MR-6906

Nonlinear Combustion Instability: Computer Experiments Using Codes COMB and TRDL

by

Samuel Burstein

Harold Schechter

Final Report

to

Jet Propulsion Laboratory

Contract No. 952505

July, 1969

RECEIVED FEB 1971 NASA STI FACILITY INPUT BRANCH

Reproduced by: NATIONAL TECHNICAL INFORMATION SERVICE Springfield, Va. 22151

Mathematical Applications Group, Inc.

180 South Broadway

White Plains, New York

FACILITY FORM 602

N71-33572 (ACCESSION NUMBER)	(THRU)
253 (PAGES)	G3 (CODE)
CR 121 472 (NASA CR OR TMX OR AD NUMBER)	33 (CATEGORY)

Nonlinear Combustion Instability: Computer Experiments Using Codes COMB and TRDL

by

Samuel Burstein

Harold Schechter

Final Report

to

Jet Propulsion Laboratory

Contract No. 952505

July, 1969

"This work was performed for the Jet Propulsion Laboratory, California Institute of Technology, as sponsored by the National Aeronautics and Space Administration under Contract NAS7-100."

"This report contains information prepared by Mathematical Applications Group, Inc. (MAGI) under JPL subcontract. Its content is not necessarily endorsed by the Jet Propulsion Laboratory, California Institute of Technology, or the National Aeronautics and Space Administration."

TABLE OF CONTENTS

I. INTRODUCTION

This report presents results from a series of diversified calculations using mathematical models developed during Contract No. 951946 to the Jet Propulsion Laboratory (Reference 1). These models resulted in two basic computer programs describing two-dimensional time dependent hydrodynamic motion of a reacting system. A pancake model is used to compute the motion of a gas dynamic field in the transverse plane of an infinite cylinder while the second model, an annular model, describes the gas dynamic motion in an annular field bounded by a fuel injector plate and a subsonic-supersonic nozzle. The independent coordinates of the pancake model are r-θ-t while the independent coordinates of the annular model are z-θ-t.

In addition to using the above models several additional programs were written to supply calculational ability for some of the problem areas that required specialized techniques not available in the two basic computer programs. Here we are referring to the blast measurements of J. M. Bonnell (Reference 2).

The first section of this report describes the tactics used to compute a blast wave in the transverse plane. Two calculations were attempted. The more difficult is a complete two-dimensional time dependent calculation of a blast center located near the periphery in the r-θ plane - the second was a calculation of a blast center located at the geometric center of the r-θ motor. The second calculation is compared to the experimental work of Bonnell.

The second section describes computer experiments performed with the pancake model for finite amplitude waves of the first transverse mode (sloshing). A second interesting calculation perturbs the finite amplitude transverse model by introducing solid body rotation corresponding to a small part of the total energy of the system.

The last section describes computer experiments carried out with the annular model. Here finite amplitude waves are allowed to interact with a droplet field that is adding energy to the system. Two types of finite amplitude disturbances are considered - solutions to the wave equation in θ-z-t coordinates and localized pressure disturbances called "pops".

II. BLAST WAVE CALCULATIONS

The problem of building stably operating liquid propellant rocket engines, engines which do not exhibit the spontaneous appearance of large amplitude pressure waves, is still pertinent. One of the methods used to rate the stability of a given rocket engine configuration is to try to "trigger" the steady flow field into a resonance condition by detonating a certain measure of explosive. Larger and larger weights of explosive are used in a progression of tests until a sustained oscillation results. However, once the test goes beyond a defined size of explosive bomb, one may not be interested in the results since that artificially induced oscillation would result from a disturbance that would occur with only a small degree of probability. Such disturbances might occur naturally through the random generation of exceptionally large slugs of fuel. Although the position at which such explosive pockets appear will be random, the generation of the resulting sustained oscillation does not appear to depend in any fundamental way on the driving source.

The tests described in Reference 2, Table I are the main concern of this section. Bonnell ignited a bomb on the axial centerline of a cylindrical chamber 5.523 inches in radius. Measurements were taken of pressure at specified positions along a radius. The time of arrival of the leading edge of the disturbance was recorded. From this primative data he derived secondary information about velocity and Mach number of the shock wave.

If assumptions are made as to the self-similarity of the motion and as to the blast geometry - cylindrical or spherical - the similarity parameter energy per unit of density can be derived from radius-time measurements. The usefulness of obtaining such a quantity is related to the question of estimating the energy release of the explosive generating the blast. This is of extreme importance in nuclear blast measurements. In rocket motor calculations it may lead to an estimate of the necessary magnitude of energy slugs needed to trigger instability. The formulas used by Bonnell to estimate these energy levels were derived under the assumption that the pressure ratio across the shock tended to infinity, i.e., infinite shock strength or equivalently the pressure in front of the shock $p_1 = 0$. In this sense, the spherical formulas

$$R = \text{const} \cdot t^{2/5} \quad , \quad S = \text{const} \cdot R^{-3/2} \tag{2.1}$$

and the cylindrical formulas

$$R = \text{const} \cdot t^{1/2} \qquad S = \text{const} \cdot R^{-1} \tag{2.2}$$

are exact solutions to the one-dimensional blast wave problem. Here R and S are the shock wave radius and velocity respectively. Each of the above constants depends only upon the ratio E/ρ_1, where $E \sim E_o$ the total blast energy and ρ_1 the density of the fluid into which the shock is propagating.

In the smooth part of the flow field the flow is determined by the equation of mass, momentum and energy or entropy conservation

$$\frac{\partial \rho}{\partial t} + \frac{\partial \rho u}{\partial r} + (\nu-1)\frac{\rho u}{r} = 0$$

$$\frac{\partial \rho u}{\partial t} + \frac{\partial(\rho u^2+p)}{\partial r} + (\nu-1)\frac{\rho u^2}{r} = 0 \tag{2.3}$$

$$\frac{\partial}{\partial t}(p\rho^{-\gamma}) + u\frac{\partial}{\partial r}(p\rho^{-\gamma}) = 0$$

The last equation expresses the fact that the entropy of a particle is constant in the space time field. The index $\nu = 3$ for spherical symmetric flow, $\nu = 2$ for cylindrical symmetry and $\nu = 1$ for plane flow.

The smooth part of the flow is connected to the constant state with sound speed a_1, into which the shock is propagating with speed S through the Rankine-Hugoniot conditions

$$u_2 = \frac{2}{\gamma+1} S\left(1-\frac{a_1^2}{S^2}\right)$$

$$p_2 = \frac{2}{\gamma+1}\rho_1 S^2\left(1-\frac{\gamma-1}{2\gamma}\frac{a_1^2}{S^2}\right) \tag{2.4}$$

$$\rho_2 = \frac{\gamma+1}{\gamma-1}\rho_1\left(1+\frac{2}{\gamma-1}\frac{a_1^2}{S^2}\right)^{-1}$$

The intense explosion limit $a_1/S \to 0$ yields the familiar value for the density discontinuity $\rho_2/\rho_1 \to \gamma+1/\gamma-1$.

The fundamental parameters are r, t, p_1, ρ_1, E with E a constant whose units are energy per unit area in the spherical case, energy per unit length in the cylindrical case and the energy in the plane case. If one considers the non-dimensional number

$$\tau = \frac{p_1^{5/6} t}{E^{1/3} \rho_1^{1/2}}$$

that can be formed from the fundamental parameters of the problem then in the limit as $t, p_1 \to 0$ we see that τ vanishes when compared to a second dimensionless grouping given below.

It is possible to introduce another dimensionless parameter λ such that

$$\lambda = \frac{r}{\left(\frac{E}{\rho_1}\right)^{1/(2+\nu)} t^{2/(2+\nu)}} \tag{2.5}$$

We can arbitrarily set $\lambda = 1$; then Equations (2.1) and (2.2) follow by setting $\nu = 3$ and $\nu = 2$ respectfully. Here when $\lambda =$ $R = r$ is the position of the shock while $\dot{r} = dr/dt = S$ is the shock speed.

We wish to carry out all our computations in the transverse plane of a right circular cylinder, and since it is reasonable to assume that the explosive will behave as a linear charge at least for small time. Equations (2.3) with $\nu = 2$ become

$$\rho_t + (\rho u)_r + \frac{\rho u}{r} = 0$$

$$(\rho u)_t + (\rho u^2+p)_r + \frac{\rho u^2}{r} = 0 \tag{2.6}$$

$$(p\rho^{-\gamma})_t + u(p\rho^{-\gamma})_r = 0$$

We now use the relation

$$\frac{\partial}{\partial t}(\ ) = \dot{r}\,\frac{\partial}{\partial r}(\ )$$

to replace the System (2.6) by

$$(\dot{r}+u)\ \rho_r/\rho + u_r = -\,u/r$$

$$(\dot{r}+u)\ u_r + p_r/\rho = 0 \tag{2.7}$$

$$\frac{p_r}{p} - \gamma\,\frac{\rho_r}{\rho} = 0$$

These equations may be rewritten as the system

$$p_r/p = -\gamma(S+u)u/\left[(S+u)^2 - \gamma\ p/\rho\right]r$$

$$\rho_r/\rho = \frac{1}{\gamma}\,p_r/p \tag{2.8}$$

$$u_r/u = -\frac{1}{r} - (S+u)\ \rho_r/\rho u$$

where we have used $S = \dot{r} = \frac{dr}{dt} = \frac{1}{2}\,\frac{r}{t}$ from Equation (2.5).

The problem of computing a blast wave then consists in fixing the shock radius and computing the shock speed. This will be discussed later on. The jump conditions, Equations (2.4), are used to determine the conditions on the boundary of the smooth part of the flow. Equations (2.8) use these values as starting conditions (to be substituted on the right hand side of (2.8)). Integration of System (2.8) proceeds until $r \to 0$. The condition $u(r)=0$ at $r=0$ is satisfied. This calculation then is used to compute

$$p(r),\ \rho(r) \text{ and } u(r) \qquad\qquad 0 \leq r \leq r_o$$

where $r_o \sim 0.2 \cdot r$ chamber. The values of $p(r)$, $\rho(r)$ and $u(r)$ $r_o \leq r \leq r$ chamber are the quiescent conditions given in the tests of Reference (2). This calculation defines a time $t_o = r_o/S$ which is the time when the calculation is continued by program COMB (Ref. 1). In actual operation it was found that a more practical approach was to modify COMB to take into account the polar symmetry of the calculation. Hence for bombs located at the geometric centerline of the combustion chamber, in which the tangential velocity component vanishes, a special version of COMB is used.

This approach differs from the approach taken by Goldstine and Von Neumann (Ref. 3) in their paper on the calculation of blast waves.

The main difference between our approach and Reference (3) is that the latter paper maintains the sharp discontinuity of the blast wave by iteratively satisfying the coupling between the jump conditions across the discontinuity and the differential equations behind the discontinuity. Our approach allows the shock to be smeared by the difference equations but preserving the proper shock strength and propagation velocity, i.e., the correct jump conditions. Of course, the same difference equations are used to compute the smooth flow field behind the shock. The difference equations for the complete two-dimensional time dependent problem are given in Reference (1). By reprogramming, only one ray is used in the code for the case when the tangential velocity component vanishes. When the blast center is not at the geometric center of the chamber, a self-similar calculation would only be valid until the shock wave was reflected from the chamber wall. For the conditions of this report that would be the instant t_o. Hence, except for the initial condition calculation described above, the entire calculation must be a two-dimensional time dependent one with symmetry about the angle $\theta=\theta_o$, the centerline of the bomb, and with no simplification due to self-similarity of the fluid field.

In essence, the method that is described above was a successful computational algorithm. The pressure gradients that are generated from the bomb blast were extremely large. In order to compute the resultant time dependent motion, due to the blast, a finer mesh than that required for the calculations in Reference (1) was built into the code. Whereas 10 to 20 grid points in the radial direction were used in COMB, this calculation required up to 500 zones.

The main results of cold bomb tests presented in Reference (2) are contained in Table 1 of that report. Figure (1.1) shows comparison between Test 3 of Reference (2) and several sets of initial data. It was found that if System (2.8) were integrated subject to the condition that the pressure ratio $p_1/p(o)=20$, where $p(o)$ is the pressure at $r=0$ the resultant pressure levels behind the shock were too low at the initial portion of the calculation and there was little decay in the shock in the latter stage up to the point at which the shock strikes the chamber wall. If one took $p_1/p(0.681)=20$ as the boundary condition, the resultant pressure ratio as a function of radius is shown in the upper curve. These two conditions are met by choosing the proper starting shock strength S. By using the shock conditions (2.4), the initial data for Equations (2.8) is obtained and integration proceeds until the desired pressure ratio is achieved. This procedure can be used iteratively to construct the desired pressure ratio.

Figure (1.1) shows the calculation of three shock waves of strength $p_2/p_1=5,15$ and 20 for a 9.02 inch radius chamber. This pressure ratio is measured at $r=1.0$ inch. The self-similar solution is shown in solid line while the time dependent numerical solution that follows are in dashed lines. At this writing it is not clear why the upper curve has a break. It is related to the problem of generating insufficient printouts of data concerning the maximum pressure ratio versus distance and time that was generated by the program. Figure (1.2) is for a 5.52 inch chamber.

It eliminated the problem of Figure (1.1) because more closely selected points were chosen. In this case data was available from Bonnell for the cold chamber bomb case and can be compared. It appears that the case $p_2/p_1=15$ is a reasonable fit to the data. In Figure (1.3) the wave arrival times are compared. Our curves are for actual arrival times while Bonnell used an average value over an inteval. Table (1) are computed arrival times for the 9.02 inch motor.

The calculation of a bomb pulse placed near the chamber walls was not attempted due to the rather large storage requirements that were required even for the one-dimensional problem presented above. However, the listing of the code for the eccentric bomb is included in the appendix. The approach used to generate the initial data is the main difference between the two bomb codes. For the eccentric code the initial data is generated by the following procedure.

A table of values for the density, pressure and radial velocity as functions of radial distance from a bomb center are given. These values are for a coordinate system located at the center of the blast, and hence, the tangential velocity is 0. This bomb center is placed at a given point (r,θ) of the chamber. The maximum radius, r_B, of the blast is 1/5 the radius of the chamber.

