR             FT

 2.0253         1       .313     .0175    .00123         1.000000        .617796       1.000000
 2.0253         2       .625     .0175    .00216         1.000000        .402424       1.000000
 2.0253         3       .938     .0175    .00216         1.000000        .000000       1.000000
```

```
DAIE      25 JUL 70              \T LODE2 RHOWING EFFECTS OF DEADBAND NONLINEARITY                       PAGE    43

         RESULTS OF INJECTION DISTRIBUTION EFFECTS
                                            OMEGA       AVN        BVN        CVN        CVN
                                                        REAL       REAL       REAL       IMAG


                                            1.6939     1.2745      .4300      .0000    -1.8125
                                            1.7675     1.2745      .4158      .0000    -1.8125
                                            1.8412     1.2745      .4020      .0000    -1.8125
                                            1.9148     1.2745      .3881      .0000    -1.8125
                                            1.9885     1.2745      .3744      .0000    -1.8125
                                            2.0253     1.2745      .3676      .0000    -1.8125

                                 ***********  THESE VALUES PERTAIN TO A SPINNING MODE  **********
```

Report 20672-P3D

N71-20417

Appendix A

ANALYSIS OF THE LOW FREQUENCY STABILITY OF A GAS-TAPOFF CYCLE USING GASEOUS HYDROGEN AND GASEOUS OXYGEN

TABLE OF CONTENTS

SECTION I - PROBLEM

A. ABSTRACT

This analysis deals with a feed system coupled one-dimensional stability problem for a gas tapoff cycle using gaseous hydrogen and oxygen. The tapoff cycle considers that a portion of the combustion product gases are "tapped off" for use elsewhere. This program can be run with or without the tapoff cycle included. It includes the wave motion and heat motion (entropy wave) in the chamber. The burning process is treated as a process of temperature, molecular weight and specific heat change. A very general expression relating injection rate to burning rate is used.

The many equations obtained are solved using matrices. The matrix operations however are used only on the equations involving real variables. The complex arithmetic used on the complex equations has been built into the program so the program can be used on a variety of computers.

The final result of the analysis is a system characteristic equation. Stability of the system can be determined using the Nyquist criteria, Ref. 1.

B. TECHNICAL DESCRIPTION

The feed system coupled stability problem at low frequencies can be analyzed conveniently by dividing the problem into two parts, the feed system and the chamber. Each part can be analyzed separately and then combined to determine the system stability. Method of analyzing the feed system is well developed, Ref. 2, but the traditional method of analyzing the chamber is quite primitive. The traditional method, Ref. 3 and 4 , considers the volume of propellant injected into the chamber to be negligible and the burning to be a process of gas generation. In other words the volume of unburned propellant is assumed negligible compared to the volume of burned gases. This is a good assumption for liquid propellants burning at low chamber pressures. But when the propellants are gaseous, especially gaseous hydrogen, this assumption is not good. Actually unburned gases flow into the chamber and when burning occurs gas properties (such as molecular weight, temperature and specific heat) change. The same weight flow of gas exists before and after burning.

B, Technical Description (cont.)

This analysis looks only at the chamber analysis but has provision for adding the feed system effects. For simplicity the combustion is assumed to occur at one discrete plane. This is justified since the combustion does occur over a length that is small compared to the wave length of the oscillation. After combustion the gas goes through the sonic exhaust nozzle. The dynamics of a sonic exhaust nozzle have been well analyzed in reference 5 . The results of that analysis will be used here. In addition to the propellant feed lines an additional feature must be accounted for in the gas tapoff cycle, the dynamics of the tapoff duct. To analyze the combustion front the following relationships must be used:

Conservation of mass

$$\dot{w}_t + \dot{w}_x + \dot{w}_f = \dot{w}_p$$

Conservation of Momentum

$$P_f = P_x$$

$$P_x = P_p$$

Conservation of energy

$$\dot{w}_x h_x + \dot{w}_f h_f + \dot{w}_t h_t = \dot{w}_p h_p$$

Definition of enthalpy for a chemical system

$$h_x = hO_x + \int_{T_R}^{T_x} CP_x \; dT$$

$$h_f = hO_f + \int_{T_R}^{T_f} CP_f \; dT$$

B, Technical Description (cont.)

$$h_p = hO_p + \int_{T_R}^{T_p} CP_p \; dT$$

$$hO_p = Y_{H_2} \; hO_{H_2} + Y_{H_2O} \; hO_{H_2O}$$

Definition of the product specific heat in terms of its composition

$$CP_p = Y_{H_2} \; CP_{H_2} + Y_{H_2O} \; CP_{H_2O}$$

Definition of the mass fraction of species in the products

$$Y_{H_2O} = \frac{9}{8} \; \frac{\dot{w}_x}{\dot{w}_p}$$

$$Y_{H_2} = 1 - Y_{H_2O}$$

Equation of state

$$P = R \rho T$$

$$R = \frac{\mathcal{R} g}{MW}$$

Definition of the molecular weight of the products

$$MW_p = X_{H_2} \; MW_{H_2} + X_{H_2O} \; MW_{H_2O}$$

B, Technical Description (cont.)

Definition of the mole fraction of species in products

$$X_{H_2} = \frac{\dfrac{Y_{H_2}}{MW_{H_2}}}{\dfrac{Y_{H_2}}{MW_{H_2}} + \dfrac{Y_{H_2O}}{MW_{H_2O}}}$$

$$X_{H_2O} = 1 - X_{H_2}$$

The analysis of the sonic exhaust nozzle results in a relationship between perturbations in pressure, velocity and entropy:

$$\gamma_p \; v_p' + \alpha p'_p + \beta \; s_p' = 0$$

The entropy can be related to other properties through the thermodynamics equation

$$T \; ds = dh - \frac{1}{g\rho} \; dp$$

The injection rates are related to the feed line dynamics by the following equations:

B, Technical Description (cont.)

$$\frac{w'_x}{p'_x} = \frac{e^{-i\omega\tau_x}}{Z_x}$$

$$\frac{w'_f}{p'_f} = \frac{e^{-i\omega\tau_f}}{Z_f}$$

$$\frac{w'_t}{P'_x} = \frac{1}{Z_t}$$

The final equation which completely characterizes the system stability can be written as

$$\frac{F\ e^{-i\omega\tau_x}}{Z_x} + \frac{G\ e^{-i\omega\tau_f}}{Z_f} + \frac{H}{Z_t} + 1 = 0$$

The problem then amounts to the determination of F, G and H from the previous equations. To do this, all variables must be written as a mean value plus a perturbation value which varies with time. Each equation can then be divided into a mean equation and a perturbation equation. There are 27 perturbation equations which must be solved simultaneously. This is done with the help of matrices. Since 22 of the equations contain only real coefficients they will be solved using matrices as follows:

$$\left[AA\right]\left[BB\right] + \left[CC\right]\dot{w}_x' + \left[DD\right]\dot{w}_f' + \left[EE\right]\dot{w}_t' + \left[FF\right]P'_p + \left[GG\right]S'_p = \left[0\right]$$

The matrix $\left[AA\right]$ is a 22 x 22 matrix and the others are column matrices. All the coefficients of these matrices are real. The remaining 5 equations involve complex variables and are solved simultaneously to obtain the final equation.

There are some important assumptions that are implied by the basic equations presented. First in the momentum and energy equations, the gas momentum and kinetic energy terms have been dropped. This assumes that the Mach number is small compared to unity. Also, in developing the perturbation equations, products of perturbations are neglected which means that only small amplitude oscillations can be treated (small compared to the mean variables). In defining the properties of the products only hydrogen and water are assumed to be in the products which infers that the engine must be running fuel rich. The analysis is limited to an engine using gaseous oxygen and gaseous hydrogen.

C. EQUATIONS

The following equations are used to determine some of the mean variables which are used as coefficients in the perturbation equations:

$$\bar{T}_p = \bar{T}_o \left(\frac{\gamma_p - 1}{2} M_p^2 + 1 \right)^{-1}$$

$$\bar{C}_p = \left(\gamma_p \frac{\mathcal{R} g}{MW_p} \bar{T}_o \right)^{1/2}$$

$$\bar{w}_f = \bar{w}_p (1 + MR)^{-1}$$

$$\bar{w}_x = \bar{w}_f (MR)$$

$$\bar{C}_x^2 = \gamma_x g \frac{\mathcal{R}}{MW_x} \bar{T}_x$$

$$\bar{C}_p = \bar{C}_o \left(\frac{\gamma_p - 1}{2} M_p^2 + 1 \right)^{-1/2}$$

$$\bar{v}_p = M_p \bar{C}_p$$

C, Equations (cont.)

$$\bar{\rho}_p = \frac{\bar{P}_p \; \overline{MW}_p}{\mathcal{R}_g \; \bar{T}_p}$$

$$\bar{h}_x = CP_{O_2} \; (\bar{T}_x - T_R)$$

$$\bar{h}_f = CP_{H_2} \; (\bar{T}_f - T_R)$$

$$\bar{\rho}_f = \frac{\bar{P}_f \; \overline{MW}_f}{\mathcal{R}_g \; \bar{T}_f}$$

$$\bar{\rho}_x = \frac{\bar{P}_x \; MW_x}{\mathcal{R}_g \; \bar{T}_x}$$

$$\bar{h}_p = (\bar{h}_x \, \bar{\dot{w}}_x + \bar{h}_f \, \bar{\dot{w}}_f) \; / \; \bar{\dot{w}}_p$$

$$\bar{Y}_{H_2} = \left(\bar{\dot{w}}_f - \frac{\bar{\dot{w}}_x}{8} \right) / \; \bar{\dot{w}}_p$$

$$\bar{Y}_{H_20} = 1 - \bar{Y}_{H_2}$$

$$R_x = \frac{\mathcal{R}_g}{MW_x}$$

$$\bar{R}_p = \frac{\mathcal{R}_g}{\overline{MW}_p}$$

$$R_f = \frac{\mathcal{R}_g}{MW_f}$$

$$\alpha = \frac{\bar{C}_o}{\bar{P}_o} \mathcal{A} \, \gamma_p$$

$$\beta = \frac{\bar{C}_o}{CP_p} \mathcal{C} \, \gamma_p$$

The perturbation equations are:

$$w_t' + w_x' + w_f' - w_p' = 0$$

$$P_x' - P_p' = 0$$

$$\bar{h}_x \dot{w}_x' + \dot{\bar{w}}_x h_x' + \dot{w}_f' \bar{h}_f + \dot{\bar{w}}_f h_f' + \dot{\bar{w}}_t h_t + \dot{w}_t' \bar{h}_t - \dot{\bar{w}}_p h_p' - \dot{w}_p' \bar{h}_p = 0$$

$$h_x' - CP_x T_x' = 0$$

$$h_f' - CP_f T_f' = 0$$

$$h_p' - h0_p' - \overline{CP}_p T_p' - CP_p' (\bar{T}_p - \bar{T}_R) = 0$$

$$h0_p' - Y_{H'_2} \overline{h0}_{H_2} - Y'_{H_20} \overline{h0}_{H_20} = 0$$

$$CP'_p - Y_{H'_2} \overline{CP}_{H_2} - Y'_{H_20} \overline{CP}_{H_20} = 0$$

$$Y'_{H_2} + Y_{H'_20} = 0$$

$$Y'_{H_20} - \frac{9}{88} \frac{\dot{w}_x}{\bar{\dot{w}}_p} + \frac{9}{8} \frac{\dot{\bar{w}}_x}{\bar{\dot{w}}_p^2} \dot{w}'_p = 0$$

$$P_x' - R_x \bar{\rho}_x T_x' - R_x \bar{T}_x \rho_x' = 0$$

$$P_f' - R_f \bar{\rho}_f T_f' - R_f \bar{T}_f \rho_f' = 0$$

$$P_x' - \frac{\gamma_x \bar{P}_x}{\bar{\rho}_x} \rho_x' = 0$$

$$v_x' - \frac{\dot{w}_x'}{\bar{\rho}_x g A_x} + \frac{\bar{v}_x}{\bar{\rho}_x} \rho_x' = 0$$

$$v_p' - \frac{\dot{w}_p'}{\bar{\rho}_p A_c \; g} + \frac{\bar{v}_p}{\bar{\rho}_p} \rho_p' = 0$$

$$P_p' - \bar{R}_p \bar{\rho}_p T_p' - R_p \bar{T}_p \rho_p' - \bar{\rho}_p \bar{T}_p R_p' = 0$$

$$\bar{T}_p S_p' - h_p' + \frac{P_p'}{g \bar{\rho}_p} = 0$$

$$R_p' + \frac{g}{\overline{MW}_p^{\,2}} MW_p' = 0$$

$$MW'_p - X'_{H_2} MW_{H_2} - X'_{H_2O} MW_{H_2O} = 0$$

$$X'_{H_2} = Y'_{H_2} \left[\frac{\frac{1}{MW_{H_2}}}{\frac{\bar{Y}_{H_2}}{MW_{H_2}} + \frac{\bar{Y}_{H_20}}{MW_{H_2O}}} - \frac{\frac{\bar{Y}_{H_2}}{MW_{H_2}^{\,2}}}{\left(\frac{\bar{Y}_{H_2}}{MW_{H_2}} + \frac{\bar{Y}_{H_20}}{MW_{H_20}} \right)^2} \right]$$

