https://ntrs.nasa.gov/search.jsp?R=19690009401 2020-03-12T07:49:16+00:00Z

N 69 18739

NASA-CR-72494

CASE FILE COPY

COMBUSTION INSTABILITY RESPONSE WITH ASYMMETRIC PRESSURE DISTURBANCES

JANUARY 1969

T.C. KOSVIC, B.P. BREEN, J. LEVINE, AND D.E. COATS

DYNAMIC SCIENCE A DIVISION

OF MARSHALL INDUSTRIES

MONROVIA, CALIFORNIA

PREPARED UNDER CONTRACT NAS 7-467

FOR

LEWIS RESEARCH CENTER

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

COMBUSTION INSTABILITY RESPONSE WITH ASYMMETRIC PRESSURE DISTURBANCES

by

T. C. Kosvic, B. P. Breen, J. Levine, and D. E. Coats
DYNAMIC SCIENCE, a Division
of Marshall Industries
Monrovia, California

NAS7-467

NATIONAL AERONAUTICS & SPACE ADMINISTRATION
LEWIS RESEARCH CENTER

SUMMARY

In order to more closely simulate rocket engine stability testing, provisions were made for the input of arbitrary shaped pressure and velocity waves in the Priem-Guentert Combustion Instability Model. A series of sample cases have shown the engine stability map to be sensitive to the shape and strength of the input disturbance with asymmetric pulses showing lower stability. An asymmetric steep fronted initial disturbance develops into a single pulse traveling wave more closely resembling experiments.

Response and gain factor calculations have been included in the program to show the time and space averaged output/input ratio of the vaporization process. Results show the average gain factor over a wave cycle to be less than one in a stable case and greater than one for an unstable case.

TABLE OF CONTENTS

INTRODUCTION

In recent years the use of the Priem Guentert (Ref. 1) mechanistic instability model has become a useful tool to relate the influences of various injector and propellant variables to tangential mode combustion instability. Dynamic Science (Ref. 2), Rocketdyne (Ref. 3), the AFRPL (Ref. 4), and Purdue University (Ref. 5) among others have performed stability analyses of several rocket engine systems with a good degree of success. The basic analysis procedure (Ref. 2) requires a coupling of results from a Steady-State Combustion Program with either a general stability map generated from the Instability Program or by direct application of the program to several annular nodes of the engine. The stability of the system is then related to that of the most sensitive node (usually the zero drop/gas relative velocity point).

All results to date have been generated using a sinusoidal initial pressure disturbance. This input results in a symmetrical standing wave which results from the addition of two waves traveling in opposite directions (Ref. 6). The purpose of this study was to determine the effects on stability due to the input of initial disturbances of arbitrary shape for both pressure and velocity. These inputs will more closely simulate engine stability testing and the subsequent response of the engine combustion process as experimentally observed.

ANALYSIS

Initial runs with asymmetric steep fronted initial disturbances showed severe numerical problems are encountered when calculating through regions of steep gradients. The "wavelets" as reported by several authors (Ref. 3 and 6) who used symmetrical sinusoidal disturbances appeared much more extreme in these cases and were the source of the numerical problems. Because of this, a rather substantial amount of the program effort was expended toward their elimination. This effort was repaid, however, by the increased confidence in the validity of the data that were subsequently generated. Appendix A (Details of Analysis) describes the process followed to solve the "wavelet" problem.

The shape of the asymmetric initial disturbance chosen for study is given by:

$$\frac{\Delta P}{P} = A + A_p \sin^{n}\left(\frac{\theta}{2}\right), \quad 0 \le \theta \le \pi$$

$$\frac{\Delta P}{P} = A + A_p \sin^{15n}\left(\frac{\theta}{2}\right), \quad \pi \le \theta \le 2\pi$$

where $$A = 1 - \int_{0}^{\pi} A_p \operatorname{Sin}^{n}\left(\frac{\theta}{2}\right) d\theta - \int_{0}^{2\pi} A_p \operatorname{Sin}^{15n}\left(\frac{\theta}{2}\right) d\theta$$

where A_p, and n are input variables. In general this equation leads to a relatively shallow rise until $\theta = \pi$ and a steep decay to $\theta = 2\pi$.

In an attempt to provide a common basis between the linear and nonlinear instability models, the calculation of a gain and response factor was included in the instability program. The response factor N was defined as:

$$N_{\bar{\theta}}(t) = \frac{\int_{\theta=0}^{\theta=2\pi} \left(\frac{\omega_i - \bar{\omega}}{\bar{\omega}}\right)\left(\frac{P_i - \bar{P}}{\bar{P}}\right) d\theta}{\int_{\theta=0}^{\theta=2\pi} \left(\frac{P_i - \bar{P}}{\bar{P}}\right)^2 d\theta}$$

where

ω – nondimensional burning or vaporization rate, ω/ω_o

P – nondimensional chamber pressure, P/P_o

subscript i - instantaneous value

$(\bar{\ })$ - average value between $\theta = 0$ and $\theta = 2\pi$

θ - angular position

t - time

The point values of N were also averaged over a wave period to determine $\bar{N}_\theta$, the average response factor per cycle. One problem encountered when applying this technique was that in the case of the steep fronted input, the wave period was not a constant at all times (frequency varied). For this reason a running integral of response factor with time was maintained and the average value determined over a wave cycle period that was estimated by inspection.

In addition, the calculation of a gain factor was included. This parameter was defined by:

$$G_{\bar{\theta}}(t) = \frac{\left[\int_0^{2\pi} \left(\frac{\omega_i - \bar{\omega}}{\bar{\omega}}\right)^2\right]^{1/2}}{\left[\int_0^{2\pi} \left(\frac{P_i - \bar{P}}{\bar{P}}\right)^2\right]^{1/2}}$$

RESULTS

Arbitrary Shaped Input

The calculated results indicated that an asymmetric initial disturbance did indeed grow into a single pulse traveling wave. Figures 1 through 5 trace the pressure time history of a steep fronted initial disturbance. As can be seen in Figure 5 (time from 8.0 to 9.5 radians) the resulting wave is composed of a relatively flat low pressure portion and a single detonation-like spike occupying about 60° of the chamber circumference. Another interesting result is that the average theta velocity is in the direction opposite to the propagation of the pressure pulse.

In order to demonstrate the effect of a steep fronted initial pressure disturbance, a stability map was generated. The input conditions were a monopropellant case, without drag, a drop Reynolds number of 1000, and relative velocity of 0.01 (same as Fig. 4 of Ref. 6). The steep fronted initial disturbance used for study had a value of n = 3; giving one portion of the wave increasing as $\sin^{3}(1/2\ \theta)$ and a steep portion decreasing as $\sin^{45}(1/2\ \theta)$.

As can be seen from Figure 6. the steep fronted disturbance is substantially more unstable. The reasons for this behavior are not completely understood, however, the large gradients associated with a steep fronted wave appear to have a significant destabilizing effect. Another factor involved is that with the symmetrical (sine wave) input the two opposite moving waves developed each travel through gas that has been processed in the same manner. In the asymmetric wave case the two waves travel through gas processed differently, coalesce into a single traveling wave, essentially concentrating the wave energy. This single travel wave, however, more closely resembles experimental results.

Gain and Response Factor Calculation

The results of two calculations in which the initial pressure disturbances employed were slightly above, and slightly below the stability limit are shown in Figure 7. The response and gain factors did indeed grow in an unstable case and decay in a stable one. In addition, the average response factor over the first cycle was about 0.5, 0.9, and 1.5 for cases which were, respectively, stable, about neutral, and unstable.

An examination of the calculations used to prepare the stability map (Figure 6) indicated that calculations with average response factors over the first wave cycle greater than 0.9 - 1.0 were unstable, while lower calculated values yielded stable results. This agrees with the premise of Reference 7, where it was shown that a combustion process with a response factor greater than 0.9 was necessary to overcome the nozzle loss, and drive an engine unstable. This response factor criterion proved to be of particular value in determining the stability of waveforms with low values of burning rate parameter ($\mathcal{L}$), where previously, the case had to be run to 12 time radians with pressure used as a criterion, stability could now be measured in the first wave cycle (about 3.1 radians). For moderate (0.5 to 1) and high burning rate parameters, stability could still be estimated after a shorter time period with a pressure criterion. It should be pointed out that the above conclusions were drawn from calculations made with a restricted range of Re_d and ΔV; therefore, their universal validity should not be taken for granted.

CONCLUSIONS AND RECOMMENDATIONS

An asymmetric steep-fronted initial pressure disturbance was found to substantially reduce the pressure pulse required to trigger an instability when compared to the pulse size required to trigger an initial symmetrical sinusoidal input. The resulting wave is a single pulse traveling wave which appears more physically realistic. It appears that the large gradients associated with the steep-fronted waves contribute to the reduced stability.

A response factor calculation included in the program has shown instability to occur whenever its average value over a cycle is greater than 0.9-1.0.

As a result of this work, it can be recommended that calculations should be made to define:

1. Reasons behind the reduced stability with an asymmetric wave over wide range of operating conditions.
2. Whether program results using as input spatial initial pressure and velocity pulses measured during bomb and pulse gun tests (Ref. 8) can be correlated with measured stability.
3. A further understanding and experimental correlation of the transient and steady-state factors governing the response factor. This would enable one to make a priori estimates of the response factor which would be related directly to injector parameters.

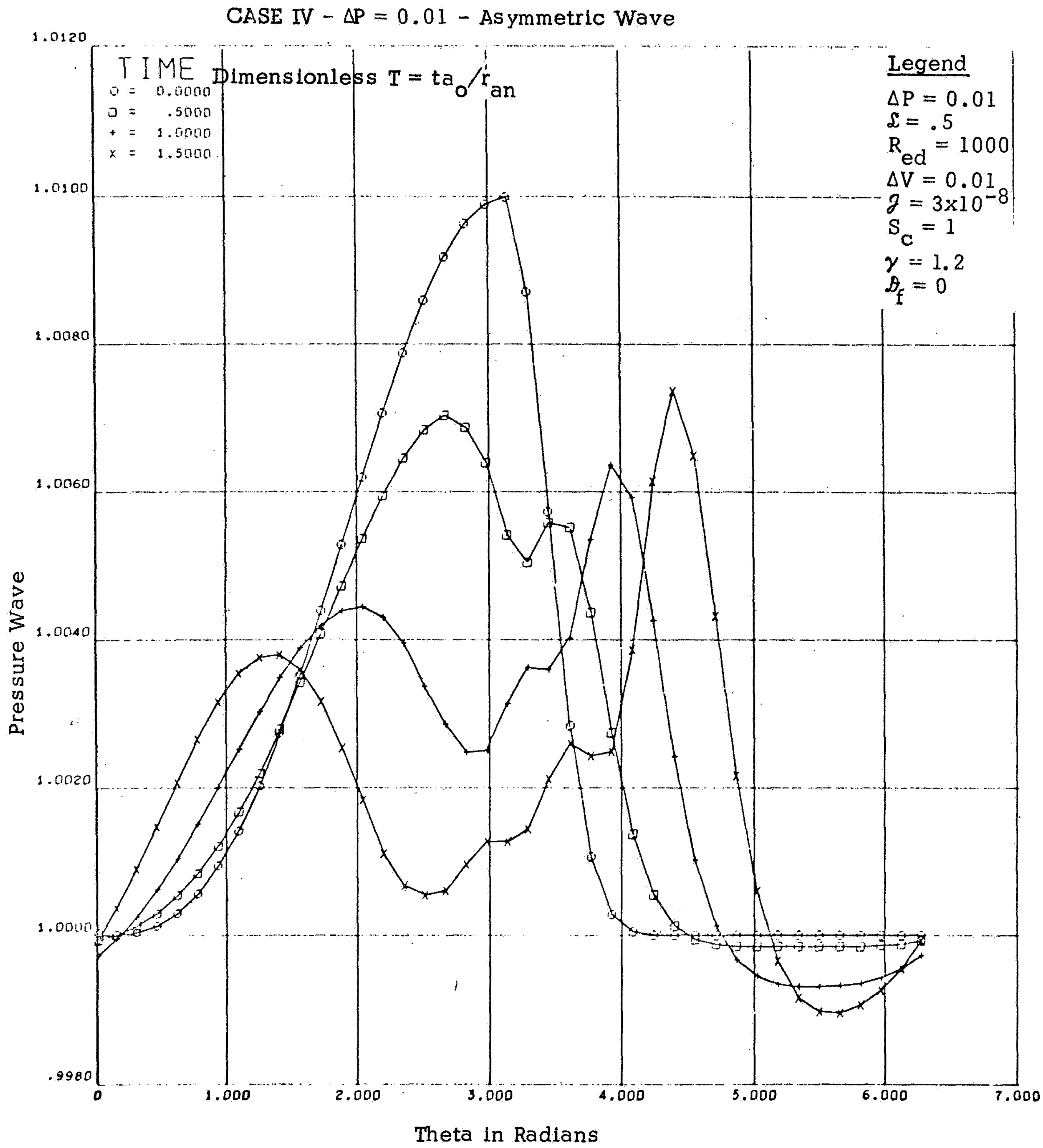

Figure 1. Growth of an Asymmetric Wave into a Single Traveling Pulse (time = 0 - 1.5)

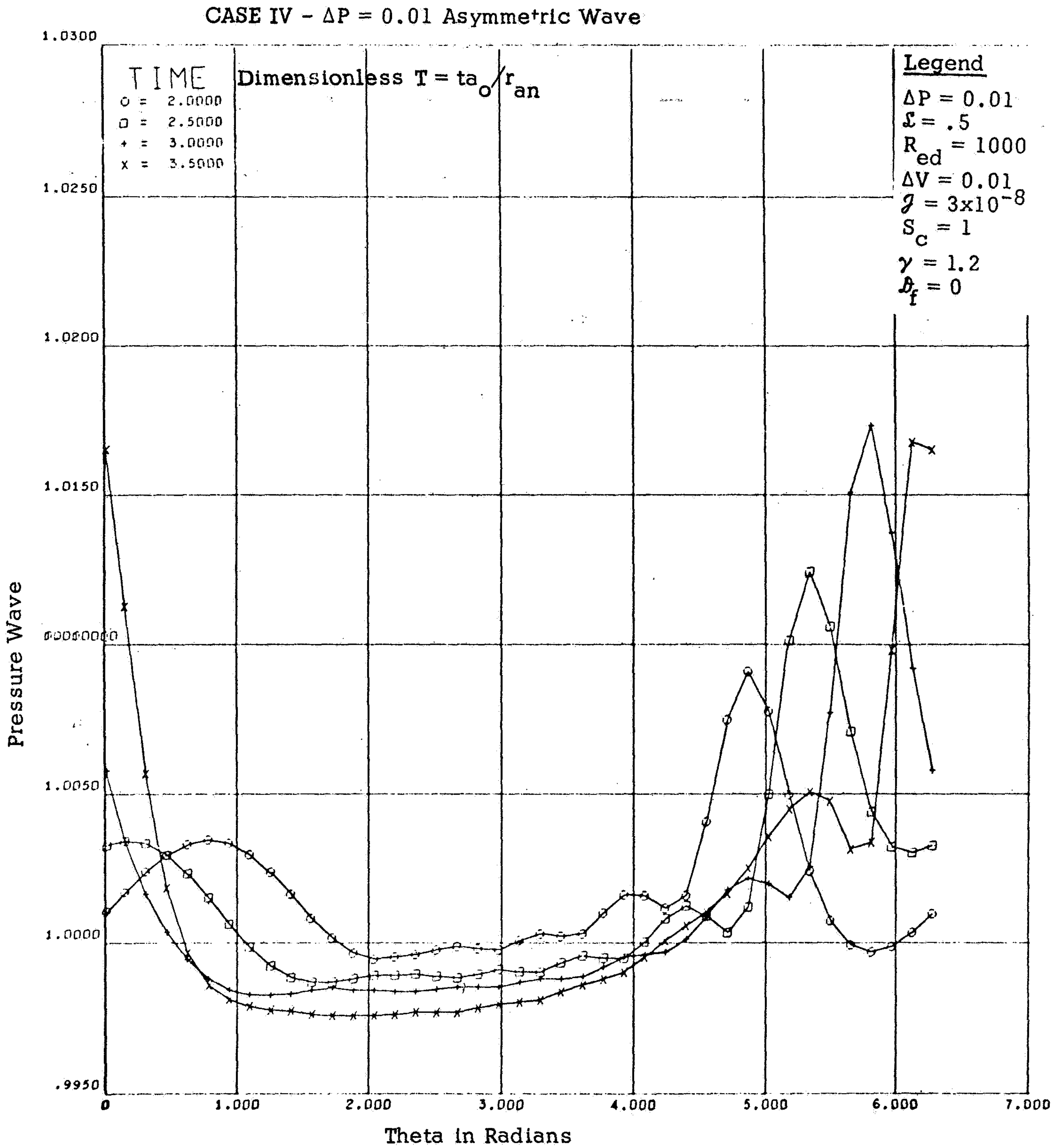

Figure 2. Growth of an Asymmetric Wave into a Single Traveling Pulse (time = 2.0 - 3.5)

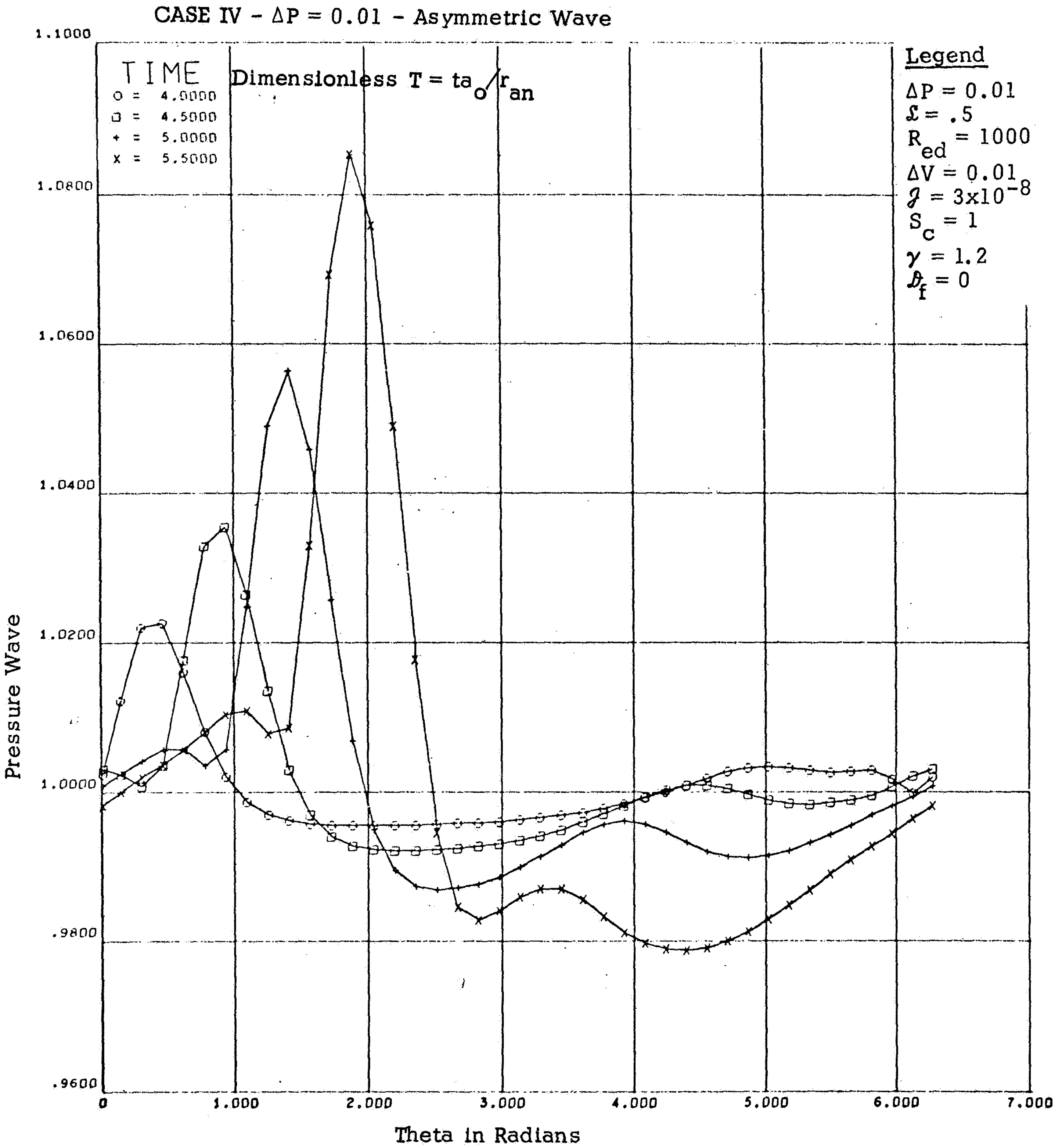

Figure 3. Growth of an Asymmetric Wave into a Single Traveling Pulse (time = 4.0 - 5.5)

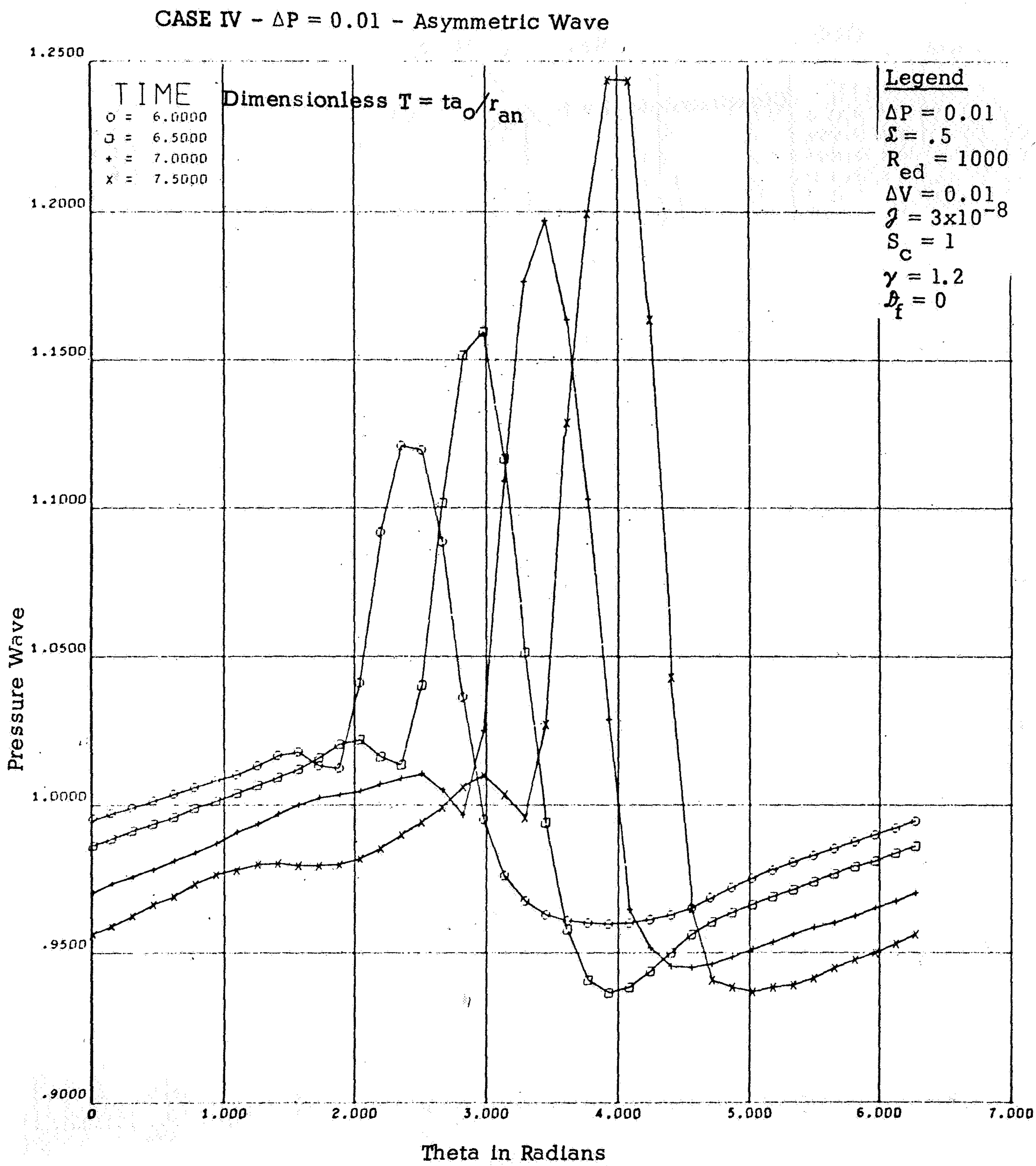

Figure 4. Growth of an Asymmetric Wave into a Single Traveling Pulse (time = 6.0 - 7.5)