Conditions in the coordinate system of the chamber are obtained as follows. If a point is a distance greater than r_B from the center of the blast it is assigned the initial conditions of the undisturbed gas. If it is within a distance r_B of the center of the blast then the density, pressure, and radial velocity at the point in the coordinate system of the blast are found by linear interpolation along the radius from the given tables. The pressure and density are scalar quantities and hence independent of the coordinate system. The velocity (u,v) in the coordinate system of the motor is found by transforming the velocity $(\bar{u},\bar{v})$ in the blast coordinate system to a cartesian coordinate system with the same center. The cartesian velocity (u',v') remains the same when the center of the coordinate system is translated to the center of the motor. The velocity (u',v') is then transformed to the polar coordinate system with center at the center of the motor. The formulas that are used relating the above quantities are given by System (2.9):

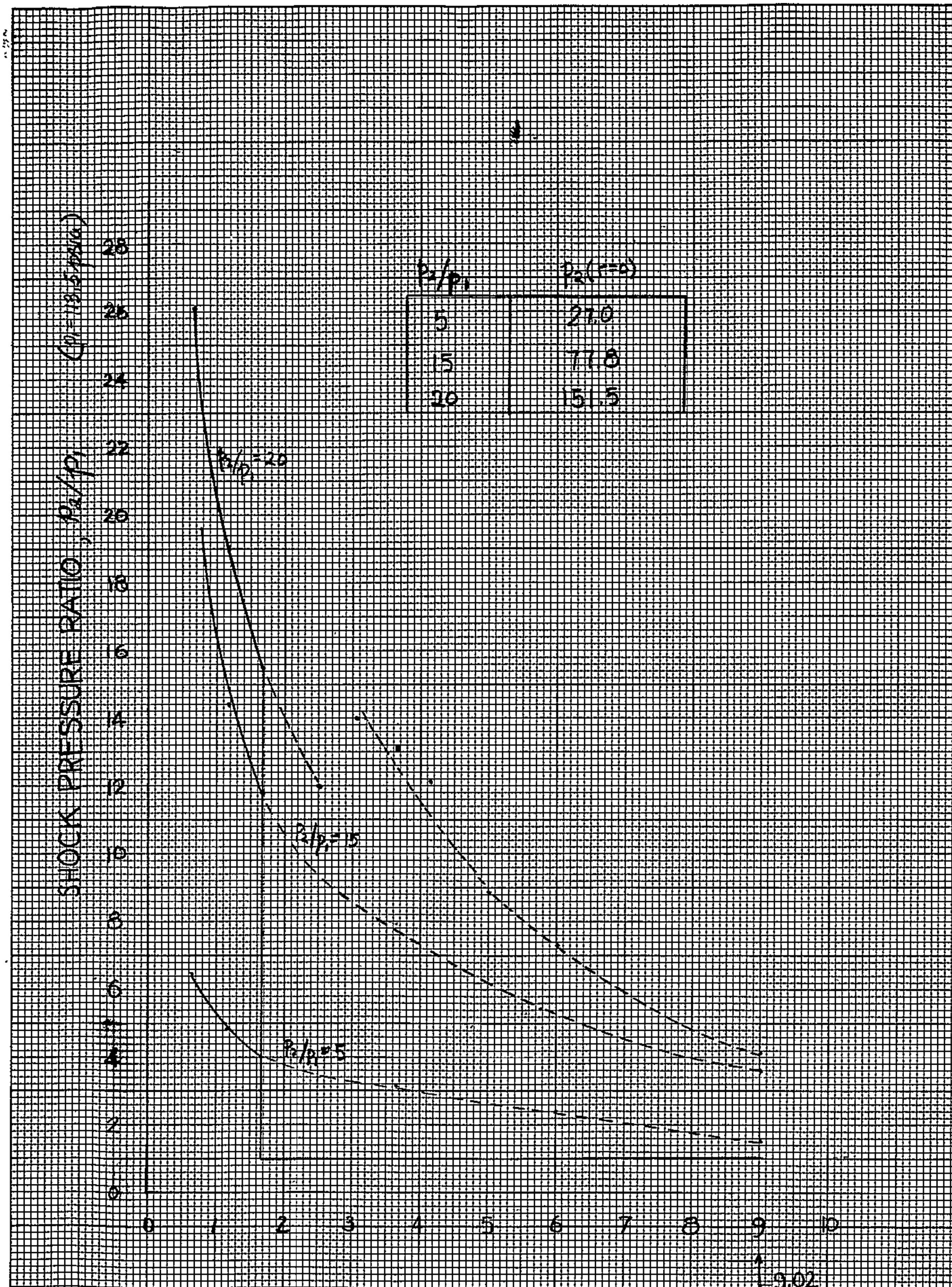
$$\begin{aligned}
u' &= \bar{u} \cos \bar{\theta} - \bar{v} \sin \bar{\theta} = \bar{u} \cos \bar{\theta} \\
v' &= \bar{u} \sin \bar{\theta} + \bar{v} \cos \bar{\theta} = \bar{u} \sin \bar{\theta} \\
u &= u' \cos \theta + v' \sin \theta \\
v &= u' \sin \theta + v' \cos \theta
\end{aligned} \tag{2.9}$$

TABLE 1

Shock Wave Arrival Times for

9.02 Inch Motor (Microseconds)

p_1/p_2 R, inches	5	15	20
3.15	---	---	65
3.681	155	91	79
5.0	---	157	115
6.0	269	---	141
9.02	418	251	225



SHOCK PRESSURE RATIO, p_2/p_1
($p_1 = 13.5$ psia)
RADIAL DISTANCE r, in.
p_2/p_1
$p_2(r=0)$
5
27.0
15
77.8
20
151.5
$p_2/p_1 = 20$
$p_2/p_1 = 15$
$p_2/p_1 = 5$
9.02

FIG. 1.1

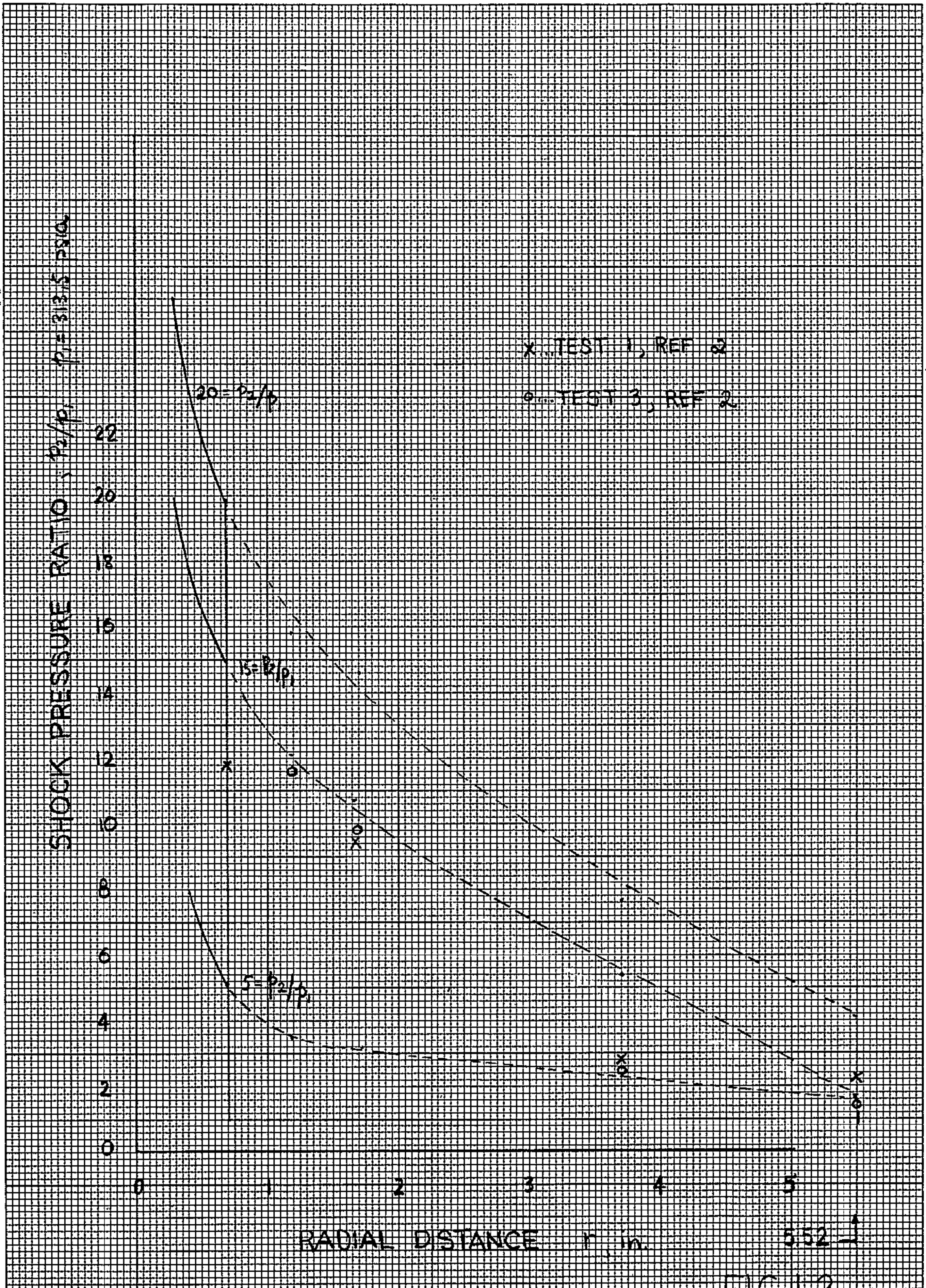
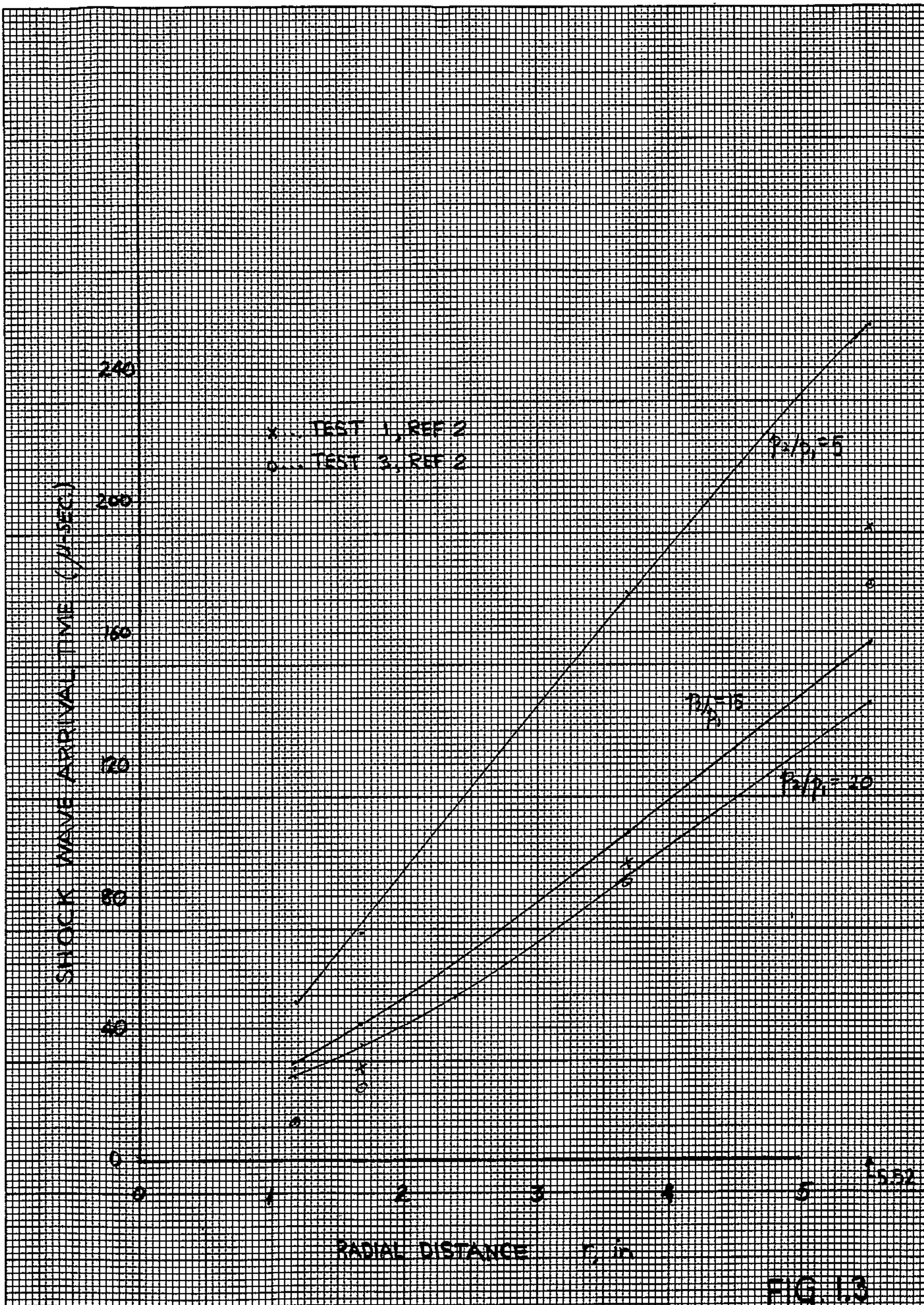

SHOCK PRESSURE RATIO, p_2/p_1 $p_1 = 313.5$ psia
x...TEST 1, REF 2
o...TEST 3, REF 2
$20 = p_2/p_1$
$15 = p_2/p_1$
$5 = p_2/p_1$
22
20
18
16
14
12
10
8
6
4
2
0
0
1
2
3
4
5
RADIAL DISTANCE r, in.
5.52

FIG. 1.2



x... TEST 1, REF 2
o... TEST 3, REF 2
$p_2/p_1=5$
$p_2/p_1=15$
$p_2/p_1=20$
SHOCK WAVE ARRIVAL TIME (μ-SEC.)
240
200
160
120
80
40
0
0
1
2
3
4
5
5.52
RADIAL DISTANCE r, in

FIG. 1.3

III. FIRST TRANSVERSE MODE INTERACTION

In this section we describe a calculation of the motion of a transverse pressure wave in the plane z=constant of a right circular cylinder. The value of the maximum pressure is initially 450 psia while the base chamber pressure is 300 psia. The basic set of partial differential equations that are solved by finite differences are

$$\rho_t + \frac{1}{r}(r\rho u)_r + \frac{1}{r}(\rho v)_\theta = 0 \tag{3.1}$$

$$(\rho u)_t + \frac{1}{r}(r(\rho u^2+p))_r + \frac{1}{r}(\rho uv)_\theta = \frac{p+\rho v^2}{r} \tag{3.2}$$

$$(\rho v)_t + \frac{1}{r}(r\rho uv)_r + \frac{1}{r}(\rho v^2+p)_\theta = \frac{-\rho uv}{r} \tag{3.3}$$

$$E_t + \frac{1}{r}((E+p)u)_r + \frac{1}{r}((E+p)v)_\theta = 0 \tag{3.4}$$

Here ρ, u, v and E are the mass per unit volume, the radial velocity component, the tangential velocity component and the total energy per unit volume, i.e.

$$E = \rho(e+\tfrac{1}{2}(u^2+v^2))$$

The pressure p is related to the specific internal energy e by the equation of state

$$p = P(e,\rho)$$

We expand Equation (3.2) by carrying out the indicated differentiation, i.e.

$$\rho u_t + u\rho_t + \dots$$

and subtract Equation (3.1) premultiplied by u from the above to obtain

$$u_t + uu_r + \frac{v}{r} u_\theta = -\frac{1}{\rho} p_r + \frac{v^2}{r} \tag{3.5}$$

The same procedure is applied to Equation (3.3) with v replacing u when Equation (3.1) is premultiplied. The tangential momentum equation takes the form

$$v_t + uv_r + \frac{v}{r} v_\theta = -\frac{1}{\rho} p_\theta - \frac{uv}{r} \tag{3.6}$$

The continuity equation can be expanded by performing the indicated differentiation and collecting the terms in the following way

$$\left(\frac{\partial}{\partial t} + u \frac{\partial}{\partial r} + \frac{v}{r} \frac{\partial}{\partial \theta}\right) \rho + \rho \left(\frac{1}{r} \frac{\partial (ru)}{\partial r} + \frac{1}{r} \frac{\partial v}{\partial \theta}\right) = 0 \tag{3.7}$$