$$+ Y'_{H_20} \frac{\frac{\bar{Y}_{H_2}}{MW_{H_20} \; MW_{H_2}}}{\left(\frac{\bar{Y}_{H_2}}{MW_{H_2}} + \frac{\bar{Y}_{H_20}}{MW_{H_20}} \right)^2} = 0$$

$$X'_{H_20} + X'_{H_2} = 0$$

The 22 equations above will be solved using matrices. The remaining perturbation equations are:

$$P_x' - P_f' = 0$$

$$\dot{w}_x' - \frac{e^{-i\omega\tau_x}}{Z_x} P_x' = 0$$

$$\dot{w}_f' - \frac{e^{-i\omega\tau_f}}{Z_f} P_f' = 0$$

$$\dot{w}_t' - \frac{1}{Z_t} P_t' = 0$$

$$v_p' + \alpha P_p' + \beta S_p' = 0$$

There are also several constants which are built into the program:

$$\gamma_x = 1.4$$

$$\gamma_f = 1.4$$

$$MW_x = 32 \frac{\text{lb}}{\text{lb - mole}}$$

$$MW_f = 2 \frac{\text{lb}}{\text{lb - mole}}$$

$$\mathcal{R} = 1545 \frac{\text{lb}}{\text{lb - mole } {}^\circ\text{R}}$$

$$g = 32.172 \text{ ft/sec}^2$$

$$C_{p_x} = \frac{\gamma_x \, \mathcal{R}}{(\gamma_x - 1)\, MW_x} = 169 \, \frac{\text{ft-lb}}{\text{lb} \, ^o\text{R}}$$

$$C_{p_f} = \frac{\gamma_f \, \mathcal{R}}{(\gamma_f - 1)\, MW_f} = 2710 \, \frac{\text{ft-lb}}{\text{lb} \, ^o\text{R}}$$

$$h0_{H_2 0} = -4.498 \times 10^6 \, \frac{\text{ft-lb}}{\text{lb}}$$

$$CP_{H_2 0} = 555 \, \frac{\text{ft-lb}}{\text{lb} \, ^o\text{R}}$$

$$T_r = 537 \, ^o\text{R}$$

$$MW_{H_2} = 2$$

$$MW_{H_2 0} = 18$$

$$h0_{H_2} = 0.0$$

For convenience a diagram of the matrices is shown on page 13. The coefficients $A_{21,\,12}$, $A_{21,\,13}$, $A_{22,\,12}$, $A_{22,\,13}$ are defined as follows:

$$A_{21,12} = - \frac{\dfrac{1}{MW_{H_2}}}{\dfrac{\bar{Y}_{H_2}}{MW_{H_2}} + \dfrac{\bar{Y}_{H_2 0}}{MW_{H_2 0}}} + \frac{\dfrac{\bar{Y}_{H_2}}{MW^2_{H_2}}}{\left(\dfrac{\bar{Y}_{H_2}}{MW_{H_2}} + \dfrac{\bar{Y}_{H_2 0}}{MW_{H_2 0}} \right)^2}$$

$$A_{21,\ 13} = \frac{\dfrac{\bar{Y}_{H_2}}{MW_{H_2}} \quad \dfrac{1}{MW_{H_2O}}}{\left(\dfrac{\bar{Y}_{H_2}}{MW_{H_2}} + \dfrac{\bar{Y}_{H_2O}}{MW_{H_2O}}\right)^2}$$

$$A_{22,\ 12} = \frac{\dfrac{\bar{Y}_{H_2O}}{MW_{H_2O}} \quad \dfrac{1}{MW_{H_2}}}{\left(\dfrac{\bar{Y}_{H_2}}{MW_{H_2}} + \dfrac{\bar{Y}_{H_2O}}{MW_{H_2O}}\right)^2}$$

$$A_{22,\ 13} = -\frac{\dfrac{1}{MW_{H_2O}}}{\dfrac{\bar{Y}_{H_2O}}{MW_{H_2O}} + \dfrac{\bar{Y}_{H_2}}{MW_{H_2}}} + \frac{\dfrac{\bar{Y}_{H_2O}}{\left(MW_{H_2O}\right)^2}}{\left(\dfrac{\bar{Y}_{H_2}}{MW_{H_2}} + \dfrac{Y_{H_2O}}{MW_{H_2O}}\right)^2}$$

$$
\left[\begin{array}{c|cccccccccccccccccccccc}
 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 & 21 & 22 \\
\hline
1 & 1.0 & & & & & & & & & & & & & & & & & & & & & \\
2 & & -\overline{\dot{W}}_x & -gA_c & & & & & & & & & & & & & & & & & & & \\
3 & -\overline{h}_p & & & \overline{\dot{W}}_x & \overline{\dot{W}}_f & \overline{\dot{W}}_r-\overline{\dot{W}}_p & & & & & & & & & & & & & & & & \\
4 & & & & 1.0 & & & -\overline{CP}_x & & & & & & & & & & & & & & & \\
5 & & & & & 1.0 & & & -\overline{CP}_f & & & & & & & & & & & & & & \\
6 & & & & & & 1.0 & & & -1.0 & -\overline{CP}_p & \overline{T}_R-\overline{T}_p & & & & & & & & & & & \\
7 & & & & & & & & & 1.0 & & & & -\overline{hO}_{H_2O} & & & & & & & & & \\
8 & & & & & & & & & & & 1.0 & -\overline{CP}_{H_2} & -\overline{CP}_{H_2O} & & & & & & & & & \\
9 & & & & & & & & & & & & 1.0 & 1.0 & & & & & & & & & \\
10 & \frac{g\overline{\dot{W}}_x}{8\overline{\dot{W}}_p^2} & & & & & & & & & & & & 1.0 & & & & & & & & & \\
11 & & & 1.0 & & & & -R_x\overline{\rho}_x & & & & & & & -R_x\overline{T}_x & & & & & & & & \\
12 & & & & & & & & -R_f\overline{\rho}_f & & & & & & & 1.0 & -R_f\overline{T}_f & & & & & & \\
13 & & & 1.0 & & & & & & & & & & & -\gamma_x\frac{\overline{P}_x}{\overline{\rho}_x} & & & & & & & & \\
14 & & & & & & & & & & & & & & & 1.0 & -\gamma_f\frac{\overline{P}_f}{\overline{\rho}_f} & & & & & & \\
15 & & 1.0 & & & & & & & & & & & & \frac{\overline{V}_x}{\overline{\rho}_x} & & & & & & & & \\
16 & \frac{-1}{\rho_p A_c g} & & & & & & & & & & & & & & & & 1.0 & \frac{\overline{V}_p}{\overline{\rho}_p} & & & & \\
17 & & & & & & & & & & -R_p\overline{\rho}_p & & & & & & & & -\overline{R}_p\overline{T}_p & -\overline{\rho}_p\overline{T}_p & & & \\
18 & & & & & & & & & & \overline{CV}_p & & & & & & & & \frac{-\overline{P}_p}{g\overline{\rho}_p^2} & & & & \\
19 & & & & & & & & & & & & & & & & & & & 1.0 & \frac{R}{MW_p} & & \\
20 & & & & & & & & & & & & & & & & & & & & 1.0 & -MW_{H_2} & -MW_{H_2O} \\
21 & & & & & & & & & & & & A_{21,12} & A_{21,13} & & & & & & & & 1.0 & \\
22 & & & & & & & & & & & & A_{22,12} & A_{22,13} & & & & & & & & & 1.0
\end{array}\right]
\begin{bmatrix} \dot{W}_p' \\ V_x' \\ P_x' \\ h_x' \\ h_f' \\ h_p' \\ T_x' \\ T_f' \\ hQ_p' \\ T_p' \\ CP_p' \\ Y_{H_2}' \\ Y_{H_2O}' \\ \rho_x' \\ \rho_f' \\ \rho_F' \\ V_p' \\ \rho_p' \\ R_p' \\ MW_p' \\ X_{H_2}' \\ X_{H_2O}' \end{bmatrix}
=
\begin{bmatrix} 1.0 \\ -\overline{V}_x \\ +\overline{h}_x \\ \\ \\ \\ \\ \\ \\ \frac{1}{8\overline{\dot{W}}_p} \\ \\ \\ \\ \\ \frac{1}{\rho_x A_x g} \\ \\ \\ \\ \\ \\ \\ \; \end{bmatrix} \dot{W}_x' +
\begin{bmatrix} 1.0 \\ \\ -\overline{h}_f \\ \\ \\ \\ \\ \\ \\ \frac{-1}{\overline{\dot{W}}_p} \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \; \end{bmatrix} \dot{W}_f' +
\begin{bmatrix} 1.0 \\ \\ \overline{h}_p \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \; \end{bmatrix} \dot{W}_r' +
\begin{bmatrix} \\ -gA_c \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ 1.0 \\ \\ \\ \\ \\ \; \end{bmatrix} P_p' +
\begin{bmatrix} \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ -\overline{T}_p \\ \\ \\ \\ \; \end{bmatrix} S_p'
$$

Figure 1. Diagram of Matrix Multiplication

D. DEFINITION OF TERMS

A	Area
AA	22 x 22 real matrix
$\mathcal{A}$	nozzle admittance coefficient
BB	column matrix multiplying AA
C	speed of sound
c*	characteristic exhaust velocity
$\mathcal{C}$	nozzle admittance coefficient
CC	column matrix multiplying $\dot{w}_x'$
CP	specific heat at constant pressure
DD	column matrix multiplying $\dot{w}_f'$
e	base of the natural logarithm
EE	column matrix multiplying $\dot{w}_t'$
F	coefficient in characteristic equation
FF	column matrix multiplying P_p'
g	acceleration of gravity
G	coefficient in characteristic equation
GG	column matrix multiplying S_p'
h	enthalpy
h0	enthalpy of element at reference temperature of 77°F
H	coefficient in characteristic equation
i	$\sqrt{-1}$
M	Mach number
MR	mixture ratio
MW	molecular weight
P	pressure
$\mathcal{R}$	universal gas constant
R	gas constant

D. Definition of Terms (cont.)

S	entropy
T	temperature
v	velocity
$\dot{w}$	weight flow rate
X	mole fraction
Y	mass fraction
Z	axial coordinate
α	nozzle admittance coefficient
β	nozzle admittance coefficient
γ	ratio of specific heats
ρ	density
τ	total time lag between injection and combustion
ω	frequency

Subscripts

x	oxidizer
f	fuel
t	tapoff
H_2	hydrogen
H_20	water
p	products
o	stagnation conditions
R	reference

E. REFERENCES

1. Marks, Mechanical Engineer's Handbook, 6th Ed., McGraw-Hill Book Co., Inc., 1958.

2. "Apollo Service Module Engine Acoustic Impedance Characteristics of the AJ10-137 Injector", Report Number 3865-529, Aerojet-General Corp., Sacramento, Calif.

3. Summerfield, M., "A Theory of Unstable Combustion In Liquid Propellant Rocket Systems", ARS Journal, Vol. 21, Sept. 1951, pp. 108-114.

4. Crocco, L., Cheng, S. I., "Theory of Combustion Instability In Liquid Propellant Rocket Motors", AGARD Monograph No. (8), Butterworths Scientific Publications, Ltd., London, 1956.

5. Crocco. L., Sirignano, W. A., "Behavior of Supercritical Nozzles Under Three-Dimensional Osciliatory Conditions", AGARDograph 117, North Atlantic Treaty Organization, 1967.

6. "Contributions to the Solution of Linear Equations and the Determination of Eigenvalues". Applied Mathematics Series V. 39, Government Printing Office, 1954, Washington, D.C.

7. "Stability Characterization of Advanced Injectors" Contract NAS 8-20672, Report 20672-II-M6, Aerojet-General Corp., Liquid Rocket Operations, Sacramento, Calif. 7, March 1969.

8. "Stability Characterization of Advanced Injectors Design Guide", Vol. 2, Operation of the Computer Program 20672-P2D January 1970.

9. General Reference Manual, UNIVAC 1108, Sperry-Rand Corp. 1966.

SECTION II - PROGRAMMING

A. SUBROUTINES

This program consists mostly of subroutines. Subroutine CMPLX is really part of the main line of the program and is therefore flow charted in a subsequent section of this report. The subroutines in this program are used for matrix inversion, matrix multiplication, complex multiplication, addition, subtraction, division, and to output results of the programs. These subroutines go by the following names:

MINV
MVMLT
CMULT
CCMLT
CADD
CSUB
CDVID
OUTPT
GSTAB (Main Program)

The subroutine MINV is used to invert the square matrix AA. This subroutine was developed at Aerojet-General Corp. The method was taken from Ref. 6. The following is excerpted from an Aerojet reference Manual.

IDENTIFICATION

MINV - Matrix Inversion Subroutine

Written by F. Yee
Aerojet-General Corporation, Sacramento, California
17 April 1967

PURPOSE

To find the inverse of an n x n real matrix .

RESTRICTIONS

The subroutine could compute an inverse to a singular matrix because of round-off erros introduced in the numerical computations.

METHOD

This subroutine uses the Gauss-Jordan method with complete pivoting.

USAGE

The subroutine is called with the statement:

CALL MINV (A, N, DET, NDET, NA, ITEM)

where:

A = initially the input matrix, finally, the matrix inverse.

N = the input matrix order.

DET = the scaled determinant.

NDET = exponent in powers of ten for the scaled determinant

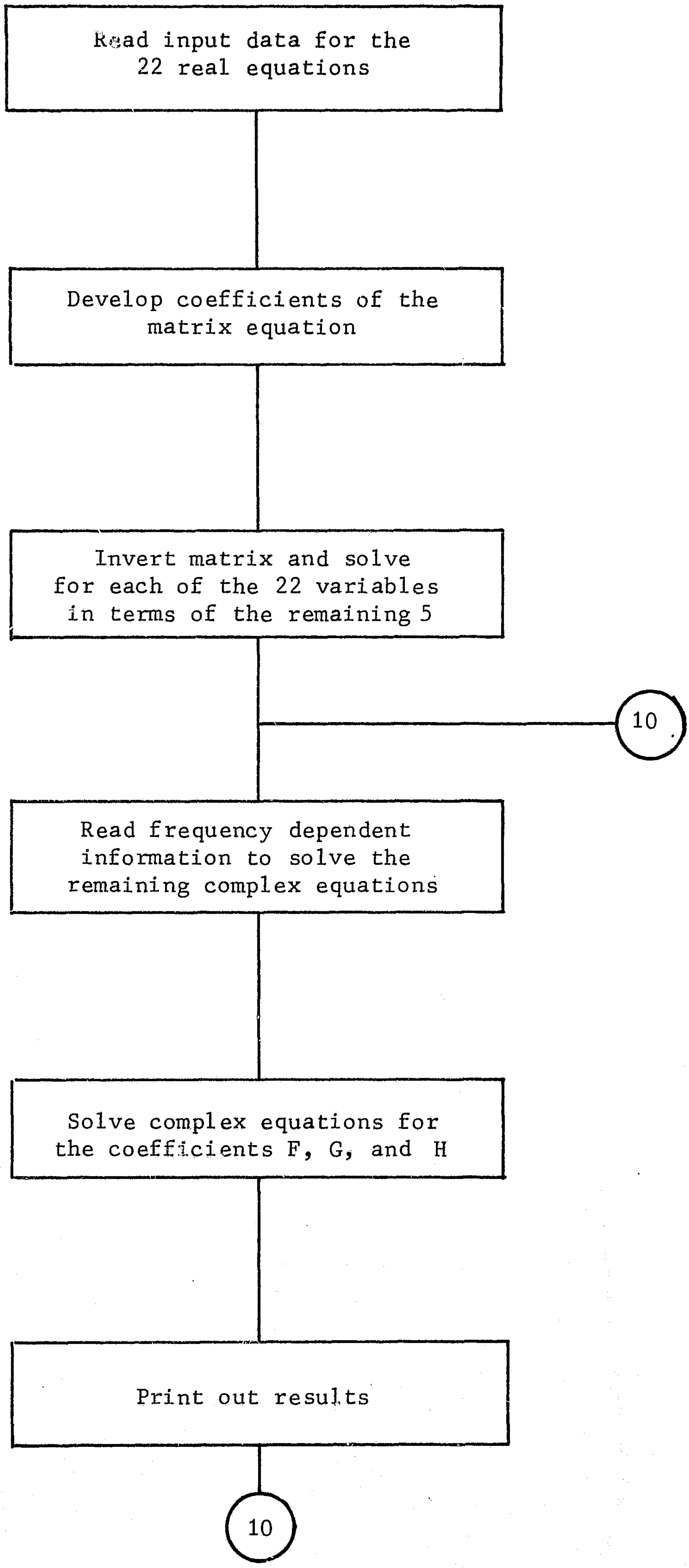
NA = 1st dimension of A; i. e., A(NA, ...)

ITEM = temporary storages with dimension size at least equal to N. After the return from MINV, if ITEM (1) = -1, the inverse has been computed; if ITEM (1) = 0, 1, 2, . . . , the matrix is singular and the positive integer is the rank of the matrix.

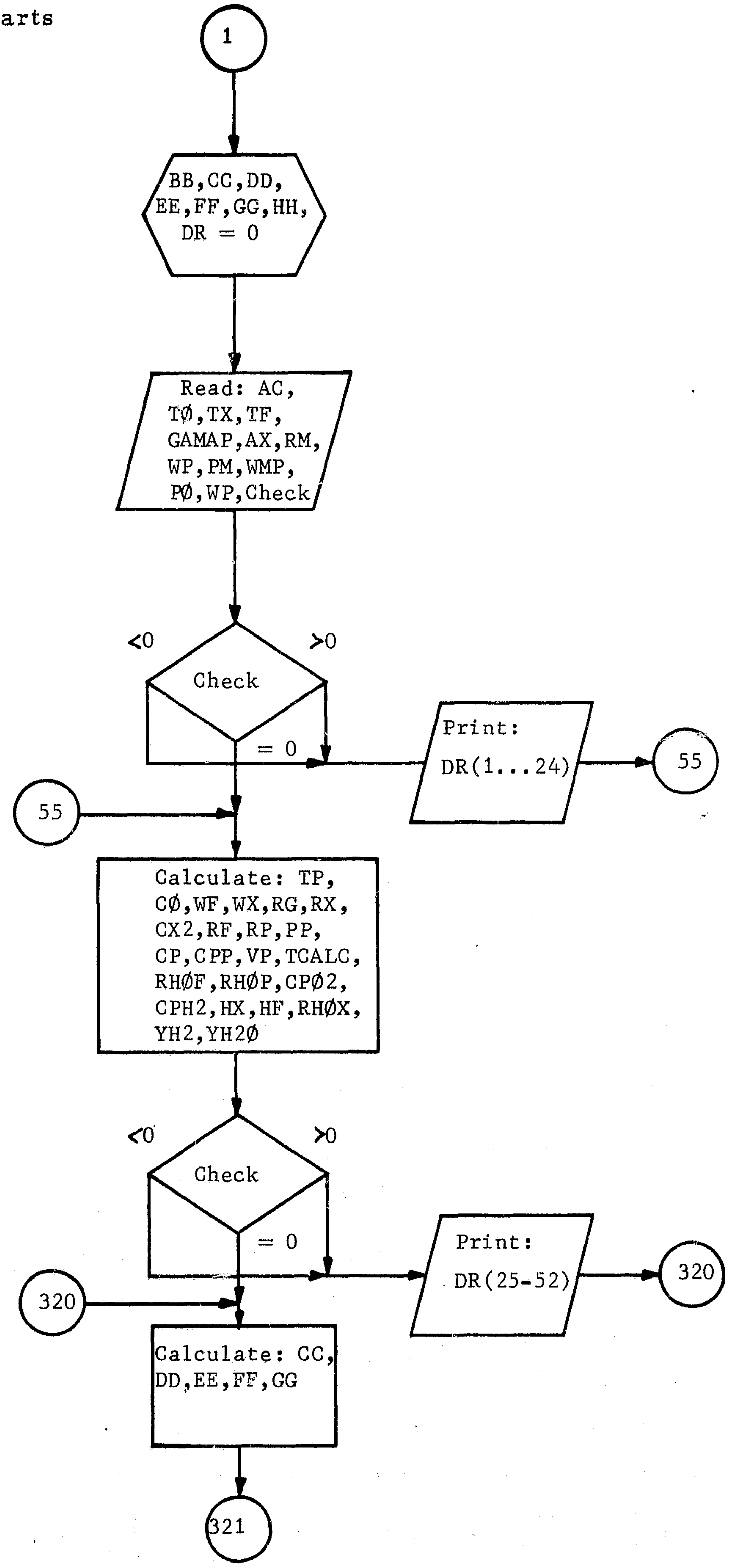
The elements a_{ij} of the matrix A are stored according to the matrix convention

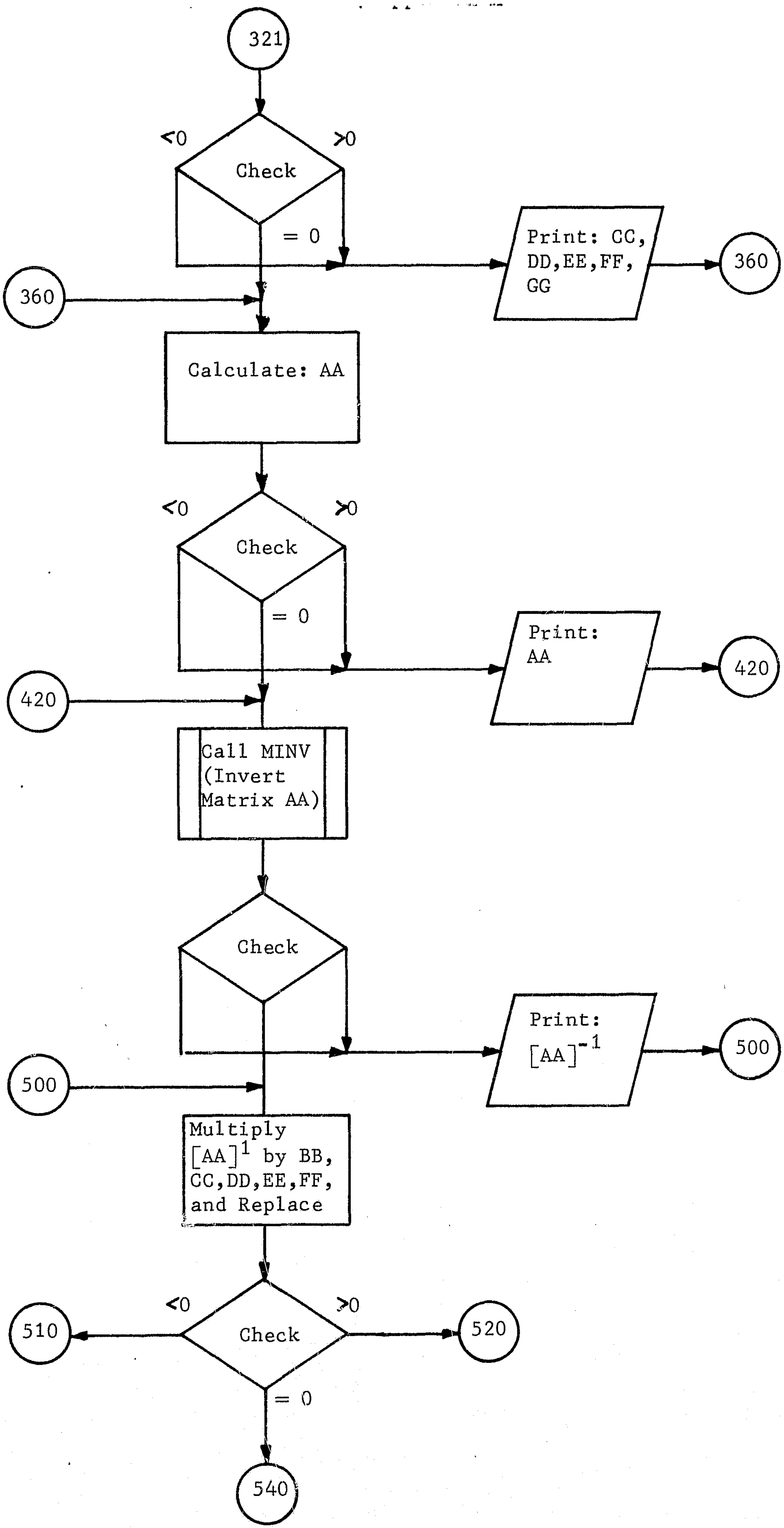
$$A(I, J) = a_{ij} , \quad I \leq N, \quad J \leq N.$$

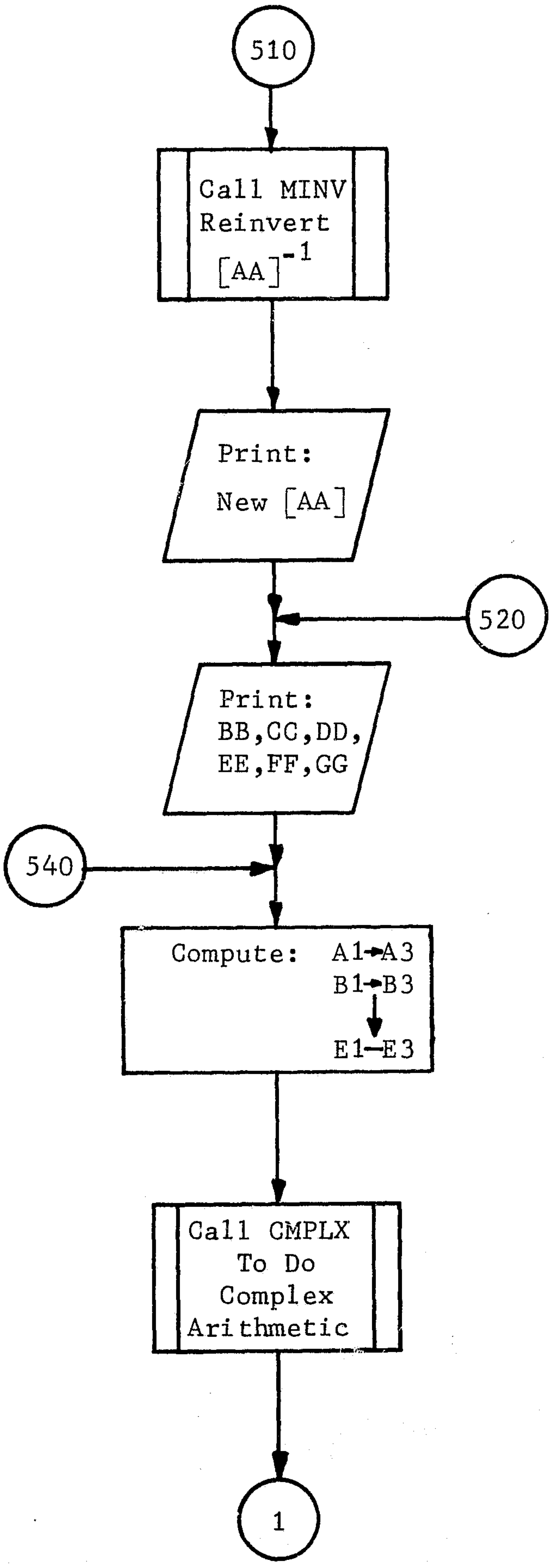
B. Block Diagrams

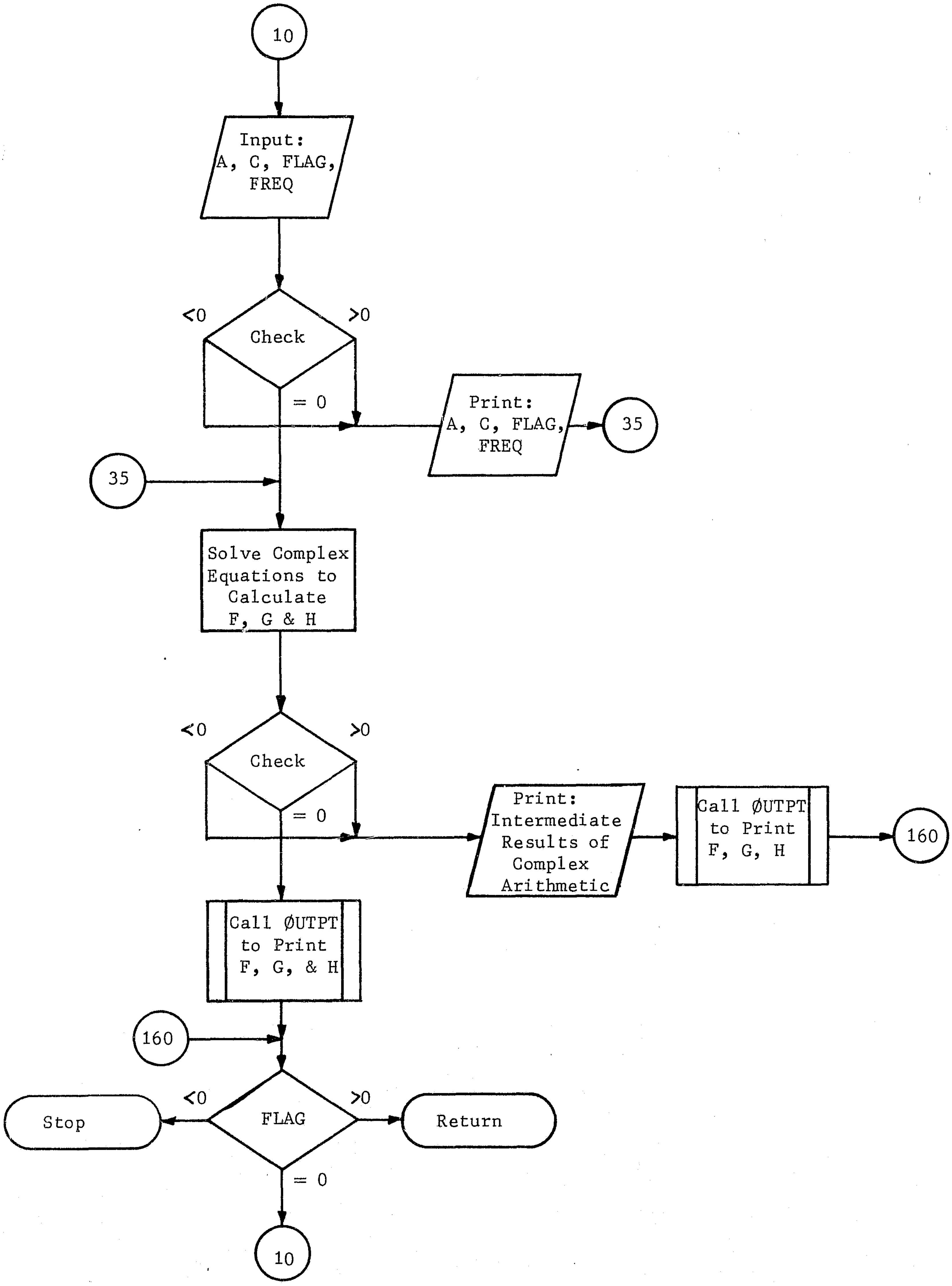


C. Flow Charts




321
<0
>0
Check
= 0
Print: CC, DD,EE,FF, GG
360
360
Calculate: AA
<0
>0
Check
= 0
Print: AA
420
420
Call MINV (Invert Matrix AA)
Check
Print: [AA]$^{-1}$
500
500
Multiply [AA]^{1} by BB, CC,DD,EE,FF, and Replace
<0
>0
510
Check
520
= 0
540


510
Call MINV Reinvert $[AA]^{-1}$
Print: New [AA]
520
Print: BB,CC,DD, EE,FF,GG
540
Compute: A1→A3
B1→B3
E1→E3
Call CMPLX To Do Complex Arithmetic
1


10
Input:
A, C, FLAG,
FREQ
<0
>0
Check
= 0
Print:
A, C, FLAG,
FREQ
35
35
Solve Complex
Equations to
Calculate
F, G & H
<0
>0
Check
= 0
Print:
Intermediate
Results of
Complex
Arithmetic
Call ØUTPT
to Print
F, G, H
160
Call ØUTPT
to Print
F, G, & H
160
<0
>0
Stop
FLAG
Return
= 0
10