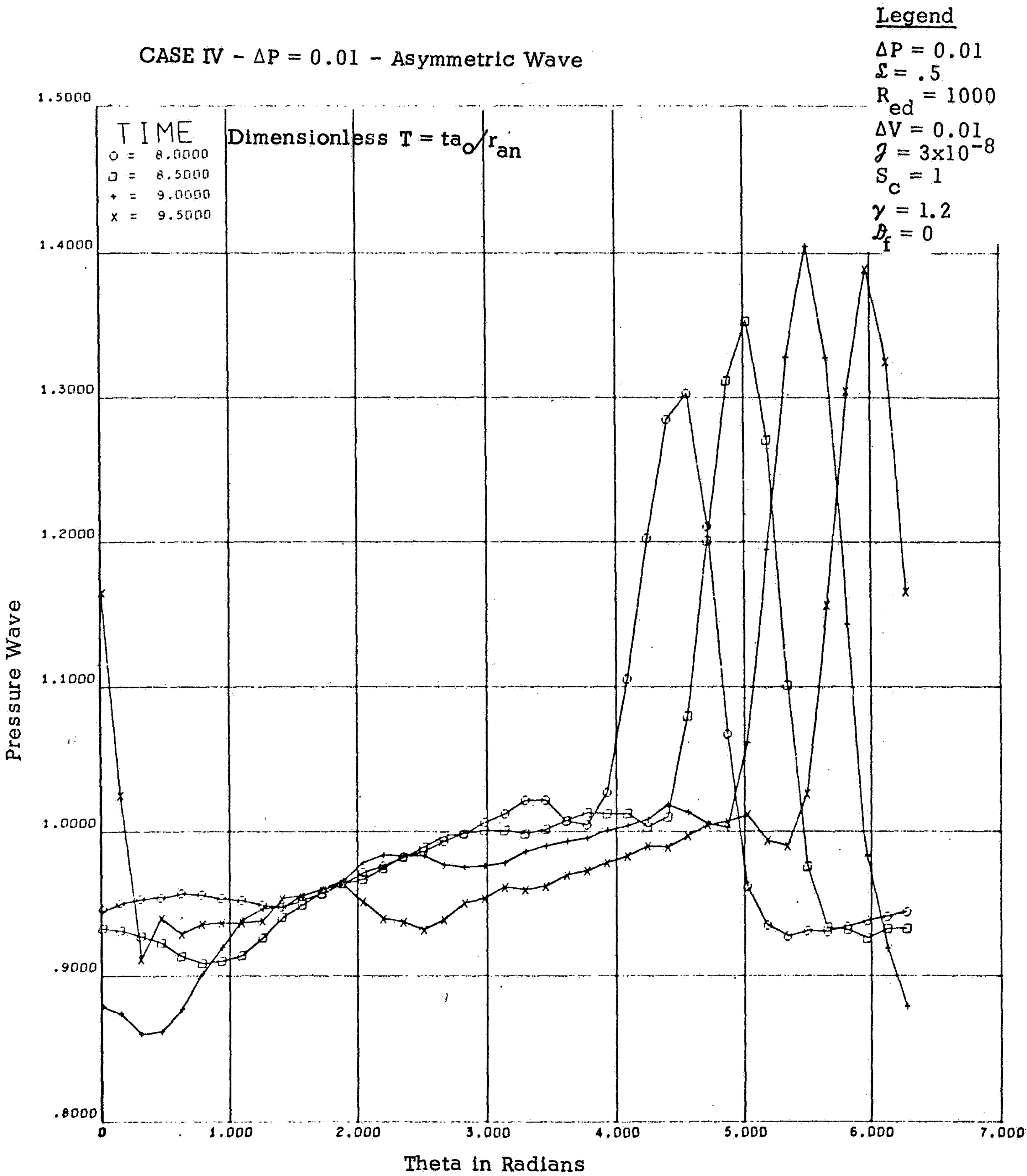

Figure 5. Growth of an Asymmetric Wave into a Single Traveling Pulse (time = 8.0 - 9.5)

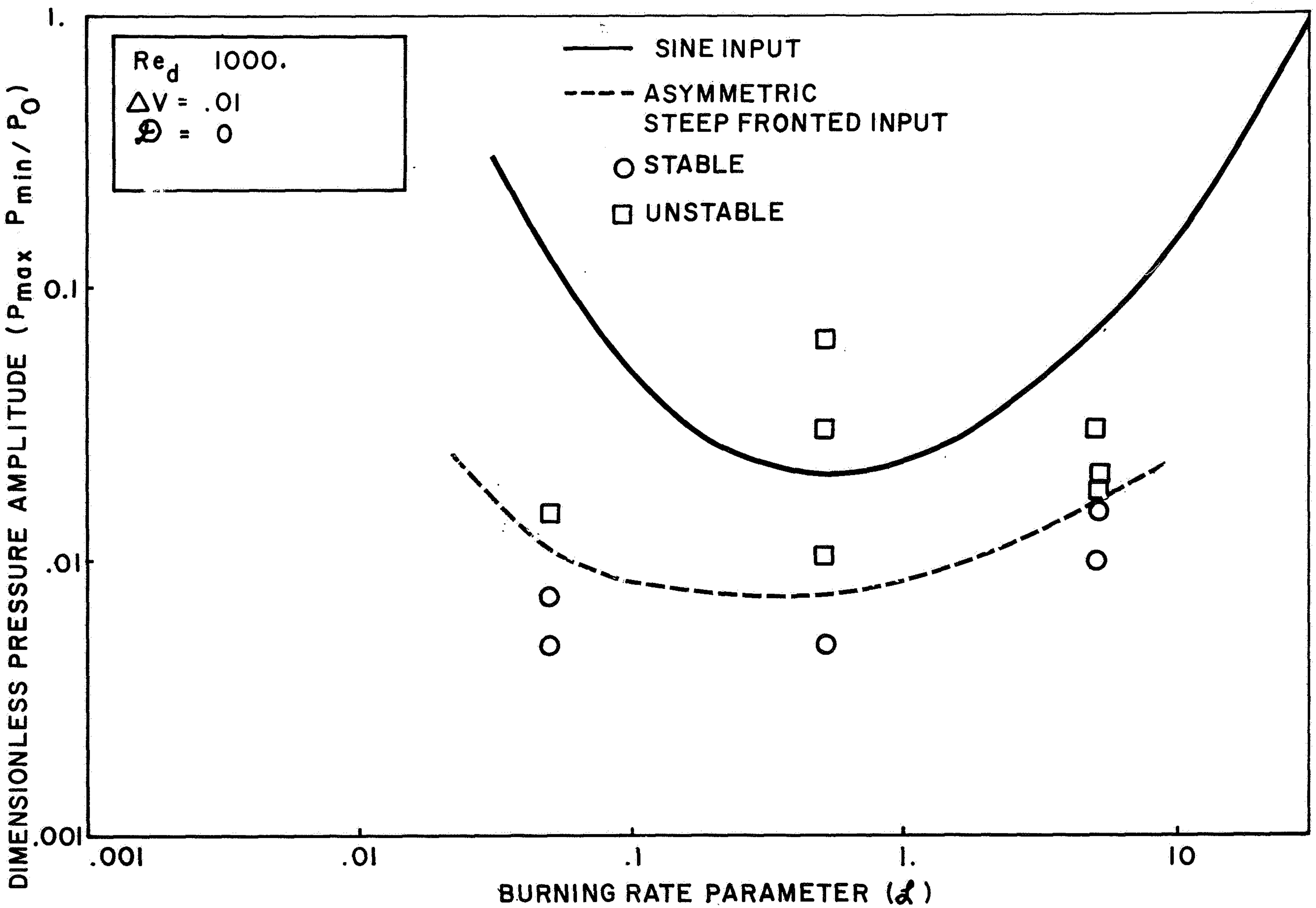

Figure 6. Effect of Asymmetric Wave on Stability Map.

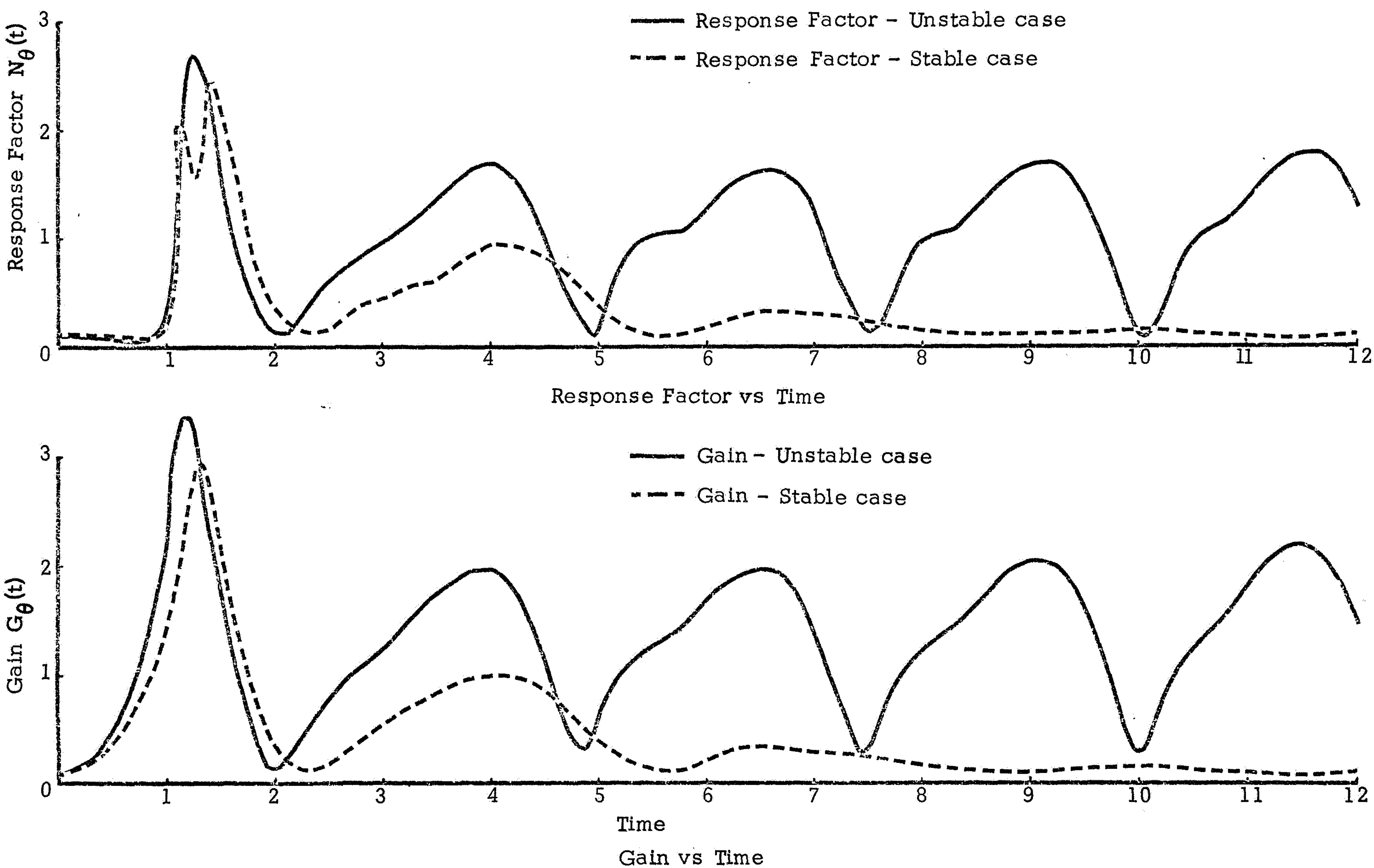

Figure 7. Time Histories of the Vaporization Response and Gain Factor.

REFERENCES

1. Priem, R. J., and Guentert, D. C.: Combustion Instability Limits Determined by a Nonlinear Theory and a One-Dimensional Model, NASA TN D-1409, October 1962.

2. Beltran, M. R., Wright, R. O., Breen, B. P., et al: Combustion Stability Limits Calculated Utilizing a Nonlinear Model" Final Report NAS7-366, NASA-Lewis Research Center, August 1966.

3. Campbell, D. T., Chadwick, W. D.: Combustion Instability Analysis at High Chamber Pressure, AFRPL-TR-68-179, August 1968.

4. Abbe, C. J., McLaughlin, C. W., and Weiss, R. R.: Influence of Storable Propellant Liquid Rocket Design Parameters on Combustion Instability, J. Spacecraft & Rockets, Vol. 5, No. 5, p. 584, May 1968.

5. Netzer, D. W., Osborn, J. R.: Investigation of Combustion Characteristics of a Bi-Phase Rocket, AFOSR753-66, April 1968 Jet Propulsion Center, Purdue University.

6. Hoffman, R. J. Wright, R. O., Breen, B. P.: Combustion Instability Prediction Using a Nonlinear Bipropellant Vaporization Model, NASA CR-920, January 1968.

7. Heidmann, M. F., Wieber, P. R.: Analysis of Frequency Response Characteristics of Propellant Vaporization, NASA TN-D-3749, December, 1966.

8. Combs, L. P., Schumacher, P. E., Kesselring, and Kuluva, N.M.: Improvement of Bombs and Pulse Guns as Combustion Stability Rating Devices, Final Report AF Contract F04611-67-C-0073, Rocketdyne, March 1968.

9. Clayton, R.M., Rogero, R. S., Sotter, J.G.: An Experimental Description of Destructive Liquid Rocket Resonant Combustion, JPL Technical Report No. 32-1293, July 1968.

APPENDIX A

DETAILS OF ANALYSIS

To provide the data and guide the analysis of the severe "wavelet" formation occurring with asymmetric input disturbances, a general SC4020 plotting routine was prepared. This permitted the rapid and economical display of all program variables versus any other program variables after an arbitrary number of calculation cycles, e.g., pressure, velocity, and burning rate against theta position at various times. This form of the output proved to be especially useful as a diagnostic tool in analyzing mathematical problems.

In preparation for including the capability for handling arbitrary initial conditions of pressure and velocity (traveling waves) for all levels of input disturbances, a series of test cases showed numerical problems with pressure pulses from 0.08 and higher depending on the values of other input parameters ($\mathcal{L}, Re_d, \Delta V$). For many cases of practical interest, this difficulty was overcome during the previous contract (NAS7-442) by the implementation of an implicit first-order corrector iteration scheme. However, for cases with high pressure disturbances numerical difficulties continued to exist, and the problem could only be remedied by appropriately decreasing the mesh size in regions of difficulty. Decreasing the mesh size led to excessive computer running time. An examination of plots of the variables ω (burning rate), V_θ, and P versus θ such as shown in Figures 8-10 at successive time steps showed the error propagating from the region where $|\partial V_\theta/\partial \theta|$ was large and in particular the node where $V_\theta = 0$.

In Figure 10, the error propagating from the $V_\theta = 0$ nodes (i.e., $\theta = \pi/2$ and $3\pi/2$) in both directions and the every other point nature of the resulting instability suggested that the error was propagated by the $\partial/\partial\theta$ difference operator used to calculate $\partial V_\theta/\partial \theta$, $\partial T/\partial \theta$, and $\partial \rho/\partial \theta$. Further examination (Fig. 8) showed the theta derivative of ω to be undefined and ω to be discontinuous at the $V_\theta = 0$ node by a time of 0.7500 radians. This led to a suggested fix whereby the burning rate at the $V_\theta = 0$ nodes was interpolated rather than calculated directly, so as to remove the possibility of numerical discontinuity. The interpolation formula used is

$$\omega_i = \bar{\omega}_1 - \frac{1}{3}(\bar{\omega}_2 - \bar{\omega}_1)$$

$$\bar{\omega}_1 = \frac{1}{2}(\omega_{i+1} + \omega_{i-1})$$

$$\bar{\omega}_2 = \frac{1}{2}(\omega_{i+2} + \omega_{i-2})$$

i.e., averaged parabolic interpolation. A check run showed five figure agreement with the integral of ω over theta as previously calculated showing no change in the energy addition rate. The improvement in the results can be seen by a comparison of Figures 10 and 11 where the first figure shows the results from the program in the Final Report on contract NAS7-442 and Figure 11 shows the results with ω interpolated at the $V_\theta = 0$ nodes. The error is seen to be considerably reduced.

The every other point nature of the "wavelets" also suggested a modification of the theta direction difference operator. Therefore the operator

$$\left.\frac{\partial f}{\partial \theta}\right|_i = \frac{f_{i+1} - f_{i-1}}{2\theta}$$

was replaced by

$$\left.\frac{\partial f}{\partial \theta}\right|_i = \frac{1}{8\Delta\theta} \quad 2f_{i+1} - 2f_{i-1} + f_{i+2} - f_{i-2}$$

thereby tying together 4 positions; 2 on either side of the one being calculated. Figure 12 shows the further improvement in the results for the same case as Figures 10 and 11 with both the modified $\partial/\partial\theta$ difference operator and interpolation for ω at $V_\theta = 0$.

To integrate the equations a fully implicit method of the Crank-Nicolson type was selected and the newly developed program, including arbitrary shaped input is documented in Appendix B. Some interesting results were obtained from the checkout cases run with the fully implicit program. Initial runs were concerned with duplicating results from the previous version of the program (Ref. 6). In fact as the number of nodes in the old program was increased from 40 - 80 - 160, the solution approached the new solution calculated using 40 nodes. Figure 13 shows the case shown previously when calculated with the fully implicit program. The improvement, noted by comparing Figures 10-13, is most significant. Figures 14-17 show the history of an asymmetric pressure wave given initially by:

$$P = 1 + 0.4 \sin^3(\tfrac{1}{2}\,\theta),\quad 0 \le \theta \le 3.55$$

$$P = 1.0 \qquad\qquad ,\ 3.55 \le \theta \le 6.28$$

and shown as circle symbols in Figure 14. In Figure 14 it can be seen that the artificial pressure discontinuity introduced by the initial conditions caused numerical difficulties in the region of the discontinuity as evidenced by the numerical oscillations in the subsequent time profiles. Tracing the history of the wave through the following figures, it may be seen that the effect of the discontinuity is continuously damped and has effectively disappeared by a time of four radians (Fig. 17). Considering the magnitude of the initial pressure pulse and the size of the original discontinuity, the numerical stability demonstrated by these results is quite remarkable.

In addition, several cases were run with various step sizes in an effort to determine optimum step size critical to employment of the Crank-Nicolson integration subroutine. It was found that due to the greatly enhanced stability characteristics, the stability of the solution no longer is critically dependent upon the step size (both in absolute value and the ratio $\theta t/\Delta\theta$). The sole criteria for choosing step sizes to eliminate the "wavelets" described in References 2 and 6 is, therefore, that enough nodes be taken to insure that the truncation errors remain within tolerable bounds, or in other words, enough nodes be taken to assure a complete point description of the resulting waves. This, of course, is a function of the initial conditions and the resulting gradients of the physical parameters. It was found that 40 nodes were adequate in the case of stable runs as these typically do not generate large gradients. When the solution goes unstable (physically), steep waves and large gradients are produced, therefore 40 nodes were found to be sufficient to accurately predict stability or instability, however, if one wished to continue the integration after the instability has formed, a larger number of nodes is required depending on the severity of the instability.

As can be seen from Figures 14 thru 14d the resulting wave following a steep fronted input becomes a nearly single pulse traveling wave more closely resembling the experimental results as reported in Ref.9 as far as shape and frequency.

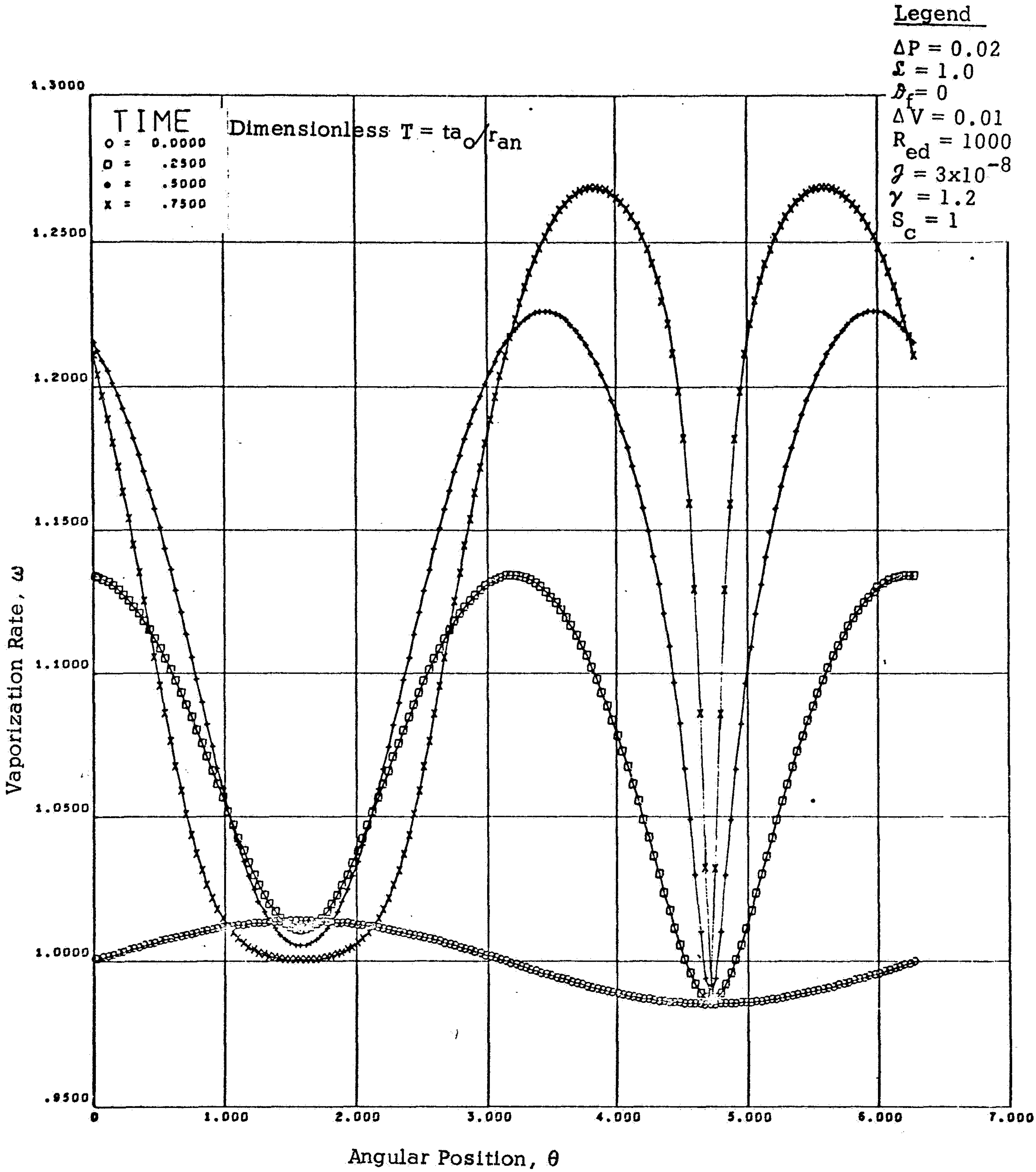

Figure 8. Time Growth of Vaporization Rate.