The operator in the first parenthesis is the substantial or particle derivative in polar coordinates $\frac{D}{Dt}$; the second term is the divergence of the fluid velocity in polar coordinates. Now, one writes the variables as the sum of a mean value and perturbation value

$$\begin{aligned} \rho &= \bar{\rho} + \rho' \\ p &= \bar{p} + p' \\ u &= \bar{u} + u' \\ v &= \bar{v} + v' \end{aligned} \tag{3.8}$$

and substitutes these values into Equations (3.5), (3.6), and (3.7). We take the value of $\overline{u}=\overline{v}=o$ which implies that the state about which the perturbation occurs is a state of rest. The perturbed quantities, designated by primes, is assumed to be much smaller than the average quantities denoted by the bars. Under this assumption, we neglect quantities of second order in smallness so that Equations (3.5), (3.6) and (3.7) become

$$u'_t = - \frac{1}{\overline{\rho}} p'_r \tag{3.9}$$

$$v'_t = - \frac{1}{\overline{\rho}} p'_\theta \tag{3.10}$$

$$\rho'_t + \overline{\rho} \ (u'_r+\frac{1}{r}u'+\frac{1}{r}v'_\theta \ ) = 0 \tag{3.11}$$

The perturbed component of pressure and density can be related through the mean sound speed $\overline{c}$ by

$$\overline{c}^2 = p'/\rho'$$

so that Equation (3.11) becomes

$$p'_t = - \overline{\rho c}^2 \ \mathrm{div}\, \underset{\sim}{u} \ , \ \underset{\sim}{u} = \begin{pmatrix} u' \\ v' \end{pmatrix} \tag{3.12}$$

where div is the divergent operator defined from Equation (3.7). Equations (3.9) and (3.10) state that the components of the perturbed velocity vector u is given by the gradient of the scalar perturbation pressure p'. The entropy of the system is constant which means that the curl of the perturbed velocity vector is zero. This implies the existence of a potential ψ such that

$$u' = \psi_r \qquad v' = \frac{1}{r} \psi_\theta \tag{3.13}$$

Then the perturbed pressure given from Equation (3.12) becomes

$$p'_t = -\overline{\rho c}^2 \nabla^2 \psi \tag{3.14}$$

where

$$\nabla^2 \psi = (\psi_{rr} + \frac{1}{r} \psi_r + \frac{1}{r^2} \psi_{\theta\theta})$$

Comparing Equations (3.13) with Equations (3.9) and (3.10) we conclude that

$$\psi_t = - \frac{1}{\bar{\rho}} p'$$

or

$$\psi_{tt} = - \frac{1}{\bar{\rho}} p'_t$$

which together with Equation (3.14) results in the wave equation for the potential ψ:

$$\psi_{tt} = \bar{c}^2 \nabla^2 \psi \tag{3.15}$$

This equation is subject to the boundary condition of the vanishing of the normal velocity,

$$\psi_r = 0 \tag{3.16}$$

on the boundary of the region in which Equation (3.15) is defined.

The solution to Equation (3.15) subject to condition (3.16) is

$$\psi = \frac{\bar{c}}{k} \varepsilon J_1(kr) \sin (k\bar{c}t) \cos \theta \tag{3.17}$$

with the constant k determined from the condition

$$\frac{\partial}{\partial r} J_1(kr) = 0 \qquad \text{on the boundary}$$

The parameter ε is used to control the amplitude of ψ. The prescribed initial data derived from Equation (3.17) is:

$$\begin{aligned} p &= \bar{p} + p' = \bar{p} - \bar{\rho}\psi_t \\ &= \bar{p}\ (1-\gamma\varepsilon J_1(kr)\ \cos\ (k\bar{c}t)\ \cos\ \theta) \\ u &= u' = \psi_r = \bar{c}\varepsilon\ J_{1_r}(kr)\ \sin\ k\bar{c}t\ \cos\ \theta \\ v &= v' = \frac{1}{r}\psi_\theta = -\ \bar{c}\varepsilon\ \frac{J_1(kr)}{kr}\ \sin\ k\bar{c}t\ \sin\ \theta \end{aligned} \tag{3.18}$$

The initial conditions are defined at time t=0; hence the first transverse mode is initiated by defining the fluid field by allowing t to be zero in System (3.18) to obtain

$$\begin{aligned} p(r,\theta,t) &= \bar{p}\ (1-\varepsilon\gamma J_1(kr)\ \cos\ \theta) \\ u(r,\theta,0) &= \bar{v}\ (r,\theta,0) = 0 \\ (r,\theta,t) &= \bar{p}(r,\theta,t)\ ^{1}/_{\gamma} \end{aligned} \tag{3.19}$$

The pressure and density conditions of System (3.19) are shown plotted in Figure (2.1). One could have started the calculation with the equivalent system

$$
\begin{aligned}
p(r,\theta,\tfrac{\pi}{2k\bar{c}}) &= \bar{p} \\
u(r,\theta,\tfrac{\pi}{2k\bar{c}}) &= \bar{c}\varepsilon J_{1_r}(kr)\cos\theta \\
v(r,\theta,\tfrac{\pi}{2k\bar{c}}) &= -\bar{c}\varepsilon\,\frac{J_1(kr)}{kr}\,\sin\theta \\
\rho(r,\theta,\tfrac{\pi}{2k\bar{c}}) &= \bar{p}^{\,1/\gamma}
\end{aligned}
\tag{3.20}
$$

In following the time dependent behavior of the pressure and density field of system (3.18), after a time inteval $\Delta t=\pi/2k\bar{c}$ they translate to that of system (3.20), i.e., these two apparently different specifications just specify the same transverse motion at somewhat different times.

The next sequence of figures show the time dependent history of the first transverse (sloshing) mode with a finite amplitude, where $\varepsilon=0.5$ and $\gamma=1.2$. The Figures (2.1) to (2.30) are selected print-outs of the calculation. There is complete symmetry about $\theta=0$ and π for the duration of the calculation out to 2000 cycles and corresponding to about 1.5 milliseconds of real time motion of the flow field. There is no indication of any net rotational motion appearing up to the point at which the calculation was terminated. The calculation shows that pressure variations induce a velocity field which is seen by following a particle path as in Figure (2.19). The important thing to notice is that the scale in $y(\theta=0)$ $\sim 10^{-5}$. Hence, since the range of motion in x is like 0.1, the streakline plots indicate purely one-dimensional motion. As Figure (2.25) indicates, the peak pressure on the chamber boundary is almost 460 psia, very close to the starting value. This indicates, at least in the absence of shock waves, that the follow will continue to oscillate interchanging potential energy in the form of pressure with kinetic energy and back again.

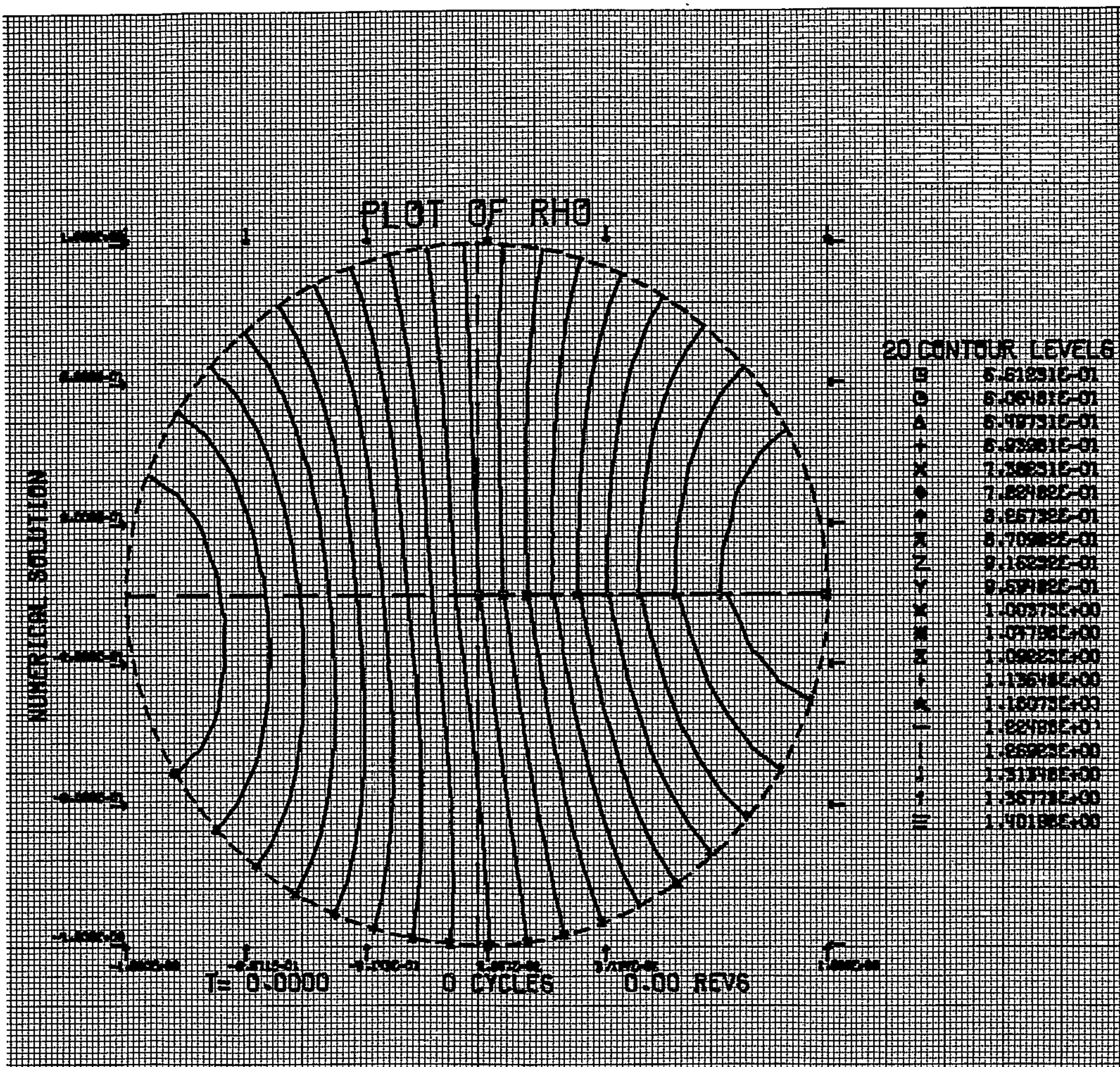

PLOT OF RHO
NUMERICAL SOLUTION
20 CONTOUR LEVELS
T= 0.0000
0 CYCLES
0.00 REVS

FIG. 2.1

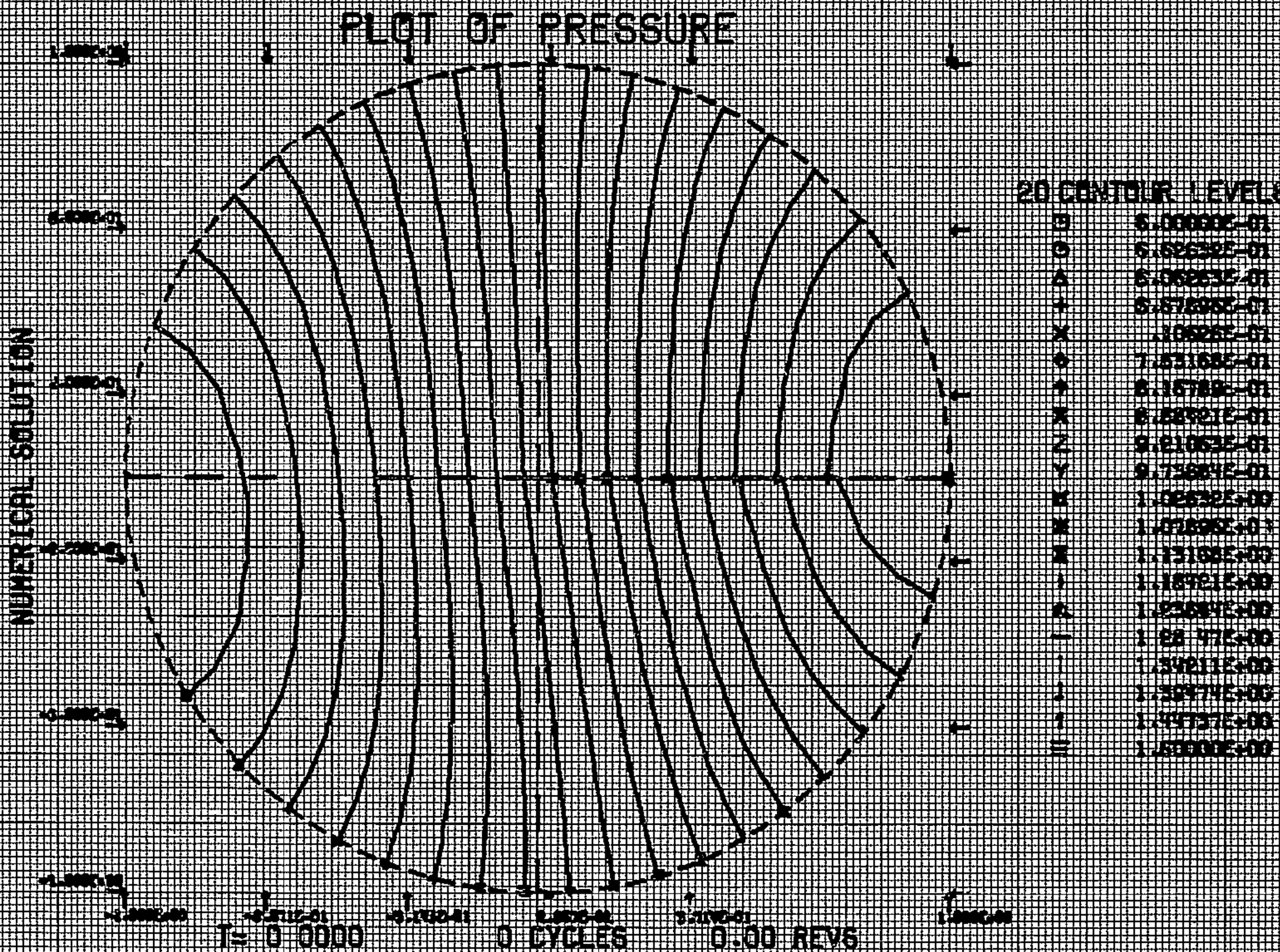


FIG. 2.2

(CALCOM 02)