```
w ELT GSTAB,1,700416, 60795

          C      PROGRAM 021504      GAS-GAS FEED SYSTEM COUPLED STABILITY PROGRAM
          C                                                 CODED BY    R.H. MARLER
          C                                                 CUSTOMER    R.C. WAUGH
          C
                 DIMENSION LL(22),MM(22),HH(22)
                 DIMENSION AA(22,22),BB(22),CC(22),DD(22),EE(22),FF(22),GG(22)
                 COMMON    DR(72)
                X          ,A(2),C(2),ALPHA(2),BETA(2)
                X          ,F(2),G(2),H(2)
                 EQUIVALENCE
                X     (GAMAX,DR( 1)), (GAMAF,DR( 2)), (WMX  ,DR( 3)), (WMF  ,DR( 4))
                X   , (R    ,DR( 5)), (GRVTY,DR( 6)), (CPX  ,DR( 7)), (CPF  ,DR( 8))
                X   , (HOH2O,DR( 9)), (CPH2O,DR(10)), (TR   ,DR(11)), (WMH2 ,DR(12))
                X   , (WMH2O,DR(13)), (AC   ,DR(14)), (TO   ,DR(15)), (TX   ,DR(16))
                X   , (TF   ,DR(17)), (GAMAP,DR(18)), (RM   ,DR(19)), (WP   ,DR(20))
                 EQUIVALENCE
                X     (PM   ,DR(21)), (WMP  ,DR(22)), (PO   ,DR(23)), (AX   ,DR(24))
                X   , (TP   ,DR(25)), (CO   ,DR(26)), (WX   ,DR(27)), (WF   ,DR(28))
                X   , (RX   ,DR(29)), (RF   ,DR(30)), (PP   ,DR(31)), (CP   ,DR(32))
                X   , (CPP  ,DR(33)), (VP   ,DR(34)), (PF   ,DR(35)), (CAY1 ,DR(36))
                X   , (CVP  ,DR(37)), (PX   ,DR(38)), (RHOP ,DR(39)), (HX   ,DR(40))
                 EQUIVALENCE
                X     (HF   ,DR(41)), (RHOF ,DR(42)), (RHOX ,DR(43)), (VX   ,DR(44))
                X   , (HP   ,DR(45)), (YH2  ,DR(46)), (YH2O ,DR(47)), (DENOM,DR(48))
                X   , (CPO2 ,DR(49)), (CPH2 ,DR(50)), (CX2  ,DR(51)), (RP   ,DR(52))
                X   , (A1   ,DR(53)), (B1   ,DR(54)), (C1   ,DR(55)), (D1   ,DR(56))
                X   , (E1   ,DR(57)), (A2   ,DR(58)), (B2   ,DR(59)), (C2   ,DR(60))
                 EQUIVALENCE
                X     (D2   ,DR(61)), (E2   ,DR(62)), (A3   ,DR(63)), (B3   ,DR(64))
                X   , (C3   ,DR(65)), (D3   ,DR(66)), (E3   ,DR(67)),(FREQ ,DR(68))
                X   , (WT   ,DR(69))                                                   *NEW
          C
                 I IN =     5
                 I OUT=     6
                 GAMAX=     1.4
                 GAMAF=     1.4
                 WMX  =    32.0
                 WMF  =     2.0
                 R    = 1545.0
                 GRVTY=    32.172
                 CPX  =  169.
                 CPF  = 2710.
                 HOH2O=    -4.498 E+06
                 HOH2 = 0.
                 CPH2O=  555.
                 TR   =  537.
                 WMH2 =     2.
                 WMH2O=    18.
          C
               1 DO 10 J=1,22
                 BB(J)=0.
                 CC(J)=0.
                 DD(J)=0.
                 EE(J)=0.
                 FF(J)=0.
```

```
      GG(J)=0.
      DO 10 I=1,22
   10 AA(I,J)=0.
      DO 20 I=14,62
   20 DR(1) = 0.
C
      READ (IIN,40)AC,TO,TX,TF,GAMAP,AX,RM,WP,PM,WMP,PO,WT,CHECK                 *NEW
   40 FORMAT(1X,4E15.6)                                                         **-1
      IF(CHECK)45,55,45
   45 WRITE(I OUT,50)
   50 FORMAT( 16H1 INITIAL VALUES /)
      WRITE(I OUT,40)(DR(I),I=1,20)
C
   55 AC = AC / 144.
      AX = AX / 144.
      PO = PO * 144.
      CON1 =(GAMAP-1.0  )*PM/2.0  *PM + 1.0
      IEQ= 1
      IF(CON1 )140,70,100
   70 WRITE(I OUT,80)IEQ
   80 FORMAT(21H0 DIVIDE CHECK IN EQ ,I2)
   90 WRITE(I OUT,40)(  DR(I),I=25,52)
   91 READ(IIN,92) FLAG
   92 FORMAT(61X,F2.0)
      IF(FLAG)93,91,1
   93 STOP
  100 TP = TO/CON1
                                                             IEQ =  2
      IF(WMP)120,70,120
  120 TCALC=GAMAP*R/WMP*GRVTY*TO
      IF(TCALC)140,170,160
  140 WRITE(I OUT,150)IEQ
  150 FORMAT(29 H0 SQRT OF -X ATTEMPTED IN EQ ,I2)
      GO TO 90
  160 CO =  SQRT(TCALC)
C
C
  170 WF = WP/(RM+1.0   )
      WX = WF* RM
      RG = R*GRVTY
      RX = RG/WMX
      CX2=GAMAX*RX*TX
      RF = RG/WMF
      RP = RG/ WMP
      PP = PO/CON1**(GAMAP/(GAMAP-1.     ))
      CP = CO /  SQRT(CON1)
      CPP= GAMAP/(GAMAP-1.0   ) * R / WMP
      VP = CP* PM
      TCALC = GRVTY * AC * AX
      IF(TCALC)190,180,190
  180                                                        IEQ = 12
      GO TO 70
  190 CAY1=WX**2 /TCALC*R / WMX *TX
C
      TCALC= PP**2-4.0  *CAY1
                                                             IEQ = 11
      IF(TCALC)140,210,200
  200 TCALC= SQRT(TCALC)
  210 PX=PO                                                                     *NEW
```

```
                                                                 IEQ = 13        **-1
      IF(GAMAP)220,70,220
  220 CVP=CPP/GAMAP
      PF = PX
                                                                 IEQ = 15
      IF(TP)230,70,230
  230 RHOP = PP/RG * WMP/ TP
      CPO2= CPX
      CPH2= CPF
      HX=(TX-TR) * CPO2
      HF=(TF-TR) * CPH2
                                                                 IEQ = 18
      IF(TF)240,70,240
  240 RHOF = PF / RG * 2.0  /TF
                                                                 IEQ = 19
      IF(TX)250, 70,250
  250 RHOX = PX /RG *32.0  / TX
      TCALC=RHOX*GRVTY*AX
                                                                 IEQ = 20
      IF(TCALC)260,70 ,260
  260 VX = WX / TCALC
                                                                 IEQ = 21
      IF(WP)270, 70,270
  270 HP=(HX*WX+WF*HF)/WP                                                        *NEW
                                                                 IEQ = 22        **-1
      TCALC = WF + WX
      IF(TCALC)280, 70,280
  280 YH2 = (WF - WX/8.  )/TCALC
      YH20= 1.   -YH2
C
      DENOM = YH20/WMH20 + YH2 / WMH2
                                                                 IEQ = 24
      IF(DENOM)290, 70,290
C
  290 IF(CHECK)300,320,300
  300 WRITE(I OUT,310)
  310 FORMAT( 21H0 COMPUTED PARAMETERS )
      WRITE(I OUT,40)(  DR(I),I=25,52)
  320 CC(1)=-1.0                                                                 *NEW
      CC(3)=+HX                                                                  *NEW
      CC(10)=-9./8./WP                                                           *NEW
      CC(15)=-1.0/GRVTY/AX/RHOX                                                  *NEW
      DD(1)=-1.0                                                                 *NEW
      DD(3)=HF                                                                   *NEW
      EE(1)=-1.0                                                                 *NEW
      EE(3)=HP                                                                   *NEW
      FF(2)=GRVTY*AC                                                             *NEW
      FF(17)=1.0                                                                 *NEW
      FF(18)=1./RHOP/GRVTY                                                       *NEW
      GG(18)=TP                                                                  *NEW
C                                                                                *-13
      IF(CHECK)330,360,330
  330 WRITE(I OUT,340)
  340 FORMAT( 53H0 COLUMN VECTORS  CC, DD, EE, FF AND GG, RESPECTFULLY )
  350 FORMAT(1H., 5E15.6)
      WRITE(I OUT,350)(CC(I),DD(I),EE(I),FF(I),GG(I),I=1,22)
C
  360 TCALC = RHOF*RHOP
                                                                 IEQ = 25
```