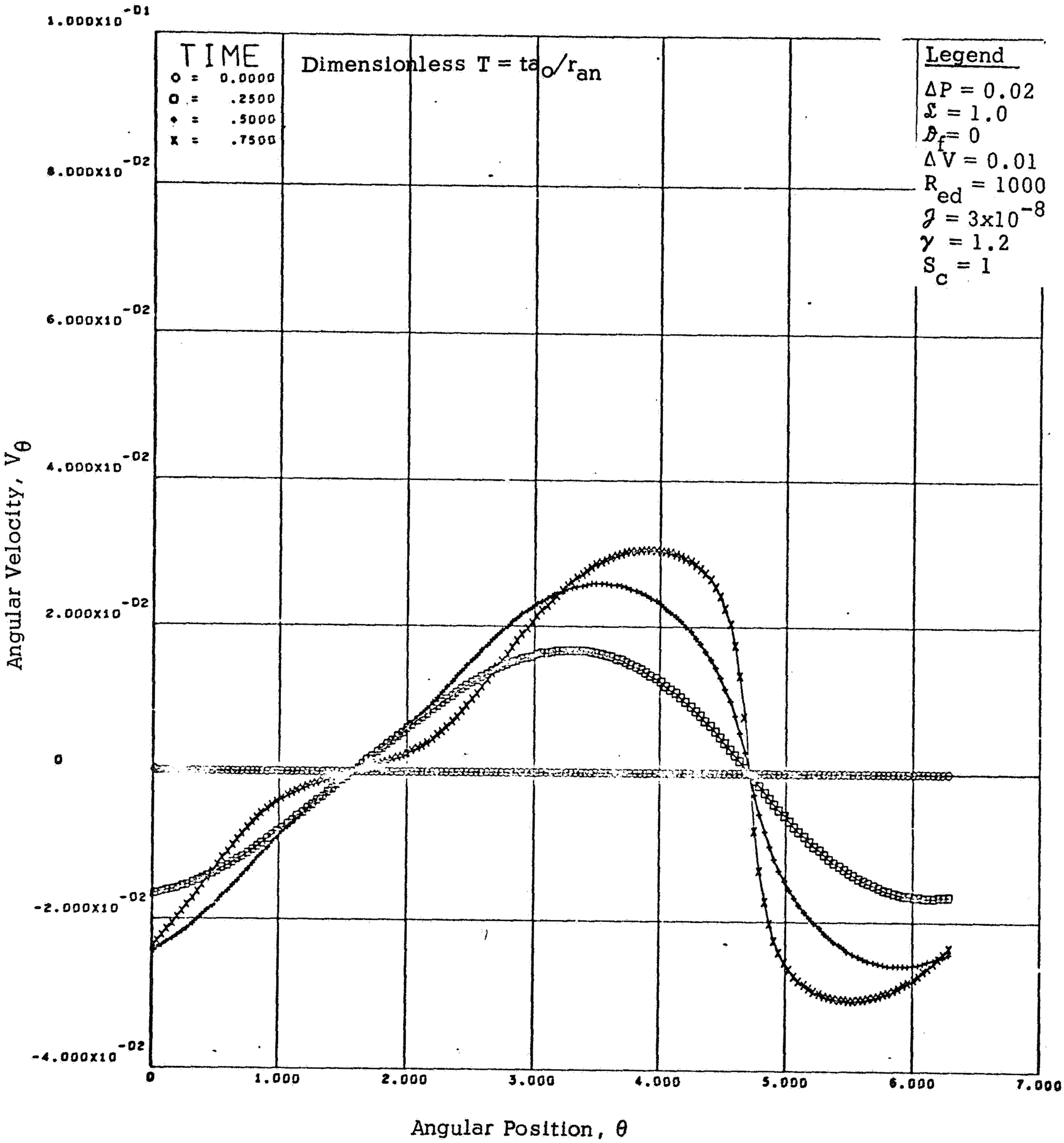

Figure 9. Time Growth of Angular Velocity, V_θ.

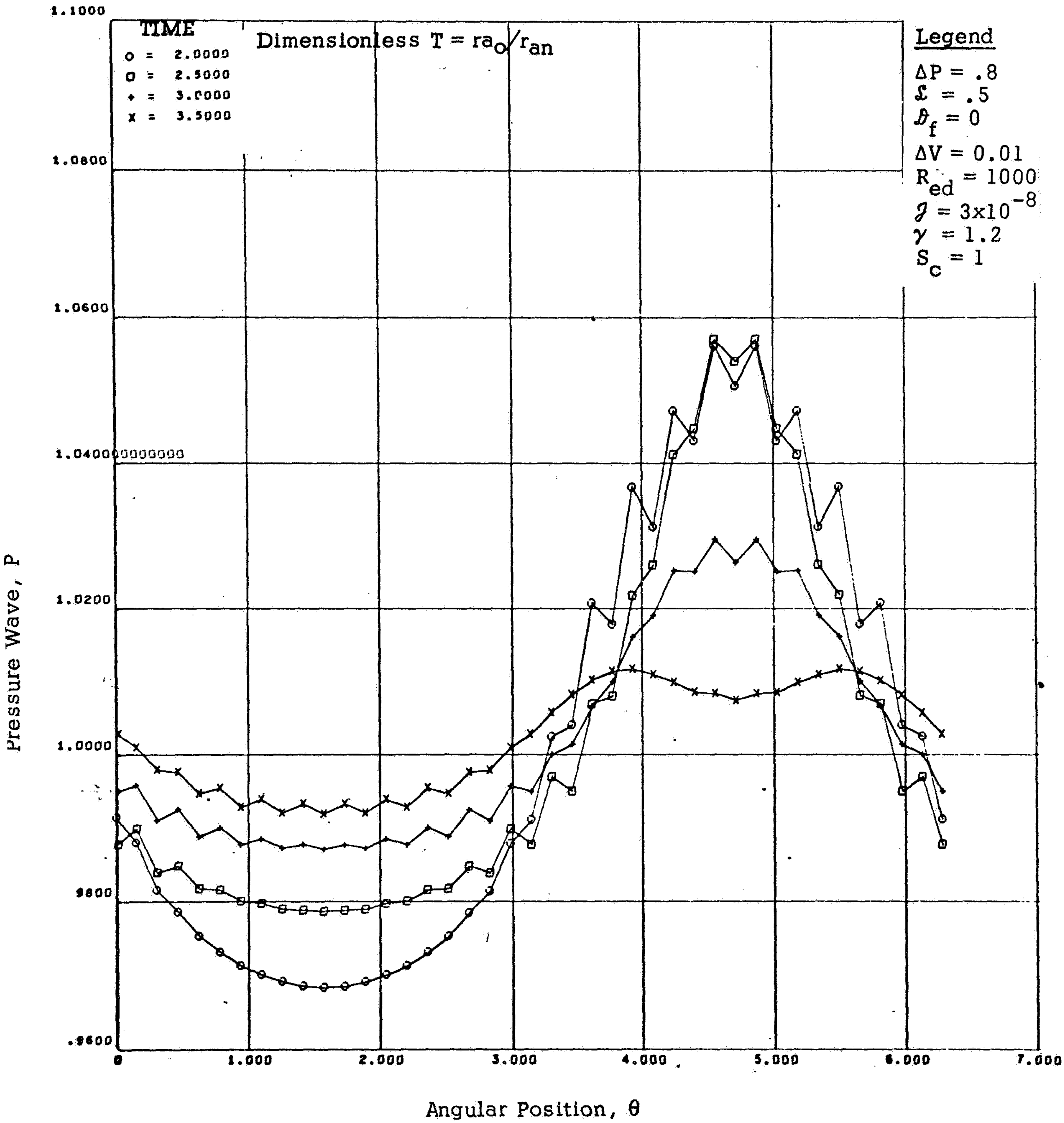

Figure 10. The Formation of "Wavelets" in the Theta Pressure/Time Profile with Program of Reference 6.

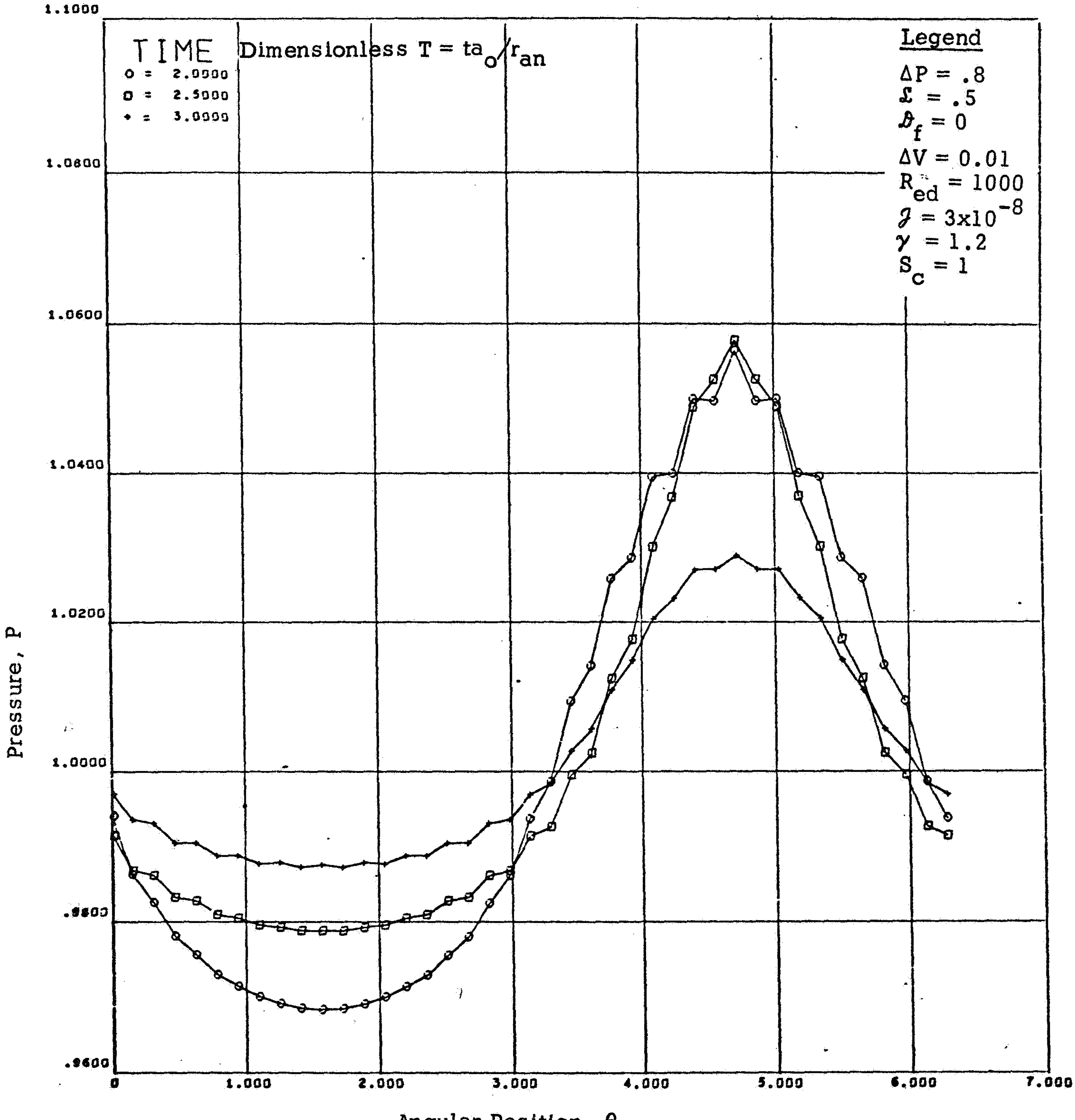

Figure 11. The Decrease in Wavelet Formation due to ω interpolated at $V_\theta = 0$ Nodes.

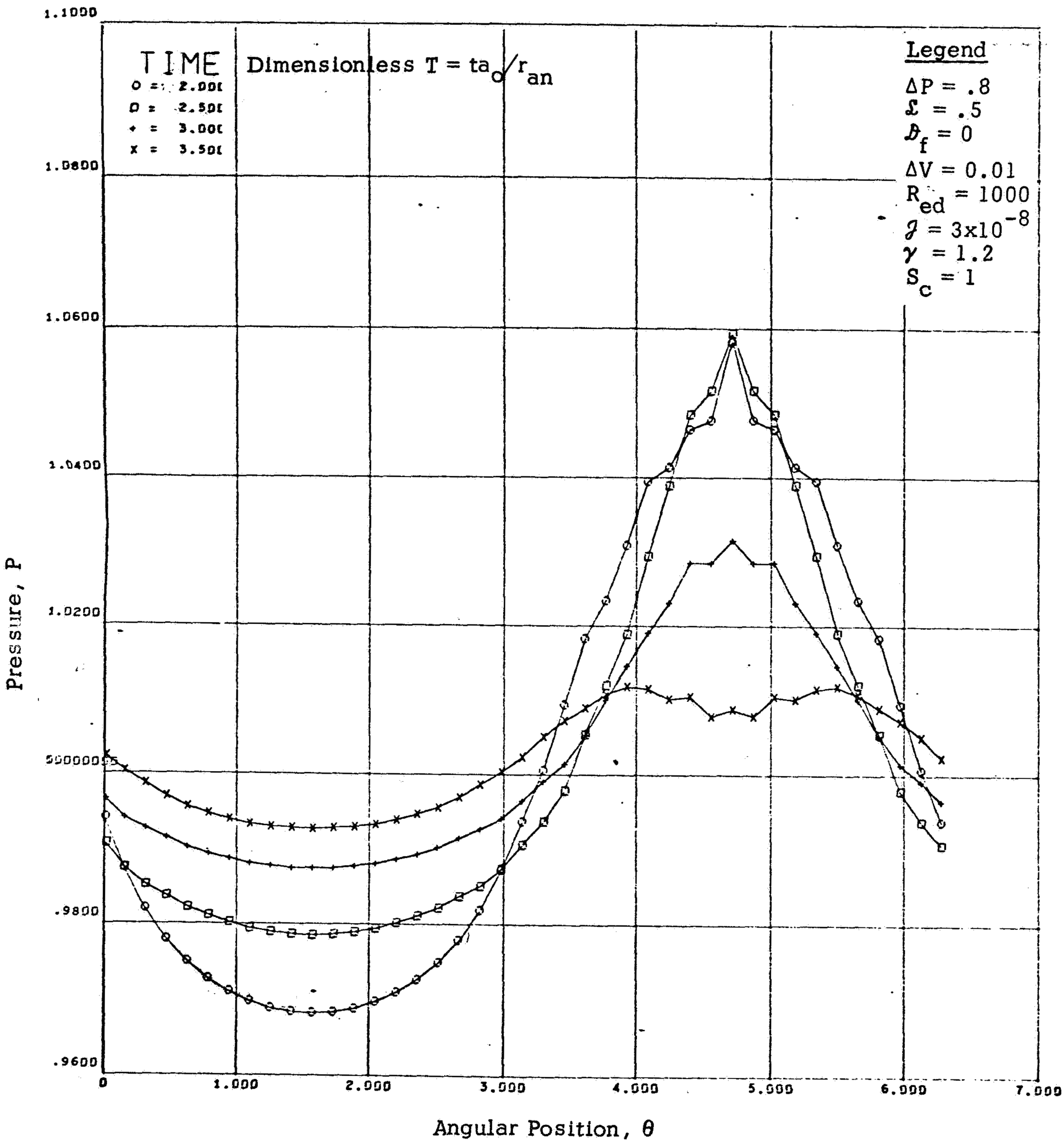

Figure 12. The Further Decrease in "Wavelet" Formation Due to ω interpolated at $V_\theta = 0$ nodes and the Improved Difference Operator.

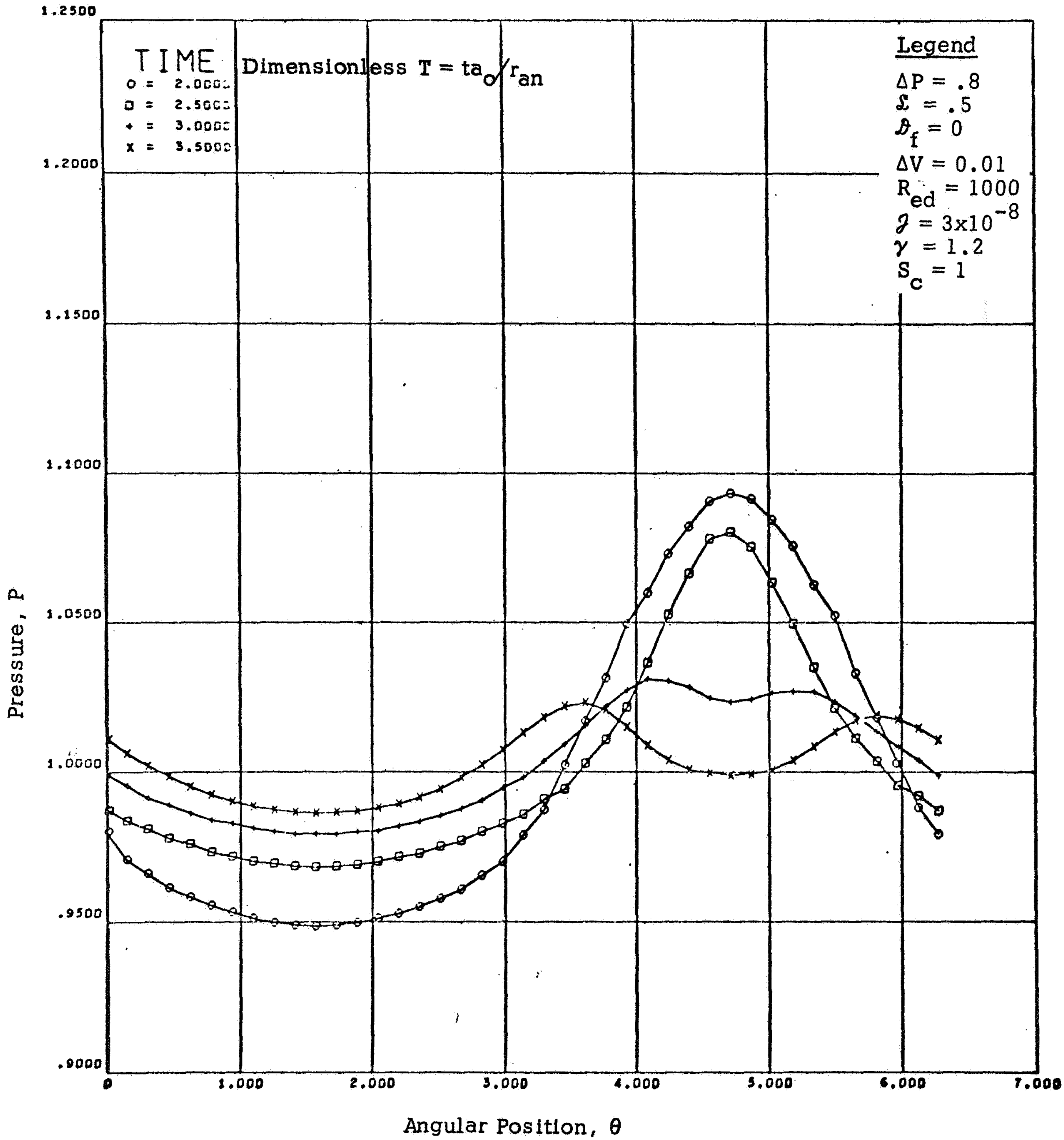

Figure 13. The Elimination of "Wavelet" for Motion due to Fully Implicit Integration Scheme.

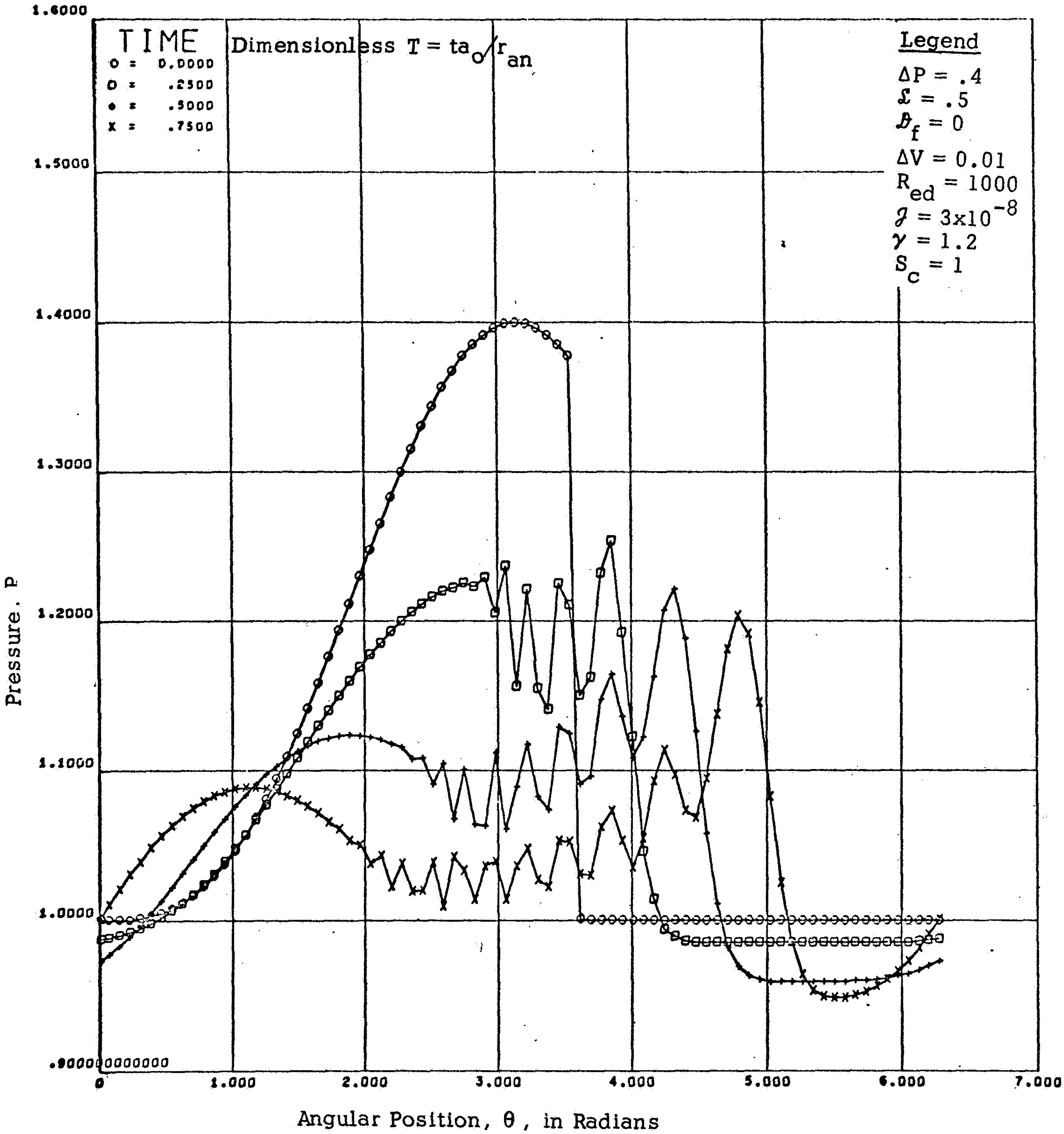

Figure 14a. Time History of Discontinuous Pressure Disturbance (time = 0-0.75).