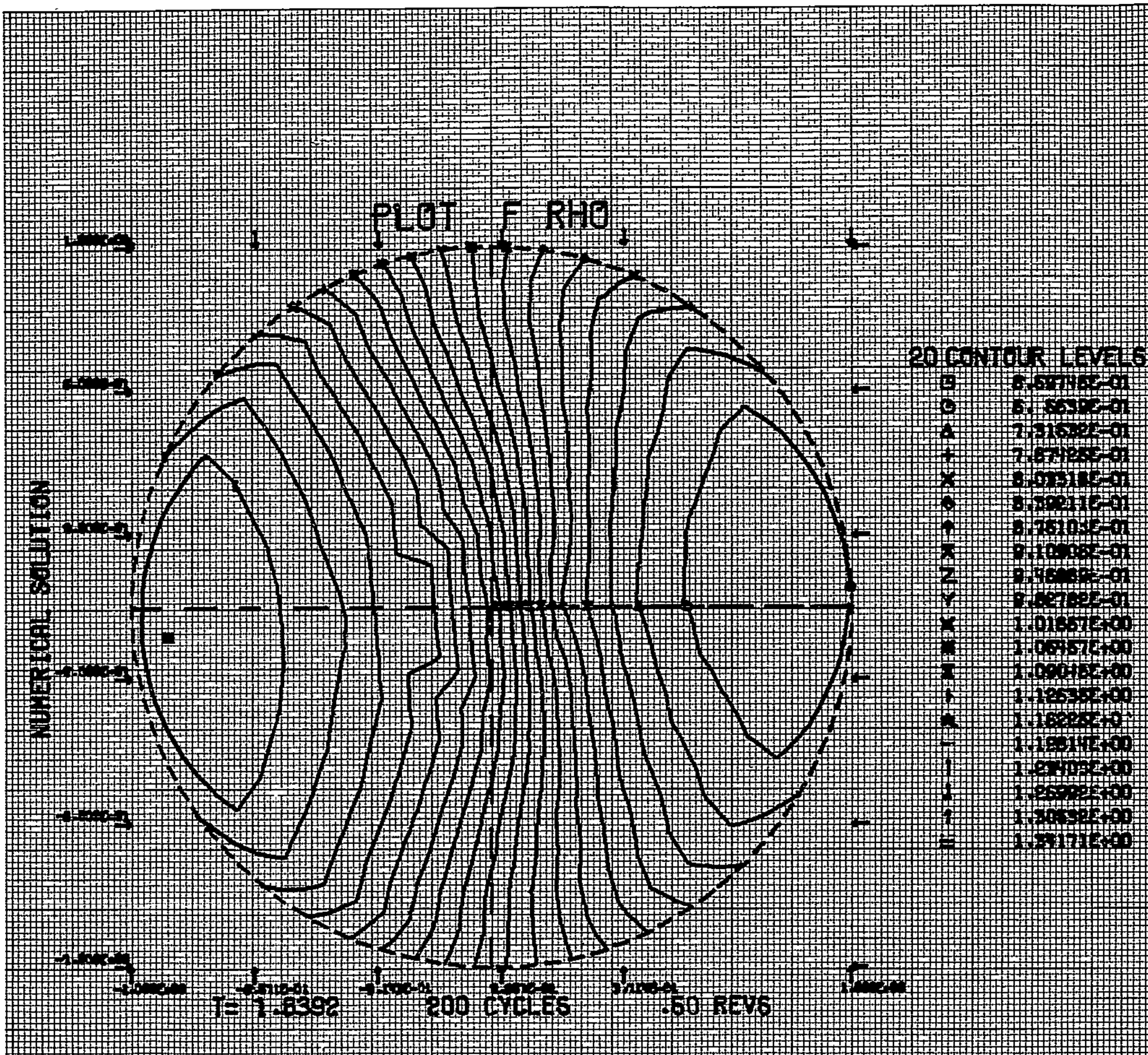

PLOT F RHO
NUMERICAL SOLUTION
20 CONTOUR LEVELS
T= 1.8392
200 CYCLES
.50 REVS

FIG 2.3

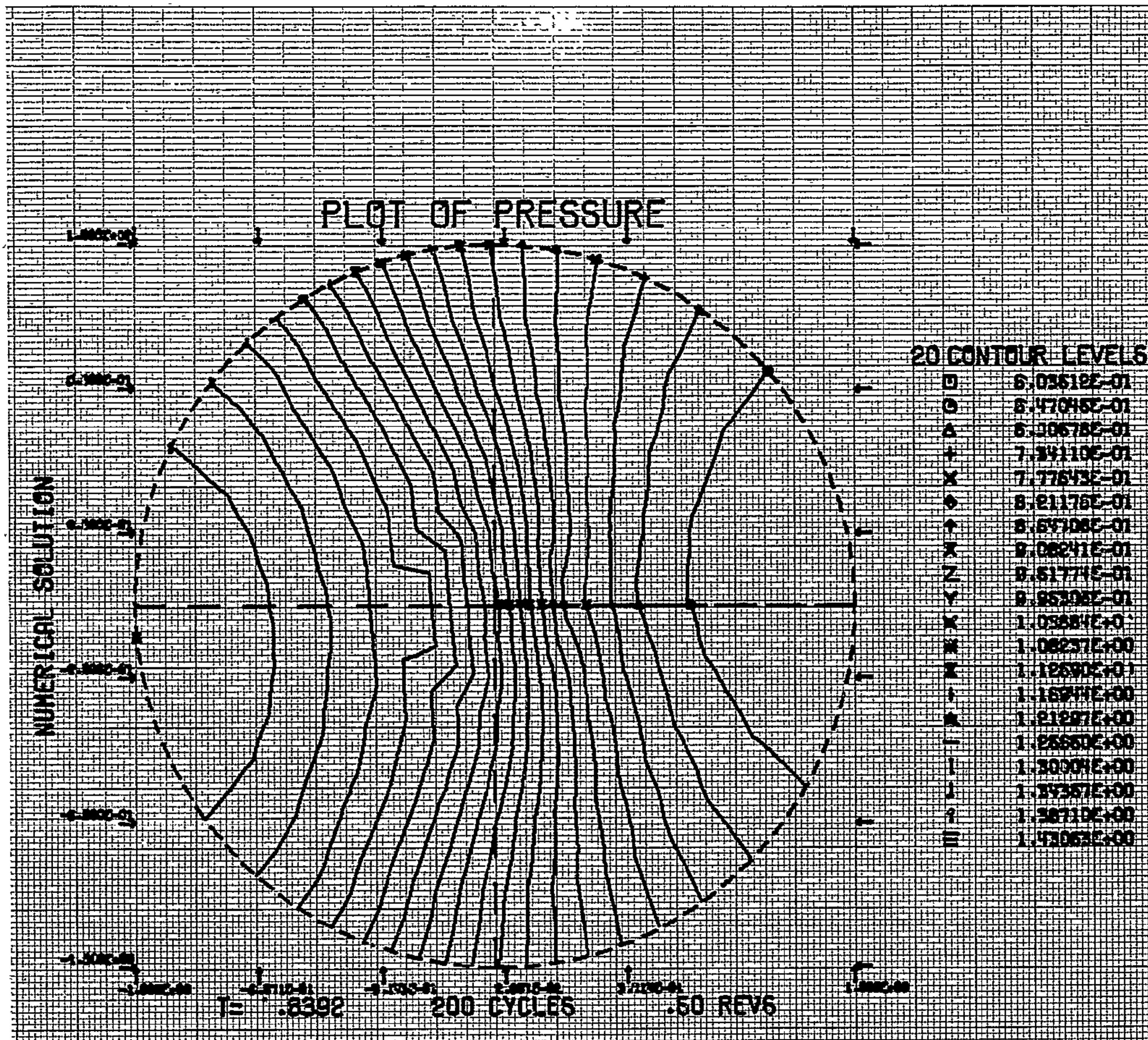
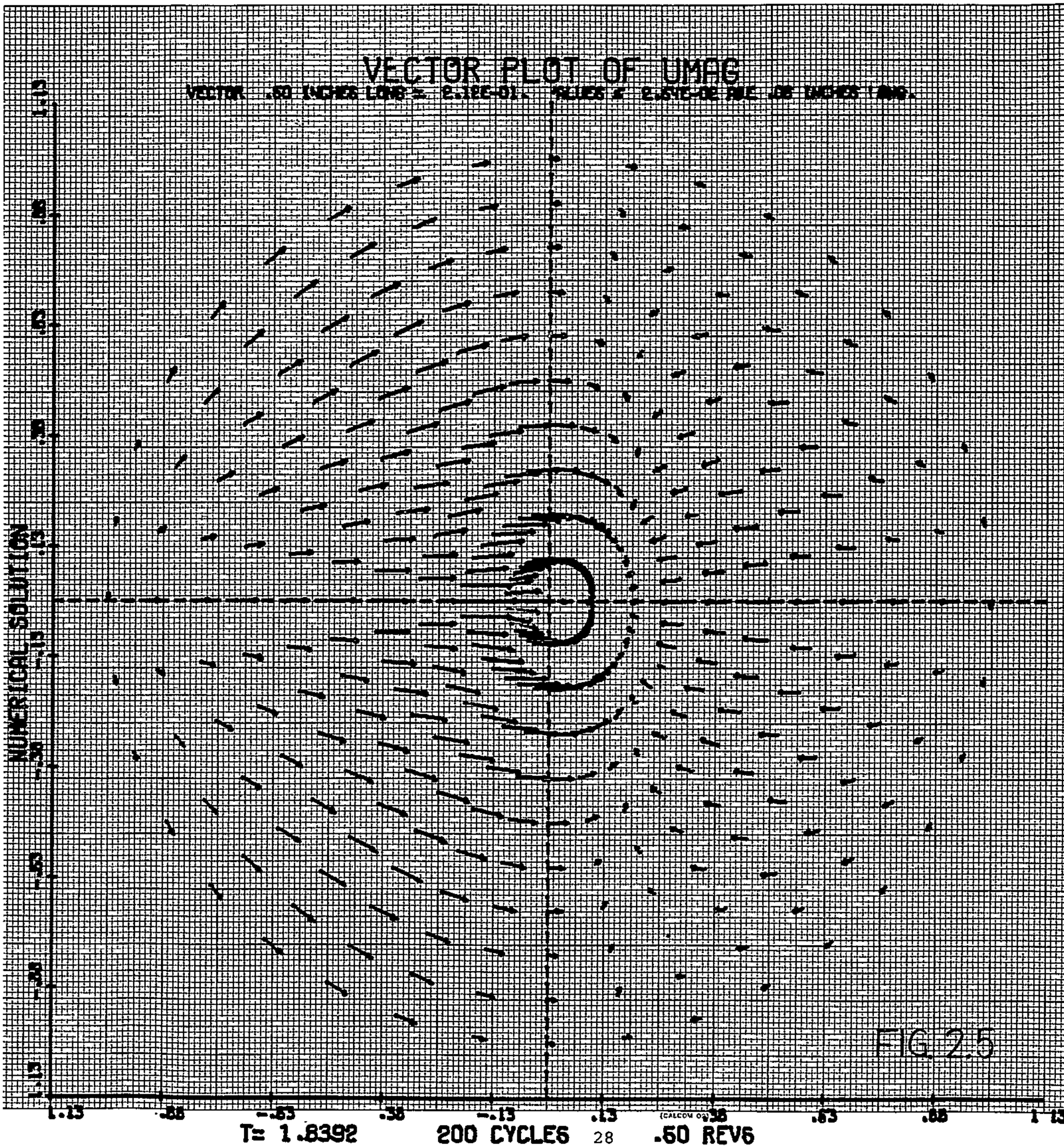

PLOT OF PRESSURE
NUMERICAL SOLUTION
20 CONTOUR LEVELS
6.03612E-01
6.47045E-01
6.90678E-01
7.34110E-01
7.77543E-01
8.21176E-01
8.64708E-01
9.08241E-01
9.51774E-01
9.95306E-01
1.03884E+00
1.08237E+00
1.12590E+00
1.16944E+00
1.21297E+00
1.25650E+00
1.30004E+00
1.34357E+00
1.38710E+00
1.43063E+00
T= .8392
200 CYCLES
.50 REVS