```
      IF(TCALC)370, 70,370
  370 AA( 1, 1)= 1.0
      AA( 2, 3)= 0.  -GRVTY * AC                                  **-1
      AA( 3, 1)= 0.  -HP
      AA( 3, 4)= WX                                               **-1
      AA( 3, 5)= WF
      AA( 3, 6)= WT-WP                                            *NEW
      AA( 4, 4)= 1.0                                              **-1
      AA( 4, 7)= 0.  -CPX
      AA( 5, 5)= 1.0
      AA( 5, 8)= 0.  -CPF
      AA( 6, 6)= 1.0
      AA( 6, 9)= 0.  -1.0
      AA( 6,10)= 0.  -CPP
      AA( 6,11)= TR-TP                                            *NEW
      AA( 7, 9)= 1.0                                              **-1
      AA( 7,12)= 0.  -HOH2
      AA( 7,13)= 0.  -HOH20
      AA( 8,11)= 1.0
      AA( 8,12)= 0.  -CPH2
      AA( 8,13)= 0.  -CPH20
      AA( 9,12)= 1.0
      AA( 9,13)= 1.0
      AA(10, 1)= WX / WP * 1.125  / WP
      AA(10,13)= 1.0
      AA(11, 3)= 1.0
      AA(11, 7)= 0.  -RX * RHOX
      AA(11,14)= 0.  -RX * TX
      AA(12, 8)= 0.  -RF * RHOF
      AA(12,15)= 1.0
      AA(12,16)= 0.  -RF * TF
      AA(13, 3)= 1.0
      AA(13,14)= 0.  -GAMAX / RHOX * PX
      AA(14,15)= 1.0
      AA(14,16)= 0.  -GAMAF / RHOF * PF
      AA(15, 2)= 1.0
      AA(15,14)= VX / RHOX
      AA(16, 1)=      -1.0/ GRVTY /AC / RHOP
      AA(16,17)= 1.0
      AA(16,18)= VP / RHOP
      AA(17,10)= 0.  -RP * RHOP
      AA(17,18)= 0.  -RP * TP
      AA(17,19)= 0.  -RHOP * TP
      AA(18, 6)=-1.0                                              *NEW
      AA(19,19)= 1.0                                              **-2
      AA(19,20)= RG / WMP**2
      AA(20,20)= 1.0
      AA(20,21)= 0.  -WMH2
      AA(20,22)= 0.  -WMH20
      TEM1 = 1./WMH2 /DENOM
      AA(21,12)= (TEM1*YH2 -1.  )*TEM1
      AA(21,13)= TEM1/DENOM / WMH20 * YH2
      AA(21,21)= 1.0
      TEM1 = 1./WMH20/DENOM
      AA(22,13)= (TEM1 * YH20 -1.  )* TEM1
      AA(22,12)= TEM1/DENOM / WMH2 * YH20
      AA(22,22)= 1.0
C
      IF(CHECK)380,420,380
```

```
  380 WRITE(IOUT, 390)
  390 FORMAT( 28H0 MATRIX AA BEFORE INVERSION)
  400 FORMAT(1H 12/(3H   5E15.6))
      DO 410 I=1,22
  410 WRITE(I OUT,400)I,(AA(I,J),J=1,22)
C
  420 CALL MINV(AA,22,DET,LL,MM)
      IF(CHECK)430,450,430
  430 WRITE(IOUT,480)
      DO 440 I=1,22
  440 WRITE(IOUT,400)I,(AA(I,J),J=1,22)
  450 IF(DET)500,460,500
  460 WRITE(I OUT,470)
  470 FORMAT( 23H0 MATRIX AA IS SINGULAR  )
  480 FORMAT( 27H0 MATRIX AA AFTER INVERSION )
      GO TO 91
  500 CALL MVMLT(AA,BB,HH,22,22)
      CALL MVMLT(AA,CC,HH,22,22)
      CALL MVMLT(AA,DD,HH,22,22)
      CALL MVMLT(AA,EE,HH,22,22)
      CALL MVMLT(AA,FF,HH,22,22)
      CALL MVMLT(AA,GG,HH,22,22)
      IF(CHECK)510,540,520
  510 CALL MINV(AA,22,DET,LL,MM)
      WRITE(IOUT,512) DET
  512 FORMAT('1MATRIX AA RE-INVERTED,   DET='E13.5//)
      DO 514 I=1,22
  514 WRITE(IOUT,400)I,(AA(I,J),J=1,22)
  520 WRITE(IOUT,530)(BB(I),CC(I),DD(I),EE(I),FF(I),GG(I),I=1,22)
  530 FORMAT('1H0VECTORS BB,CC,DD,EE,FF,GG,AFTER MULTIPLICATION'/
     X      (1H 6E15.6))
  540 CONTINUE
      A1=-CC(3)                                                           *NEW
      B1=-DD(3)                                                           *NEW
      C1=-EE(3)                                                           *NEW
      D1=-FF(3)                                                           *NEW
      E1=-GG(3)                                                           *NEW
      A2=-CC(17)                                                          *NEW
      B2=-DD(17)                                                          *NEW
      C2=-EE(17)                                                          *NEW
      D2=-FF(17)                                                          *NEW
      E2=-GG(17)                                                          *NEW
      A3=-CC(15)                                                          *NEW
      B3=-DD(15)                                                          *NEW
      C3=-EE(15)                                                          *NEW
      D3=-FF(15)                                                          *NEW
      E3=-GG(15)                                                          *NEW
      CALL CMPLX(IIN,IOUT,CHECK)                                          *-15
      GO TO 1
      END
```

```
W ELT CMPLX,1,700724, 29535

      SUBROUTINE CMPLX(IIN,IOUT,CHECK)
      DIMENSION  CREG1(2),CREG2(2),CREG3(2),CREG4(2)
      DIMENSION  A4(2),B4(2),C4(2),D4(2),E4(2)
      COMMON   DR(72)
     X        ,A(2),C(2),ALPHA(2),BETA(2)
     X        ,F(2),G(2),H(2)
      EQUIVALENCE
     X    (GAMAX,DR( 1)), (GAMAF,DR( 2)), (WMX  ,DR( 3)), (WMF  ,DR( 4))
     X  , (R    ,DR( 5)), (GRVTY,DR( 6)), (CPX  ,DR( 7)), (CPF  ,DR( 8))
     X  , (HOH20,DR( 9)), (CPH20,DR(10)), (TR   ,DR(11)), (WMH2 ,DR(12))
     X  , (WMH20,DR(13)), (AC   ,DR(14)), (TO   ,DR(15)), (TX   ,DR(16))
     X  , (TF   ,DR(17)), (GAMAP,DR(18)), (RM   ,DR(19)), (WP   ,DR(20))
      EQUIVALENCE
     X    (PM   ,DR(21)), (WMP  ,DR(22)), (PO   ,DR(23)), (AX   ,DR(24))
     X  , (TP   ,DR(25)), (CO   ,DR(26)), (WX   ,DR(27)), (WF   ,DR(28))
     X  , (RX   ,DR(29)), (RF   ,DR(30)), (PP   ,DR(31)), (CP   ,DR(32))
     X  , (CPP  ,DR(33)), (VP   ,DR(34)), (PF   ,DR(35)), (CAY1 ,DR(36))
     X  , (CVP  ,DR(37)), (PX   ,DR(38)), (RHOP ,DR(39)), (HX   ,DR(40))
      EQUIVALENCE
     X    (HF   ,DR(41)), (RHOF ,DR(42)), (RHOX ,DR(43)), (VX   ,DR(44))
     X  , (HP   ,DR(45)), (YH2  ,DR(46)), (YH20 ,DR(47)), (DENOM,DR(48))
     X  , (CPO2 ,DR(49)), (CPH2 ,DR(50)), (CX2  ,DR(51)), (RP   ,DR(52))
     X  , (A1   ,DR(53)), (B1   ,DR(54)), (C1   ,DR(55)), (D1   ,DR(56))
     X  , (E1   ,DR(57)), (A2   ,DR(58)), (B2   ,DR(59)), (C2   ,DR(60))
      EQUIVALENCE
     X    (D2   ,DR(61)), (E2   ,DR(62)), (A3   ,DR(63)), (B3   ,DR(64))
     X  , (C3   ,DR(65)), (D3   ,DR(66)), (E3   ,DR(67)),(FREQ ,DR(68))
     X  , (WT   ,DR(69))
   10 READ(IIN,20)A,C,FLAG,FREQ
   20 FORMAT(1X,4E15.6,F2.0,F9.2)
      IF(CHECK)25,35,25
   25 WRITE(IOUT,30)
   30 FORMAT(34H1INITIAL DATA,  COMPLEX EQUATIONS )
      WRITE(IOUT,20)A,C,FLAG,FREQ
  35                                                            IEQ = 26
      IF(PO) 80, 40,80
   40 WRITE(IOUT,50)IEQ
   50 FORMAT( 20H0DIVIDE CHECK IN EQ ,I3)
      WRITE(I OUT,70)(DR(I),I=53,68),ALPHA,BETA
   70 FORMAT(1H ,4E15.6)
      GO TO 160
   80 TCALC = CO/PO*GAMAP                                                         *NEW
      CALL CCMLT(TCALC,A,ALPHA)                                                   **-1
                                                                IEQ = 27
      IF(CPP)90,40,90
   90 TCALC = CO/CPP*GAMAP
      CALL CCMLT(TCALC,C,BETA)
                                                                IEQ = 28
      IF(E3)92,40,92
   92 E4(1) =(BETA(1)/GAMAP + E2) / E3
      E4(2) = BETA(2)/GAMAP / E3
                                                                IEQ = 29
      CREG2(1)=ALPHA(1)/GAMAP + D2
      CREG2(2)=ALPHA(2)/GAMAP
      CALL CCMLT(D3, E4  ,CREG3)
```

```
      CALL CSUB (CREG2,CREG3,D4)
                                                        IEQ = 30
      CALL CCMLT(C3, E4  ,CREG3)
      C4(1) = C2 - CREG3(1)
      C4(2) =    - CREG3(2)
                                                        IEQ = 31
      CALL CCMLT(B3, E4  ,CREG3)
      B4(1) = B2 - CREG3(1)
      B4(2) =    - CREG3(2)
                                                        IEQ = 32
      CALL CCMLT(A3, E4  ,CREG3)
      A4(1) = A2 - CREG3(1)
      A4(2) =    - CREG3(2)
                                                        IEQ = 33
      A5= A1-E1/E3*A3
      B5= B1-E1/E3*B3
      C5= C1-E1/E3*C3
      D5= D1-E1/E3*D3
      E5= E1/E3-1.0
      IF(D5)105,40,105
  105 TCALC = 1.0/ D5
      CALL CCMLT(TCALC,D4,CREG4)
      CALL CCMLT(A5,CREG4,CREG1)
      CALL CCMLT(B5,CREG4,CREG2)
      CALL CCMLT(C5,CREG4,CREG3)
      CALL CCMLT(E5,CREG4,CREG4)
      CALL CSUB(A4,CREG1,CREG1)
      CALL CSUB(B4,CREG2,CREG2)
      CALL CSUB(C4,CREG3,CREG3)
      CALL CSUB(E4,CREG4,CREG4)
      CALL CCMLT(144.00,CREG4,CREG4)
      CALL CDVID(CREG1,CREG4,F,IERR)
      IF(IERR)40,110,40
  110 CALL CDVID(CREG2,CREG4,G,IERR)
      CALL CDVID(CREG3,CREG4,H,IERR)
      IF(CHECK)120,150,120
  120 WRITE(IOUT,130)
  130 FORMAT('0COMPUTED VALUES OF A4,B4,C4,D4,CREG1,CREG2,CREG3,CREG4, (
     CDR(I),I=5,68),A5,B5,C5,D5,E5 ' )
      WRITE(IOUT,140)A4,B4,C4,D4,E4,CREG1,CREG2,CREG3,CREG4,(DR(I),I=5,
     X    67),A5,B5,C5,D5,E5
  140 FORMAT(1H E16.6,E14.6,E16.6,E14.6)
  150 CALL OUTPT(IOUT)
  160 IF(FLAG)180, 10,170
  170 RETURN
  180 STOP
      END
```

```
w ELT OUTPT,1,700416, 60799

      SUBROUTINE OUTPT(IOUT)
      COMMON   DR(72)
     X         ,A(2),C(2),ALPHA(2),BETA(2)
     X         ,F(2),G(2),H(2)
      EQUIVALENCE
     X    (GAMAX,DR( 1)), (GAMAF,DR( 2)), (WMX  ,DR( 3)), (WMF  ,DR( 4))
     X  , (R    ,DR( 5)), (GRVTY,DR( 6)), (CPX  ,DR( 7)), (CPF  ,DR( 8))
     X  , (HOH2O,DR( 9)), (CPH2O,DR(10)), (TR   ,DR(11)), (WMH2 ,DR(12))
     X  , (WMH2O,DR(13)), (AC   ,DR(14)), (TO   ,DR(15)), (TX   ,DR(16))
     X  , (TF   ,DR(17)), (GAMAP,DR(18)), (RM   ,DR(19)), (WP   ,DR(20))
      EQUIVALENCE
     X    (PM   ,DR(21)), (WMP  ,DR(22)), (PO   ,DR(23)), (AX   ,DR(24))
     X  , (TP   ,DR(25)), (CO   ,DR(26)), (WX   ,DR(27)), (WF   ,DR(28))
     X  , (RX   ,DR(29)), (RF   ,DR(30)), (PP   ,DR(31)), (CP   ,DR(32))
     X  , (CPP  ,DR(33)), (VP   ,DR(34)), (PF   ,DR(35)), (CAY1 ,DR(36))
     X  , (CVP  ,DR(37)), (PX   ,DR(38)), (RHOP ,DR(39)), (HX   ,DR(40))
      EQUIVALENCE
     X    (HF   ,DR(41)), (RHOF ,DR(42)), (RHOX ,DR(43)), (VX   ,DR(44))
     X  , (HP   ,DR(45)), (YH2  ,DR(46)), (YH2O ,DR(47)), (DENOM,DR(48))
     X  , (CPO2 ,DR(49)), (CPH2 ,DR(50)), (CX2  ,DR(51)), (RP   ,DR(52))
     X  , (A1   ,DR(53)), (B1   ,DR(54)), (C1   ,DR(55)), (D1   ,DR(56))
     X  , (E1   ,DR(57)), (A2   ,DR(58)), (B2   ,DR(59)), (C2   ,DR(60))
      EQUIVALENCE
     X    (D2   ,DR(61)), (E2   ,DR(62)), (A3   ,DR(63)), (B3   ,DR(64))
     X  , (C3   ,DR(65)), (D3   ,DR(66)), (E3   ,DR(67)),(FREQ ,DR(68))
     X  , (WT   ,DR(69))                                                    *NEW
   10 FORMAT(1H1//////,47X  27HAEROJET-GENERAL CORPORATION
     X       /1H ,46X  27HCOMPUTING SCIENCES DIVISION
     Y       /1H ,48X  23HSACRAMENTO,  CALIFORNIA
     W       /1H0,52X,15HPROGRAM  E21504
     Z       /1H ,37X  45HGAS-GAS FEED SYSTEM COUPLED STABILITY PROGRAM )
   20 FORMAT(//1H0,49X21HREAL INPUT CONDITIONS
     X         /1H0 18X2HAC14X2HTO14X2HTF11X7HGAMMA P12X2HAX14X2HMR)
   30 FORMAT(13XE13.6,5E16.6
     X       /1H0,34X2HWP14X2HMP12X4HMW P14X2HPO14X2HWT                    *NEW
     Y       /28XE13.6,4E16.6  )                                            *NEW
   40 FORMAT(//1H0,41X38HCOMPLEX NOZZLE ADMITTANCE COEFFICIENTS             **-2
     X         /1H0,34X4HREAL11X4HIMAG14X4HREAL11X4HIMAG
     Y         /     40X8HSCRIPT A25X8HSCRIPT C
     Z         /     30XE13.6,E15.6,E18.6,E15.6
     W         /1H0 40X5HALPHA29X4HBETA
     Z         /     30XE13.6,E15.6,E18.6,E15.6      )
   50 FORMAT(//1H0 41X36HCOMPUTED COMPLEX COEFFICIENTS IN THE
     X         /1H  48X23HCHARACTERISTIC EQUATION
     Y  /1H039X41H(F)* WX' + (G)* WF' + (H)* WT' + PX' = 0. /
     Z     1H0 45X 19HFOR A FREQUENCY OF ,F10.3/)
   60 FORMAT(//20X 8H( F ) = E14.7,3H + E14.7,6H * I ,3HOR E14.7,18H TIM   *NEW
     VES E TO THE I E14.7/                                                  *NEW
     W            20X 8H( G ) = E14.7,3H + E14.7,6H * I ,3HOR E14.7,18H TIM *NEW
     XES E TO THE I E14.7/                                                  *NEW
     Y            20X 8H( H ) = E14.7,3H + E14.7,6H * I ,3HOR E14.7,18H TIM *NEW
     ZES E TO THE I E14.7 )                                                 *NEW
C                                                                           **-3
C
      WRITE(IOUT,10)
```

```
      WRITE(IOUT,20)
      TPO = PO / 144.
      TAC = AC * 144.
      TAX = AX * 144.
      FA = SQRT(F(1)**2+F(2)**2)                                           *NEW
      FP = ATAN(F(2)/F(1))                                                 *NEW
      IF (F(1)) 101,102,102                                                *NEW
  101 FP = FP+3.1415927                                                    *NEW
  102 CONTINUE                                                             *NEW
      GA = SQRT(G(1)**2+G(2)**2)                                           *NEW
      GP = ATAN(G(2)/G(1))                                                 *NEW
      IF (G(1)) 201,202,202                                                *NEW
  201 GP = GP+3.1415927                                                    *NEW
  202 CONTINUE                                                             *NEW
      HA = SQRT(H(1)**2+H(2)**2)                                           *NEW
      HP = ATAN(H(2)/H(1))                                                 *NEW
      IF (H(1)) 301,302,302                                                *NEW
  301 HP = HP+3.1415927                                                    *NEW
  302 CONTINUE                                                             *NEW
      WRITE(IOUT,30)TAC,TO,TF,GAMAP,TAX,RM,WP,PM,WMP,TPO,WT                *NEW
      WRITE(IOUT,40)A,C,ALPHA,BETA                                         **-1
      WRITE(IOUT,50) FREQ
      WRITE(IOUT,60)F,FA,FP,G,GA,GP,H,HA,HP                                *NEW
      RETURN                                                               **-1
      END
```