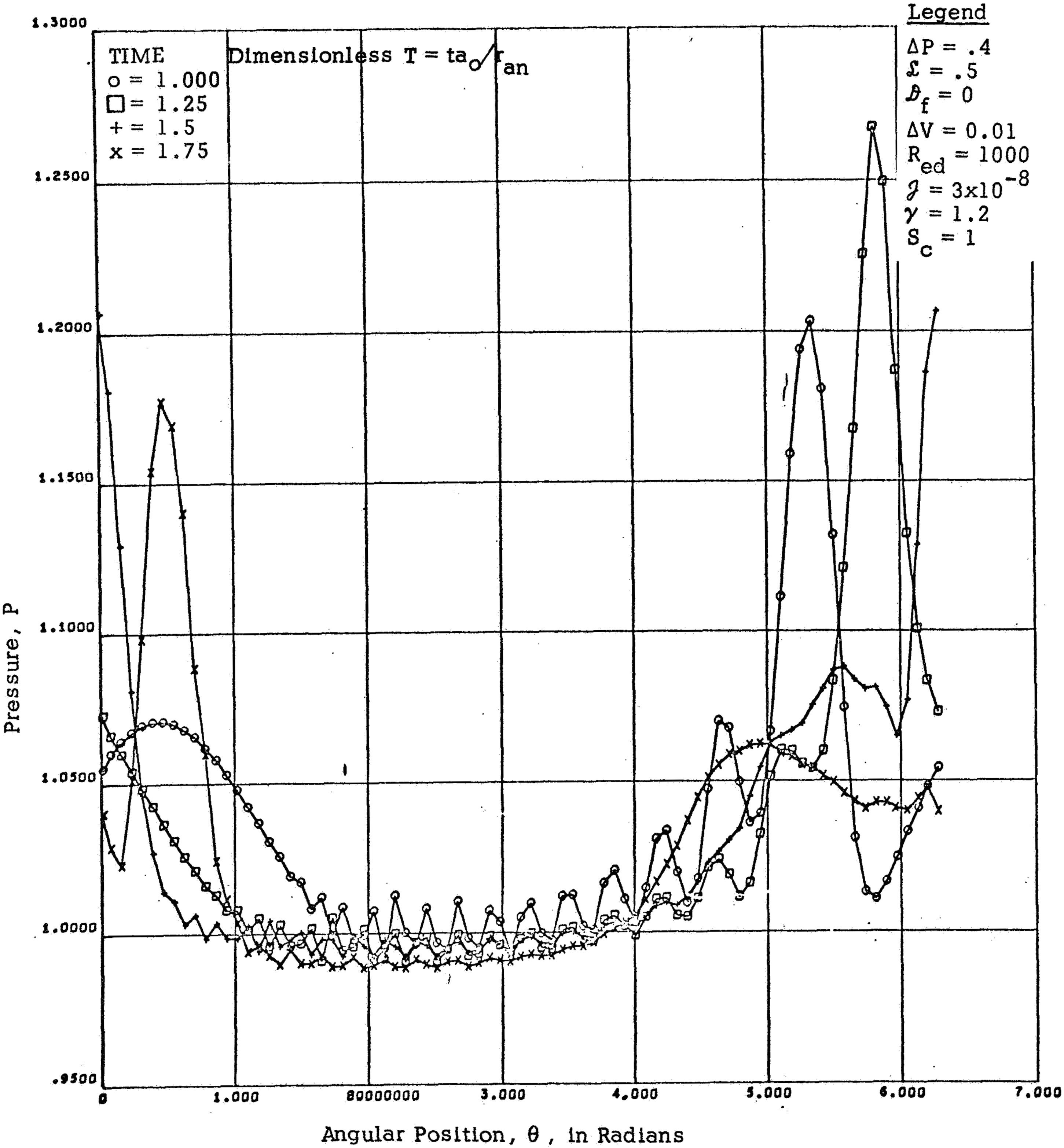

Figure 14b. Time History of a Discontinuous Pressure Disturbance (time = 1.0-1.75).

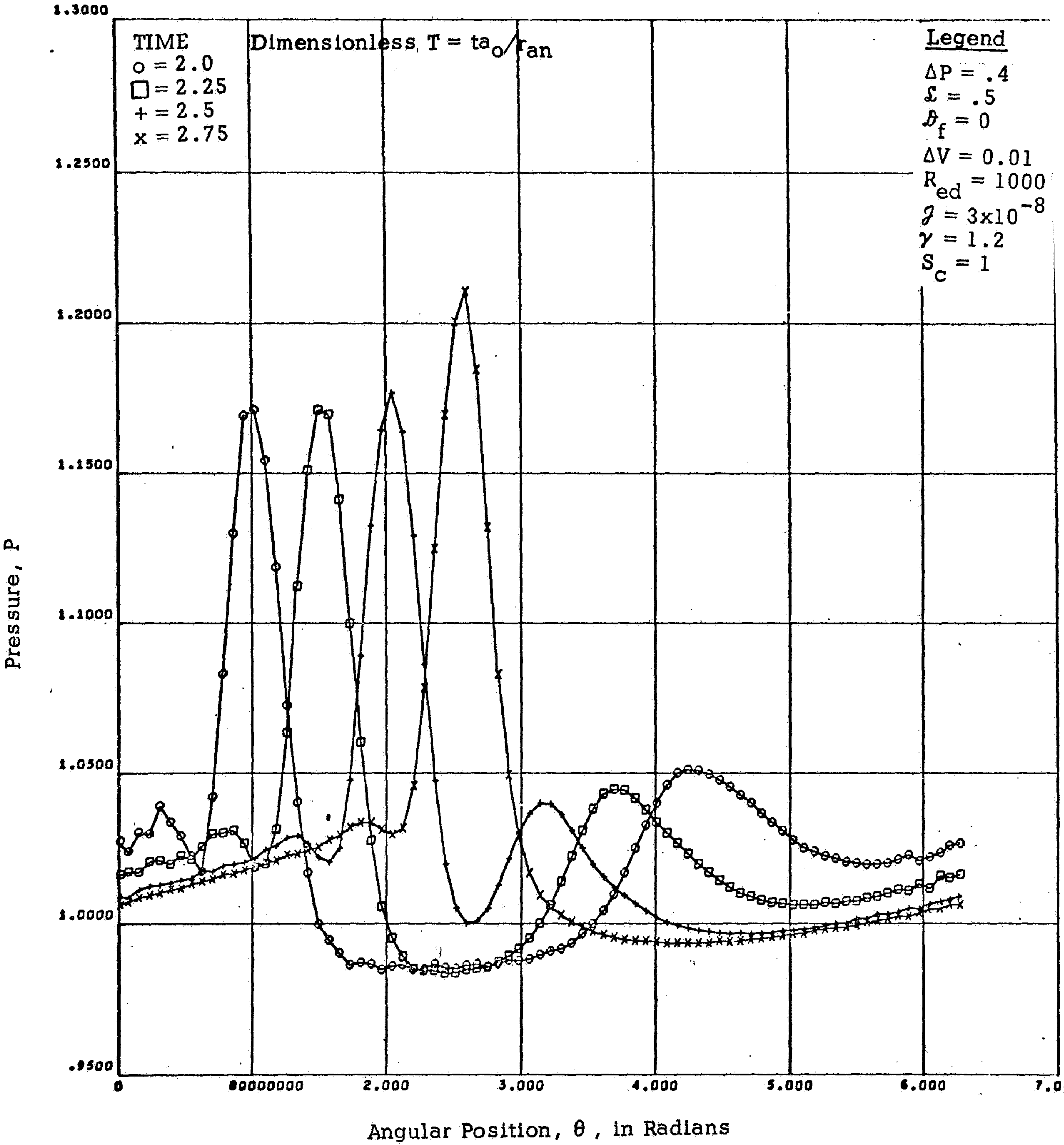

Figure 14c. Time History of a Discontinuous Pressure Disturbance (time=2.0-2.75).

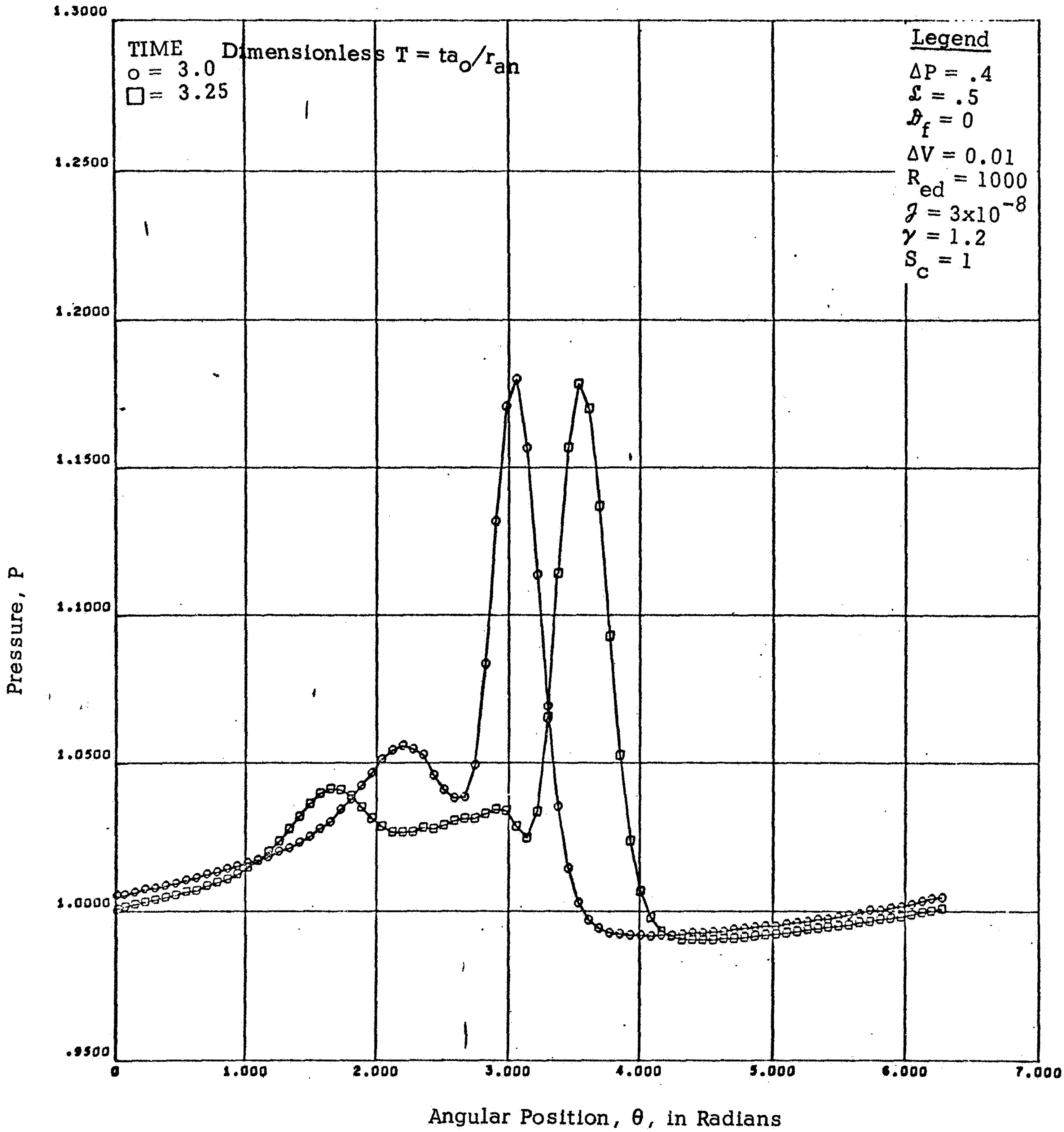

Figure 14d. Time History of a Discontinuous Pressure Disturbance (time =3.0-3.25).

APPENDIX B

COMPUTER PROGRAM

Statement of Work

The implicit integration scheme chosen for incorporation into the computer program is of the Crank-Nicolson type. This scheme was chosen for the following reasons:

- the scheme is implicit.
- it is a second order method, i.e., the truncation error is of $O(\Delta t^2) + O(\Delta\theta^2)$.
- the stability and applicability of the method in solving parabolic partial differential equations in general has been well demonstrated.
- the ease and speed in which the method could be implemented into the program and applicability to arbitrary initial conditions.

The basis of the Crank-Nicolson type of integration scheme is that the values of the dependent variables at the next time step should depend on the time derivatives evaluated at one half the time step, i.e.,

$$f(\theta, t + \Delta t) = f(\theta, t) + \Delta t \frac{\partial\left(f(\theta, t + \frac{\Delta t}{2})\right)}{\partial t}$$

The above equation becomes second order in truncation error when compared to a Talyor series expansion if the difference operator for the partial derivatives in time is

$$\frac{\partial\left(f(\theta, t + \frac{\Delta t}{2})\right)}{\partial t} = \frac{1}{2}\,\frac{f(\theta, t + \Delta t/2) - f(\theta, t)}{\Delta t/2} + \frac{1}{2}\,\frac{f(\theta, t+\Delta t) - f(\theta, t + \frac{\Delta t}{2})}{\Delta t/2}$$

$$= \frac{f(\theta, t+\Delta t) - f(\theta, t)}{\Delta t}$$

and accordingly, the derivatives in the θ direction are also evaluated at time $t + \frac{\Delta t}{2}$. Thus the difference operators for the θ derivatives are:

$$\frac{\partial f_i}{\partial \theta} = \frac{\frac{\partial f_i^{k+1}}{\partial \theta} + \frac{\partial f_i^k}{\partial \theta}}{2} = \frac{f_{i+1}^{k+1} - f_{i-1}^{k+1} + f_{i+1}^k - f_{i-1}^k}{4\Delta\theta}$$

$$\frac{\partial^2 f_i}{\partial \theta^2} = \frac{\frac{\partial^2 f_i^{k+1}}{\partial \theta^2} + \frac{\partial^2 f_i^k}{\partial \theta^2}}{2} = \frac{f_{i+1}^{k+1} - 2f_i^{k+1} + f_{i-1}^{k+1} + f_{i+1}^k - 2f_i^k + f_{i-1}^k}{4\Delta\theta^2}$$

where the subscript i refers to the i^{th} node and the superscript k refers to the k^{th} time step.

Linear terms appearing in the equations are evaluated as

$$g_i = \bar{g}_i = \frac{{}^j g_i^{k+1} + g^k}{2}$$

where j refers to the j^{th} iterant.

Nonlinear terms are handled as either

$$g_i f_i = \frac{{}^j g_i^{k+1} \cdot {}^j f_i^{k+1} + g_i^k \cdot f_i^k}{2} \quad \text{or} \quad g_i \cdot f_i = \bar{g}_i \, \bar{f}_i$$

depending on the order of magnitude of the two terms.

In this type of integration scheme, it is desirable to solve for the values of the dependent variables at time $t + \Delta t$ directly. However, since the equations are nonlinear, the differenced equations result in a nonlinear algebraic system of equations. These equations may be solved by linearizing the set and iterating until the solution converges. After substitution of the difference operators into the partial differential equations as described in NASA CR 920 and solving for the values of ρ, V_θ, and T at time $t + \Delta t$, the linearized form of the equations are:

Notation

Notation for variables is the same as in NASA CR 920, Appendix C, except that the primes have been dropped for notational clarity.

Continuity

$$- \left\{ \frac{V_i^{k+1}}{4\Delta\theta} \right\} \rho_{i-1}^{k+1} + \left\{ \frac{1}{\Delta t} + \frac{1}{2} \left(\frac{\partial V_{\theta i}^{k+1}}{\partial \theta} + \frac{\partial V_Z^{k+1}}{\partial Z} \right) \right\} \rho_i^{k+1}$$

$$+ \left\{ \frac{V_{\theta_i}^{k+1}}{4\Delta\theta} \right\} \rho_{i+1}^{k+1} = \frac{\rho_i^k}{\Delta t} - \frac{\rho_i^k}{2} \left\{ \frac{\partial V_{\theta i}^k}{\partial \theta} + \frac{\partial V_Z^k}{\partial Z} \right\} - \frac{V_i^k}{2} \frac{\partial \rho_i^k}{\partial \theta}$$

$$- V_Z \overline{\frac{\partial \rho}{\partial Z}} + (\mathcal{L}_f \bar{\omega}_f + \mathcal{L}_{ox} \bar{\omega}_{ox}) f(\gamma)$$

Momentum

$$\left\{ - \frac{\rho_i^{k+1} V_{\theta i}^{k+1}}{4\Delta\theta} - \frac{2}{3} \mathcal{J} \frac{f(\gamma)}{\Delta\theta^2} \right\} V_{\theta_{i-1}}^{k+1}$$

$$+ \left\{ \frac{\bar{\rho}_i}{\Delta t} + \frac{4}{3} \mathcal{J} \frac{f(\gamma)}{\Delta\theta^2} + \frac{\mathcal{L}_f \omega_f^{k+1} + \mathcal{L}_{ox} \omega_{ox}^{k+1}}{2} f(\gamma) \right.$$

$$\left. + \frac{1}{2} \left(\frac{D_f}{r_{d,f}} + \frac{D_{ox}}{r_{d,ox}} \right) \rho_i^{k+1} \left| V_{\theta_i}^{k+1} \right| \right\} V_{\theta_i}^{k+1}$$

$$+ \left\{ \frac{\rho_i^{k+1} V_{\theta i}^{k+1}}{4\Delta\theta} - \frac{2}{3} \mathcal{J} \frac{f(\gamma)}{\Delta\theta^2} \right\} V_{i+1}^{k+1} =$$

$$= \frac{\bar{\rho}_i V_{\theta i}^k}{\Delta t} - \frac{\rho_i^k V_{\theta i}^k}{2} \frac{\partial V_i^k}{\partial \theta} - \frac{1}{\gamma} \frac{\partial \bar{P}}{\partial \theta}$$

$$+ \frac{2}{3} \mathcal{J} f(\gamma) \frac{\partial^2 V_{\theta i}^k}{\partial \theta^2} - \frac{V_{\theta i}^k}{2} \left\{ \mathcal{L}_f \omega_f^k + \mathcal{L}_{ox} \omega_{ox}^k \right\} f(\gamma)$$

$$- \frac{1}{2} \left(\frac{D_f}{r_{d,f}} + \frac{D_{ox}}{r_{d,ox}} \right) \rho_i^k \left| V_{\theta_i}^k \right| V_{\theta_i}^k$$

Energy

$$\left\{-\frac{\rho_i^{k+1} V_{\theta i}^{k+1}}{4\Delta\theta} + \mathcal{J}\frac{f(\gamma)}{2\Delta\theta^2}\right\} T_{i-1}^{k+1}$$

$$+\left\{\frac{\bar{\rho}_i}{\Delta t} + \mathcal{J}\frac{f(\gamma)}{\Delta\theta^2} + \frac{\mathcal{L}_{ox}\,\omega_{ox}^{k+1} + \mathcal{L}_f\omega_f^{k+1}}{2} f(\gamma)\right\} T_i^{k+1}$$

$$+\left\{\frac{\rho_i^{k+1} V_{\theta i}^{k+1}}{4\Delta\theta} - \mathcal{J}\frac{f(\gamma)}{2\Delta\theta^2}\right\} T_{i+1}^{k+1} =$$

$$= \frac{\bar{\rho}_i T_i^k}{\Delta t} - \frac{\rho_i^k V_{\theta i}^k}{2}\frac{\partial T_i^k}{\partial\theta} - \frac{\rho_i^{k+1} V_Z}{2}\frac{\partial T_i^{k+1}}{\partial Z}$$

$$-\frac{\rho_i^k V_Z}{2}\frac{\partial T^k}{\partial Z} + \mathcal{J}\frac{f(\gamma)}{2}\frac{\partial^2 T_i^k}{\partial\theta^2}$$

$$-\frac{(\mathcal{L}_{ox}\omega_{ox}^k + \mathcal{L}_f\omega_f^k) f(\gamma) T_i^k}{2}$$

$$-|\gamma-1|\;\bar{P}_i\left(\frac{\partial\bar{V}_{\theta_i}}{\partial\theta} + \frac{\partial\bar{V}_Z}{\partial Z}\right)$$

$$+\frac{4}{3}|\gamma(\gamma-1)|\;\mathcal{J}\left[\frac{\partial\bar{V}_{\theta i}^2}{\partial\theta^2} + \frac{\partial\bar{V}_Z^2}{\partial Z} - \frac{\partial\bar{V}_{\theta i}}{\partial\theta}\frac{\partial\bar{V}_Z}{\partial Z}\right] f(\gamma)$$

$$+\mathcal{L}_t\,\bar{\omega}_c\, f(\gamma)\,\gamma,$$

$$+\frac{(\gamma-1)\gamma}{2} f(\gamma)\left\{\mathcal{L}_f\bar{\omega}_f[\bar{V}_{\theta_i}^2 + (V_Z - V_{f,Z})^2] + \mathcal{L}_{ox}\bar{\omega}_{ox}[\bar{V}_{\theta_i}^2 + (V_Z - V_{ox,Z})]^2\right\}$$

$$+\gamma(\gamma-1)\,\bar{\rho}_i\left\{D_f\frac{[\bar{V}_{\theta_i}^2 + (V_Z - V_{f,Z})^2]^{3/2}}{r_{d,f}} + D_{ox}\frac{[\bar{V}_{\theta_i}^2 + (V_Z - V_{Z,ox})^2]^{3/2}}{r_{d,ox}}\right\}$$

where the k+1 terms within the coefficients are evaluated at the previous iteration.

It should be noted that the resulting system of equations would be a tridiagonal system except for the first and last node. The system of equations is thus solved approximately as a tridiagonal set using Gaussian elimination by estimating the values of the variables at the first and last nodes.