FIG. 2.4



VECTOR PLOT OF UMAG
NUMERICAL SOLUTION
T= 1.8392
200 CYCLES
.50 REVS

FIG. 2.5

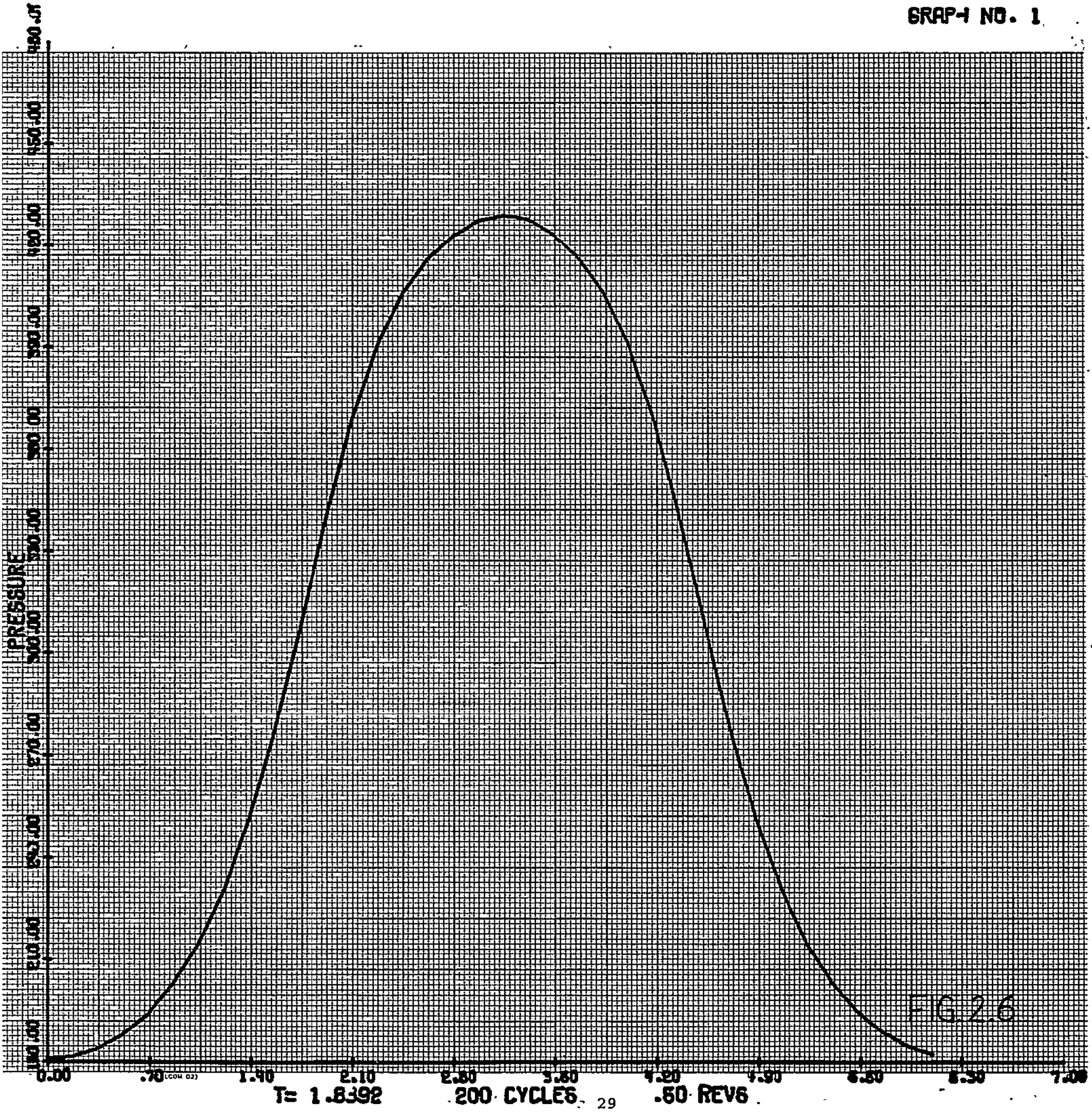

GRAPH NO. 1
PRESSURE
0.00
.70
1.40
2.10
2.80
3.50
4.20
4.90
5.60
6.30
7.00
T= 1.8392
200 CYCLES
.50 REVS

FIG. 2.6

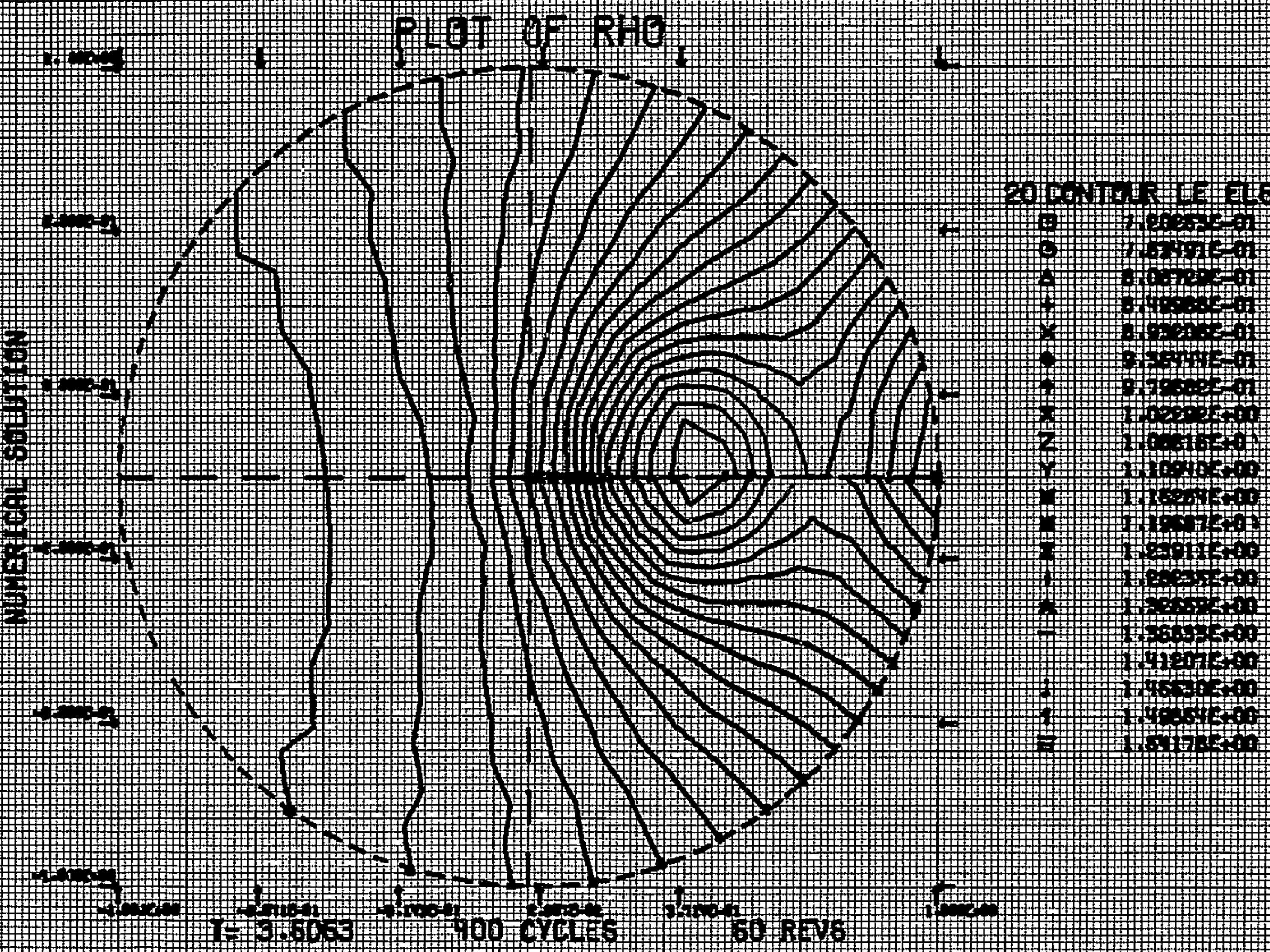


FIG. 2.7

(CALCOM 02)

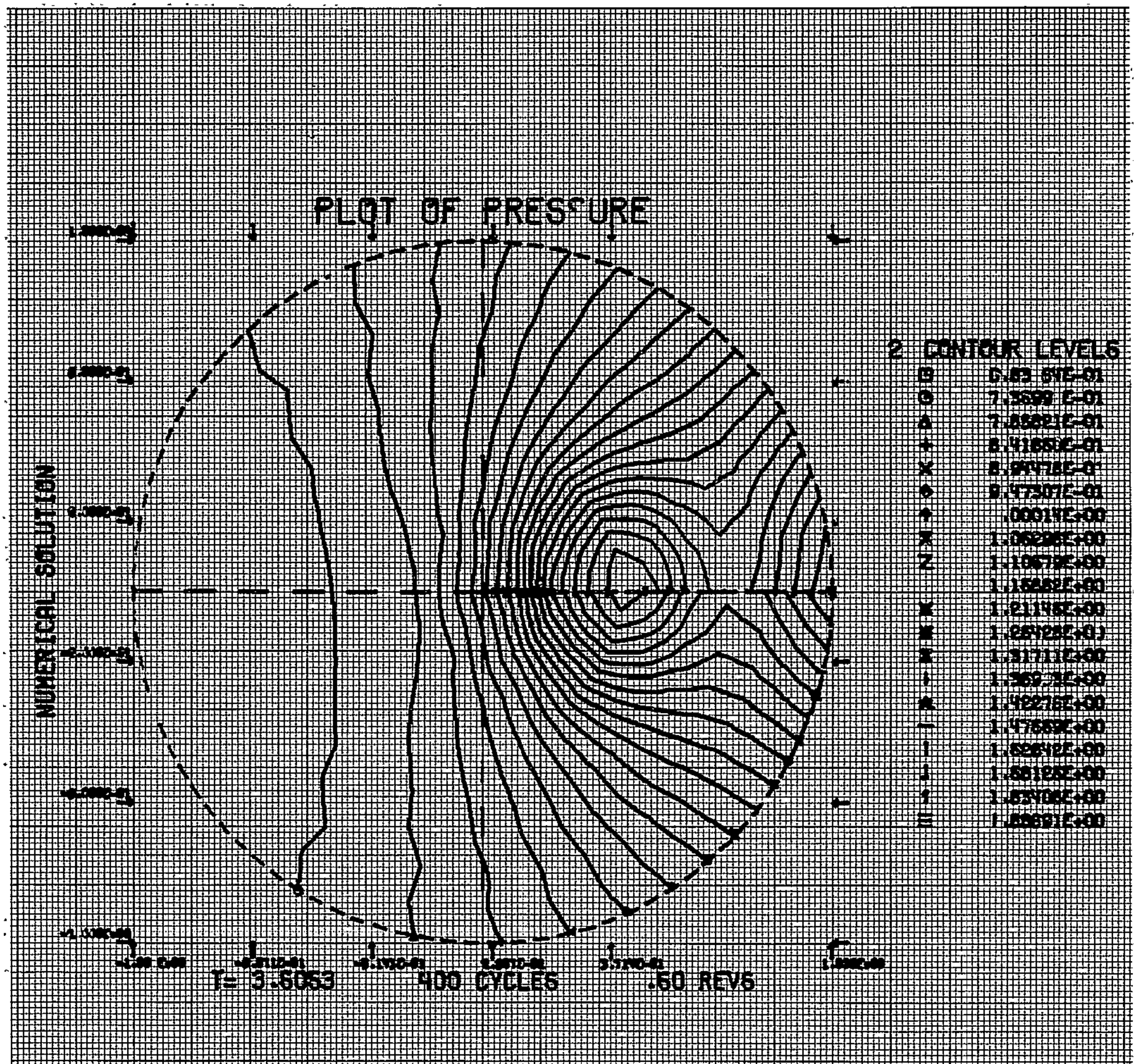


FIG. 2.8

(CALCOM 02)