```
@ ELT MINV,1,700724, 29525

      SUBROUTINE MINV(A,N,D,L,M)
      DIMENSION A(1),L(1),M(1)
C        SEARCH FOR LARGEST ELEMENT
      D=1.0
      NK=-N
      DO 80 K=1,N
      NK=NK+N
      L(K)=K
      M(K)=K
      KK=NK+K
      BIGA=A(KK)
      DO 20 J=K,N
      IZ=N*(J-1)
      DO 20 I=K,N
      IJ=IZ+I
   10 IF( ABS(BIGA)- ABS(A(IJ))) 15,20,20
   15 BIGA=A(IJ)
      L(K)=I
      M(K)=J
   20 CONTINUE
C        INTERCHANGE ROWS
      J=L(K)
      IF(J-K) 35,35,25
   25 KI=K-N
      DO 30 I=1,N
      KI=KI+N
      HOLD=-A(KI)
      JI=KI-K+J
      A(KI)=A(JI)
   30 A(JI) =HOLD
C        INTERCHANGE COLUMNS
   35 I=M(K)
      IF(I-K) 45,45,38
   38 JP=N*(I-1)
      DO 40 J=1,N
      JK=NK+J
      JI=JP+J
      HOLD=-A(JK)
      A(JK)=A(JI)
   40 A(JI) =HOLD
C        DIVIDE COLUMN BY MINUS PIVOT (VALUE OF PIVOT ELEMENT IS
C        CONTAINED IN BIGA)
   45 IF(BIGA) 46,46,48
   46 D=0.0
      RETURN
   48 DO 55 I=1,N
      IF(I-K) 50,55,50
   50 IK=NK+I
      IF(A(IK))52,55,52
   52 A(IK)=A(IK)/(-BIGA)
   55 CONTINUE
C        REDUCE MATRIX
      DO 65 I=1,N
      IK=NK+I
      IJ=I-N
```

```
      DO 65 J=1,N
      IJ=IJ+N
      IF(I-K) 60,65,60
   60 IF(J-K) 62,65,62
   62 KJ=IJ-I+K
      IF(A(IK))63,65,63
   63 IF(A(KJ))64,65,64
   64 A(IJ)=A(IK)*A(KJ)+A(IJ)
   65 CONTINUE
C        DIVIDE ROW BY PIVOT
      KJ=K-N
      DO 75 J=1,N
      KJ=KJ+N
      IF(J-K) 70,75,70
   70 IF(A(KJ))72,75,72
   72 A(KJ)=A(KJ)/BIGA
   75 CONTINUE
C        PRODUCT OF PIVOTS
      D=D*BIGA
C        REPLACE PIVOT BY RECIPROCAL
      A(KK)=1.0/BIGA
   80 CONTINUE
C        FINAL ROW AND COLUMN INTERCHANGE
      K=N
  100 K=(K-1)
      IF(K) 150,150,105
  105 I=L(K)
      IF(I-K) 120,120,108
  108 JQ=N*(K-1)
      JR=N*(I-1)
      DO 110 J=1,N
      JK=JQ+J
      HOLD=A(JK)
      JI=JR+J
      A(JK)=-A(JI)
  110 A(JI) =HOLD
  120 J=M(K)
      IF(J-K) 100,100,125
  125 KI=K-N
      DO 130 I=1,N
      KI=KI+N
      HOLD=A(KI)
      JI=KI-K+J
      A(KI)=-A(JI)
  130 A(JI) =HOLD
      GO TO 100
  150 RETURN
      END
```

```
@ ELT MVMLT,1,700724, 29526

      SUBROUTINE MVMLT(A,B,C,N,M)
C  SPARSE MATRIX * VECTOR MULTIPLY ROUTINE--MOVES B INTO C, THEN B=A*C
C    N = NO. ROWS IN A
C    M = NO. COLS IN A = NO. ELEMENTS IN B
C    A = N*M MATRIX STORED SEQUENTIALLY BY COLUMN
C    B = VECTOR OR COLUMN MATRIX OF LENGTH M
C    C = RESULTANT PRODUCT VECTOR OF LENGTH M (MUST BE DIFFERENT FROM B)
C         MAP OF MATRIX A STORGE --
C
C             1   N+1  2N+1   .   .  (M-1)N+1
C             2   N+2  2N+2   .   .  (M-1)N+2
C             3   N+3  2N+3   .   .  (M-1)N+2
C             .    .     .    .   .      .
C             .    .     .    .   .      .
C             N    2N    3N   .   .     M*N
C
      DIMENSION A(1),B(1),C(1)
      DO 5 J=1,M
      C(J)=B(J)
    5 B(J)=0.
      DO 30 I=1,N
      DO 30 J=1,M
      IF(C(J))10,30,10
   10 IJ=(J-1)*N+I
      IF(A(IJ))20,30,20
   20 B(I)=B(I)+C(J)*A(IJ)
   30 CONTINUE
      RETURN
      END
```

```
@ ELT CADD,1,700724, 29528

      SUBROUTINE CADD(A,B,C)
COMPLEX ADDITION  C = A + B  WHERE EACH VARIABLE IS A 2-CELL
C      I.E.,    CONSIDER THE COMPLEX NUMBER   X = U+V*I.
C               THEN    X(1)=U    AND  X(2)=V.
C
      DIMENSION A(2),B(2),C(2)
      C(1) = A(1) + B(1)
      C(2) = A(2) + B(2)
      RETURN
      END
```

```
@ ELT CSUB,1,700724, 29529

      SUBROUTINE CSUB(A,B,C)
C COMPLEX SUBTRACTION  C=A-B
C
      DIMENSION A(2),B(2),C(2)
      C(1) = A(1) - B(1)
      C(2) = A(2) - B(2)
      RETURN
      END
```

```
@ ELT CMULT,1,700724, 29530

      SUBROUTINE CMULT(A,B,C)
COMPLEX MULTIPLICATION  C = A * B
C
      DIMENSION A(2),B(2),C(2)
      CR   = A(1)*B(1) - A(2)*B(2)
      CI = A(1)*B(2) + A(2)*B(1)
      C(1)=CR
      C(2)=CI
      RETURN
      END
```

```
w ELT CCMLT,1,700304, 44516

      SUBROUTINE CCMLT(C,X,Y)
COMPLEX SCALAR MULTIPLICATION  Y = C*X
C     wHERE    X,Y ARE COMPLEX,
C                C IS A REAL SCALAR
      DIMENSION X(2), Y(2)
      Y(1) = X(1) * C
      Y(2) = X(2) * C
      RETURN
      END
```

```
@ ELT CDVID,1,700304, 44517

                SUBROUTINE CDVID(A,B,C,IERR)
          COMPLEX DIVISION C = A / D
          C     IERR = 0  IF DIVISION SUCCESSFUL,  IERR=-1 IF DIVIDE CHECK OCCURRS
                DIMENSION  A(2),B(2),C(2)
                DEN = B(1)**2 + B(2)**2
                IF(DEN )30,20,30
             20 IERR=-1
                GO TO 40
             30 CR   = (A(1)*B(1) +  A(2)*B(2)) / DEN
                CI   = (A(2)*B(1) -  A(1)*B(2)) / DEN
                C(1)=CR
                C(2)=CI
                IERR = 0
             40 RETURN
                END
```

E. METHOD OF VERIFICATION

The results of this program at zero frequency should be reasonably close to the simple quasi-steady analysis using C* and the change in C* with mixture ratio (Ref. 7). This was the guide used in checking out the program. There is no other known reference to measure the validity of these results.

SECTION III - DECK SETUP

A. COMPUTER CONFIGURATION

1. This program should be able to run on any of at least 3 computers: IBM 360-65, IBM 1130, UNIVAC 1108. Only two variables must be changed to change from one computer to another, IIN and IOUT. These variables are defined in the first part of the main program and refer to the UNIT numbers in READ and WRITE statements.

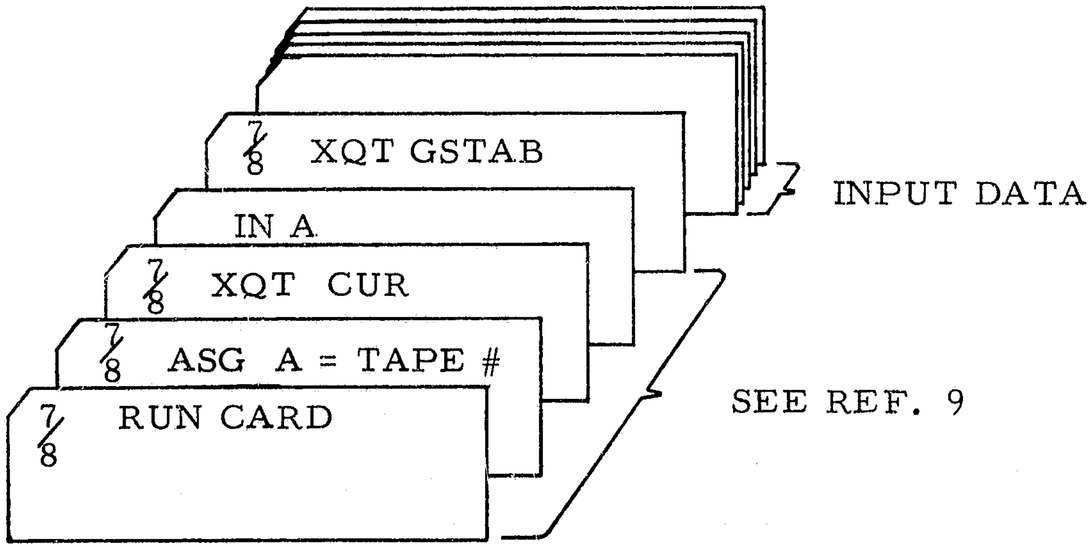
2. FORTRAN language is used in this program.

B. RUN TIME

Run time for one frequency including compilations is 13 sec on the UNIVAC 1108.

C. DECK SEQUENCE

The deck sequence shown below include the control cards as required by the EXEC II Monitor system for the UNIVAC 1108 as described in Ref. 9. For use with other computers the instruction manual for the particular computer system used should be consulted.



To run multiple cases see FLAG under INPUT DATA.

D. INPUT DATA

The following is a list with definitions and units of all input variables including special flags with an explanation of their purpose:

AC	Cross-sectional area of the chamber in square inches.
TO	Stagnation temperature of the burned exhaust gases in degrees Rankine. This comes from thermochemical calculations.
TX	Temperature of gaseous oxygen entering the chamber in degrees Rankine.
TF	Temperature of gaseous hydrogen entering the chamber in degrees Rankine
GAMAP	Ratio of specific heats for the burned exhaust gases. This would be obtained from a thermochemical calculation. It is primarily a function of mixture ratio.
AX	Total oxidizer injection area in square inches.
RM	Mixture Ratio
WP	Total weight flow of exhaust gases, pounds per second.
PM	Mach number of the chamber. This can be calculated from the chamber contraction ratio and GAMAP.
WMP	Molecular weight of the burned exhaust gases. This also comes from thermochemical calculations.
PO	Stagnation pressure in the chamber in pounds per square inch.
WT	Weight flow in tapoff line in pounds per second (normally negative).
CHECK	This is a flag which controls the output of the program. If CHECK is less than zero all input variables and variables needed to calculate the coefficients of the matrix are listed. The coefficients of the matrices, and the inverse are listed. The inverted matrix is then reinverted to check the accuracy of the inversion. This result is also listed. If CHECK is greater than zero all of the above is done except the reinversion is not performed.

	If CHECK is zero only the input variables and final results are listed. CHECK is normally zero.
FLAG	If FLAG is negative only one case is run If FLAG is zero a new frequency with the same chamber is run If FLAG is positive a complete new case is run
$\mathcal{A}$, $\mathcal{C}$, FREQ.	These are longitudinal admittance coefficients and their corresponding frequency. $\mathcal{A}$ and $\mathcal{C}$ are complex. They can be obtained from the computer program described in Reference 8.

E. RESTRICTIONS AND LIMITATIONS

1. Low Mach number in chamber and feed lines.
2. Fuel rich (mixture ratio less than 8).
3. Gaseous hydrogen and gaseous oxygen are used as propellants.
4. Perturbations are assumed to be small.
5. No pressure sensitive combustion is included.

F. DIAGNOSTICS

Any division by zero will cause the program to print DIVIDE CHECK IN EQ, IEQ where IEQ is an equation number in the program. If a square root operation is attempted on a negative number a message is printed SQRT OF -X ATTEMPTED IN* EQ, IEQ. If the matrix is singular a message is printed MATRIX AA IS SINGULAR.

G. QUANTITY OF OUTPUT

If CHECK is zero one page of output per frequency is obtained. If CHECK is not zero 20 to 30 pages of output per frequency are obtained.

H. OUTPUT DEFINITION

LABEL	SYMBOL	UNITS	DESCRIPTION
AC	A_c	in^2	Chamber area
ALPHA	α	--	Nozzle admittance coefficient
AX	A_x	in^2	Oxidizer injection area
BETA	β	--	Nozzle admittance coefficient
F	F	sec/in^2	Coefficient of characteristic equation
G	G	sec/in^2	Coefficient of characteristic equation
GAMMA P	γ_p	--	ratio of specific heats of products of combustion
H	H	sec/in^2	Coefficient of characteristic equation
I	i	--	$\sqrt{-1}$
MP	M_p	--	Mach number of products of combustion
MR	MR	--	Mixture ratio
MWP	MW_p	--	Molecular weight of products of combustion
PO	P_o	lb/in^2	stagnation pressure of products of combustion
SCRIPT A	$\mathcal{A}$	--	nozzle admittance coefficient
SCRIPT C	$\mathcal{C}$	--	nozzle admittance coefficient
TF	T_F	^oR	temperature of fuel
TO	T_o	^oR	stagnation temperature of products of combustion
WT	$\dot{w}_t$	lb/sec	weight flow rate in tapoff line
WP	$\dot{w}_p$	lb/sec	weight flow rate of products of combustion

I. INPUT FORM

The format of the input data cards is shown on Page 47 and a listing of the input data cards for the sample case included herein is given on Page 48.

JOB NUMBER	REPORT TITLE	DATE	PAGE OF

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80

AC	TO	TX	TF
GAMAP	AX	RM	WP
PM	WMP	PO	WT
CHECK			

NOZZLE ADMITTANCE INPUT FLAG

A REAL	A IMAG	C REAL	C IMAG		FREQ

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80

AGCS 0140-106

```
177.0          6800.          530.           530.
1.2            29.0           5.0            290.0
.274           11.74          1000.          -5.0

-.026049       -.083588       -.12841        .048249        177.89
-.042862       -.24008        -.069302       .11453         553.67
```

J. SAMPLE OUTPUT

The following pages consist of the output of a typical case for the tapoff cycle. The input data for this case is that given in the previous section.

```
                              AEROJET-GENERAL CORPORATION
                              COMPUTING SCIENCES DIVISION
                                SACRAMENTO,  CALIFORNIA

                                    PROGRAM  E21504
                         GAS-GAS FEED SYSTEM COUPLED STABILITY PROGRAM


                                 REAL INPUT CONDITIONS

     AC              TO              TF            GAMMA P            AX              MR
 .177000+03      .680000+04      .530000+03      .120000+01      .290000+02      .500000+01

                     WP              MP             MW P              PO              WT
                 .290000+03      .274000+00      .117400+02      .100000+04     -.500000+01


                            COMPLEX NOZZLE ADMITTANCE COEFFICIENTS

                   REAL           IMAG                 REAL           IMAG
                        SCRIPT A                            SCRIPT C
                 -.260490-01    -.835880-01         -.128410+00     .482490-01

                          ALPHA                               BETA
                 -.127592-02    -.409427-02         -.114705+01     .430995+00


                            COMPUTED COMPLEX COEFFICIENTS IN THE
                                  CHARACTERISTIC EQUATION

                         (F)* WX' + (G)* WF' + (H)* WT' + PX' = 0.

                              FOR A FREQUENCY OF     177.890


  ( F ) =    -.1657541+01 +     .5268061+00 * I ,OR     .1739243+01 TIMES E TO THE I     .2833865+01
  ( G ) =    -.8678599+01 +     .2761513+01 * I ,OR     .9107361+01 TIMES E TO THE I     .2833525+01
  ( H ) =    -.8683373+01 +     .2761445+01 * I ,OR     .9111890+01 TIMES E TO THE I     .2833691+01
```

```
                         AEROJET-GENERAL CORPORATION
                         COMPUTING SCIENCES DIVISION
                           SACRAMENTO,  CALIFORNIA

                               PROGRAM  E21504
                     GAS-GAS FEED SYSTEM COUPLED STABILITY PROGRAM


                              REAL INPUT CONDITIONS
     AC              TO              TF           GAMMA P           AX              MR
 .177000+03      .680000+04      .530000+03      .120000+01      .290000+02      .500000+01

                     WP              MP             MW P             PO              WT
                 .290000+03      .274000+00      .117400+02      .100000+04     -.500000+01


                      COMPLEX NOZZLE ADMITTANCE COEFFICIENTS

                 REAL            IMAG                 REAL            IMAG
                       SCRIPT A                             SCRIPT C
              -.428620-01    -.240080+00          -.693020-01     .114530+00

                          ALPHA                                BETA
              -.209945-02    -.117595-01          -.619056+00     .102306+01


                      COMPUTED COMPLEX COEFFICIENTS IN THE
                            CHARACTERISTIC EQUATION

                  (F)* WX' + (G)* WF' + (H)* WT' + PX' = 0.