Asymmetric Wave Input

The pressure pulse wave input shape was arbitrarily chosen to be

$$\frac{\Delta P}{P} = A + A_p \sin^{n}\left(\frac{\theta}{2}\right), \quad 0 \leq \theta \leq \pi$$

$$\frac{\Delta P}{P} = A + A_p \sin^{15n}\left(\frac{\theta}{2}\right), \quad \pi \leq \theta \leq 2\pi$$

where $$A = 1 - \int_0^{\pi} A_p \sin^{n}\left(\frac{\theta}{2}\right) d\theta - \int_0^{2\pi} A_p \sin^{15n}\left(\frac{\theta}{2}\right) d\theta$$

where A_p and n are input constants and A is a constant calculated so that there is zero mass and energy addition into the annulus. The resulting wave is a very steep fronted asymmetrical wave with continuous derivatives in the θ direction as shown in Figures 1 through 5.

Calculation of Gain and Response Factor

The following calculations were added to the computer program.

Gain

$$G_{\bar{\theta}}(t) = \sqrt{\frac{\int_0^{2\pi} \left(\frac{\omega(\theta) - \bar{\omega}}{\bar{\omega}}\right)^2 d\theta}{\int_0^{2\pi} \left(\frac{P(\theta) - \bar{P}}{\bar{P}}\right)^2 d\theta}}$$

Response Factor

$$N_{\bar{\theta}}(t) = \frac{\int_0^{2\pi} \left(\frac{\omega - \bar{\omega}}{\bar{\omega}}\right)\left(\frac{P - \bar{P}}{\bar{P}}\right) d\theta}{\int_0^{2\pi} \left(\frac{P - \bar{P}}{\bar{P}}\right)^2 d\theta}$$

Root Mean Square Burning Rate

$$\omega_{RMS,\theta} = \sqrt{\frac{\int_0^{2\pi} \left(\frac{\omega - \bar{\omega}}{\bar{\omega}}\right)^2 d\theta}{2\pi}}$$

Root Mean Square Pressure

$$P_{RMS,\theta} = \sqrt{\frac{\int_0^{2\pi} \left(\frac{P - \bar{P}}{\bar{P}}\right)^2 d\theta}{2\pi}}$$

Integral of the Response Factor

$$I(t) = \int_{t_o}^{t} N_{\bar{\theta}}(t)\, dt$$

The integral of the response factor was evaluated using the Trapezoidal rule. The other definite integrals were evaluated using Wedells rule.

The average response factor at any time, t, can be calculated as

$$N_{\bar{\theta},\tilde{t}}(t) = \frac{I(t) - I(t - \tau)}{\tau} \qquad \text{for all } t \geq \tau$$

once the period, τ, has been determined by inspection.

Program Input

The computer program input is standard FØRTRAN IV NAMELIST. Familiarity with this standard input procedure is assumed.

The input list of variables is as follows.

$DATA

AP	=	,	AMPLITUDE OF PRESSURE PULSE
DRAGF	=	,	$\mathcal{D}_f$
DRAGØ	=	,	$\mathcal{D}_{ox}$
GAM	=	,	γ
HMAX	=	,	TIME STEP
MP	=	,	MP = 2 - FUEL CONTROLLED BURNING MP = 4 - OXIDIZER CONTROLLED BURNING
ND	=	,	NØ ØF NØDES (must be multiple of 20 plus 1)
NØ	=	,	Print every NØth time step
NJ	=	,	Print every NJth NODE
REFD	=	,	$R_{e,f}$
REØD	=	,	$R_{e,ox}$
SC	=	,	Schmit No.
TI	=	,	Initial Time
TSTØP	=	,	Final Time
VFZ	=	,	$\Delta V_{Z,f}$
VØZ	=	,	$\Delta V_{Z,ox}$
VZ	=	,	V_Z
XJ	=	,	$\mathcal{J}$
XLF	=	,	$\mathcal{L}_f$
XLØ	=	,	$\mathcal{L}_{ox}$
IFPLØT(I)	=	,	Plot control for the I^{th} Plot

PLØT	if IFPLØT(I) =
P	1
ρ	2
T	3
V	4
ω_f	5
ω_{ox}	6
nothing	0

ITGEN	=		if ITGEN = 0	input plot times
			if ITGEN ≠ 0	plot times generated
TPLØT(I)	=	,	inputted plot times if ITGEN = 0	
TDELTA	=	,	time increment for plots if ITGEN ≠ 0	
TFIRST	=	,	starting time for plots if ITGEN ≠ 0	
TLAST	=	,	end time for plots if ITGEN ≠ 0	
EMIN	=	,	error tolerance for integration	
MAXIT	=	,	maximum no. of iterations per integration step	
VSTART	=	,	initial value of V_θ	
SYM	=	,	SYM = 0.0,	sine wave input
			SYM ≠ 0.0,	$\sin^{XEXP}$ (XK·θ) input
XEXP	=	,	exponent for asymmetric wave	
XK	=	,	XK = 1.0,	sine wave
			XK = 0.5,	asymmetric wave

$END

All program input parameters are dimensionless.

Program Description

The following paragraphs give a brief description of each subprogram in the computer program and the major common blocks. The descriptions are followed by a source language listing of the program.

MAIN PRØGRAM BIPRØP

Controls the calculation procedure and print out.

SUBRØUTINE REED

Reads data, writes out the header for each case, and sets up the plotting.

SUBRØUTINE RSET

Calculates constants and computes the initial data line for the sine wave.

SUBRØUTINE FIT

Calculates the burning rate at the V_θ=0 nodes using average parabolic interpolation

SUBRØUTINE SCALE

Determines the optimal scale for plotting

SUBRØUTINE MAXMIN

Determines the maximum and minimum values of a given array.

SUBRØUTINE ØRG

Sets the counters for printed control.

SUBRØUTINE SETMAP

Sets up limits of calculations, determines maximum pressure node, and calculates the initial data line for the asymmetric wave.

SUBRØUTINE ASET

Calculates the burning rates and the coefficients used in evaluating the Z derivatives.

FUNCTIØN WEDS

Definite integral evaluator using Wedells rule.

SUBRØUTINE ZDIR

Calculates the Z derivatives.

SUBRØUTINE ABCD

Calculates the coefficients in the band matrices.

SUBRØUTINE DEINT

Solves for the values of ρ, V_θ, and T at the next time step.

SUBRØUTINE INVTRI

Linear equation solver for tridiagonal sets of equations.

SUBRØUTINE MAPPER

Calculates auxiliary variables and maps some variables depending on the type of initial wave used.

SUBRØUTINE BARRED

Calculates average values.

SUBRØUTINE AUXCAL

Calculates the gain, response factor, integral of the response factor, and traps certain variables for printing and plotting.

SUBRØUTINE PLTSUB

Controls the SC4020 plotting.

SUBRØUTINE SECØND

Machine clock routine.

SUBRØUTINE SHIFT

Shifts and updates index pointers. Also checks on number of iterations.

SUBRØUTINE PURGE

Purges the plot buffers.

The major common regions used in the program are:

COMMON REGION: CØMMØN/INDEX/

VARIABLE NAME	VARIABLE DESCRIPTION
NT	pointer to the column in three-dimensional arrays in which values of variables at time t are stored.
NTP	pointer to the column in three-dimensional arrays in which values of variables at time $t+\Delta t$ are stored.
NTB	pointer to the column in three-dimensional arrays in which averaged values of variables are stored.
KTP	pointer to the column in two-dimensional arrays in which values of variables at time $t+\Delta t$ are stored.
KTB	pointer to the column in two-dimensional arrays in which averaged values of variables are stored.
KT	pointer to the column in two-dimensional arrays in which values of variables at time t are stored.
NCØN	pointer to the column in which the control burning rate values are stored.

COMMON REGION: CØMMØN/B/

VARIABLE	DIMENSIØN	DESCRIPTIØN
T	181,3	T
RHØ	181,3	ρ
V	181,3	V_θ
P	181,2	P
DVTH	181,2	$\frac{\partial V_\theta}{\partial \theta}$
DPTH	181,2	$\frac{\partial P}{\partial \theta}$
WØX	181,2	ωox
WF	181,2	ωf
VSQ	181,2	V_{θ^2}
DTTH	181,2	$\frac{\partial T}{\partial \theta}$
WP	181	$(\mathcal{L}ox\ \omega ox^{k+1} + \mathcal{L}_f \omega_f^{k+1})\ f(\gamma)$

COMMON REGION: CØMMØN/CØNST/

VARIABLE	DESCRIPTIØN
CØN(1)	$\Delta t/2$
CØN(2)	$1/4\Delta\theta$
CØN(3)	$\Delta t/4\Delta\theta$
CØN(4)	$\mathcal{G}f(\gamma)/\Delta\theta^2$
CØN(5)	$2/3\mathcal{G}f(\gamma)\Delta\theta^2$
CØN(6)	$4/3\mathcal{G}f(\gamma)/\Delta\theta^2$
CØN(7)	$1/2\mathcal{G}\cdot f(\gamma)/\Delta\theta^2$
CØN(8)	$\mathcal{L}\cdot\gamma\cdot f(\gamma)$
CØN(9)	$\mathcal{L}_f\cdot f(\gamma)/2$
CØN(10)	$\mathcal{L}_{ox}\cdot f(\gamma)/2$
CØN(11)	$(\mathcal{D}_{ox}+\mathcal{D}_f)/2$
CØN(12)	$\gamma - 1$
CØN(13)	$4/3\ \mathcal{G}f(\gamma)\cdot\gamma\cdot(\gamma-1)$
CØN(14)	$1/8\Delta\theta$
CØN(15)	$1/\Delta t$
CØN(16)	$1/2\Delta\theta$