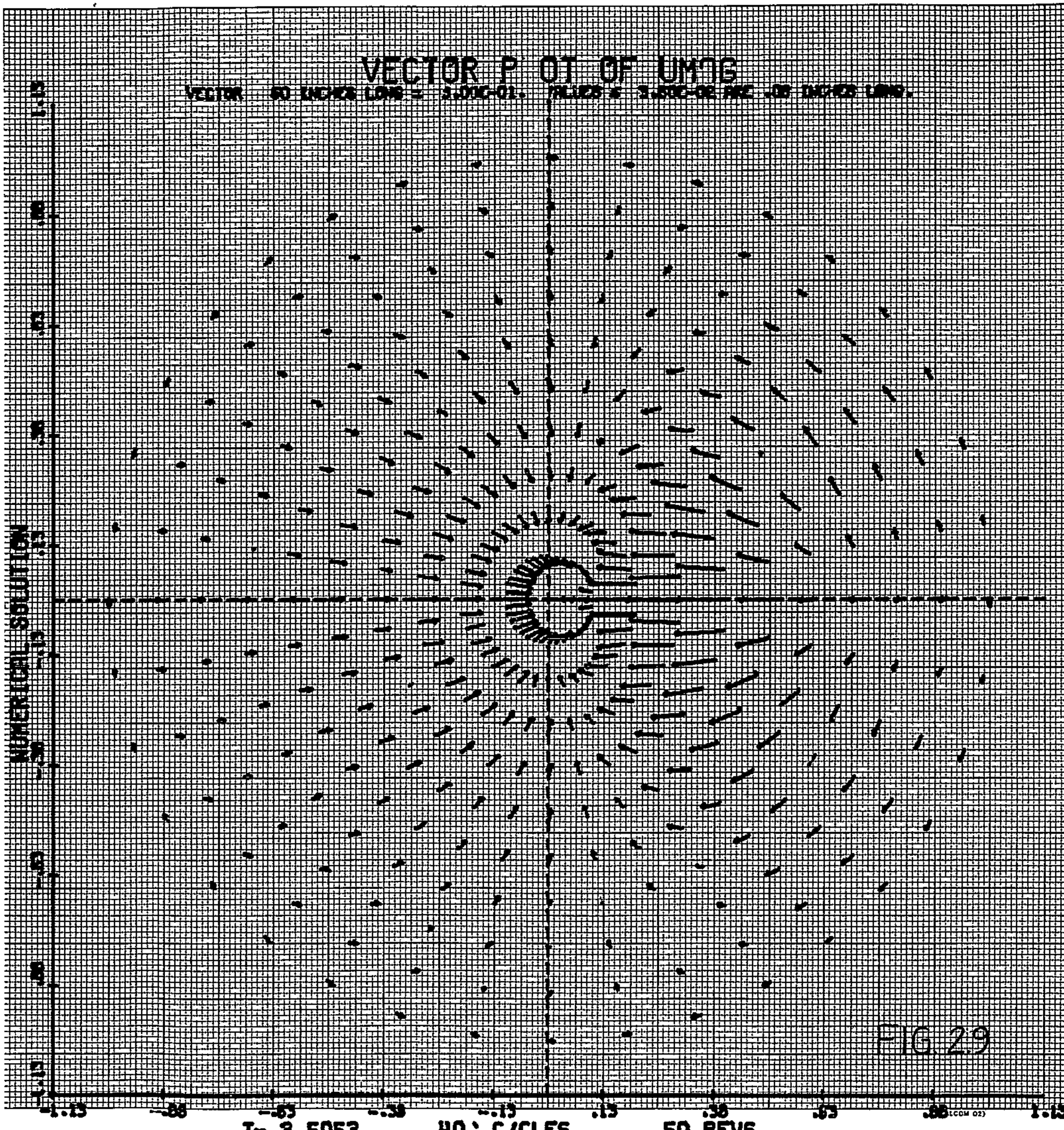

VECTOR PLOT OF UMAG
NUMERICAL SOLUTION
T= 3.6053
40 CYCLES
.60 REVS

FIG.29

GRA·H NO. 2

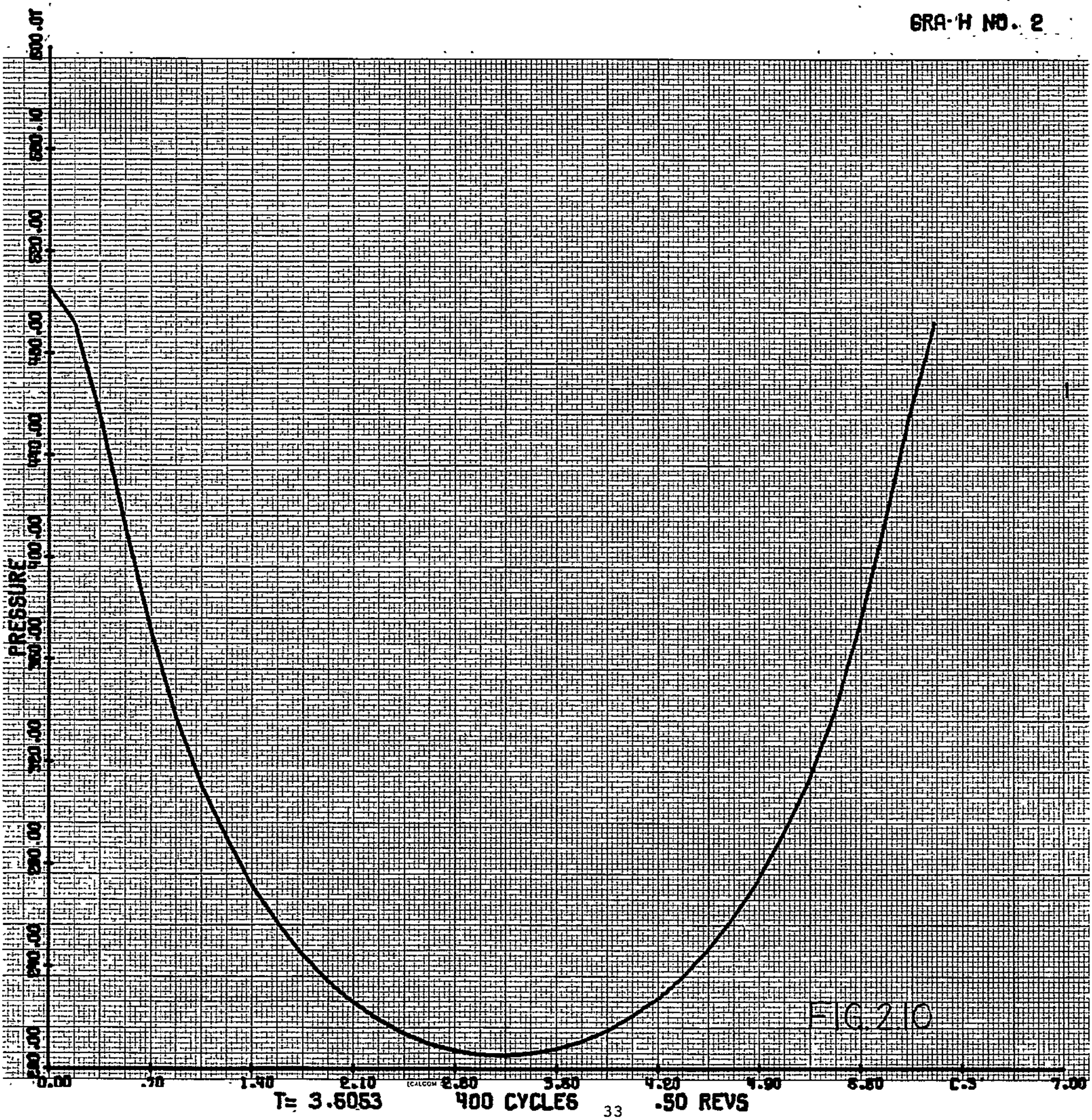


FIG. 2.10

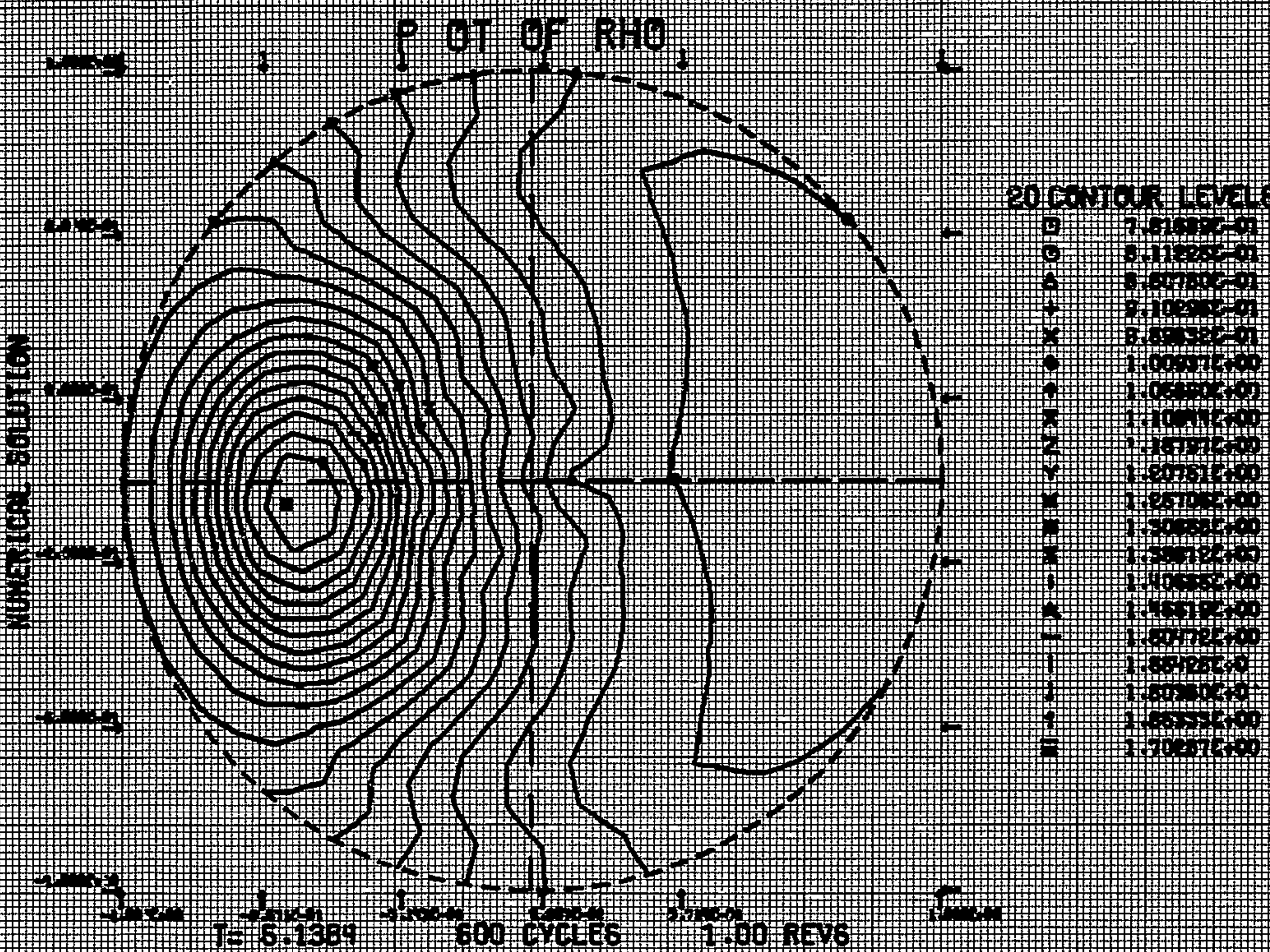


FIG. 2.11

(CALCOM 02)