                         FOR A FREQUENCY OF    553.670


  ( F ) =   -.9958138+00 +    .8208317+00 * I ,OR    .1290508+01 TIMES E TO THE I    .2452220+01
  ( G ) =   -.5208834+01 +    .4300611+01 * I ,OR    .6754791+01 TIMES E TO THE I    .2451413+01
  ( H ) =   -.5214180+01 +    .4301570+01 * I ,OR    .6759525+01 TIMES E TO THE I    .2451807+01
```

Report 20672-P3D

N71-20418

Appendix B

ANALYSIS OF THE LONGITUDINAL HIGH FREQUENCY STABILITY OF A STAGED COMBUSTION CYCLE

TABLE OF CONTENTS

Page

I PROBLEM

II PROGRAMMING

III DECK SETUP

SECTION I -- PROBLEM

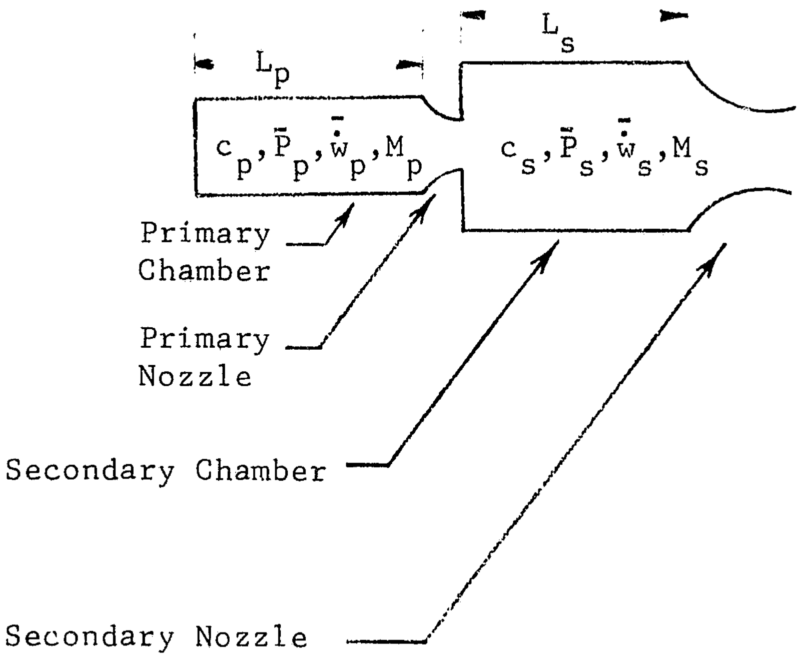
A ABSTRACT

This analysis uses the sensitive time lag theory in both combustors of a staged combustion cycle. It considers only longitudinal modes and no feed system effects are included. The primary and secondary combustors are coupled by including in the secondary analysis the flow oscillation generated in the primary combustor. As in other analyses using the sensitive time lag theory an n and τ curve for neutral stability is obtained (Ref. 1). Since a staged combustion cycle has two combustors there are two sets of n's and τ's, one set for the primary combustor and one set for the secondary combustor. In this analysis the n and τ for the primary are assumed known and and the analysis solves for the n and τ of the secondary.

The only input parameters needed by the program which are somewhat obscure are the n and τ for the primary combustor and the specific acoustic admittance of the sonic secondary exhaust nozzle, α_s. The first of these, n and τ, must come from experiments on primary combustors fired alone. What limited data that exists is published in Reference 2. The second, the specific acoustic admittance of a sonic nozzle, can be obtained from a computer program described in Reference 1.

B. TECHNICAL DESCRIPTION

There are many possible variations in configuration of staged combustion cycles. The one analyzed in this program is shown on Page 2. Figure 1 also includes dimensions and variables used in the input to the program. The purpose of the primary combustor is to generate gases to drive the turbine which in turn drives the propellant feed pumps. In this analysis the turbine is simulated by a pressure drop which is assumed to be proportional to the square of the flow rate through the turbine. No effect of turbine speed change or vibration is included. The distance between the turbine inlet and the point where the primary gases are injected into the secondary chamber is assumed to be negligible. The flow coming from the turbine that is injected



SCHEMATIC OF STAGED COMBUSTION CYCLE

Figure 1

B, Technical Description (cont.)

into the secondary combustor is assumed to have a time delay τ_{fs} between injection and combustion. This time delay is a mean total time delay as distinguished from the sensitive time lags τ_s and τ_p. The chambers and nozzles are analyzed using one-dimensional acoustics with a finite but small Mach number.

Both chambers are analyzed acoustically the same way so the following development applies to both the primary and secondary combustors. The following equations are used in developing the model:

Conservation of mass.

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial Z} (\rho u) = 0 \qquad \text{Eq. 1}$$

Conservation of momentum.

$$\rho \frac{\partial u}{\partial t} + \rho u \frac{\partial u}{\partial Z} + \frac{\partial p}{\partial Z} = 0 \qquad \text{Eq. 2}$$

Conservation of energy.

$$\frac{\partial}{\partial t} (\rho \; e_o) + \frac{\partial}{\partial Z} (\rho u \; h_o) = 0 \qquad \text{Eq. 3}$$

where,

$$e_o = c_v T + \frac{u^2}{2} \qquad \text{Eq. 4}$$

$$h_o = c_\rho T + \frac{u^2}{2} \qquad \text{Eq. 5}$$

$$c_v = \frac{RT}{\gamma - 1} \qquad \text{Eq. 6}$$

$$c_\rho = \frac{\gamma RT}{\gamma - 1} \qquad \text{Eq. 7}$$

B, Technical Description (cont.)

Ideal gas law.

$$p = \rho R T \qquad \text{Eq. 8}$$

The energy equation is simplified if the kinetic energy terms are neglected. These terms become important only when Mach number is near 1. With this assumption the energy equation becomes:

$$\frac{\partial p}{\partial t} + \gamma p \frac{\partial u}{\partial Z} + \gamma u \frac{\partial p}{\partial Z} = 0 \qquad \text{Eq. 9}$$

The above equations have built into them the assumption that the area of the chamber does not change with the distance Z. It has also assumed no combustion. The combustion will be assumed to occur at the injector face and will be included in the analysis as a boundary condition.

All the dependent variables in the above equations will be assumed to be composed of a mean part which does not vary with time and a perturbation part which does vary with time.

$$p = \overline{p} + p' e^{St}$$

$$\rho = \overline{\rho} + \rho' e^{St} \qquad \text{Eq. 10}$$

$$u = \overline{u} + u' e^{St}$$

The variable S is the Laplace variable and is in general complex. When these are substituted into the equations and products of perturbations are neglected (which implies that only small amplitude oscillations will be considered) the following equations result:

$$S\rho' + \overline{u} \frac{d\rho'}{dZ} + \overline{\rho} \frac{du'}{dZ} = 0 \qquad \text{Eq. 11}$$

B, Technical Description (cont.)

$$\bar{\rho} S u' + \bar{\rho}\bar{u} \frac{du'}{dZ} + \frac{dp'}{dZ} = 0 \qquad \text{Eq. 12}$$

$$Sp + \gamma \bar{p} \frac{du'}{dZ} + \gamma \bar{u} \frac{dp'}{dZ} = 0 \qquad \text{Eq. 13}$$

In obtaining the above equations $\frac{d\bar{u}}{dZ}$, $\frac{d\bar{p}}{dZ}$, $\frac{d\bar{\rho}}{dZ}$ were all set equal to zero since the area is not changing and no combustion is occurring.

These three equations can be combined to give:

$$\frac{c^2}{S^2} \frac{d^2}{dZ^2} \left(\frac{p'}{p}\right) - (1-\gamma M^2) \frac{p'}{p} = \frac{\bar{u}}{S} (\gamma+1) \frac{d}{dZ} \left(\frac{p'}{p}\right) \qquad \text{Eq. 14}$$

where M is the Mach number and c is the acoustic velocity, $c = \sqrt{\gamma g R T}$. If the Mach number is small enough γM^2 is negligible compared to 1. For an even smaller Mach number the right-hand side can be neglected. The governing equation then becomes the homogeneous Helmholtz equation:

$$\frac{d^2}{dZ^2} \left(\frac{p'}{p}\right) - \frac{S^2}{c^2} \left(\frac{p'}{p}\right) = 0 \qquad \text{Eq. 15}$$

From Equations 11 thru 14, velocity and density can be obtained in terms of pressure.

$$\frac{u'}{\bar{u}} = \frac{1}{\gamma} \frac{p'}{\bar{p}} - \frac{1}{\gamma} \frac{c^2}{S\bar{u}} \frac{d}{dZ} \left(\frac{p'}{\bar{p}}\right) \qquad \text{Eq. 16}$$

$$\frac{\rho'}{\bar{\rho}} = \frac{1}{\gamma} \frac{p'}{\bar{p}} \qquad \text{Eq. 17}$$

Equation 15 is a second order equation and requires two boundary conditions. At the nozzle entrance the velocity and density are related by the specific acoustic admittance.

$$\frac{u'/\bar{u}}{\rho'/\bar{\rho}} = \alpha \qquad \text{Eq. 18}$$

B, Technical Description (cont.)

At the injector the burning rate and pressure are related by the sensitive time lag burning expression.

$$\frac{(\rho u)'}{\bar{\rho}\bar{u}} = \mathcal{P}\,\frac{p'}{\bar{p}} \qquad \text{where} \qquad \mathcal{P} = n\,(1-e^{-S\tau}) \qquad \text{Eq. 19}$$

The solution to equation 15 is:

$$\frac{p'}{\bar{p}} = c_1\, e^{\frac{SZ}{c}} + c_2\, e^{-\frac{SZ}{c}} \qquad \text{Eq. 20}$$

Combining Equation 20 with the 2 boundary conditions, after considerable algebra, the following expression is obtained.

$$M\,(1-\gamma\mathcal{P}) = \frac{1 - B\exp\left(\frac{2SL}{c}\right)}{1 + B\exp\left(\frac{2SL}{c}\right)} \qquad \text{where } B = \frac{1+\alpha M}{1-\alpha M} \qquad \text{Eq. 21}$$

This expression applies without qualification to the primary combustor:

$$M_p\,(1-\gamma\mathcal{P}_p) = \frac{1 - B_p\exp\left(\frac{2SL_p}{c_p}\right)}{1 + B_p\exp\left(\frac{2SL_p}{c_p}\right)} \qquad \text{where } B_p = \frac{1+\alpha_p M_p}{1-\alpha_p M_p} \qquad \text{Eq. 22}$$

For the secondary combustor Equation 21 must be modified slightly to include the flow injected from the turbine exit as well as the pressure sensitive combustion:

$$M_S\left[1-\gamma\left(G\,e^{-S\tau_{fs}} + \mathcal{P}_s\right)\right] = \frac{1 - B_s\exp\left(\frac{2SL_s}{c_s}\right)}{1 + B_s\exp\left(\frac{2SL_s}{c_s}\right)} \qquad \text{Eq. 23}$$

$$\text{where} \quad B_s = \frac{1+\alpha_s M_s}{1-\alpha_s M_s}$$

B, Technical Description (cont.)

The factor $e^{-S\tau_{fs}}$ is the combustion time delay for the fuel entering the secondary. The term G is determined as follows: Define $\hat{G}_p = \frac{\dot{w}_p'}{P_p'}$. This can be related to α_p by:

$$\hat{G}_p = \frac{1}{\gamma}\,\frac{\bar{\dot{w}}_p}{\bar{P}_p}\,(1+\alpha_p) \qquad \text{Eq. 24}$$

The admittance upstream of the turbine is related to the admittance downstream of the turbine by:

$$\frac{1}{\hat{G}_{fs}} = \frac{1}{\hat{G}_p} - \frac{2\,(\bar{P}_p - \bar{P}_s)}{\bar{\dot{w}}_p} \qquad \text{Eq. 25}$$

The G in Equation 23 is non-dimensional. To convert $\hat{G}_{fs}$ to G divide by $\frac{\bar{\dot{w}}_s}{\bar{P}_s}$

$$G = \hat{G}_{fs}\,\frac{\bar{P}_s}{\bar{\dot{w}}_s} = \frac{\bar{P}_s}{\bar{\dot{w}}_s}\;\frac{\dfrac{\bar{\dot{w}}_p}{2\,(\bar{P}_p-\bar{P}_s)}}{\dfrac{1}{\hat{G}_p}\,\dfrac{\bar{\dot{w}}_p}{2\,(\bar{P}_p-\bar{P}_s)} - 1}$$

$$G = \frac{-R_s}{1+R_s}\;\frac{\bar{P}_s}{2\,(\bar{P}_p-\bar{P}_s)}\left[\frac{1}{1-\left(\dfrac{\bar{P}_p}{2\,(\bar{P}_p-\bar{P}_s)}\right)\left(\dfrac{\gamma}{1+\alpha_p}\right)}\right] \qquad \text{Eq. 26}$$

where R_s is the mixture ratio of the secondary chamber.

Equations 22, 23 and 26 are the governing equations for the system. These equations can now be solved simultaneously for $\mathscr{P}_s$.

C. EQUATIONS

Equation 22 must first be solved for α_p

$$p = \frac{\Omega\ (1+\beta) - 1+\beta}{M_p\ (\Omega\ (1-\beta) - \beta - 1)}$$

$$\text{where } \beta = \exp\left(\frac{2SL_p}{c_p}\right) \quad \text{and } \Omega = M_p\ (1 - \alpha\mathcal{P}_p)$$

Equation 26 can then be used to get G.

$$G = -\left(\frac{R_s}{1+R_s}\right)\left(\frac{\overline{P}_s}{2\ (\overline{P}_p - \overline{P}_s)}\right)\left[\frac{1}{1 - \left(\frac{\overline{P}_p}{2\ (\overline{P}_p - \overline{P}_s)}\right)\left(\frac{\gamma}{1+\alpha_p}\right)}\right]$$

Finally equation 23 can be used to solve for $\mathcal{P}_s$.

$$\mathcal{P}_s = \frac{\left[\frac{1 - B_s \exp\left(\frac{2SL_s}{c_s}\right)}{1 + B_s \exp\left(\frac{2SL_s}{c_s}\right)} - M_s\right]}{-\ \gamma M_s} - G\ e^{-S\tau_{fs}}$$

The n and τ come from the definition of $\mathcal{P}_s$.

$$\mathcal{P}_s = n_s\ (1 - e^{-S\tau_s})$$

Set $S = 0 + i\ \omega$ and solve for $n\ (\omega)$ and $\tau(\omega)$

D. DEFINITION OF TERMS

B	Intermediate term defined in equation 21
C_1--C_2	Constants in Equation 20 to be determined by the boundary conditions
c_p	Specific heat at constant pressure
c_v	Specific heat at constant volume
e	Base of Natural Logarithms
e_o	Internal energy of stagnant combustion gases
G	Nondimensional weight flow admittance injected into secondary chamber
G_p	Dimensional weight flow admittance upstream of the primary exhaust nozzle
h_o	Enthalpy of stagnant combustion gases
L	Length of cylindrical portion of chamber
M	Mach number in cylindrical portion of chamber
n	Pressure interaction index measuring the amount of interaction between pressure oscillations and burning rate oscillations
p	Pressure
R	Universal gas constant
R_s	Mixture Ratio in secondary chamber
S	Laplace variable
t	Time
T	Temperature
u	Velocity of gas
$\dot{w}$	Weight flow rate
Z	Length dimension
α	Specific Acoustic admittance (= $-\frac{\mathcal{A}\gamma}{M}$ where $\mathcal{A}$ is from Ref. 1)
γ	Ratio of specific heats
ρ	Gas density

D. Definition of Terms (cont.)

τ_{fs}	Total time lag of the fuel injected into the secondary chamber in seconds
ω	Frequency of oscillation
τ	Sensitive time lag in seconds
w	dimensional frequency in radians per second
$\wp$	$n(1 - e^{-S\tau})$

Subscripts

p	Primary combustor
s	Secondary combustor

Superscripts

—	Mean value - does not vary with time
'	Perturbation value

E. SPECIAL OPTIONS

This program is restricted to the problem described in Section B.

F. NUMERICAL METHODS OF SOLUTION

Complex arithmetic is used in this program. The specific acoustic admittance of the second combustor α_s is input as a table. An interpolation function GNINT, taken from the UNIVAC MATH PACK, is used to interpolate for values of α_s between those input in the table. A third order polynomial is used in this interpolation.

G. REFERENCES

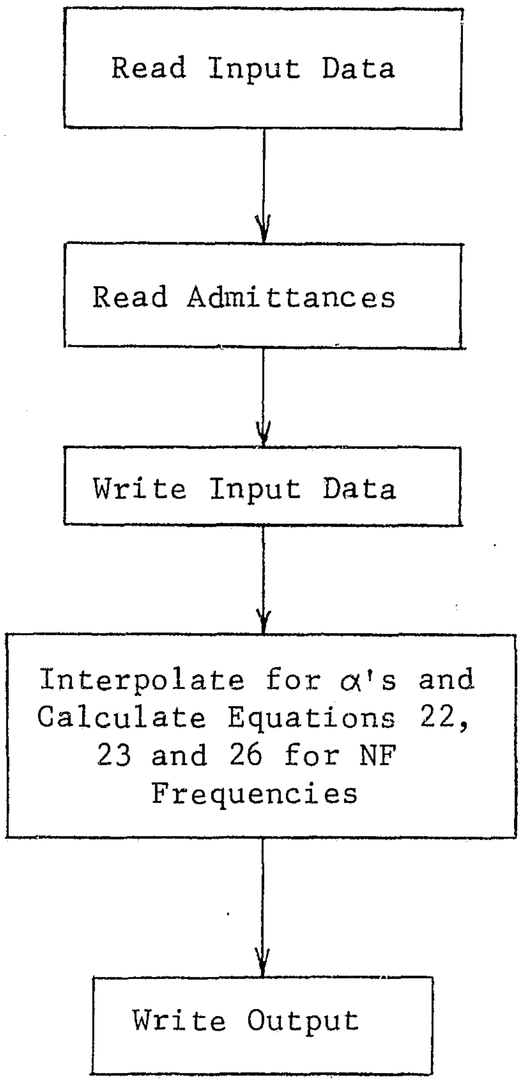
1. Stability Characterization of Advanced Injectors Design Guide Vol. 2 Operation of the Computer Program 20672-P2D Jan. 1970.

2. Stability Characterization of Advanced Injectors Design Guide for Stable H_2/O_2 Combustors. Vol 1: Design Application 20672-P2D May 1970.

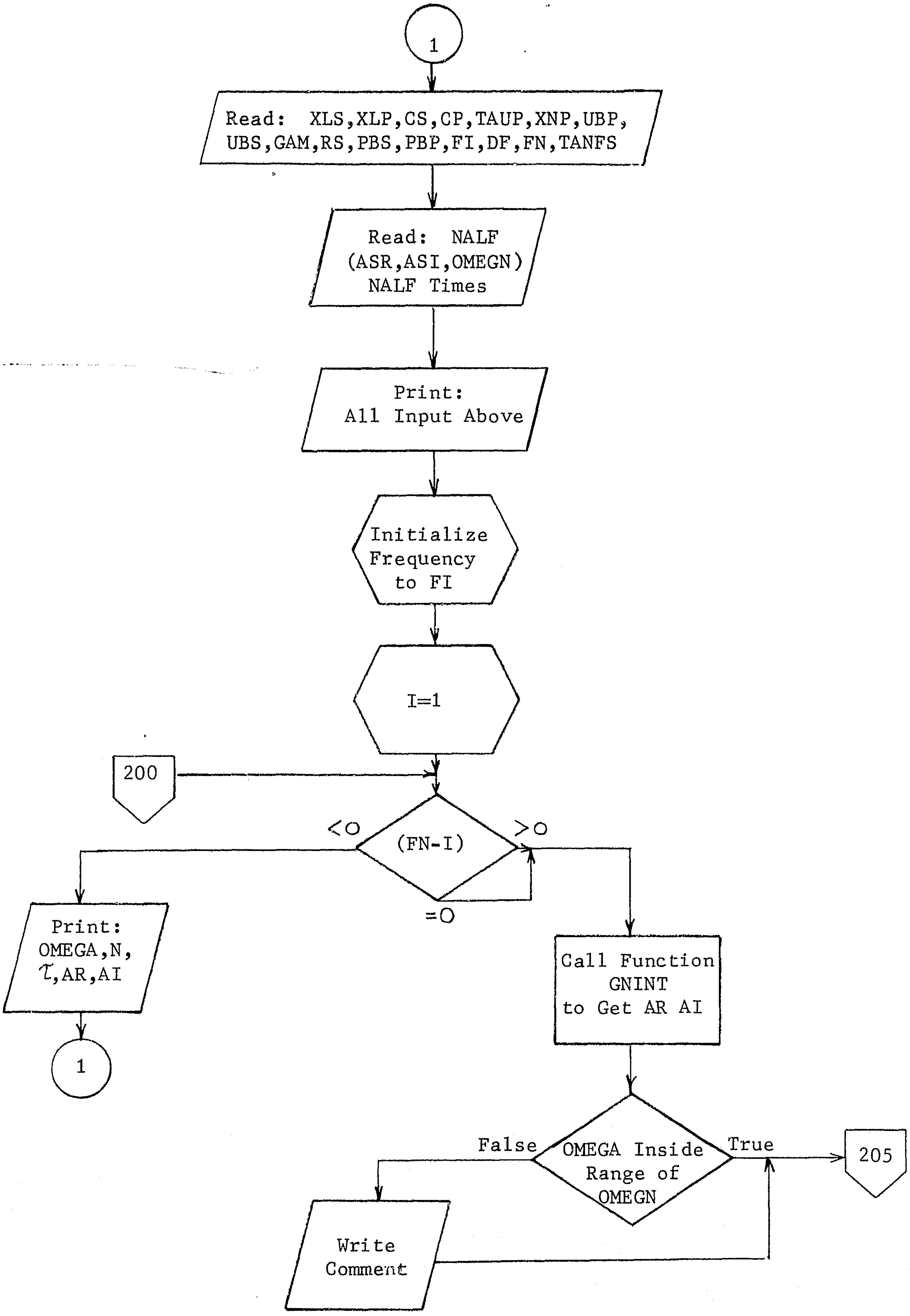
3. General Reference Manual, UNIVAC 1108, Sperry-Rand Corp., 1966.

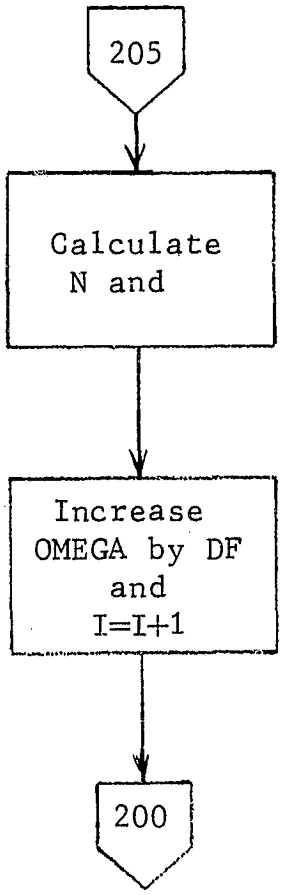
SECTION II - PROGRAMING

A. BLOCK DIAGRAM



B. FLOW CHARTS




205
Calculate N and
Increase OMEGA by DF and I=I+1
200

C. PROGRAM LISTING