COMMON REGION CØMMØN/E/

Cell	Description	Cell	Description	Cell	Description
1	Initial Time (TI)	41	v_z (VZ)	61	(SCR)
2	Not Used	42	$f(\gamma)$ (FGAM)	62	$\gamma(\gamma-1)/2.0$ (SIP)
3	Not Used	43	$a_1 = 2\pi$ [A(1)]	63	$\gamma(\gamma-1)$ (SIP2)
4	Not Used	44	$a_2 = 2\pi v_z$ [A(2)]	64	Not Used
5	Not Used	45	Not Used	65	$\mathcal{J}*f(\gamma)$ (BC)
6	Max Step (HMAX)	46	$a_4 = 2\pi v_z$ [A(4)]	66	$4/3*\mathcal{J}f(\gamma)$ (BZ)
7	Not Used	47	$a_5 = \frac{1}{\gamma}\int_0^{2\pi} T d\theta$ [A(5)]	67	$Re_{d,f}$ (REFO)
8	Min Error (EMIN)	48	$a_6 = 2\pi/\gamma$ [A(6)]	68	$Re_{d,ox}$ (REFD)
9	Not Used	49	$a_7 = (\gamma-1)\int_0^{2\pi} p d\theta$	69	$\left(Re_{d,ox}\right)^{1/2}$ (SRD2)
10	Not Used	50	$a_8 = 2\pi v_z$	70	2(DTH) (D2)
11	Not Used	51	c_1 [C(1)]	71	$(DTH)^2$ (DSQ)
30	Not Used	52	c_2 [C(2)]	72	$(\Delta v_{ox,z})^2$ (DELIV)
31	Time Stop (TSTOP)	53	c_3 [C(3)]	73	Not Used
32	$\mathcal{L} = \mathcal{L}_f + \mathcal{L}_{ox}$ (XL)	54	Theta Step (DTH)	74	$\mathcal{L}_f$ (XLF)
33	Not Used	55	$\partial v/\partial z$ [X(1)]	75	$\mathcal{L}_{ox}$ (XLO)
34	Parameter, $\mathcal{J}$(XJ)	56	$\partial \rho/\partial z$ [X(2)]	76	$\Delta v_{f,z}$ (VFZ)
35	Not Used	57	$\partial T/\partial z$ [X(3)]	77	$\Delta v_{ox,z}$ (VOZ)
36	γ (GAM)	58	Not Used	78	$\mathcal{D}_f$ (DRAGF)
37	Init. Pres.Dist. (AP)	59	$(Re_{d,f})^{1/2}$ (SRD)	79	$\mathcal{D}_{ox}$ (DRAGO)
38	Not Used	60	.6*Sc**1/3 (SCB)	80	(SCRO)
39	Schmidt No. (SC)				
40	$(\Delta v_{f,z})^2$ (DEL2V)				

```
      PROGRAM BIPROP(INPUT,OUTPUT,PUNCH,TAPE5=INPUT,TAPE6=OUTPUT,
     1  TAPE8=PUNCH,TAPE48)
C     MAIN PROGRAM FOR COMBUSTION INSTABILITY MODEL OF DSC
C
      COMMON/GAINS/ XNSUM,RMSW,RMSP, GAIN, RESPON,NCYCLE,XNAVE,NCNT
     *       ,NSTEPS, XNSAVE(600)
      COMMON/FLGPLT/ PLOTFG
      COMMON /DEBUG/ IDEBUG
      COMMON /CNM/    IM, INDR,    IU, MALP, MGAM,    MP, MPTN, MPTS
     1               ,NCD,    ND,    NJ,    NO,  NOD, NOD1,    NZ,    N1
      COMMON/E/ FC(80)
      COMMON/CPLT/ ART(800,3)
      COMMON/B/      T(181,3) ,          RHO(181,3) ,          V(181,3) ,
     1               P(181,2) ,         DVTH(181,2) ,       DPTH(181,2) ,
     2             WOX(181,2) ,           WF(181,2) ,        VSQ(181,2) ,
     3            DTTH(181,2) ,           WP(  181)
      COMMON/INDEX/NT,NTP,NTB,KTP,KTB,KT,NCON
      DIMENSION F(200,3)
      EQUIVALENCE ( F(1,1),XNSAVE(1) )
      EQUIVALENCE
     1   (FC( 1), TI   ),      (FC( 3), H   ),       (FC(31), TSTOP)
     2  ,(FC(32), XL   ),      (FC(33), RED),        (FC(34), KJ    )
     3  ,(FC(35), DELV),       (FC(36), GAM),        (FC(38), ZIP   )
     4  ,(FC(39), SC   ),      (FC(74), TL  ),       (FC(75), TH    )
C
C     IPLCT IS CURRENT NO OF PLOTS RESIDING IN BUFFER FCTPLT(I,J,K)
      IPLCT = 0
      IZ = 1
    5 IU = 1
      NN = 1
C     IPLCT IS CURRENT NO OF PLOTS RESIDING IN BUFFER FCTPLT(I,J,K)
      IPLCT = 0
      CALL REED
C
C               SET  BEGINNING  INDEX POINTERS
C
      NT=1
      NTP=2
      NTB=1
      KTP=2
      KT=1
      KTB=1
C
C                    N  IS FOR 3 D  ARRAYS  AND K  FOR  2 D  ARRAYS
C                    THE    T  IS  FOR  TIME = T
C                    THE    TP IS  FOR  TIME = T + DT
C                    THE    TB IS  FOR  BARRED VALUES
C
```

```
      CALL RSET
         CALL ORG
      CALL SETMAP
 10      CALL ASET
         CALL  ZDIR
      CALL ABCD(NT,KT)
      NTB= MOD(NT+1,3) +1
      NTP= MOD(NT,3)+1
      KTB=KTP

      CALL DEINT(NT )
      CALL MAPPER(NTB,KTB)
      NSAVE= NTP
      NTP=NTB
      NTB=NSAVE
      GO TO  30
 20   CALL ASET
      CALL ZDIR
      CALL BARRED
      CALL ABCD(NTP,KTP)
      CALL DEINT(NTP)
      CALL MAPPER(NTB,KTB)
      GO TO 70
C
 30   CALL AJXCAL
      IF( PLOTFG,NE, 0.0) CALL PLTSUB(IPLOT)
C
C
C     TEST FOR PRINT STATION
      IF (MPTN) 60,50,60
C     BRANCH TO 50 IMPLIES PRINT POINT OBTAINED
 50   CALL SECOND(TSEC)
      WRITE(6,970) TSEC
 970  FORMAT(30X,4HSEC=F8,3)
      WRITE(6,97) ART(IU,1), ART(IU,2), IM, (FC(JM),JM=55,57)
     *     ,RESPON,XNAVE, GAIN, RMSW, RMSP
 97   FORMAT (6H0TIME=F9,5 ,10X,17H(PMAX-PMIN)/PAVE=F9,5,10X,3HIT=I2
     1 ,//,20X,42HAXIAL DERIVATIVES FOR V,RHO, AND T ......., ,/ ,
     2 ,17X,3E16,7 ,//
     *      , 8X, 16HRESPONSE FACTOR=,E17,8,5X,
     *  21HAVE, RESPONSE FACTOR=,E17,8 /
     *  8X,5HGAIN=,E17,8,5X,6HW RMS=,E17,8,5X,
     *  6HP RMS=,E17,8//
     *                        ,8X,1HP,17X,3HRHO,15X,1HT,17X
     3 ,7HV THETA,11X,6HW FUEL,12X,5HW OX, ,/)
      WRITE(6,98)(P(J,KT),RHO(J,NT),T(J,NT),V(J,NT),WF(J,KT),
     1                 WOX(J,KT), J=1,ND,NJ)
 98   FORMAT (F14,5,2F18,5,E22,5,F14,5,F18,5)
      WRITE (6,980)
 980  FORMAT(1H0,///)
 52   IU = IJ +1
```

```
C      TEST FOR  TIME STOP,
                  IF (TI - TSTOP)  60,  55, 55
C      STORE PRINT POINT FOR  PLOTTING
 55    ZIP = ART(1,2)
       IU = IJ - 1
       NN = NN-1
       WRITE(6,990)
 990   FORMAT(1H1,22X,12HTIME HISTORY////)
       WRITE(6,99) (ART(J,1),ART(J,2),ART(J,3),F(J,1),F(J,2),F(J,3),
      *      J=1,IU)
 99    FORMAT(1H0,6X,4HTIME,15X,16H(PMAX-PMIN)/PAVE,4X,8HPRESSURE
      1  ,10X,4HGAIN,16X,15HRESPONSE FACTOR,3X,
      2 27HINTEGRAL OF RESPONSE FACTOR//,(6E20,8))
C          PUNCH CARDS FOR PLTPK
       WRITE(8,905)IU
       WRITE(8,915)(ART(J,1),J=1,IU)
 905   FORMAT(5X, 4HTIME/5X, 4HGAIN/5X,4HGAIN/1X, 5H$DATA/1X, 5HNPTS=,
      1     I6,1H,/1X, 5HX(1)=)
 915   FORMAT(1X,4( E14,6,1H,))
       WRITE(8,920)
 920   FORMAT(1X, 5HY(1)=)
       WRITE(8,915) (F(J,1),J=1,IU )
       WRITE(8,925)
 925   FORMAT(1X, 4H$END)
       WRITE(8,930)
 930   FORMAT(5X, 4HTIME/5X,1HN/3X, *RESPONSE FACTOR*/1X, *$DATA*/1X,
      1  5HY(1)=)
       WRITE(8,915) ( F(J,2),J=1,IU)
       WRITE(8,925)
       IF(PLOTFG,EQ, 0,0) GO TO 5
C      CAN CALL PURGE DIRECTLY SINCE PURGE(1) HAS SPECIAL FUNCTION
       CALL PURGE (1)
           GO  TO 5
C      SUBROUTINE  SHIFT  UPDATES TERMS INVOLVED WITH INTEGRATION=
  60   MGAM= +1
       GO TO  20
  70   CALL SHIFT
       IF (MGAM)10,20,20
C
       END
```

```
      SUBROUTINE REED
C
      COMMON/FLGPLT/ PLOTFG
      COMMON /CNM/   IM, INDR,   IU, MALP, MGAM,   MP, MPTN, MPTS
     1              ,NCD,   ND,   NJ,   NO,  NOD, NOD1,   NZ,   N1
      COMMON/ITER/ MAXIT
      COMMON /DEBUG/ IDEBUG
      COMMON/E/ FC(80)
      COMMON/MESS/SYM
      COMMON/B/     T(181,3) ,         RHO(181,3) ,           V(181,3) ,
     1              P(181,2) ,        DVTH(181,2) ,        DPTH(181,2) ,
     2            WOX(181,2) ,          WF(181,2) ,         VSQ(181,2) ,
     3           DTTH(181,2) ,          WP(  181)
      COMMON/LAW/NFLAG,XK,XEXP
      DIMENSION  IPRNT(2)
      COMMON/INDEX/NT,NTP,NTB,KTP,KTB,KT,NCON
      COMMON/MARKET/ NPATH,VSTART,NPMAX
      COMMON /PLINFO/ FCTPLT(181,6,4), TITLE(5)
     1 , IFPLOT(6), TIMEP(4), THETAP(181), YT(6), YB(6), TPLOT(50)
     2,DELX , IPLOT, NLPHGH, NLPHGV, NLTHGH, NLTHGV, NPLPF
     3 , NPLT, NPHPLT, XLEFT, XRIGHT, TIMAX, IPLMAX
C
      DIMENSION  WT(20), BCD(12)
C
      EQUIVALENCE
     1   (FC( 1), TI    ),    (FC( 6), HMAX ),    (FC( 8), EMIN )
     2  ,(FC( 9), EMAX  ),    (FC(11), WT(1)),    (FC(31), TSTOP)
     3  ,(FC(32), XL    ),    (FC(33), RED  ),    (FC(34), XJ   )
     4  ,(FC(35), DELV  ),    (FC(36), GAM  ),    (FC(37), AP   )
     5  ,(FC(39), SC    ),    (FC(41), VZ   ),    (FC(54), DTH  )
     6  ,(FC(67), REFD  ),    (FC(68), REOD ),    (FC(74), XLF  )
     7  ,(FC(75), XLO   ),    (FC(76), VFZ  ),    (FC(77), VOZ  )
     8  ,(FC(78), DRAGF),     (FC(79), DRAGO)
      DATA IPRNT/10H   FUEL    ,10H OXIDIZER /
      NAMELIST /DATA/    AP, DRAGF, DRAGO, GAM, HMAX, MP, ND, NO, NJ
     1, REFD, REOD, SC, TI,TSTOP, VFZ, VOZ, VZ, XJ, XLF, XLO
     *, DEL, DELT, IFPLOT, ITGEN, NLPHGV, NLPHGH, NLTHGV, NLTHGH
     *, NPLPF, NPHPLT, NTHPLT, TPLOT
     *, TFIRST, TLAST, TDELTA
     *, TIMAX, IPLMAX
     *, IDEBUG
     *    ,EMIN,MAXIT
     *   ,XK,XEXP
     *   ,VSTART
     *    ,SYM
C
      VSTART=0.0
      IFPLOT(1)=0
      NPP = 50
```

```
      NTHPLT = 0
      NPLPF=4
      NPHPLT = 0
      DELX=0,0001
      SYM=0,0
      ITGEN =1
      MAXIT=4
      EMAX=,01
      EMIN=,0001
      NLPHGV = 10
      NLPHGH = 10
      NLTHGV = 7
      NLTHGH = 7
      IDEBUG = 1
      TI=0.0
      TFIRST=0,0
      XK=1,0
      XEXP=1,0
      NFLAG=0
      READ(5,90) (TITLE(I),I=1,5)
90    FORMAT(5A10)
      IF( EOF, 5) 99,1
1     READ(5,DATA)
      WRITE(6,91) (TITLE(I),I=1,5)
91    FORMAT (1H1,38X,41HDYNAMIC SCIENCE BIPROPELLANT INSTABILITY
     1 ,7HPROGRAM /// 5A10)
      XL=XLF+XLO
      IF(XK,NE, 1,0) NFLAG=1,0
      IF( XK,NE, 1,0 ) XK=0,5
      ND=((ND-1)/5)*5+1
      XND = ND - 1
      DTH = 6,2831853071/XND
      TSTS=HMAX/DTH
      WRITE (6,93) AP,XLF,XLO,REFD,REOD,VFZ,VOZ,DRAGF,DRAGO
     1 ,XJ,GAM,SC,VZ,TI, HMAX, TSTOP, DTH, TSTS
93    FORMAT(//,47H0INITIAL AMPLITUDE OF PRESSURE DISTURBANCE,AP =
     1 ,F9,6,//////,51X,20HSTABILITY PARAMETERS,//,51X,4HFUEL,8X
     2 ,8HOXIDIZER,///,4X,24HBURNING-RATE PARAMETER,L ,21X,F9,3,
     3 ,3X,F9,3,//,4X, 8HRE SUB D,37X,F9,0,3X,F9,0,//
     4 ,4X,26HRELATIVE VELOCITY, DELTA V,19X,F9,4,3X,F9,4,//
     5 ,4X,16HDRAG PARAMETER,D ,29X,F9,2,3X,F9,2 ,//////////
     6 ,13X,3HJ =,E12,5,/,9X,7HGAMMA =,F7,4,/,16H   SCHMIDT NO, =
     7 ,F7,4,/,7X,9HV SUB Z =,F9,6,//////
     8 ,15H INITIAL TIME =,F9,6,20X,12HTIME  STEP =,F9,6,/
     9 ,15H    FINAL TIME =,F9,6,20X,12HTHETA STEP =,F9,6,/
     1 ,34X            ,22HTIME STEP/THETA STEP =,F9,6,//)
C
C          MP=2 FUEL      OR      MP=4 OX,     CONTROLS BURNING
C
```

```
      MMP= MP/2
      WRITE(6,94)ND,IPRNT(MMP),NJ,NO
 94   FORMAT(4X,14HNO, OF NODES =,I4/ 4X,A10,17HCONTROLED BURNING//
     *       20X,13HPRINT CONTROL/4X,44HPRINT EVERY NJ TH NODE EVERY NO
     *TH TIME STEP/4X,10H,WHERE NJ=,I4,3X, 8HAND NO =,I4)
      WRITE(6,95)
 95   FORMAT(1H1)
      NPLT = 0
      IF (NTHPLT) 30,26,30
C
C     IF THETA PLOTTING HAS BEEN REQUESTED, CHECK RANGE OF REQUESTED
C     FUNCTIONS  1=P, 2=RHO, 3=T, 4=V, 5=WF, 6=WOX
C     FOR COMPUTED GO TO MUST HAVE LIMITS 1 TO 6 INCLUSIVE
C
 26   DO 27 I = 1,6
      IF (IFPLOT(I),EQ,0) GO TO 28
      IF((IFPLOT(I),LT,1),OR,(IFPLOT(I),GT,6)) GO TO 29
      NPLT = NPLT + 1
 27   CONTINUE
 28   IF(NPLT,GT,0) GO TO 30
 29   WRITE(6,200)
 200  FORMAT(53H1THETA GRAPHS DELETED DUE TO INPUT ERROR IN IFPLOT(I))
      NPLT = 0
 30   CONTINUE
C
      IF (NPLT,EQ,0) GO TO 40
      IF(ITGEN) 35,31,35
 31   IF (TFIRST,LT,TI) TFIRST = TI
      TPLOT(1) = TFIRST
 32   DO 34 I=2,NPP
      TEMP = TPLOT(I-1) + TDELTA
      IF (TEMP,LT,TLAST) GO TO 321
      TPLOT(I) = TLAST
 320  IPLMAX = I
      GO TO 35
 321  IF (TEMP,LT.TSTOP) GO TO 33
      TPLOT(I) = TSTOP
      GO TO 320
 33   TPLOT(I) = TEMP
      IPLMAX = I
 34   CONTINUE
 35   CONTINUE
      IF(IDEBUG,EQ,0) WRITE(6,100) ( I,TPLOT(I),I=1,IPLMAX)
 100  FORMAT(1H1,10HPLOT TABLE// (I5,E20,8)/)
      IPLOT = 1
C
C     GENERATE FUNCTION THETAP(361)
      THETAP(1) = 0
      DO 15 IX = 2,ND
      THETAP(IX) = THETAP(IX-1) + DTH
```

```
 15   CONTINUE
      IF(IDEBUG.EQ.0) WRITE(6,101) (I,THETAP(I),I=1,ND)
 101  FORMAT(1H1,16HTHETA(RAD) TABLE// (I5,E20.8))
 40   CONTINUE
      PLOTFG=NPLT
      RETURN
C
 99   WRITE(6,210)
 210  FORMAT(16H0STOPPED IN REED)
      STOP
      END
```

```
          SUBROUTINE   RSET
C
      COMMON/B/     T(181,3) ,          RHO(181,3) ,           V(181,3) ,
     1              P(181,2) ,         DVTH(181,2) ,        DPTH(181,2) ,
     2            WOX(181,2) ,           WF(181,2) ,         VSQ(181,2) ,
     3           DTTH(181,2) ,           WP(  181)
      COMMON/MARKET/ NPATH,VSTART,NPMAX
      COMMON/CONST/CON(20)
      COMMON /CNM/    IM, INDR,    IU, MALP, MGAM,    MP, MPTN, MPTS
     1                ,NCD,    ND,    NJ,    NO,   NOD, NOD1,    NZ,    N1
      COMMON/LAW/NFLAG,XK,XEXP
      COMMON/INDEX/NT,NTP,NTB,KTP,KTB,KT,NCON
      COMMON/E/ FC(80)
      COMMON/NODES/ NCRIT(2)
      COMMON/GAINS/ XNSUM,RMSW,RMSP, GAIN, RESPON,NCYCLE,XNAVE,NCNT
     *        ,NSTEPS, XNSAVE(600)
C
      DIMENSION A(11)
      EQUIVALENCE ( FC(6), HMAX)      ,(FC(31), TSTOP)
      EQUIVALENCE                  (FC(32), XL     ),    (FC(33), RED    )
     1,   (FC(34), XJ     ),     (FC(35), DELV   ),    (FC(36), GAM    )
     2,   (FC(37), AP     ),     (FC(38), ZIP    ),    (FC(39), SC     )
     3,   (FC(40), DEL2V  ),     (FC(41), VZ     ),    (FC(42), FGAM   )
     4,   (FC(43), A      ),     (FC(54), DTH    ),    (FC(59), SRD    )
     5,   (FC(60), SCB    ),     (FC(61), SCR    ),    (FC(62), SIP    )
     6,   (FC(63), SIP2   ),     (FC(64), BB     ),    (FC(65), BC     )
     7,   (FC(66), BZ     ),     (FC(67), REFD   ),    (FC(68), REOD   )
     8,   (FC(69), SRD2   ),     (FC(70), D2     ),    (FC(71), DSQ    )
     9,   (FC(72), DEL1V  ),     (FC(76), VFZ    ),    (FC(77), VOZ    )
     A,   (FC(78), DRAGF  ),     (FC(79), DRAGO  ),    (FC(80), SCRO   )
     *    ,(FC(74),XLF)     ,(FC(75),XLO)
C
C     SET UP  INITIAL ARRAY
C
      A(1)=6.28318531
      A(2)=A(1)*VZ
      A(4)=A(2)
      A(6)=A(1)/GAM
      A(8)=A(2)
```

```
C
      XNSAVE(1)=0,0
      XNSUM=0,0
      NCYCLE=A(1)/HMAX +1,0
      NCNT=0
      NSTEPS=TSTOP/HMAX +1
         GAM1 = GAM + 1,0
         FGAM = SQRT ((2,0/GAM1)**(GAM1/(GAM-1,0)))
               BC = XJ*FGAM
               BZ = 1,3333333333*BC
         DEL2V = VFZ**2
         DEL1V = VOZ**2
         SIP2=  GAM*(GAM - 1,0)
        SIP = SIP2/2,0
         ZIP =  GAM +  SIP*DEL2V
        SRD =SQRT (REFD)
        SRD2=SQRT (REOD)
         SCB = ,6 *SC**,33333333333
         SCR = 2,0 +  SCB*SQRT (ABS(VFZ))*SRD
         SCRO= 2,0 +  SCB*SQRT (ABS(VOZ))*SRD2
      D2 =  2,0*DTH
     DSQ=DTH**2
        NOD = 2*ND
        NOD1 = 3*ND
        NZ = ND + 1
      N1 = ND - 1
C
C              CALC MAX AND MIN NODE NUMBERS FOR BURNING RATES
C
     NCRIT(1)=N1/4+1
     NCRIT(2)=3*N1/4+1
     CON1=1,0/GAM
     CON2=1,0-CON1
C
C         CONSTANTS FOR COEFF CALC
C
     CON(1)= HMAX*0,5
     CON(2)= 0,25/DTH
     CON(3)= HMAX*CON(2)
     CON(4)=  XJ* FGAM / DTH**2
     CON(5)= 0,6666667* CON(4)
     CON(6)=  2,0 * CON(5)
     CON(7)=  0,5 * CON(4)
     CON(8)=  XL * GAM *FGAM
     CON(9)=  XLF * FGAM *0,5
     CON(10)= XLO * FGAM *0,5
     CON(11)= 0,5*( DRAGO +DRAGF)
     CON(12)= ABS(GAM -1,0)
     CON(13)=1,3333334*XJ *FGAM*CON(12)*GAM
     CON(14)= 0,125/DTH
     CON(15)= 1,0/HMAX
     CON(16)=0,5/DTH
     NEXP=XEXP
     DO 20 I=1,ND
```

```
      XI=I-1
      ZIG=XI*DTH
      SINE=SIN(XK*ZIG)
      P(I,1)=1.0+ AP*SINE**NEXP
      DPTH(I,1)= XK*XEXP*AP*COS(XK*ZIG)
      IF(XEXP.NE. 1.0) DPTH(I,1)=DPTH(I,1)*SINE**(NEXP-1)
      T(I,1)= P(I,1)**CON2
      RHO(I,1)=P(I,1)**CON1
      V(I,1)=VSTART
      DVTH(I,1)=0.0
 20   DTTH(I,1)= CON2/RHO(I,1)*DPTH(I,1)
C
      RETURN
      END
```

```
      SUBROUTINE ORG
C
      COMMON /CNM/    IM, INDR,    IU, MALP, MGAM,    MP, MPTN, MPTS
     1              ,NCD,    ND,    NJ,    NO,   NOD, NOD1,    NZ,    N1
      COMMON/E/ FC(80)
C
      DIMENSION W(20)
C
      EQUIVALENCE                  (FC( 1), T      ),   (FC( 2), RKT    )
     1 , (FC( 3), H       ),       (FC( 4), HO     ),   (FC( 5), HMIN   )
     2 , (FC( 6), HMAX    ),       (FC( 7), HZD2   ),   (FC( 8), EMIN   )
     3 , (FC( 9), EMAX    ),       (FC(10), E1     ),   (FC(11), W      )
C
      E1=0.
      H=HMAX
      IM = 0
      MGAM = -1
      MPTN=0
      MPTS=NO
      NCD=0
C
      RETURN
      END
```

```
      SUBROUTINE SETMAP
      COMMON/3/     T(181,3) ,          RHO(181,3) ,           V(181,3) ,
     1              P(181,2) ,         DVTH(181,2) ,        DPTH(181,2) ,
     2            WOX(181,2) ,           WF(181,2) ,         VSQ(181,2) ,
     3           DTTH(181,2) ,           WP(  181)
      COMMON/LIMITS/      NI1,NE1,NI2,NE2,NI3,NE3,JSTART,JEND
      COMMON/LAW/NFLAG,XK,XEXP
      COMMON /CNM/    IM, INDR,    IU, MALP, MGAM,    MP, MPTN, MPTS
     1              ,NCD,    ND,    NJ,    NO,   NOD, NOD1,    NZ,    N1
      COMMON/NODES/NCRIT(2)
      COMMON/MARKET/ NPATH,VSTART,NPMAX
      COMMON/MESS/SYM
      COMMON/E/FC(53),DTH,DUM(26)
      EQUIVALENCE ( FC(36),GAM)
      IF(SYM.NE.0.0)GO TO 300
      IF(VSTART.NE. 0.0) GO TO 300
      IF(XK.NE. 1.0) GO TO 200
C           STRAIGHT SINE WAVE INPUT
      NPATH=1
      NPMAX=NCRIT(1)
      NI1=NCRIT(1)
      NE1=NCRIT(2)
      RETURN
C          SINE 1/2 THETA INPUT
 200  NPATH=2
      NPMAX=N1/2+1
      NI1=1
      NE1=NPMAX
      DPTH(1,1)=0.0
      RETURN
C          ASYMETRICAL OR V THETA INPUT .NE. 0.0
 300  NPATH=3
      AP=FC(37)
      NPMAX=1
      NI1=1
      NE1=N1
      IF(SYM.EQ.0.0)GO TO 390
      NK=N1/2+1
      CON1=1.0/GAM
      CON2=1.0-CON1
      NEXP=15.0*XEXP
      XEXPO=NEXP
      NEXP1=NEXP-1
      DO 350 I=NK,N1
      XI=I-1
      ZIG=XI*DTH
```

```
      ANG=XK*ZIG
      SINE=SIN(ANG)
      P(I,1)=1.0+AP*SINE**NEXP
      DPTH(I,1)=XK*XEXPO*AP*COS(ANG)*SINE**NEXP1
  350 CONTINUE
      PBA=WEOS(P(1,1))/6.2831853
      AMINUS=PBA-1.0
      DO 380 I=1,ND
      P(I,1)=P(I,1)-AMINUS
      T(I,1)=P(I,1)**CON2
      RHO(I,1)=P(I,1)**CON1

      DTTH(I,1)=CON2/RHO(I,1)*DPTH(I,1)
  380 CONTINUE
  390 XMAX=P(1,1)
      DO 400 I=1,N1
      XMAX=AMAX1(XMAX,P(I,1) )
      IF( P(I,1) .EQ. XMAX) NPMAX=I
  400 CONTINUE
      RETURN
      END
```

```
         SUBROUTINE  ASET
C
C     THIS  SUBROUTINE  CALCULATES THE COEFFICIENTS FOR THE AXIAL
C     DERIVATIVE  PACKAGE AND ALSO INITIATES THE W ARRAY AND THE WZ
C     ARRAY.
C
      COMMON/LAW/NFLAG,XK,XEXP
      COMMON /CNM/    IM, INDR,    IU, MALP, MGAM,   MP, MPTN, MPTS
     1            ,NCD,    ND,    NJ,    NO,  NOD, NOD1,   NZ,   N1
      COMMON/E/ FC(80)
      COMMON/NODES/ NCRIT(2)
      COMMON/B/      T(181,3) ,          RHO(181,3) ,          V(181,3) ,
     1               P(181,2) ,         DVTH(181,2) ,       DPTH(181,2) ,
     2             WOX(181,2) ,           WF(181,2) ,        VSQ(181,2) ,
     3            DTTH(181,2) ,           WP(   181)
      COMMON/INDEX/NT,NTP,NTB,KTP,KTB,KT,NCON
C
      DIMENSION A(11),AZ(361),BVD(361),BZD(361),C(3),DD(361),PVD(361)
C
      DIMENSION W(181,4)
      EQUIVALENCE (W(1,1), WOX(1,1) )
      EQUIVALENCE
     1  (FC(32), XL   ),      (FC(33), RED  ),      (FC(34), XJ   )
     2 ,(FC(35), DELV),       (FC(36), GAM  ),      (FC(38), ZIP  )
     3 ,(FC(39), SC   ),      (FC(40), DEL2V),      (FC(41), VZ   )
     4 ,(FC(42), FGAM),       (FC(43), A    ),      (FC(51), C    )
     5 ,(FC(59), SRD ),       (FC(60), SCB  ),      (FC(61), SCR  )
     6 ,(FC(62), SIP ),       (FC(63), SIP2 ),      (FC(64), BB   )
     7 ,(FC(65), BC   ),      (FC(66), BZ   ),      (FC(67), REFD )
     8 ,(FC(68), REOD),       (FC(69), SRD2 ),      (FC(72), DEL1V)
     9 ,(FC(74), XLF ),       (FC(75), XLO  ),      (FC(76), VFZ  )
     A ,(FC(77), VOZ ),       (FC(78), DRAGF),      (FC(79), DRAGO)
     B ,(FC(80), SCRO)
```

```
C
      IF(MGAM)5,10,10
    5 K1=KT
      K2=NT
      NCON=MOD(MP,4)+KT
      GO TO 13
   10 K1=KTP
      K2=NTP
      NCON=MOD(MP,4)+KTP
C
   13 DO 25 I=1,ND
      IF(RHO(I,K2) )60,20,20
   20 VSQ(I,K1)=V(I,K2)**2
      WF(I,K1)=(2,0+SCB*SQRT(RHO(I,K2)*SQRT(VSQ(I,K1) +DEL2V))*SRD)/SCR
      WOX(I,K1)=(2,0+SCB*SQRT(RHO(I,K2)*SQRT(VSQ(I,K1) +DEL1V))*SRD2)/
     1                                                              SCRO
   25 CONTINUE
      IF(NFLAG)30,28,30
   28 CALL FIT(WOX(1,K1))
      CALL FIT(WF(1,K1))
   30 CONTINUE
      DO 40 I = 1, ND
      BVD(I)= RHO(I,K2)* V(I,K2)
      BZD(I)= BVD(I)*DTTH(I,K1)
      PVD(I)= P(I,K1)*DVTH(I,K1)
      AZ(I) =  XLF*WF(I,K1) + XLO*WOX(I,K1)
      DD(I) =  VFZ*WF(I,K1)*XLF  +VOZ*XLO*WOX(I,K1)
      WP(I)=XL*GAM*W(I,NCON)-AZ(I)*T(I,K2)
     1    +SIP*(  XLF*(VSQ(I,K1)+DEL2V)*WF(I,K1)
     2           +XLO*(VSQ(I,K1)+DEL1V)*WOX(I,K1)   )
   40 CONTINUE
C
      A(5)= WEDS( T(1,K2) )/GAM
      A(7)=(GAM-1,0)* WEDS(P(1,K1) )
      C(1)=  FGAM* WEDS(AZ)
      C(2)=-FGAM*WEDS(DD)
      C(3)=FGAM*WEDS(WP(1))  -(GAM-1,)*WEDS(PVD)-WEDS(BZD)
C
      IF(MGAM)43,47,47
   43 DO 45 I=1,ND
   45 WP(I)=FGAM*AZ(I)
   47 IF(DRAGO +DRAGF ,LT, 1,0E-15) RETURN
C          R SUB D,F = R SUB D,OX = 1,
C
      C(2)=C(2)-6,28318531*(DRAGF*ABS(VFZ)*VFZ+DRAGO*ABS(VOZ)*VOZ)
      DO 50 I=1,ND
      X1=   VSQ(I,K1) +DEL2V
      X2=   VSQ(I,K1) +DEL1V
      PVD(I)= RHO(I,K2)*X1* SQRT(X1)
      BZD(I)= RHO(I,K2)*X2* SQRT(X2)
```

```
 50   CONTINUE
      C(3)=C(3)+SIP2*(DRAGF*WEDS(PVD)+DRAGO*WEDS(BZD))
          RETURN
C
 60   WRITE(6,21)FC(1), (P(J,K1), RHO(J,K2), T(J,K2), V(J,K2),
     1                  WF(J,K1),WOX(J,K1), J=1,ND)
   21 FORMAT(1H0,1E20,8/(6E18,8))
       STOP
      END
```

```
      SUBROUTINE FIT(F)
      COMMON/NODES/NCRIT(2)
      DIMENSION F(1)
C           THIS SUBROUTINE USES AVERAGED PARABOLIC INTERPOLATION
C          TO EVALUATE PARAMETERS AT THE V THETA =0 NODES
C
      DO 100 I=1,2
      J=NCRIT(I)
      F(J)=0,66666667*( F(J+1)+F(J-1))-(F(J+2)+F(J-2))/6,0
 100  CONTINUE
      RETURN
      END
```

```
      SUBROUTINE  ZDIR
C
      COMMON/BAR/VZCZ,    RHOCZ,TCZ,VZCZB,     RHOCZB,TCZB
      COMMON /CNM/   IM, INDR,    IU, MALP, MGAM,    MP, MPTN, MPTS
     1            ,VCD,    ND,    NJ,    NO,  NOD, NOD1,    NZ,    N1
      COMMON/E/ FC(80)
      DIMENSION A(8), C(3), X(4)
      EQUIVALENCE               (FC(36), GAM    ),   (FC(41), VZ     )
     1,  (FC(43), A       ),    (FC(51), C      ),   (FC(55), X      )
C
      X(1)=(C(2)-A(5)*C(1)/A(2)-A(6)*C(3)/A(8))
     1    /(A(4)-A(5)*A(1)/A(2)-A(6)*A(7)/A(8))
      X(2)=(C(1)-A(1)*X(1))/A(2)
      X(3)=(C(3)-A(7)*X(1))/A(8)
      IF(MGAM)15,10,10
  10  RETURN
  15  VZCZ=X(1)
      RHOCZ=X(2)
      TCZ=X(3)
      VZCZB=X(1)
      RHOCZB=X(2)
      TCZB=X(3)
C
      RETURN
      END
```

```
      FUNCTION  WEDS(B)
C
C     THIS FUNCTION EMPLOYS WEDDLES RULE TO EVALUATE THE INTEGRAL(0,2PI)
C
      COMMON/CNM/ DUM1(15),N1
      COMMON/E/ FC(80)
C
      DIMENSION  B(1)
C
      EQUIVALENCE  (FC(54),DTH)
C
      SUM = 0.0
         DO 30    I = 1,N1,5
   30    SUM =  SUM +  38.*B(I) +75.*(B(I+1) + B(I+4)) + 50.*(B(I+2) +
     1   B(I+3))
      WEDS = 5.0*DTH/288.*SUM
C
      RETURN
      END
```

```
      SUBROUTINE BARRED
      COMMON/B/     T(181,3) ,          RHO(181,3) ,           V(181,3) ,
     1              P(181,2) ,         DVTH(181,2) ,        DPTH(181,2) ,
     2            WOX(181,2) ,           WF(181,2) ,         VSQ(181,2) ,
     3           DTTH(181,2) ,           WP(  181)
      COMMON/BAR/VZCZ,    RHOCZ,TCZ,VZCZB,     RHOCZB,TCZB
      COMMON/INDEX/NT,NTP,NTB,KTP,KTB,KT,NCON
      COMMON/E/ FC(80)
      COMMON /CNM/    IM, INDR,    IU, MALP, MGAM,   MP, MPTN, MPTS
     1              ,NCD,     ND,     NJ,     NO,  NOD, NOD1,    NZ,   N1
      DIMENSION X(4)
      EQUIVALENCE                 (FC(36), GAM    ),    (FC(41), VZ     )
     1   ,(FC(55),X(1))
     *    ,(FC(74),XLF)    ,(FC(75),XLO)
     *    ,(FC(42),FGAM)
C
C                    THIS SUB CALCS AVE OR BARRED VALUES
C
      DO   200 I=1,ND
      T(I,NTB)= 0.5*(T(I,NT)+T(I,NTP) )
      RHO(I,NTB)= 0.5*( RHO(I,NT)  + RHO(I,NTP) )
      V(I,NTB) = 0.5*( V(I,NT)  + V(I,NTP) )
      P(I,KTB) = 0.5*( P(I,KT)  + P(I,KTP) )
      DPTH(I,KTB)=0.5*(  DPTH(I,KT)  + DPTH(I,KTP) )
      WP(I) = FGAM*(XLO*WOX(I,KTP)  + XLF*WF(I,KTP) )
      WOX(I,KTB) =0.5*(  WOX(I,KT)   + WOX(I,KTP ) )
      WF(I,KTB)  =0.5*(  WF(I,KT)    + WF(I,KTP) )
  200 CONTINUE
      VZCZB= 0.5*( X(1)+ VZCZ )
      RHOCZB=0.5*( X(2)+ RHOCZ)
      TCZB = 0.5*( X(3)+ TCZ   )
      RETURN
      END
```

```
      SUBROUTINE ABCD(NOW,NOW2)
C
C          THIS  SUB CALCS THE COEFF FOR THE BAND MATRICES
C
      COMMON/CONST/CON(20)
      COMMON/BAR/VZCZ,    RHOCZ,TCZ,VZCZB,     RHOCZB,TCZB
      DIMENSION X(3)
      COMMON/COEFF/A(181,3) ,            B(181,3) ,          C(181,3) ,
     1              D(181,3)
      COMMON/B/     T(181,3) ,          RHO(181,3) ,          V(181,3) ,
     1              P(181,2) ,         DVTH(181,2) ,       DPTH(181,2) ,
     2            WOX(181,2) ,           WF(181,2) ,        VSQ(181,2) ,