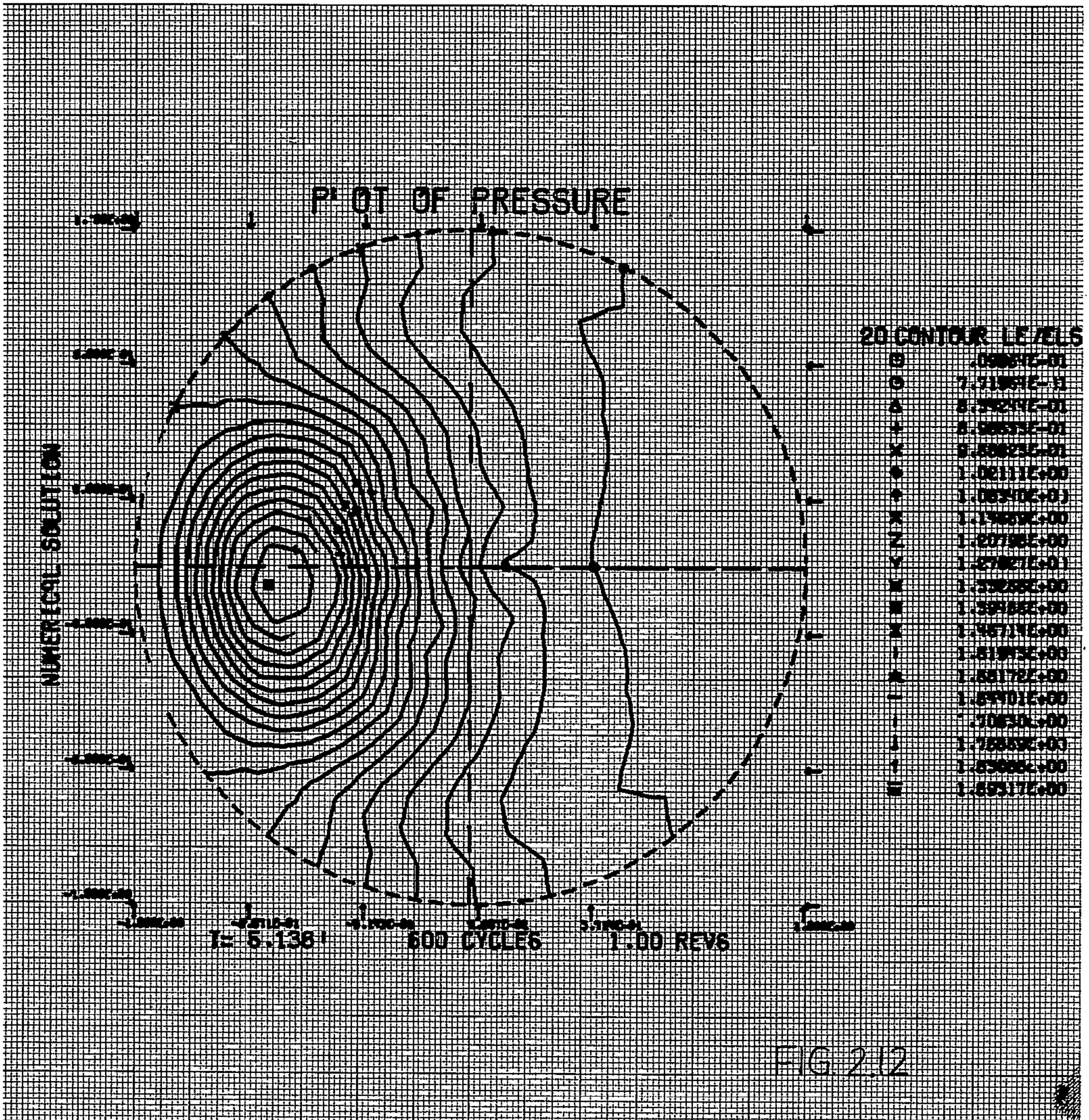

PLOT OF PRESSURE
NUMERICAL SOLUTION
20 CONTOUR LEVELS
T= 5.136
600 CYCLES
1.00 REVS

FIG. 2.12

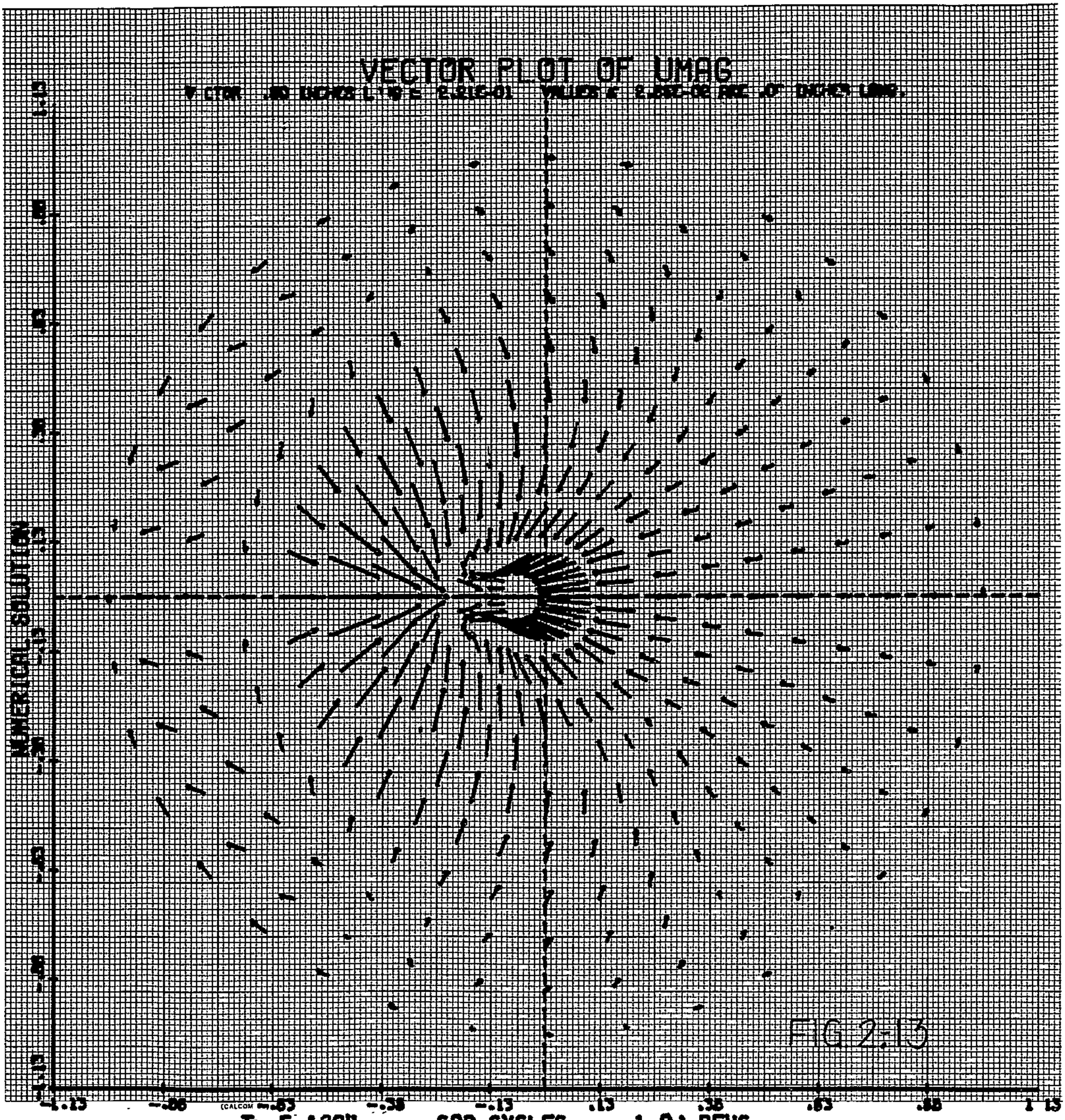

VECTOR PLOT OF UMAG
NUMERICAL SOLUTION
1.13
.88
.63
.38
.13
-.13
-.38
-.63
-.88
-1.13
T- 5.1384
600 CYCLES
1.0J REVS

FIG 2.13

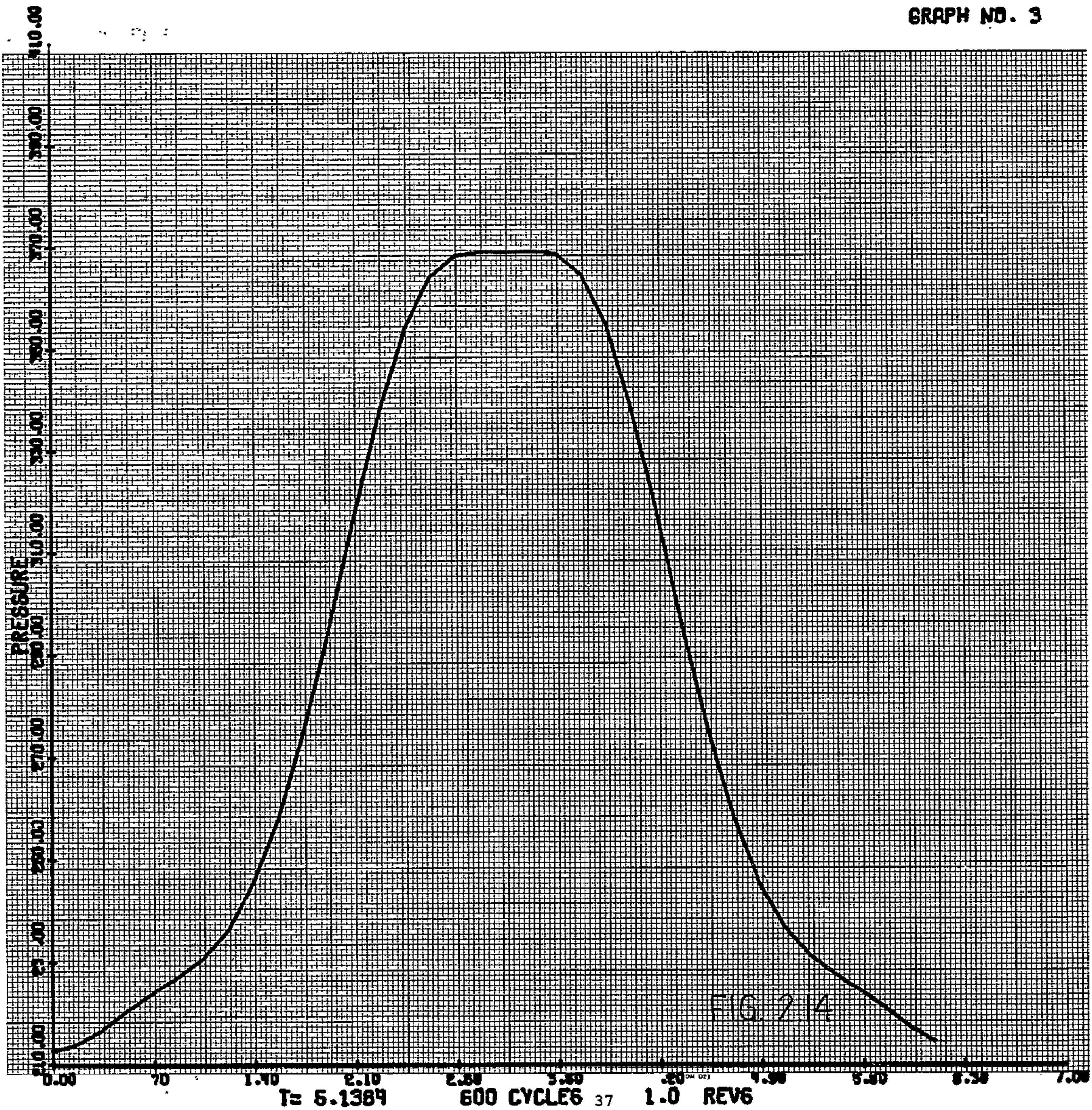


FIG. 2.14

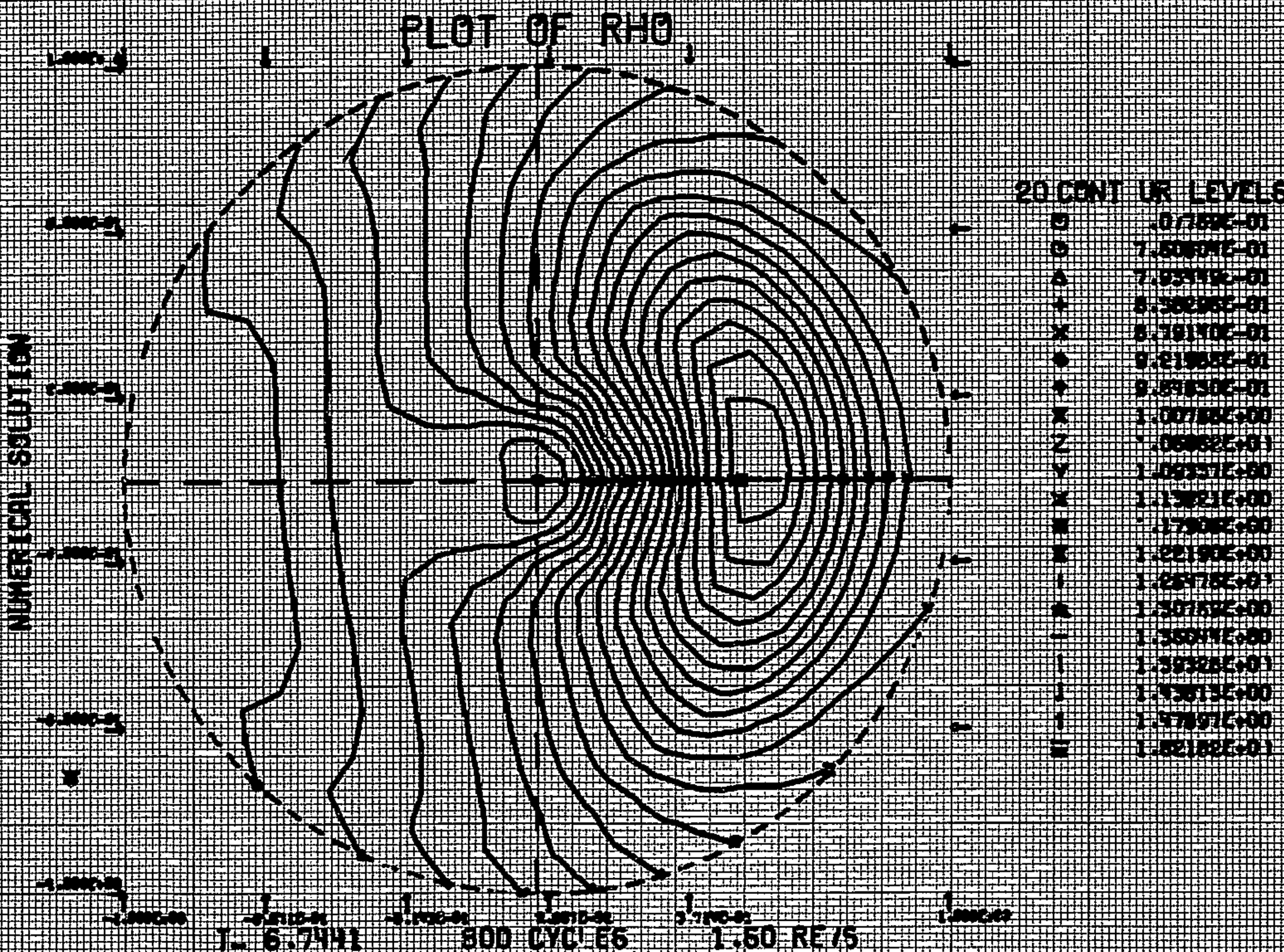

PLOT OF RHO
NUMERICAL SOLUTION
20 CONT UR LEVELS
T= 6.7441
800 CYCLES
1.50 REV/S

FIG. 2.15

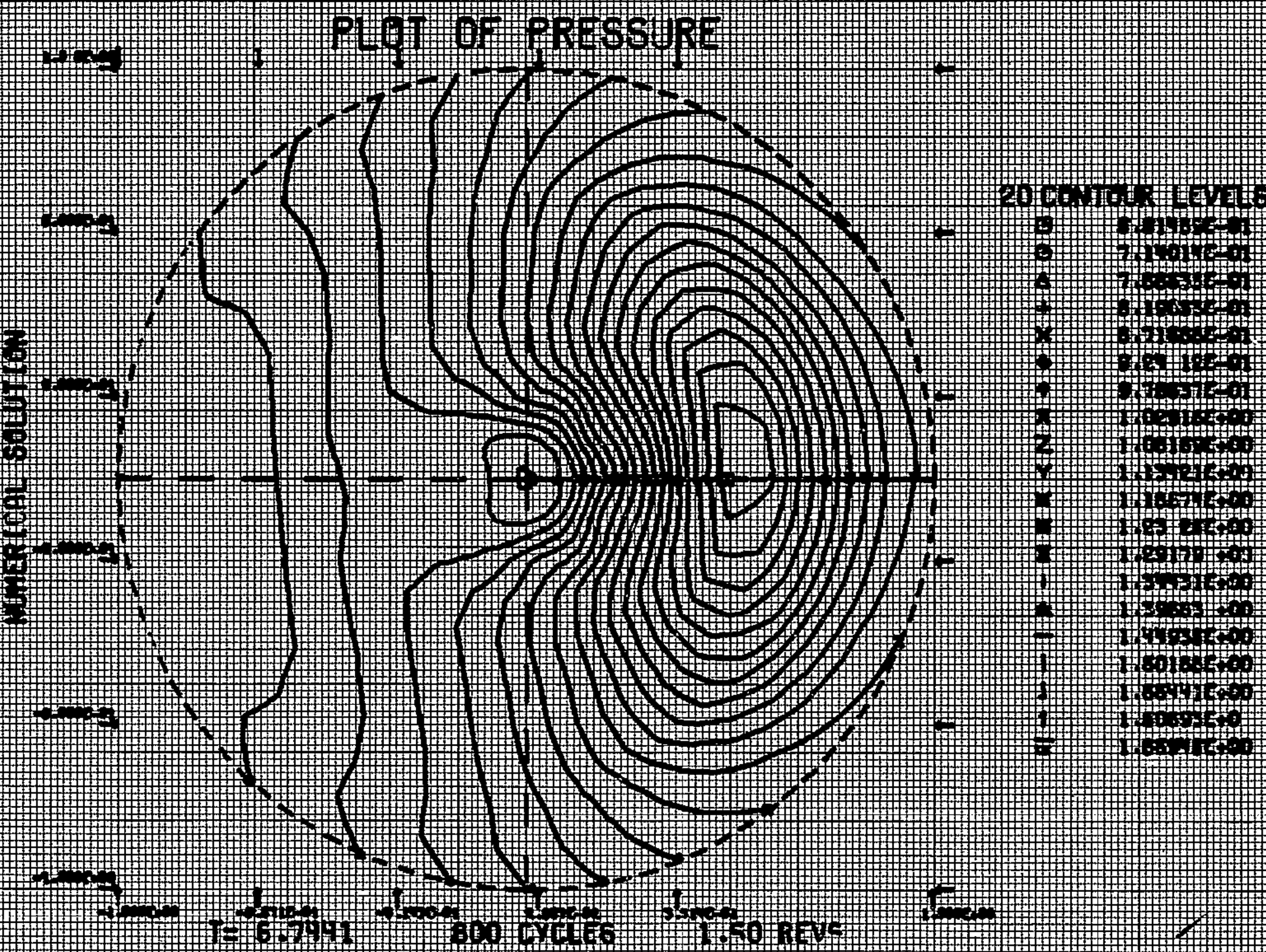


FIG. 2.16

(CALCOM 02)