```
@ ELT STAGE,1,700724, 29533

C     COMMENTS
C     THIS PROGRAM CALCULATES AN EN, TAU ZONE FOR THE SECONDARY INJECTOR OF A
C     STAGED COMBUSTION CYCLE.  THE EN AND TAU FOR THE PRIMARY ARE INPUT TO THE
C     PROGRAM ALONG WITH A FUEL TOTAL TIME LAG FOR THE SECONDARY.  THE ANALYSIS
C     IS ONE DIMENSIONAL AND APPLIES TO LONGITUDINAL MODES WITHOUT FEED SYSTEM
C     COUPLING.  THE ANALYSIS FROM WHICH THIS PROGRAM WAS WRITTEN APPEARS IN THE
C     AEROJET GENERAL REPORT NUMBER 20672-PIF.  THE INPUT PARAMETERS ARE AS
C     FOLLOWS:
C     XLS = LENGTH OF SECONDARY CHAMBER IN FEET
C     XPL = LENGTH OF PRIMARY CHAMBER IN FEET
C     CS = SPEED OF SOUND IN SECONDARY IN FEET PER SECOND
C     CP = SPEED OF SOUND IN PRIMARY IN FEET PER SECOND
C     TAUP = SENSITIVE TIME LAG IN PRIMARY CHAMBER IN SECONDS
C     XNP = INTERACTION INDEX IN PRIMARY CHAMBER (DIMENSIONLESS)
C     UBP = MACH NUMBER IN PRIMARY CHAMBER
C     UBS = MACH NUMBER IN SECONDARY CHAMBER
C     GAM = RATIO OF SPECIFIC HEATS
C     RS = MIXTURE RATIO OF SECONDARY CHAMBER
C     PBS = SECONDARY CHAMBER PRESSURE IN POUNDS PER SQUARE INCH
C     PBP = PRIMARY CHAMBER PRESSURE IN POUNDS PER SQUARE INCH
C     FI = INITIAL FREQUENCY AT WHICH THE EN AND TAU ARE TO BE CALCULATED, HZ
C     DF = INCREMENT BETWEEN FREQUENCIES
C     FN = NUMBER OF FREQUENCIES
C     TAUFS = TOTAL TIME LAG OF FUEL IN SECONDARY CHAMBER IN SECONDS
C     ASR, ASI = REAL AND IMAGINARY PARTS OF SPECIFIC ACOUSTIC ADMITTANCE AS
C     DEFINED BY CROCCO (CALLED ALPHA), DIMENSIONLESS
C     OMEGN = FREQUENCY AT WHICH THE ASR, ARE GIVEN - RADIANS PER SECOND
       DIMENSION  ASR(50),ASI(50),OMEGN(50),SN(100),TAUMS(100),HERTZ(100)
      1,D(10),ALFPR(100),ALFPI(100)
       COMPLEX SPP,EXPP,G,ALFP,EXPS,RS,ALFS,PHLFS,H,S
     1 READ(5,101)XLS,XLP,CS,CP,TAUP,XNP,UBP,UBS,GAM,RS,PBS,PBP,FI,DF,FN
      1,TAUFS
   101 FORMAT(   6F10.1)
       READ(5,102) NALF,(ASR(I),ASI(I),OMEGN(I),I=1,NALF)
   102 FORMAT(I2,/,(3F20.7))
       WRITE(6,103)XLS,XLP,CS,CP,TAUP, XNP,UBP,UBS,GAM,RS,PBS,PBP,TAUFS,
      1(OMEGN(I),ASR(I),ASI(I),I=1,NALF)
   103 FORMAT(1X,15HINPUT DATA     //1X,35HLENGTH OF SECONDARY CHAMBER IN
      1FEET=,F10.4/1X,34HLENGTH OF PRIMARY CHAMBER IN FEET=,F10.4/
      21X,55HSPEED OF SOUND IN SECONDARY CHAMBER IN FEET PER SECOND=,F10.
      34/1X,53HSPEED OF SOUND IN PRIMARY CHAMBER IN FEET PER SECOND=,F10.
      44/1X,12HTAU PRIMARY=,F10.4/1X,26HINTERACTION INDEX PRIMARY=,F10.4/
      51X,20HMACH NUMBER PRIMARY=,F10.4/1X,22HMACH NUMBER SECONDARY=,F10.
      64/1X,6HGAMMA=,F10.4/1X,24HMIXTURE RATIO SECONDARY=,F10.4/1X,33HMEA
      7N CHAMBER PRESSURE SECONDARY =,F10.4/1X,30HMEAN CHAMBER PRESSURE P
      8RIMARY=,F10.4/1X,33HTOTAL TIME LAG OF SECONDARY FUEL=,F10.4//1X,41
      9HSECONDARY NOZZLE ADMITTANCE (OMEGA,AR,AI)/(1X,3F20.4))
       WRITE(6,104)
   104 FORMAT(1X, 6HOUTPUT,/,1X,98H        FREQUENCY                 N
      1                   TAU                  APR                 API    )
       NF=FN
       OMEGI=2.*3.1415927*FI
       OMEGA=OMEGI
       DO 200 I=1,NF
       S=CMPLX(0.0,OMEGA)
```

```
      SPP=XNP*(1.0-EXP(-S*TAUP))
      EXPP=EXP(2.0*S*XLP/CP)
      AR=GNINT(OMEGN,ASR,NALF,3,OMEGA,D,,400)
      AI=GNINT(OMEGN,ASI,NALF,3,OMEGA,D,,400)
      GO TO 205
  400 WRITE(6,500)
  500 FORMAT(1X,47HINTERPOLATION FUNCTION IS TRYING TO EXTRAPOLATE   )
  205 ALFS=CMPLX(AR,AI)
      BS=(1.0+ALFS*UBS)/(1.0-ALFS*UBS)
      ALFP=(UBP*(1.0-GAM*SPP)*(1.0+EXPP)-1.0+EXPP)/(UBP*(UBP*(1.0-GAM*
     1SPP)*(1.0-EXPP)-EXPP-1.0))
      G=-RS/(1.0+RS)*PBS/2.0/(PBP-PBS)*(1.0/(1.0-(PBP/2.0/(PBP-PBS))
     1*(GAM/(1.0+ALFP))))
      PHLFS=EXP(-S*TAUFS)
      EXPS=EXP(2.0*S*XLS/CS)
      H=-((1.0-BS*EXPS)/(1.0+BS*EXPS)-UBS)/GAM/UBS-G*PHLFS
      HI=AIMAG(H)
      HR=REAL(H)
      ALFPR(I)=REAL(ALFP)
      ALFPI(I)=AIMAG(ALFP)
      SN(I)=(HR*HR+HI*HI)/HR*0.5
      WT=3.1415927-2.0*ATAN(HI/HR)
      TAUMS(I)=WT/OMEGA*1000.
      HERTZ(I)=OMEGA/6.2831853
  200 OMEGA=OMEGA +DF*6.2831853
      WRITE(6,300) (HERTZ(I),SN(I),TAUMS(I),ALFPR(I),ALFPI(I),I=1,NF)
  300 FORMAT(1X,5F20.5)
      GO TO 1
      END

 4.    LIST GNINT                                                    10:04:36
```

@ ELT GNINT,1,700722, 38216

```
      FUNCTION GNINT(X,Y,N,LD,XBAR,D,$)                                 GNINT
C-----------------------------------------------------------------------GNINT
C     GREGORY-NEWTON INTERPOLATION AND EXTRAPOLATION                    GNINT
C     THIS SUBPROGRAM DETERMINES                                        GNINT
C     1.  WHETHER EXTRAPOLATION OR INTERPOLATION IS REQUIRED.           GNINT
C     2.  WHETHER FORWARD OR BACKWARD INTERPOLATION IS TO BE PERFORMED.GNINT
C-----------------------------------------------------------------------GNINT
C     ARGUMENT DEFINITION                                               GNINT
C        X    -- ARRAY OF N INDEPENDENT VARIABLE VALUES                 GNINT
C        Y    -- ARRAY OF N DEPENDENT VARIABLE VALUES                   GNINT
C        N    -- NUMBER OF DATA POINTS                                  GNINT
C        LD   -- DESIRED DEGREE OF INTERPOLATING POLYNOMIAL             GNINT
C        XBAR -- INDEPENDENT VALUE FOR WHICH GNINT IS THE INTERPOLANT   GNINT
C                OR EXTRAPOLANT                                         GNINT
C        D    -- ARRAY OF LD+1 ELEMENTS USED FOR TEMPORARY STORAGE BY   GNINT
C                GNPOL.                                                 GNINT
C                GNINT STORES SPACING OF X ARRAY ELEMENTS IN D(1).      GNINT
C        $    -- ERROR EXIT IF EXTRAPOLATION NECESSARY, AND XBAR IS MOREGNINT
C                THAN 5 SPACES FROM LIMIT OF X ARRAY.                   GNINT
C-----------------------------------------------------------------------GNINT
C     OTHER SUBPROGRAMS REFERENCED                                      GNINT
C        GNPOL -- DETERMINATION OF INTERPOLATING POLYNOMIAL VALUE       GNINT
C                 AT XBAR.                                              GNINT
C        GNEXT -- GREGORY-NEWTON EXTRAPOLATION.                         GNINT
C-----------------------------------------------------------------------GNINT
C     REFERENCES                                                        GNINT
C        F. B. HILDEBRAND,  INTRODUCTION TO NUMERICAL ANALYSIS,         GNINT
C           MCGRAW-HILL BOOK CO., 1956.                                 GNINT
C        J. B. SCARBOROUGH,  NUMERICAL MATHEMATICAL ANALYSIS,           GNINT
C           JOHNS HOPKINS PRESS,  1962.                                 GNINT
C-----------------------------------------------------------------------GNINT
C     GNINT PERFORMS THE FOLLOWING SET OF OPERATIONS                    GNINT
C     1.  DETERMINES POSITION OF XBAR WITHIN X ARRAY.                   GNINT
C     2.  REFERENCES GNEXT IF EXTRAPOLATION IS NECESSARY.               GNINT
C     3.  DETERMINES MAXIMUM OBTAINABLE INTERPOLATING POLYNOMIAL DEGREE,GNINT
C         AS LIMITED BY N AND POSITION OF XBAR.                         GNINT
C     4.  REDUCES LD IF IT IS GREATER THAN MAXIMUM OBTAINABLE DEGREE.   GNINT
C     5.  DETERMINES WHETHER FORWARD OR BACKWARD INTERPOLATION IS TO BE GNINT
C         PERFORMED.                                                    GNINT
C     6.  INITIALIZES ARGUMENTS FOR GNPOL.                              GNINT
C     7.  REFERENCES GNPOL TO OBTAIN INTERPOLATING POLYNOMIAL VALUE     GNINT
C         AT XBAR.                                                      GNINT
C-----------------------------------------------------------------------GNINT
C     INITIALIZATION OF D(1) AS X ARRAY SPACING.                        GNINT
C     DETERMINATION WHETHER INTERPOLATION OR EXTRAPOLATION IS REQUIRED. GNINT
C-----------------------------------------------------------------------GNINT
      DIMENSION X(N),Y(N),D(1)                                          GNINT
      D(1)=X(2)-X(1)                                                    GNINT
      IF(XBAR.LT.X(1).OR.XBAR.GT.X(N)) GO TO 3                          GNINT
C-----------------------------------------------------------------------GNINT
C     INTERPOLATION REQUIRED.                                           GNINT
C     DETERMINATION OF X ARRAY ELEMENT, X(J), BELOW XBAR.               GNINT
C-----------------------------------------------------------------------GNINT
      J=(XBAR-X(1))/D(1) +1.                                            GNINT
C-----------------------------------------------------------------------GNINT
```

```
C     DETERMINATION OF DEGREE OF INTERPOLATING POLYNOMIAL.              GNINT
C     M IS MAXIMUM POSSIBLE DEGREE, LD IS DEGREE USED.                  GNINT
C-----------------------------------------------------------------------GNINT
      M=MAX0(J,N-J)                                                     GNINT
      LD=MIN0(M,LD)                                                     GNINT
C-----------------------------------------------------------------------GNINT
C     INITIALIZATION OF ARGUMENTS FOR GNPOL.                            GNINT
C-----------------------------------------------------------------------GNINT
      IF(M.EQ.J) J=J+1                                                  GNINT
      S=(XBAR-X(J))/D(1)                                                GNINT
      IF(.NOT.ABS(S).GT.0.) GO TO 2                                     GNINT
      IF(M.NE.J-1) GO TO 1                                              GNINT
      J=-J                                                              GNINT
      S=-S                                                              GNINT
C-----------------------------------------------------------------------GNINT
C     REFERENCE FUNCTION SUBPROGRAM GNPOL TO OBTAIN INTERPOLATING       GNINT
C     POLYNOMIAL VALUE AT XBAR.                                         GNINT
C-----------------------------------------------------------------------GNINT
    1 GNINT=GNPOL(Y,J,S,LD,D)                                           GNINT
      RETURN                                                            GNINT
    2 GNINT=Y(J)                                                        GNINT
      RETURN                                                            GNINT
C-----------------------------------------------------------------------GNINT
C     REFERENCE FUNCTION SUBPROGRAM GNEXT TO PERFORM FORWARD OR         GNINT
C     BACKWARD EXTRAPOLATION.                                           GNINT
    3 GNINT=GNEXT(X,Y,N,LD,XBAR,D,S4)                                   GNINT
      RETURN                                                            GNINT
    4 RETURN 7                                                          GNINT
      END                                                               GNINT
```