     3           DTTH(181,2) ,           WP(   181)
      COMMON/E/ FC(80)
      COMMON /CNM/    IM, INDR,    IU, MALP, MGAM,    MP, MPTN, MPTS
     1              ,NCO,    ND,    NJ,    NO,  NOD, NOD1,   NZ,    N1
      EQUIVALENCE                (FC(36), GAM    ),    (FC(41), VZ     )
     1    , ( FC(6), HMAX )
     2 ,  (FC(72), DEL1V ),
     3    (FC(63), SIP2  ),
     4    (FC(40), DEL2V ),
     5    (FC(78), DRAGF ),    (FC(79), DRAGO )
     6      ,(FC(55),X(1))
     *    ,(FC(74),XLF)     ,(FC(75),XLO)
     *    ,(FC(42),FGAM)
      DIMENSION W(181,4)
      EQUIVALENCE (W(1,1), WOX(1,1) )
      COMMON/INDEX/NT,NTP,NTB,KTP,KTB,KT,NCON
C
C                   THIS  THE  MASTER  LIST OF THE CON ARRAY CALCED
C                                        IN RSET
C       CON             EXPRESSION
C
C         1             DT/2,
C
C         2             1,0/(4,0*DTHETA)
C
C         3             DT/(4,0* DTHETA)
C
C         4             J * F(GAMMA)/ (DTHETA**2)
C
C         5             2,/3, * J * F(GAMMA) /(DTHETA**2)
C
C         6             4,/3, * J * F(GAMMA) /(DTHETA**2)
C
C         7             1,/2, * J * F(GAMMA) /(DTHETA**2)
C
C         8              L  * GAMMA * F(GAMMA)
C
C         9             (L SUB F)*  F(GAMMA) * 1,/2,
C
C        10            (L SUB OX) * F(GAMMA) * 1,/2,
C
C        11            1,/2, * ( DRAGO  + DRAGF)
C
C        12             GAMMA-1,0
C
```

```
C       13          4./3.*J *F(GAMMA)*GAMMA*(GAMMA-1.0)
C
C        14    1./(8.0*DTHETA)
C
C        15    1.0/DTIME
C
C       16          1.0/(2.0*DTHETA)
C
C
      VZCZB2=VZCZB**2
      K=N1
C               K=I-1 FOR THE FOLLOWING LOOP
C
      DO 500 I=1,N1
C
C                         RHO  COEFF
C
      WPT=FGAM*(XLF*WF(I,KT)+XLO*WOX(I,KT) )
      A(I,1)=-V(I,NOW)*CON(2)
      C(I,1)= -A(I,1)
      B(I,1)= CON(15)+ 0.5*( DVTH(I,NOW2)+ X(1)      )
      D(I,1)=  RHO(I,NT)*(CON(15) -0.5*( DVTH(I,KT) +VZCZ) )
     1          - V(I,NT)*( RHO(I+1,NT)-RHO(K,NT) )*CON(2)
     2          - VZ*RHOCZB     +(WP(I)+WPT)*0.5
C
C                         V COEFF
C
      C1= RHO(I,NOW)*V(I,NOW)*CON(2)
      A(I,2)=-C1-CON(5)
      C(I,2)= C1-CON(5)
      B(I,2)=  RHO(I,NTB)*CON(15) +CON(6)+WP(I)*0.5+CON(11)*RHO(I,NOW)
     1             *ABS(V(I,NOW))
      D(I,2)=RHO(I,NTB)*V(I,NT)*CON(15)-0.5*RHO(I,KT)*V(I,KT)*DVTH(I,KT)
     1       - DPTH(I,KTB)/GAM + CON(5)*(V(I+1,NT) -2.0*V(I,NT)+V(K,NT))
     2       - V(I,NT)*WPT  *0.5 -CON(11)*RHO(I,NT)*V(I,NT)*ABS(V(I,NT))
C
C                      T COEFF
C
      C1=RHO(I,NTB)*V(I,NTB)*CON(2)
      A(I,3) =-C1 -CON(7)
      C(I,3) =+C1 -CON(7)
      B(I,3) =  RHO(I,NTB)/HMAX +CON(4) +WP(I)*0.5
      VBSQ=V(I,NTB)**2
      VBSQF= VBSQ+ DEL2V
      VBSQO= VBSQ+ DEL1V
      D(I,3) = RHO(I,NTB)*T(I,NT)/HMAX -C1*(T(I+1,NT)-T(K,NT))
     1       -RHO(I,NTB)*VZ *TCZB + CON(7)*(T(I+1,NT)-2.0* T(I,NT)
     2       +T(K,NT) ) - T(I,NT)*WPT  *0.5 -CON(12)*P(I,KTB)
     3     *(DVTH(I,KTB)+VZCZB ) + CON(8)* W(I,NCON)
     4    + CON(13)*( VZCZB2 + DVTH(I,KTB)*( DVTH(I,KTB)-VZCZB) )
     5    +SIP2*(   VBSQF*( CON(9)*WF(I,KTB) + RHO(I,NTB)*DRAGF*
     6             SQRT(VBSQF) )
     7           + VBSQO*(CON(10)*WOX(I,KTB)+RHO(I,NTB)*DRAGO*
     8             SQRT(VBSQO) )    )
  500 K=I
      RETURN
      END
```

```
      SUBROUTINE DEINT(NOW)
      COMMON/B/     T(181,3) ,         RHO(181,3) ,          V(181,3) ,
     1              P(181,2) ,        DVTH(181,2) ,       DPTH(181,2) ,
     2            WOX(181,2) ,          WF(181,2) ,        VSQ(181,2) ,
     3           DTTH(181,2) ,          WP(  181)
      COMMON/LIMITS/      NI1,NE1,NI2,NE2,NI3,NE3,JSTART,JEND
      COMMON/MARKET/ NPATH,VSTART,NPMAX
      COMMON/COEFF/A(181,3) ,          B(181,3) ,           C(181,3) ,
     1             D(181,3)
      COMMON/E/ FC(80)
      COMMON/INDEX/NT,NTP,NTB,KTP,KTB,KT,NCON
      COMMON /CNM/   IM, INDR,   IU, MALP, MGAM,   MP, MPTN, MPTS
     1              ,NCD,    ND,    NJ,    NO,   NOD, NOD1,   N7,   N1
      EQUIVALENCE (FC(1),TI)
C
C
C     CALC  RHO
C
      RHO(NI1,NTB)=RHO(NI1,NOW)
      RHO(NE1,NTB)=RHO(NE1,NOW)
      N=NE1-NI1+1
      CALL INVTRI(A(NI1,1),B(NI1,1),C(NI1,1),D(NI1,1),RHO(NI1,NTB),N)
C
C     CALC  V
C
      IF(NPATH-2) 60,60,80
C        SYMETRICAL WAVE
 60   V(NE1,NTB)=0,0
      V(NI1,NTB)=0,0
      GO TO 200
C        ASYMETRICAL WAVE OR V INPUT ,NE, 0,0
 80   V(NI1,NTB)=V(NI1,NOW)
      V(NE1,NTB)=V(NE1,NOW)
 200  CALL INVTRI(A(NI1,2),B(NI1,2),C(NI1,2),D(NI1,2),V  (NI1,NTB),N)
C
C     CALC  T
C
      T(NI1,NTB)=T(NI1,NOW)
      T(NE1,NTB)=T(NE1,NOW)
      CALL INVTRI(A(NI1,3),B(NI1,3),C(NI1,3),D(NI1,3),T  (NI1,NTB),N)
      RETURN
      END
```

```
      SUBROUTINE INVTRI(A,B,C,D,X,NMAX)
      DIMENSION A(1),B(1),C(1),D(1),X(1)
      K=1
      SAVE=A(1)*X(NMAX)
      D(1)=( D(1) - SAVE )/B(1)
      D(NMAX)=D(NMAX)-C(NMAX)*X(1)
      C(1)=C(1)/B(1)
      DO 100 I=2,NMAX
      IF( ABS(A(I) ) .LT. 1.0E-10 ) GO TO 50
      B(I)=B(I)/A(I)-C(K)
      C(I)=C(I)/A(I)/B(I)
      D(I)= ( D(I)/A(I)-D(K) )/B(I)
      GO TO 100
50    C(I)=C(I)/B(I)
      D(I)=D(I)/B(I)
100   K=I
      X(K)=D(K)
      I=K-1
200   X(I)=D(I)-C(I)*X(K)
      K=I
      I=I-1
      IF(I)  300,300,200
300   RETURN
      END
```

```
      SUBROUTINE MAPPER(M,N)
      COMMON/CONST/CON(20)
      COMMON/LAW/NFLAG,XK,XEXP
      COMMON/MARKET/ NPATH,VSTART,NPMAX
      COMMON/LIMITS/     NI1,NE1,NI2,NE2,NI3,NE3,JSTART,JEND
      COMMON/R/     T(181,3) ,         RHO(181,3) ,           V(181,3) ,
     1              P(181,2) ,        DVTH(181,2) ,        DPTH(181,2) ,
     2            WOX(181,2) ,          WF(181,2) ,         VSQ(181,2) ,
     3           DTTH(181,2) ,          WP(  181)
      COMMON/INDEX/NT,NTP,NTB,KTP,KTB,KT,NCON
      COMMON /CNM/    IM, INDR,    IU, MALP, MGAM,    MP, MPTN, MPTS
     1               ,NCD,    ND,    NJ,    NO,  NOD, NOD1,    NZ,    N1
      COMMON/NODES/NCRIT(2)
C
C          CALC  NEEDED VALUES
C                   AND
C          MAP  PRIME VARIABLES
C
      GO TO (10,80,120), NPATH
 10   KE=NI1-1
      DO 20 I=1,KE
      J1=NI1-I
      K1=NI1+I
      J2=NE1+I
      K2=NE1-I
```

```
      RHO(J1,NTB)=RHO(K1,NTB)
      RHO(J2,NTB)=RHO(K2,NTB)
      T(J1,NTB)=T(K1,NTB)
      T(J2,NTB)=T(K2,NTB)
      V(J1,NTB)=-V(K1,NTB)
      V(J2,NTB)=-V(K2,NTB)
20    CONTINUE
30    CALL FIT(RHO(1,NTB))
      CALL FIT(T(1,NTB))
      GO TO 120
80    KE=NE1-1
      DO 100 I=1,KE
      J=NE1+I
      K=NE1-I
      RHO(J,NTB)=RHO(K,NTB)
      T(J,NTB)=T(K,NTB)
      V(J,NTB)=-V(K,NTB)
100   CONTINUE
120   DO 150 I=1,N1
      P(I,NTB)=RHO(I,NTB)*T(I,NTB)
150   CONTINUE
      RHO(ND,M)=RHO(1,M)
      V(ND,M)=V(1,M)
      T(ND,M)=T(1,M)
      P(ND,N)=P(1,N)
      K=N1
      DO 200 I=1,N1
      DVTH(I,N)=     CON(16)* (V(I+1,M)- V(K,M))
      DTTH(I,N)=     CON(16)* (T(I+1,M)- T(K,M))
      DPTH(I,KTP)=RHO(I,NTB)*DTTH(I,N)+T(I,NTB)*CON(16)*(RHO(I+1,NTB)-
     1   RHO(K,NTB))
      K=I

200   CONTINUE
      DVTH(ND,N)=DVTH(1,N)
      DPTH(ND,N)=DPTH(1,N)
      DTTH(ND,N)=DTTH(1,N)
      RETURN
      END
```

```
      SUBROUTINE AUXCAL
      DIMENSION PDIS(361),WDIS(361), PWDIS(361)
      COMMON/B/     T(181,3) ,          RHO(181,3) ,           V(181,3) ,
     1              P(181,2) ,         DVTH(181,2) ,        DPTH(181,2) ,
     2            WOX(181,2) ,           WF(181,2) ,         VSQ(181,2) ,
     3           DTTH(181,2) ,           WP(  181)
      COMMON/MARKET/ NPATH,VSTART,NPMAX
      COMMON/INDEX/NT,NTP,NTB,KTP,KTB,KT,NCON
      COMMON/GAINS/ XNSUM,RMSW,RMSP, GAIN, RESPON,NCYCLE,XNAVE,NCNT
     *       ,NSTEPS, XNSAVE(600)
      COMMON /CNM/    IM, INDR,    IU, MALP, MGAM,    MP, MPTN, MPTS
     1             ,NCD,    ND,    NJ,    NO,  NOD, NOD1,   NZ,   N1
      COMMON/E/ FC(80)
      COMMON/NODES/NCRIT(2)
      COMMON/CPLT/ ART(800,3)
      DIMENSION F(200,3)
      EQUIVALENCE ( F(1,1),XNSAVE(1) )
C
      EQUIVALENCE  (NC,NCRIT(1))
      EQUIVALENCE  (FC(1), TI)
      DATA PI/3,1416/
C       VARIBLES INITILIZED IN RSET
      PSUM=P(1,KT)
      WDIS(1)=WOX(1,KT)+WF(1,KT)
      WSUM=WDIS(1)
      XMAX=P(1,KT)
      XMIN=P(1,KT)
      DO 50 I=2,N1
      PSUM=PSUM+ P(I,KT)
      XMAX= AMAX1(XMAX, P(I,KT) )
      XMIN=AMIN1(XMIN,P(I,KT))
      WDIS(I)= WOX(I,KT)+ WF(I,KT)
   50 WSUM= WSUM +WDIS(I)
      XN1=N1
      WAVE=WSUM/XN1
      PAVE=PSUM/XN1
      DO 100 I=1,N1
      PDIS(I)= P(I,KT)/PAVE -1,0
      WDIS(I)=WDIS(I)/WAVE-1,0
      PWDIS(I)=WDIS(I)*PDIS(I)
      WDIS(I)=WDIS(I)**2
      PDIS(I)=PDIS(I)**2
  100 CONTINUE
      PINTSQ=WEDS(PDIS)
      RESPON=WEDS(PWDIS)/PINTSQ
      IF(NCNT,NE, 0) GO TO 150
      NCNT=NCNT+1
      RS1=RESPON
      XNAVE=0,0
      GO TO 210
  150 XNAVE=XNAVE+0,5*FC(6)*(RS1+RESPON)
      RS1=RESPON
```

```
210   IF( MPTN.NE. 0 )  RETURN
      ART(IU,1)= TI
      ART(IU,2)=(XMAX-XMIN)/PAVE
      ART(IU,3)= P(NPMAX,KT)
      RMSW= SQRT(WEDS(WDIS)*0.5/PI)
      RMSP = SQRT(PINTSQ*0.5/PI)
      GAIN= RMSW/RMSP
      F(IU,1)=GAIN
      F(IU,2)=RESPON
      F(IU,3)=XNAVE
      RETURN
      END
```

```
        SUBROUTINE SHIFT
C
       COMMON /CNM/    IM, INDR,    IU, MALP, MGAM,    MP, MPTN, MPTS
      1               ,NCD,    ND,    NJ,    NO,  NOD, NOD1,    NZ,    N1
       COMMON/DAMPER/DEL,VP1,VPN,DELO
       COMMON/LIMITS/     NI1,NE1,NI2,NE2,NI3,NE3,JSTART,JEND
       COMMON/ITER/ MAXIT
       COMMON/E/ FC(80)
       COMMON/B/     T(181,3) ,        RHO(181,3) ,            V(181,3) ,
      1              P(181,2) ,       DVTH(181,2) ,         DPTH(181,2) ,
      2            WOX(181,2) ,         WF(181,2) ,          VSQ(181,2) ,
      3           DTTH(181,2) ,         WP(  181)
       COMMON/INDEX/NT,NTP,NTB,KTP,KTB,KT,NCON
       DIMENSION B(181,9)
       EQUIVALENCE ( B(1,1), T(1,1) )
C
       EQUIVALENCE                (FC( 1), TI      ),    (FC( 2), RKT    )
      1,    (FC( 3), H       ),    (FC( 4), HO      ),    (FC( 6), HMAX   )
      2,    (FC( 7), HZD2    ),    (FC( 8), EMIN    ),    (FC( 9), EMAX   )
      3,    (FC(10), E1      )
C
       IF(MPTN .EQ. 0) MPTN=MPTS
       NCD=NCD+1
C
C        ERROR CH
C
       KONVER=1
       ERMAX=0.0
       DO 60 I=1,N1
       IF(I.EQ.NI1. OR.   I.EQ.NE1) GO TO 60
       IF(ABS(V(I,NTB)).LT. 1.0E-12) GO TO 60
       ERR=(V(I,NTP)-V(I,NTB ))/V(I,NTB)
       XERR=ABS(ERR)
       ERMAX=AMAX1(ERMAX,XERR)
       IF(ERMAX.NE.XERR) GO TO 60
       ERRPT=ERR
       NMAX=I
```

```
 60    CONTINJE
       WRITE(6,900) NMAX,ERRPT
 900   FORMAT(5X,I5,E17,7)
       IF(ERMAX,GT,EMIN)KONVER=0
       IF(NCD,GE,MAXIT) GO TO 500
 150   IF(KONVER)500,200,500
 200   NSAVE=NTP
       NTP=NTB
       NTB=NSAVE
       RETURN
 500   MGAM= -1
 550   IM=NCD
       NCD=1
       TI=TI+H
       MPTN=MPTN-1
C
C              SHIFT   POINTERS
C
       KTP=KT
       NT=NTB
       KT=KTB
       RETURN
       END
```

```
      SUBROUTINE MAXMIN(Y,X,XL,XR,YT,YB,L,M,N,NPLT,NODES,NPLPF)
C
C     OBTAINS BOUNDS FOR NPLT FUNCTIONS TO BE PLOTTED NPLPF PER FRAME
C     FOR NODES POINTS.
C
      COMMON /DEBUG/ IDEBUG
      DIMENSION Y(L,M,N), X(L), YT(M), YB(M)
C
C     OBTAIN X LIMITS SINGLE FUNCTION
C
      XL = X(1)
      XR = XL
      DO 10 I = 2,NODES
      XL = AMIN1(XL,X(I))
      XR = AMAX1(XR,X(I))
 10   CONTINUE
C
C     OBTAIN LIMITS FOR NPLT Y FUNCTIONS
C
      DO 40 I = 1,NPLT
      YB(I) = Y(1,I,1)
      YT(I) = YB(I)
      DO 30 J = 1,NPLPF
      DO 20 K = 2,NODES
      YB(I) = AMIN1(YB(I),Y(K,I,J))
      YT(I) = AMAX1(YT(I),Y(K,I,J))
 20   CONTINUE
 30   CONTINUE
 40   CONTINUE
      IF(IDEBUG.NE.0) GO TO 999
      DO 200 IX = 1,NPLT
      DO 200 IY = 1,NPLPF
 200  WRITE(6,100) (Y(IZ,IX,IY),IZ=1,NODES)
 100  FORMAT(10H0Y ARRAY  //(4E20.8))
      WRITE(6,101) NPLT,NPLPF,NODES, XL, XR, (YT(IX),YB(IX),IX=1,NPLT)
 101  FORMAT(18H0NPLT,NPLPF,NODES=,3I10/4X, 6HXL,XR=,2E20.8//
     1 12H0YT(I),YB(I)//(2E20.8))
 999  RETURN
      END
```

```
      SUBROUTINE PLTSUB(IPLCT)
C
C     IPLCT IS CURRENT NUM OF PLOTS RESIDING IN BUFFER FCTPLT(I,J,K)
C     IT IS INITIALIZED TO 0 IN MAIN, INCREMENTED AND RESET IN PLTSUB
C
C     FUNCTIONS OF PLTSUB ARE.,,,,,
C     TEST FOR PLOT POINT, FILL PLOT-BUFFER, CALL PURGE TO EMPTY
C     BUFFER WHEN NECESSARY
C
C     NEED ACCESS TO CURRENT TIME (TI) TO TEST FOR PLOT POINT
C     NEED ACCESS TO NUMBER OF NODES,,,ND
C     NEED ACCESS TO FUNCTIONS TO BE PLOTED.,P,RHO,T,V,W
C
      COMMON/B/     T(181,3) ,          RHO(181,3) ,           V(181,3) ,
     1              P(181,2) ,         DVTH(181,2) ,        DPTH(181,2) ,
     2            WOX(181,2) ,           WF(181,2) ,         VSQ(181,2) ,
     3           DTTH(181,2) ,           WP(   181)
      COMMON/INDEX/NT,NTP,NTB,KTP,KTB,KT,NCON
      COMMON /CNM/    IM, INDR,    IU, MALP, MGAM,    MP, MPTN, MPTS
     1               ,NCD,    ND,    NJ,    NO,  NOD, NOD1,    NZ,    N1
      COMMON /DEBUG/ IDEBUG
      COMMON /E/ FC(80)
      COMMON /PLINFO/ FCTPLT(181,6,4), TITLE(5)
     1 , IFPLOT(6), TIMEP(4), THETAP(181), YT(6), YB(6), TPLOT(50)
     2 , DEL, IPLOT, NLPHGH, NLPHGV, NLTHGH, NLTHGV, NPLPF
     3 , NPLT, NPHPLT, XLEFT, XRIGHT, TIMAX, IPLMAX
C
C
      EQUIVALENCE (FC(1),TI)
C
C     OF NPLT = 0 PLOTTING FUNCTIONS VS THETA AT CONSTANT T NOT
C     REQUIRED.....,RETURN
C
      IF(NPLT.EQ.0) GO TO 40
C
C     MGAM IS SET TO -1 IN SHIFT UPON COMPLETION OF AN INTEGRATION
C     STEP.  CANNOT PLOT IF INTEGRATION STEP NOT COMPLETED.
      IF (MGAM.NE.-1) GO TO 40
C
C     PLOTTING REQUIRED.,, TEST FOR PLOT POINT
C
 20   TEMPA = TPLOT(IPLOT)
      TEMPB = ABS(TI-TEMPA)
C
C     IF NEXT STMT TRUE HAVE A PLOT POINT
      IF (TEMPB.LE.DEL) GO TO 25
C     MUST MAKE SURE TPLOT(IPLOT) HAS NOT BEEN BYPASSED
C     IF SO THEN MAKE TPLOT(IPLOT) CATCH UP AND PASS TI
      IF (TI.LT.TEMPA) GO TO 40
C     MAKE TPLOT CATCH UP
      IPLOT = IPLOT + 1
      IF (IPLOT.LE.IPLMAX) GO TO 20
```

```
C
C     HAVE COUNTED PAST TIME PLOT ARRAY, CUT OFF FURTHER PLOTTING
 22   NPLT = 0
      CALL PLOTND
      GO TO 40
C
C     CURRENT TIME CLOSE ENOUGH TO A PLOT STATION TO WARRANT PLOTTING
C     IF MAX PLOTS PER PAGE HAVE BEEN REACHED PLOT THAT FRAME
 25   IF (IPLCT.LT.NPLPF)  GO TO 30
C     PLOT FRAME BY PURGING PLOT ARRAY, RESET COUNTER
      CALL PURGE (0)
      IPLCT = 0
C     INCREMENT COUNTER FOR NO PLOTS PER FRAME
 30   IPLCT = IPLCT + 1
      IF(IDEBUG.EQ.0) WRITE(6,101) MPTN, NPLT, ND,IPLCT
 101  FORMAT(1H0,*IN PLTSUB BEFORE DO 37*,4I10)
C
C     FILL PLOT ARRAY(WHICH ACTS AS A BUFFER ALLOWING MULTIPLE
C     PLOTS PER FRAME) WITH FUNCTIONS TO BE PLOTTED
C
      DO 37 I = 1,NPLT
      IGOTO = IFPLOT(I)
      GO TO (31,32,33,34,35,36), IGOTO
C     31 = PRESSURE DISTURBANCE
 31   DO 311 J =1,ND
      FCTPLT(J,I,IPLCT) = P(J,KT)
 311  CONTINUE
      GO TO 37
 32   DO 322 J = 1,ND
      FCTPLT(J,I,IPLCT) = RHO(J,NT)
 322  CONTINUE
      GO TO 37
 33   DO 333 J = 1,ND
      FCTPLT(J,I,IPLCT) = T(J,NT)
 333  CONTINUE
      GO TO 37
 34   DO 344 J = 1,ND
      FCTPLT(J,I,IPLCT) = V(J,NT)
 344  CONTINUE
      GO TO 37
 35   DO 355 J = 1,ND
      FCTPLT(J,I,IPLCT) = WF(J,KT)
 355  CONTINUE
      GO TO 37
 36   DO 366 J = 1,ND
      FCTPLT(J,I,IPLCT) = WOX(J,KT)
 366  CONTINUE
 37   CONTINUE
```

```
C     BUFFER ARRAY FCTPLT(J,I,K) IS FILLED FOR
C        J = 1 TO NUMBER OF NODES + 1
C        I = 1 TO NUMBER OF FUNCTIONS TO BE PLOTTED
C        K = 1 TO CURRENT NO PLOTS PER FRAME  .LE. NPLPF
      TIMEP(IPLCT) = TI
      IF (TIMEP(IPLCT).LT.TPLOT(IPLMAX)) GO TO 39
C
C     IF LAST PLOT POINT PUT IN BUFFER.....
C      1. ADJUST AND SAVE NUMBER OF PLOTS PER FRAME.
C      2. PURGE INTERNAL BUFFER FCTPLT(I,J,K).
C      3. PURGE PLOT PACKAGE BUFFERS AND RESET NPLPF
C      4. CUT OFF FURTHER FUNCTION VS THETA PLOTTING (GO TO 22).
      ISAVE = NPLPF
      NPLPF = IPLCT
      CALL PURGE(0)

      NPLPF = ISAVE
      GO TO 22
C
 39   CONTINUE
      IF(IDEBUG.EQ.0) WRITE(6,102) MPTN,NPLT,ND
 102  FORMAT(24HO HAVE COMPLETED LOOP 37, 3I10)
C     INCREMENT TIME PLOT POINTER, CHECK IF COUNTED PAST TIME PLOT ARRAY
      IPLOT = IPLOT + 1
      IF(IPLOT.GT.IPLMAX) GO TO 22
C     GO TO 22 EMPTIES BUFFERS FOR PLOT AND CUTS OFF FURTHER PLOTTING
C
 40   CONTINUE
      RETURN
      END
```

```
      SUBROUTINE PURGE (IGOTO)
C
C     THIS SUB PLOTS NPLPF PLOTS OF A FUNCTION VS THETA ON ONE FRAME
C     UTILIZES SC4020 SOFTWARE ROUTINES
C     HAVE NPLT FUNCTIONS TO PLOT, NPLPF PER FRAME
C     IFPLOT(I) POINTS TO ITH FUNCTION
C
C     IGOTO = 0 PLOT THETA VS FUNCTION FOR CONSTANT TIME
C     IGOTO = 1 PLOT PRESSURE HISTORY PLOTS
C
      COMMON /CNM/ DUM1(2), IU, DUM2(6), ND, DUM3(6)
      COMMON/CPLT/ ART(800,3)
      COMMON /DEBUG/ IDEBUG
      COMMON /PLINFO/ FCTPLT(181,6,4), TITLE(5)
     1 , IFPLOT(6), TIMEP(4), THETAP(181), YT(6), YB(6), TPLOT(50)
     2 , DEL, IPLOT, NLPHGH, NLPHGV, NLTHGH, NLTHGV, NPLPF
     3 , NPLT, NPHPLT, XLEFT, XRIGHT, TIMAX, IPLMAX
      COMMON /PTEXT/ PTX(3),ROTX(3),TTX(3),VTHX(3),WFTX(3),WOTX(3)
      DIMENSION BCDTXT(1), MRKPT(5)
      EQUIVALENCE (BCDTXT(1),PTX(1))
      DATA (MRKPT(I),I=1,5) / 38,63,16,55,44/
C     38 = CIRCLE, 63 = SQUARE, 16 = +, 55 = X, 44 = *
      DATA IDL, IDM, IDN / 181,6,4 /
      DATA PTX  /24HPRESSURE WAVE PERTURB   /
      DATA ROTX /24HDENSITY  WAVE PERTURB   /
      DATA TTX  /24HT        WAVE PERTURB   /
      DATA VTHX /24HV THETA  WAVE PERTURB   /
      DATA WFTX /24HW FUEL   WAVE PERTURB   /
      DATA WOTX /24HW OXID   WAVE PERTURB   /
      DATA L800,M3,NONE /800,3,1/
C
      IF (IGOTO) 1,1,60
 1    CONTINUE
C
C     STEP 1 IS TO DRAW GRID, NEED MAX-MIN
C     OBTAIN MAXMIN FOR X AND FOR Y(NUMBER OF FUNCTIONS) (NPLT FUNC)
C
      CALL MAXMIN(FCTPLT(1,1,1),THETAP(1),XLEFT,XRIGHT,YT(1),YB(1)
     1 , IDL,IDM,IDN, NPLT, ND, NPLPF)
C
C
C     OBTAIN GRID SPACING AND ADJUSTED MAX AND MIN FOR X
      XI = FLOAT(NLTHGV)
      CALL SCAL (XRIGHT,XLEFT,XI,DX,XMIN,DZZ)
C     ADJUST CURRENT X LIMITS
      XLEFT = XMIN
      XRIGHT = XLEFT + DX * XI
C     BEGIN PLOTTING OVER LOOP (NPLT = NO FUNCTIONS TO BE PLOTTED)
      DO 50 IP = 1,NPLT
      IF(IDEBUG,EQ,0) WRITE(6,130)IP,XLEFT,XRIGHT,YT(IP),YB(IP),DX,DY
 130  FORMAT(27H0IN PURGE, PLOT FUNCTION IS,I5// 14H0XLEFT,XRIGHT=,
     1  2E20,8// 15H0YT(IP),YB(IP)=,2E20,8// 7H0DX,DY=,2E20,8)
```

```
C       SET EXTRA MARGIN SPACING
        CALL SETMIV ( 40,100, 40, 40)
C       OBTAIN GRID SPACING AND ADJUSTED MAX AND MIN FOR Y FUNCTIONS
        XI = FLOAT(NLTHGH)
        CALL SCAL (YT(IP),YB(IP),XI,DY,YMIN,DZZ)
C
        IF (DY.GE.0.5E-3) GO TO 15
        DY = 0.5E-3
        YMIN = YMIN + DY
 15       CONTINUE
C
C       ADJUST CURRENT Y LIMITS
        YB(IP) = YMIN
        YT(IP) = YMIN + DY *XI
C
C       MINUS NY CAUSES Y AXIS TO BE LABELED IN SCIENTIFIC NOTATION
C       PLUS NY CAUSES Y AXIS TO BE LABELED IN FIXED POINT NOTATION
C NOTE IN FIXED POINT NOTATION LEADING ZEROS TO RIGHT OF . SUPPRESSED
C
        NY = +6
        IF (YT(IP).LT.0.1 .OR. YB(IP).LT.0.1) NY = -4
C
C       FOR THE FIRST PLOT.IE.GRID.ADVANCE FILM
        CALL GRID1V(3,XLEFT,XRIGHT,YB(IP),YT(IP),DX,DY,-0,-0,-1,-1,+5,NY)
C
        CALL RITE2V(125,990,1020,90,1,40,1,TITLE(1),IER)
C
C       AFTER OBTAINED GRID DO LOOP OVER NPLPF
C
        DO 30 IQ = 1,NPLPF
C       PLOT POINTS FOR 1 TIME INTERVAL AT A TIME
C       CONNECT POINTS FOR 1 TIME INTERVAL
C       FIRST POINTS
        CALL APLOTV (ND,THETAP(1),FCTPLT(1,IP,IQ),1,1,1,MRKPT(IQ),KK)
        IX1 = NXV(THETAP(1))
        IX2 = NXV(THETAP(2))
        IY1 = NYV(FCTPLT(1,IP,IQ))
        IY2 = NYV(FCTPLT(2,IP,IQ))
        CALL LINEV(IX1,IY1,IX2,IY2)
C       CONNECT ALL OTHER POINTS
        NUP = ND - 1
        DO 20 IR = 2,NUP
        IX1 = NXV(THETAP(IR))
        IX2 = NXV(THETAP(IR+1))
        IY1 = NYV(FCTPLT(IR,IP,IQ))
        IY2 = NYV(FCTPLT(IR+1,IP,IQ))
        CALL LINEV ( IX1,IY1, IX2,IY2)
 20     CONTINUE
 30     CONTINUE
C       NPLPF PLOTS HAVE BEEN GENERATED FOR THIS FRAME.PROCEED TO
C       LABEL IT. 1. LABEL AXES
C                 2. LABEL CHARACTERS VS TIME POINTS
C       LABEL X AXIS
        CALL RITE2V(300,10,1000,90,1,16,-1,16HTHETA IN RADIANS,IER)
C       LABEL Y AXIS
```

```
C     IFPLOT(1) CONTAINS POINTER FOR TYPE OF PLOT VS THETA FOR FNC 1.
      INT = IFPLOT(IP)
      INT = INT * 3 - 2
      CALL RITE2V(10,300,1000,180,1,24,+1,BCDTXT(INT),IER)
C     LABEL OUTSIDE MARGINS TO IDENTIFY CHARACTERS
C     SET RASTER POSITIONS
      CALL RITE2V(950,980,1023,90,1, 4,-1, 4HTIME,IER)
      IX = 940
      IY = 960
      DO 40 IJP = 1,NPLPF
      NS = MRKPT(IJP)
      CALL PLOTV (IX,IY,NS)
C     MOVE X POSIT OVER 16 RASTERS
      IX1 = IX + 16
C     PRINT AN = SIGN   (11 = DECIMAL EQUIV FOR =)
      CALL PLOTV(IX1,IY,11)
C     NOW PLOT VALUE THAT CHARACTER EQUALS AFTER SPACE
      IX1 = IX1 + 16
      CALL LABLV (TIMEP(IJP),IX1,IY, 7, 1, 2)
C     MOVE DOWN 1 LINE
      IY = IY - 18
 40   CONTINUE
      IF(IDEBUG.EQ.0) WRITE(6,110)IPLOT,NLTHGH,NLTHGV,NPLPF,NPLT
 110  FORMAT(1H0,31HA PLOT FRAME COMPLETED, VAR ARE, 6I6)
 50   CONTINUE
      GO TO 999
 60   CONTINUE
C
      IF (NPHPLT.NE.0) GO TO 999
C     PRESSURE HISTORY PLOT REQUESTED
C     IU IS NUMBER OF PLOT POINTS
C     ART(IU,1) IS TIME
C     ART(IU,2) IS (PMAX-PMIN)/PAVG FOR TIME ART(IU,1)
C     ART(IU,3) IS PRESSURE AT TIME ART(IU,1)
C     FRAME 1 IS TIME VS (PMAX-PMIN)//PAVG, 1 PLOT PER FRAME
C     FRAME 2 IS TIME VS PRESSURE, 1 PLOT PER FRAME
      CALL MAXMIN(ART(1,1),  ART(1,1),  XL,XR,YT(1),YB(1),L800,M3
     1 ,NONE,3,IJ,1)
C     YT(1),YB(1) = TIME LIMITS
C     YT(2),YB(2) = (PMAX-PMIN)/PAVG LIMITS
C     YT(3),YB(3) = PRESSURE LIMITS
C     **** X IS NOT USED IN THESE PLOTS
      IF(IDEBUG.EQ.0) WRITE(6,140) (I,YT(I),YB(I),I=1,3)
 140  FORMAT(26H0IN PURGE(1),I,YT(I),YB(I)// (I10,2E20.8))
C     OBTAIN GRID AND SCALE FACTORS
      XTIME = FLOAT(NLPHGV)
      XFUNC = FLOAT(NLPHGH)
      IT = 2
      CALL SCAL (YT(1),YB(1),XTIME,DX,XMIN,DZZ)
      CALL SCAL (YT(2),YB(2),XFUNC,DY1,YMIN1,DZZ)
      CALL SCAL (YT(3),YB(3),XFUNC,DY2,YMIN2,DZZ)
      YB(1) = XMIN
      YT(1) = XMIN + DX * XTIME
      YB(2) = YMIN1
      YT(2) = YMIN1 + DY1 * XFUNC
```

```
      YB(3) = YMIN2
      YT(3) = YMIN2 + DY2 * XFUNC
      IF(IDEBUG,EQ,0) WRITE(6,141) (I,YT(I),YB(I),I=1,3)
 141  FORMAT(17H0 ADJUSTED LIMITS// (I10,2E20,8))
C     MARGIN LIMITS REMAIN FROM PRIOR CALL TO SETMIV
C     PLOT (PMAX-PMIN)/PAVG VS T ON FIRST FRAME
      XLL = YB(1)
      XRR = YT(1)
      DELY= DY1
 65   CONTINUE
      NY = +6
      IF (YT(IT),LT,0,1 ,OR, YB(IT),LT,0,1) NY = -4
      CALL GRID1V(3,XLL,XRR,YB(IT),YT(IT),DX,DELY,-0,-0,-1,-1,+6,NY)
      CALL RITE2V(125,990,1020,90,1,40,1,TITLE(1),IER)
C     PLOT FUNCTION
      CALL APLOTV (IU,ART(1,1),ART(1,IT),1,1,1, MRKPT,KK)
C     CONNECT LINES
      IX1 = NXV(ART(1,1))
      IX2 = NXV(ART(2,1))
      IY1 = NYV(ART(1,IT))
      IY2 = NYV(ART(2,IT))
      CALL LINEV(IX1,IY1,IX2,IY2)
      NUP = IJ - 1
      DO 70 IQ = 2,NUP
      IX1 = NXV(ART(IQ,1))
      IX2 = NXV(ART(IQ+1,1))
      IY1 = NYV(ART(IQ,IT))
      IY2 = NYV(ART(IQ+1,IT))
      CALL LINEV(IX1,IY1,IX2,IY2)
 70   CONTINUE
C     LABEL X AXIS
      CALL RITE2V(300,10,1000,90,1,12,-1,12HT IN RADIANS,IER)
C     LABEL Y AXIS
      IF(IT,EQ,2) GO TO 75
      CALL RITE2V(10,300,999,180,1,16,-1,16HPRESSURE HISTORY,IER)
      GO TO 80
 75   CALL RITE2V(10,300,1000,180,1,17,-1,17H(PMAX-PMIN)/PAVGE,IER)
 80   IF (IT,EQ,3) GO TO 998
      IT = 3
      DELY = DY2
      GO TO 65
C     PURGE BUFFERS FOR ALL PREVIOUS PLOTS THIS RUN
 998  CALL PLOTND
 999  RETURN
      END
```

```
      SUBROUTINE SCAL (XMAX,XMIN,XI,DX,X0,XE)
C
C     FORTRAN IV PLOT SCALE OPTIMIZATION ROUTINE.
C
      DIMENSION S(3)
C
      DATA S/1.,2.,5./
C
C
      DX=0.0
      X0=0.0
      XE=0.0
      W=(XMAX-XMIN)/XI
      IF(W)21,21,6
    6 DO 20 I=1,3
      A=1.0+ALOG10(W/S(I))
      B=AINT(A)
      IF((A.EQ.B).OR.(A.LT.0.0))B=B-1.0
      C=S(I)*10.0**B
      D=AINT(XMIN/C)
      IF(C*D.GT.XMIN)D=D-1.0
      D=C*D
      IF(XI*C-XMAX+D)20,15,15
   15 E=W/C
      IF(XE-E)17,20,20
   17 DX=C
      X0=D
      XE=E
   20 CONTINUE
   21 RETURN
      END
```