```
5.     LIST GNPOL                                                10:04:36
```

@ELT GNPOL,1,700722, 38217

```
      FUNCTION GNPOL(Y,J,S,LD,D)                                        GNPOL
C-----------------------------------------------------------------------GNPOL
C     CALCULATION OF GREGORY-NEWTON INTERPOLATING POLYNOMIAL VALUE      GNPOL
C-----------------------------------------------------------------------GNPOL
C     INITIALIZATION.  D ARRAY INITIALIZED AS Y ARRAY VALUES.           GNPOL
C-----------------------------------------------------------------------GNPOL
      DIMENSION Y(1),D(1)                                               GNPOL
      COEFF=1.                                                          GNPOL
      L=IABS(J)                                                         GNPOL
      GNPOL=Y(L)                                                        GNPOL
      IF(J.LT.0) L=L-LD                                                 GNPOL
      DO 1 K=0,LD                                                       GNPOL
    1 D(K+1)=Y(L+K)                                                     GNPOL
C-----------------------------------------------------------------------GNPOL
C     LOOP THROUGH ALL DIFFERENCES                                      GNPOL
C-----------------------------------------------------------------------GNPOL
      DO 3 K=1,LD                                                       GNPOL
      LDM=LD-K+1                                                        GNPOL
C-----------------------------------------------------------------------GNPOL
C     CALCULATE KTH COLUMN OF DIFFERENCES                               GNPOL
C-----------------------------------------------------------------------GNPOL
      DO 2 L=1,LDM                                                      GNPOL
    2 D(L)=D(L+1)-D(L)                                                  GNPOL
C-----------------------------------------------------------------------GNPOL
C     COMPUTE DIFFERENCE COEFFICIENT(COEFF), POLYNOMIAL TERM(TERM),     GNPOL
C     AND POLYNOMIAL SUM(GNPOL).  TERM SUBTRACTED FOR ODD BACKWARD DIFFSGNPOL
C-----------------------------------------------------------------------GNPOL
      COEFF=COEFF*(S-K+1.)/K                                            GNPOL
      IF(J.GT.0) LDM=1                                                  GNPOL
      TERM=COEFF*D(LDM)                                                 GNPOL
      IF(J.LT.0.AND.MOD(K,2).EQ.1) TERM=-TERM                           GNPOL
    3 GNPOL=GNPOL+TERM                                                  GNPOL
      RETURN                                                            GNPOL
      END                                                               GNPOL
```

```
6.    LIST GNEXT                                                   10:04:36
```

@ ELT GNEXT,1,700722, 38219

```
      FUNCTION GNEXT(X,Y,N,LD,XBAR,D,$)
C-----------------------------------------------------------------------GNEXT
C     GREGORY-NEWTON EXTRAPOLATION                                      GNEXT
C     INITIALIZATION OF VARIABLES AND LOOP PARAMETERS.                  GNEXT
C-----------------------------------------------------------------------GNEXT
      DIMENSION TEMP(5),Y(N),X(N),D(1)                                  GNEXT
      SAVE=XBAR                                                         GNEXT
      H=D(1)                                                            GNEXT
      IF(XBAR.LT.X(1)) GO TO 1                                          GNEXT
      J=N                                                               GNEXT
      I=-1                                                              GNEXT
      N1=1                                                              GNEXT
      GO TO 2                                                           GNEXT
    1 J=1                                                               GNEXT
      I=1                                                               GNEXT
      N1=2                                                              GNEXT
    2 ITEMP=0                                                           GNEXT
      FI=I                                                              GNEXT
      N2=N1+N-2                                                         GNEXT
      NN=N+1-J                                                          GNEXT
      S=(XBAR-X(J))/H*FI                                                GNEXT
C-----------------------------------------------------------------------GNEXT
C     IF MAGNITUDE OF S LT 1, TABLE DOES NOT HAVE TO BE EXTENDED.       GNEXT
C     RECOMPUTE S.                                                      GNEXT
C-----------------------------------------------------------------------GNEXT
      IF(ABS(S).GT.5.) GO TO 6
      IF(.NOT.ABS(S).GT.1.) GO TO 5                                     GNEXT
      XBAR=X(J)-FI*H                                                    GNEXT
    3 GNEXT=GNPOL(Y,I*J,FI,LD,D)                                        GNEXT
      ITEMP=ITEMP+1                                                     GNEXT
      TEMP(ITEMP)=Y(NN)                                                 GNEXT
      DO 4 K=N1,N2                                                      GNEXT
    4 Y(K)=Y(K-I)                                                       GNEXT
      Y(J)=GNEXT                                                        GNEXT
      S=(SAVE-XBAR)/H*FI                                                GNEXT
      IF(.NOT.ABS(S).GT.1.) GO TO 5                                     GNEXT
      XBAR=XBAR-FI*H                                                    GNEXT
      GO TO 3                                                           GNEXT
C-----------------------------------------------------------------------GNEXT
C     ONLY ONE (MORE) EXTRAPOLATION TO BE PERFORMED.                    GNEXT
C-----------------------------------------------------------------------GNEXT
    5 XBAR=SAVE                                                         GNEXT
      GNEXT=GNPOL(Y,I*J,S,LD,D)                                         GNEXT
C-----------------------------------------------------------------------GNEXT
C     RESHIFT Y ARRAY (IF NECESSARY) SO THAT IT IS ORIGINAL INPUT ARRAY.GNEXT
C-----------------------------------------------------------------------GNEXT
      IF(ITEMP.EQ.0) RETURN                                             GNEXT
      N1=N1-I                                                           GNEXT
      N2=N2-I                                                           GNEXT
      IF(I.EQ.-1) GO TO 9
      J=N2                                                              GNEXT
      N2=N1
      N1=J
    9 DO 7 K=ITEMP,1,-1
      DO 6 J=N2,N1,I
    6 Y(J)=Y(J+I)                                                       GNEXT
    7 Y(NN)=TEMP(K)                                                     GNEXT
      RETURN                                                            GNEXT
    8 RETURN 7                                                          GNEXT
      END                                                               GNEXT
```

D. METHODS OF VERIFICATION

The only method of verification of the results of this program is based on the experience of the engineer with the sensitive time lag theory and its application. The correctness of the numbers can only be evaluated as reasonable by an engineer who has become familiar with the similar analysis for a single chamber as presented in Program B, described in the main portion of this report. The test case given herein is a convenient method of checking the program and computer for correct operation.

SECTION III - DECK SETUP

A. COMPUTER CONFIGURATION

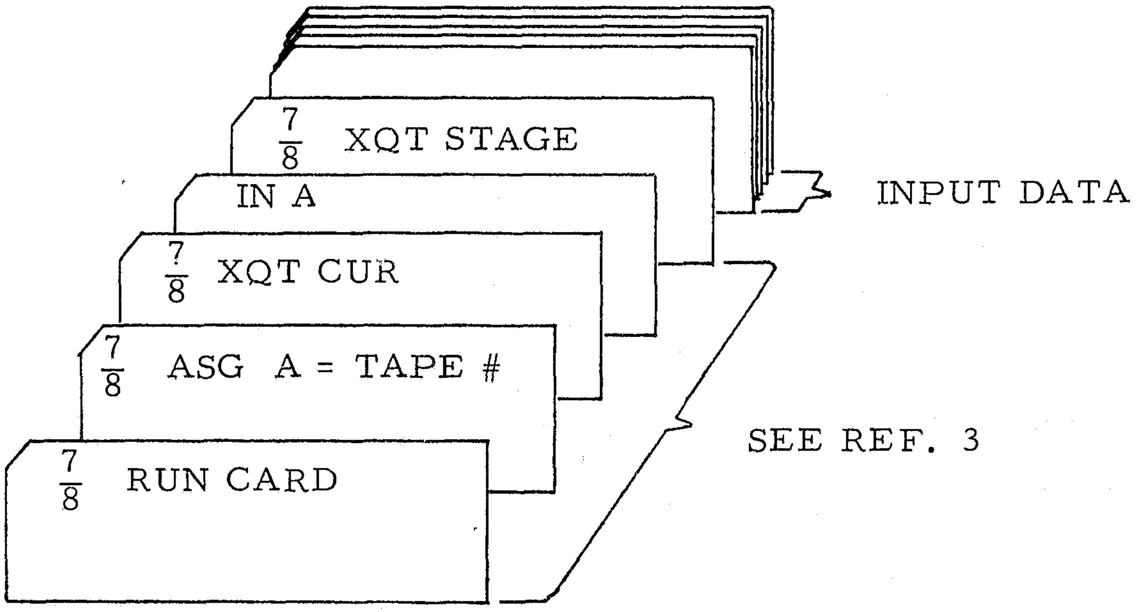
1. Univac 1108 computer, 65 K of core minimum
2. 46211_8 for code, 72245_8 for data
3. FORTRAN V
4. Executive II monitor system
5. No plot output required
6. No punch output required

B. ESTIMATED RUNNING TIME

The sample case presented in this section took 11 seconds to run including compilation. This case can be assumed to be a representative case as far as computer run time is considered.

C. DECK SEQUENCE

The deck sequence shown below includes the control cards as required by the EXEC II Monitor system for the UNIVAC 1108 as described in Ref. 3.



Multiple cases may be run by stacking the data cards for subsequent cases immediately behind the data cards for the original case.

D. INPUT DATA

Symbol	Name in Program	Units	Definition
L_s	XLS	ft.	Length of cylindrical portion of secondary chamber
L_p	XLP	ft.	Length of cylindrical portion of primary chamber
c_s	CS	ft/sec	Speed of sound of burned gases in the secondary chamber
c_p	CP	ft/sec	Speed of sound of burned gases in the primary chamber
τ_p	TAUP	sec	Mean sensitive time lag for the primary
n_p	XNP	—	Combustion interaction index for the primary chamber
M_p	UBP	—	Mach number in primary chamber
M_s	UBS	—	Mach number in secondary chamber
γ	GAM	—	Ratio of specific heats
R_s	RS	—	Mixture Ratio of secondary chamber
$\bar{P}_s$	PBS	psi	Mean pressure in secondary chamber
$\bar{P}_p$	PBP	psi	Mean pressure in primary chamber
F_i	FI	(H_z)	Initial calculation frequency
ΔF	DF	Hz	Frequency increment of calculation
N_f	FN	—	Number of frequencies to be calculated
τ_{fs}	TAUFS	sec	Total time lag of fuel entering the secondary chamber

Symbol	Name in Program	Units	Definition
—	NALF	—	Number of admittance values input
α_s	ASR	—	Real part of specific acoustic admittance
	ASI	—	Imaginary part of specific admittance
ω	OMEGN	$\frac{\text{rad}}{\text{sec}}$	Frequency at which specific acoustic admittance is given

E. RESTRICTION AND LIMITATIONS

Number of input admittances is limited to 50 per case. Number of n, t points output is limited to 100 per case. Applies only to longitudinal modes. Mach number in both chambers must be low.

F. OUTPUT DEFINITION

All input parameters are output as well as the following:

SYMBOL	NAME ON OUTPUT	UNITS	DEFINITION
f	F	Hertz	Frequency
n	N	—	Combustion interaction index for the secondary chamber
τ	TAU	sec	Mean sensitive time lag for the secondary chamber
α_p	APR	—	Real part of specific acoustic admittance of primary nozzle
	API	—	Imaginary part of specific acoustic of primary nozzle

G. INPUT FORM

The format of the input data cards is shown in Figure 2 and a listing of the input data cards for the sample case included herein is given in Figure 3.

REPORT TITLE	DATE	PAGE	of

XLS	XLP	CS	CP	τ_p	NP
$\bar{u}_p$	$\bar{u}_s$	γ	R_s	$\bar{P}_s$	$\bar{P}_p$
FI	DF	FN	τ_{FS}		

ω_N	α_r	α_i

AGCS 8143-10

PLEASE PRINT CLEARLY - USE BLACK PENCIL

Figure 2

```
.01          .625        6520.       5500.        .0005        .6
.04          .274         1.2        6.0          3000.        5660.
200.         100.        59.         .001
19
0.11384                  0.36194                 1120.
0.42671                  1.9776                  5600.
0.97068                  3.0084                  7840.
2.3036                   4.0984                  10100.
4.9362                   3.9735                  12320.
6.1355                   1.1506                  14560.
3.8882                   -0.44747                17920.
3.0435                   0.010772                20150.
2.7955                   0.64416                 22400.
2.9859                   1.2256                  24640.
3.5515                   1.5667                  26900.
4.2895                   1.3871                  29020.
4.6474                   0.65966                 31400.
4.5706                   0.30943                 32300.
4.2807                   -0.033799               33600.
3.9166                   -0.15899                35150.
3.5948                   -0.10287                36500.
3.3702                   0.068232                37900.
3.6929                   -0.13763                38600.
```

Figure 3. Listing of Input for Sample Problem

H. SAMPLE OUTPUT

The following pages consist of the output of a typical case for a staged combustor. The input data for this case is that given the previous section.

```
INPUT DATA

LENGTH OF SECONDARY CHAMBER INFEET=      .0100
LENGTH OF PRIMARY CHAMBER IN FEET=      .0250
SPEED OF SOUND IN SECONDARY CHAMBER IN FEET PER SECOND= 6520.0000
SPEED OF SOUND IN PRIMARY CHAMBER IN FEET PER SECOND= 5500.0000
TAU PRIMARY=      .0005
INTERACTION INDEX PRIMARY=      .6000
MACH NUMBER PRIMARY=      .0400
MACH NUMBER SECONDARY=      .2740
GAMMA=    1.2000
MIXTURE RATIO SECONDARY=    6.0000
MEAN CHAMBER PRESSURE SECONDARY = 3000.0000
MEAN CHAMBER PRESSURE PRIMARY= 5060.0000
TOTAL TIME LAG OF SECONDARY FUEL=      .0010

SECONDARY NOZZLE ADMITTANCE (OMEGA,AR,AI)
             1120.0000              .1138              .3619
             5600.0000              .4267             1.9776
             7840.0000              .9707             3.0084
            10130.0000             2.3036             4.0984
            12320.0000             4.9362             3.9735
            14560.0000             6.1353             1.1506
            17920.0000             3.8882             -.4475
            20150.0000             3.0435              .0108
            22400.0000             2.7955              .6442
            24640.0000             2.9859             1.2256
            26900.0000             3.5515             1.5667
            29020.0000             4.2895             1.3871
            31400.0000             4.6474              .6597
            32300.0000             4.5706              .3094
            33600.0000             4.2807             -.0338
            35150.0000             3.9166             -.1590
            36500.0000             3.5948             -.1029
            37900.0000             3.3702              .0682
            38600.0000             3.6929             -.1376
OUTPUT
        FREQUENCY                  N                  TAU                APR               API
          200.00000              .61595             2.48136            -.87603          -3.15922
          300.00000              .48537             1.52847            -.72905          -4.82700
          400.00000              .50185              .77736            -.53712          -6.59965
          499.99999             1.06957              .34667            -.31220          -8.51178
          599.99999             1.90287              .22842            -.06779         -10.59970
          699.99998             2.26283              .21349             .18126         -12.90380
          799.99998             2.23552              .22351             .41825         -15.47175
          899.99998             2.05917              .23514             .62316         -18.36239
          999.99997             1.80242              .24281             .77019         -21.65088
         1099.99995             1.61241              .23425             .82390         -25.43552
         1199.99995             1.54153              .20549             .73361         -29.84760
         1299.99994             1.72290              .15976             .42476         -35.06677
         1399.99994             2.32886              .11506            -.21465         -41.34697
         1499.99992             3.22977              .08932           -1.35799         -49.06401
         1599.99992             3.85329              .08289           -3.28803         -58.80960
         1699.99991             4.41323              .09258           -6.50885         -71.59144
         1799.99991             4.14735              .10339          -12.04921         -89.29786
         1899.99991             3.83689              .11416          -22.44437        -115.90587
         1999.99989             3.56174              .12230          -45.82713        -161.12933
         2099.99988             3.35459              .12624         -121.87021        -252.18555
         2199.99988             3.24237              .12540         -584.02924        -286.57150
```

2299.99988	3.25262	.12065	-226.79218	387.01318
2399.99985	3.27948	.13994	-36.76532	209.29594
2499.99985	3.30218	.13662	-8.20351	132.80283
2599.99985	3.36707	.13541	-.88121	94.64927
2699.99982	3.42553	.13704	1.33372	72.09322
2799.99979	3.44515	.14110	1.88899	57.23720
2899.99976	3.41340	.14653	1.81951	46.71402
2999.99976	3.32791	.15214	1.50421	38.85959
3099.99973	2.65861	.16858	1.10022	32.76035
3199.99969	2.41349	.17043	.67847	27.87432
3299.99966	2.13358	.16929	.27351	23.86003
3399.99963	1.86012	.16381	-.09607	20.49134
3499.99960	1.65897	.15394	-.41863	17.61157
3599.99957	1.58462	.14252	-.68556	15.10699
3699.99954	1.62669	.13385	-.88988	12.89088
3799.99954	1.81538	.11898	-1.02619	10.89354
3899.99951	1.89397	.11885	-1.09149	9.05626
3999.99948	1.88617	.12024	-1.08607	7.32792
4099.99945	1.79064	.12099	-1.01427	5.66337
4199.99939	1.62669	.11874	-.88460	4.02308
4299.99939	1.46037	.11025	-.70916	2.37307
4399.99933	1.60701	.09273	-.50249	.68477
4499.99933	1.50722	.10389	-.28028	-1.06560
4599.99933	1.53693	.08613	-.05824	-2.89825
4699.99927	1.79609	.07843	.14847	-4.83124
4799.99927	2.00608	.07721	.32532	-6.88208
4899.99921	2.11058	.07907	.45801	-9.06914
4999.99921	2.11510	.08183	.53173	-11.41282
5099.99915	2.04268	.08393	.53055	-13.93654
5199.99915	2.08463	.08307	.43685	-16.66763
5299.99915	1.98179	.08083	.23146	-19.63477
5399.99908	1.95117	.07610	-.10564	-22.89042
5499.99908	2.03983	.07056	-.59362	-26.47582
5599.99902	2.22894	.06650	-1.24876	-30.47047
5699.99902	2.44108	.06507	-2.08274	-34.98930
5799.99896	2.60493	.06597	-3.10316	-40.21666
5899.99896	2.69594	.07208	-4.31850	-46.45850
5999.99896	2.70202	.07512	-5.75063	-54.23857