APPENDIX C

Distribution

NASA Lewis Research Center
21000 Brookpark Road
Cleveland, Ohio 44135
Attn: Dr. R. J. Priem (2)
Dr. C. E. Feiler (2)

NASA Lewis Research Center
21000 Brookpark Road
Cleveland, Ohio 44135
Attn: Norman T. Musial

NASA Lewis Research Center
21000 Brookpark Road
Cleveland, Ohio 44135
Attention: Library (2)

NASA Lewis Research Center
21000 Brookpark Road
Cleveland, Ohio 44135
Attn: Report Control Office

NASA Scientific and Technical
Information Facility
P.O.Box 33
College Park, Maryland 20740 (6)
Attention: NASA Representative

NASA Lewis Research Center
21000 Brookpark Road
Cleveland, Ohio 44135
Attn: E. W. Conrad

NASA Headquarters
6th & Independence Avenue, S.W.
Washington, D.C. 20546
Attn: R. S. Levine, Code RPL

NASA
George C. Marshall Space Flight Ctr.
R-P&VE-PA
Huntsville, Alabama 35812
Attn: R. J. Richmond

NASA
Manned Spacecraft Center
Houston, Texas 77058
Attn: J. G. Thibadaux

AFRPL(RPRRC)
Edwards, California 93523
Attn: B. R. Bornhorst

AFRPL(RPPZ)
Edwards, California 93523
Attn: Capt. C. J. Abbe

Air Force Office of Scientific Research
1400 Wilson Blvd.
Arlington, Virginia 22209
Attn: B. T. Wolfson

Chemical Propulsion Information Agency
8621 Georgia Avenue
Silver Spring, Maryland 20910
Attn: T. W. Christian

U.S. Naval Weapons Center
China Lake, California 93555
Attn: D. Couch

Office of Naval Research
Navy Department
Washington, D. C. 20360
Attn: R. D. Jackel, 429

U.S. Naval Weapons Center
China Lake, California 93555
Attn: E. W. Price, Code 508

U. S. Army Missile Command
AMSMI-RKL, Attn: W. W. Wharton
Redstone Arsenal, Alabama 35808

ARL(ARC)
Attn: K. Scheller
Wright-Patterson AFB
Dayton, Ohio 45433

University of California
Department of Chemical Engineering
Attn: A. K. Oppenhiem
6161 Etcheverry Hall
Berkeley, California 94720

University of California
Mechanical Engineering, Thermal Systems
Attn: Dr. R. F. Sawyer
Berkeley, California 94720

University of California
Aerospace Engineering Department
Attn: F. A. Williams
P.O.Box 109
LaJolla, California 92038

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

Jet Propulsion Laboratory
California Institute of Technology
Attn: R. M. Clayton
4800 Oak Grove Drive
Pasadena, California 91103

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

Dartmouth University
Attn: P. D. McCormack
Hanover, New Hampshire 03755

Georgia Institute of Technology
Aerospace School
Attn: B. T. Zinn
Atlanta, Georgia 30332

Illinois Institute of Technology
Rm 200 M.H.
Attn: T. P. Torda
3300 S. Federal Street
Chicago, Illinois 60616

Massachusetts Institute of Technology
Dept. of Mechanical Engineering
Attn: T. Y. Toong
Cambridge, Massachusetts 02139

University of Michigan
Aerospace Engineering
Attn: J. A. Nicholls
Ann Arbor, Michigan 48104

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

The Pennsylvania State University
Mechanical Engineering Department
Attn: G. M. Faeth
207 Mechanical Engineering Boulevard
University Park, Pa. 16802

Princeton University
James Forrestal Campus Library
Attn: I. Glassman
P.O.Box 710
Princeton, New Jersey 08540

Princeton University
James Forrestal Campus Library
Attn: D. Harrje
P.O.Box 710
Princeton, New Jersey 08540

Purdue University
School of Mechanical Engineering
Attn: J. R. Osborn
Lafayette, Indiana 47907

Sacramento State College
School of Engineering
Attn: F. H. Reardon
60000 J. Street
Sacramento, California 95819

Purdue University
Jet Propulsion Center
Attn: R. Weiss
W. Lafayette, Indiana 47907

University of Southern California
Attn: M.Gerstein, Dept.Mechanical Eng.
University Park
Los Angeles, California 90007

University of Wisconsin
Mechanical Engineering Department
Attn: P. S. Myers
1513 University Avenue
Madison, Wisconsin 53705

Aerojet-General Corporation
Attn: R. McBride
P.O.Box 296
Dept. 4921 Bldg. 160
Azusa, California 91703

Aerospace Corporation
Attn: O. W. Dykema
P.O.Box 95085
Los Angeles, California 90045

Bell Aerosystems Company
Attn: L. M. Wood
P. O. Box 1
Mail Zone J-81
Buffalo, New York 14205

Dynamic Science, a Division
of Marshall Industries
Attn: B. P. Breen
1900 Walker Avenue
Monrovia, California 91016

Multi-Tech., Inc.
Attn: F. B. Cramer
601 Glenoaks Blvd.
San Fernando, California 91340

Pratt & Whitney Aircraft
Florida Research & Development Ctr.
Attn: G. D. Garrison
P.O. Box 2691
West Palm Beach, Florida 33402

Rocketdyne
A Division of North American
Aviation
Attn: E. C. Clinger
6633 Canoga Avenue
Canoga Park, California 91304

Rocketdyne
A Division of North American
Aviation
Attn: R. B. Lawhead
6633 Canoga Avenue
Canoga Park, California 91304

TRW Systems
Attn: G. W. Elverum
1 Space Park
Redondo Beach, California 90278