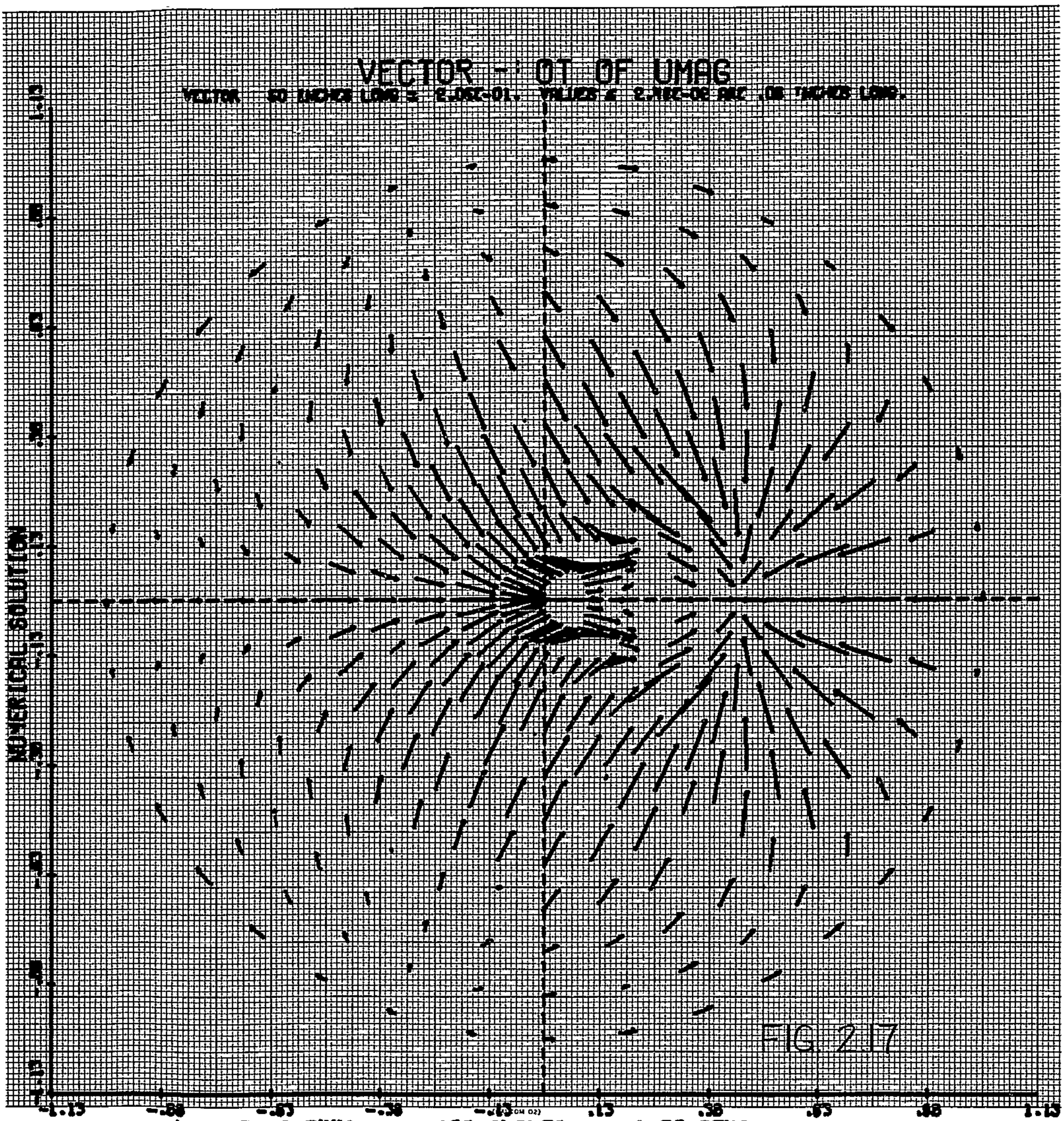


FIG. 2.17

T= 6.7441 800 CYCLES 1.60 REVS

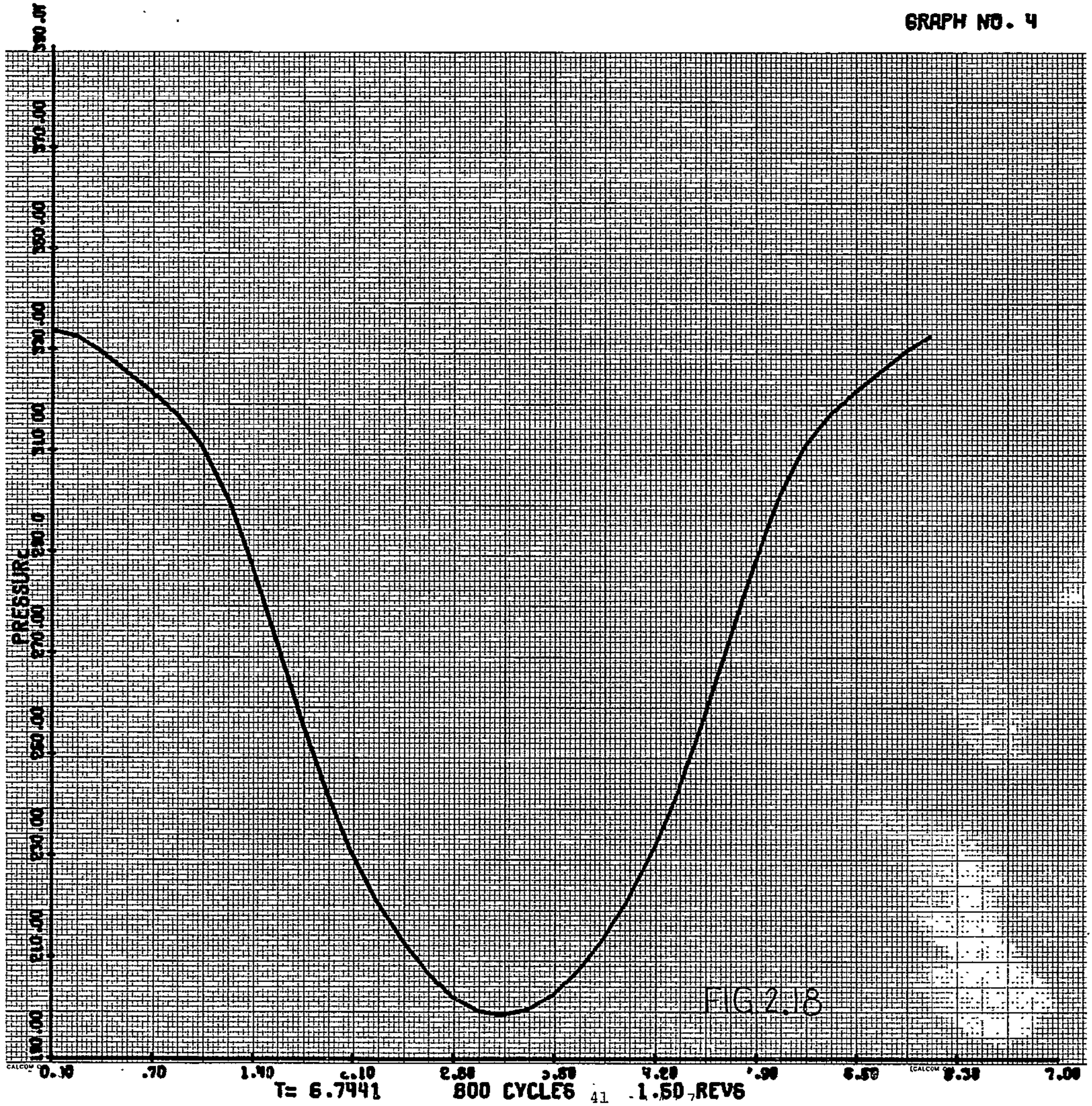
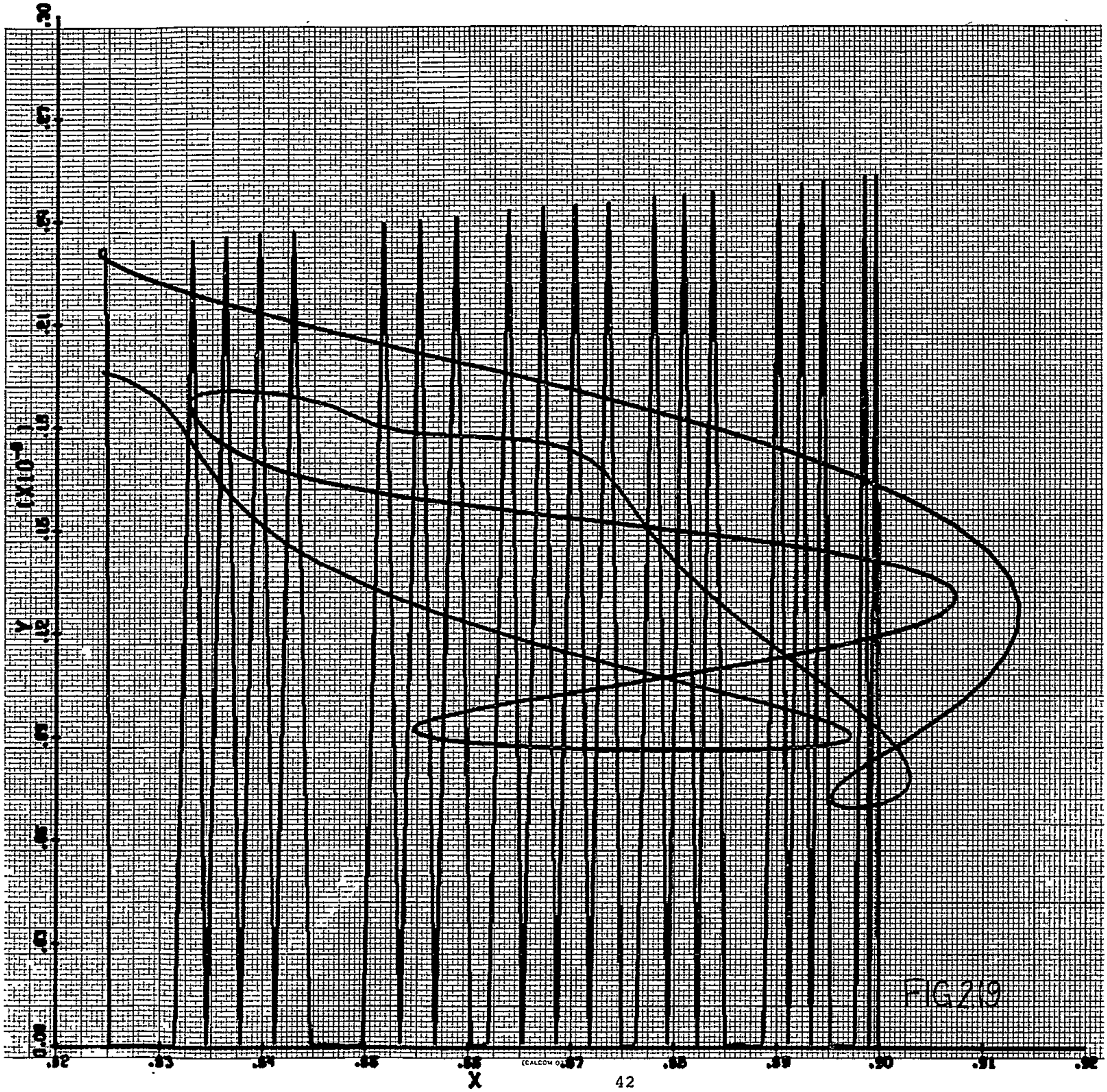
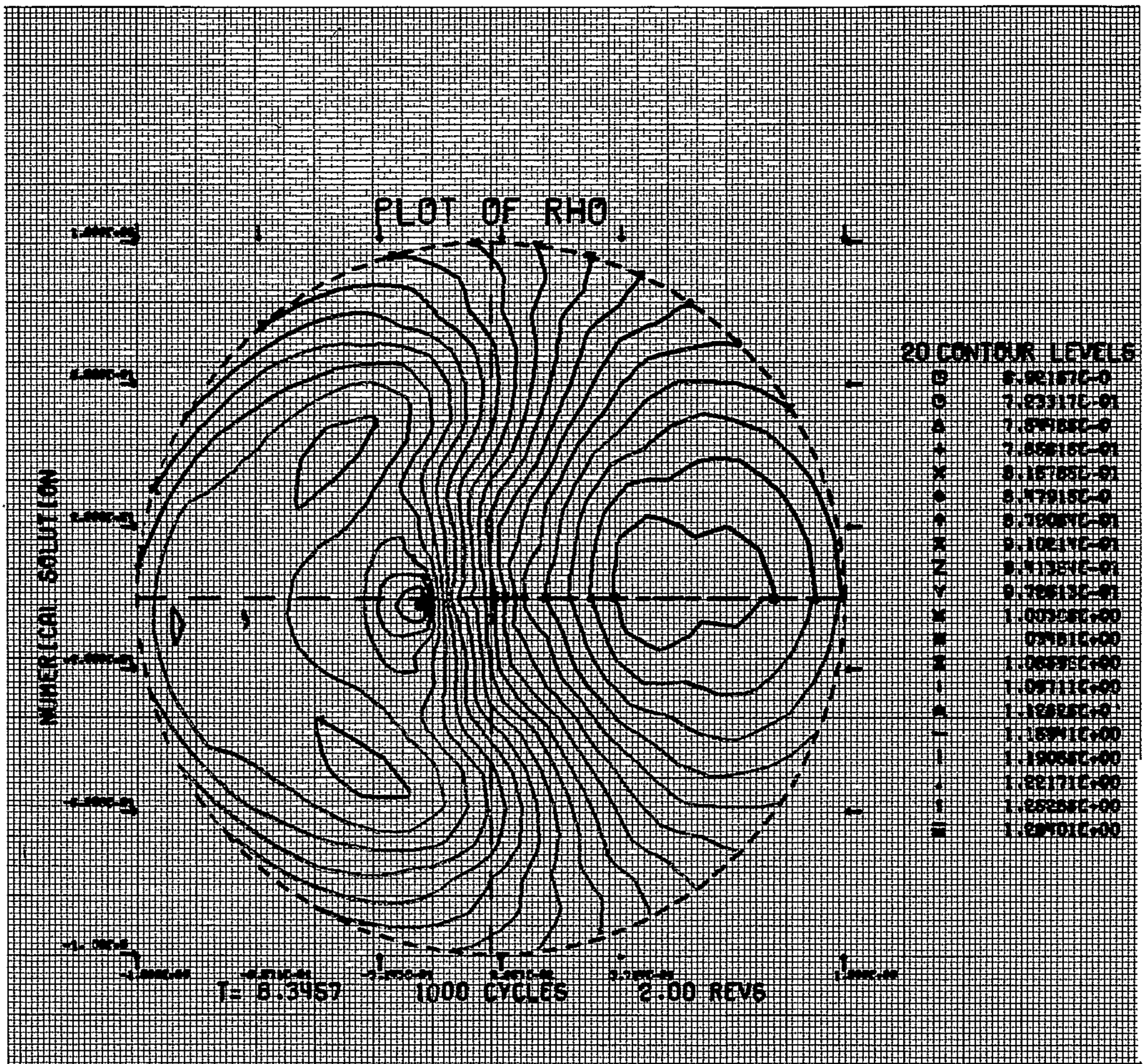


FIG 2.18

GRAPH NO. 5





PLOT OF RHO
NUMERICAL SOLUTION
20 CONTOUR LEVELS
T= 8.3457
1000 CYCLES
2.00 REVS

FIG. 2.20

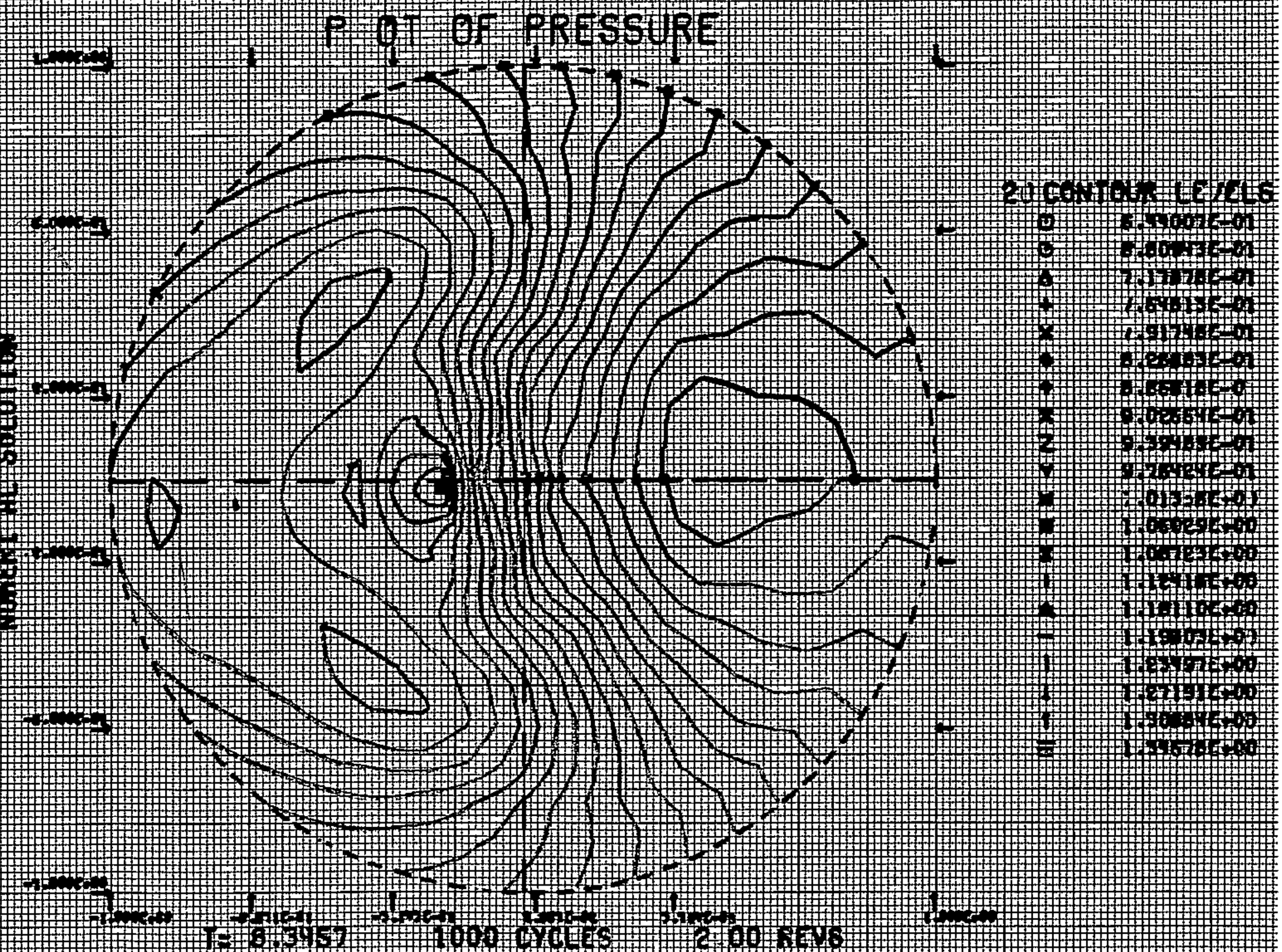
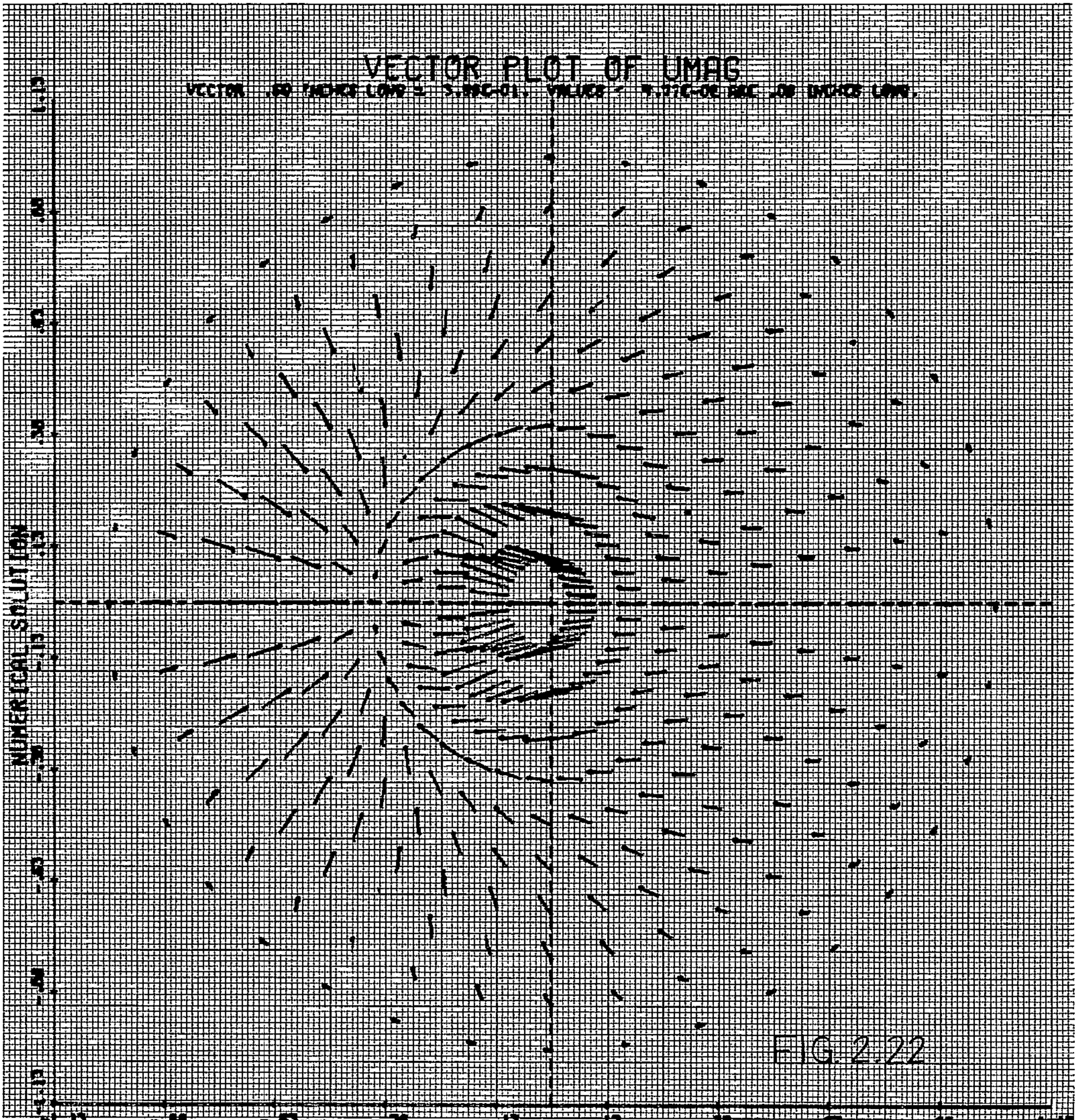
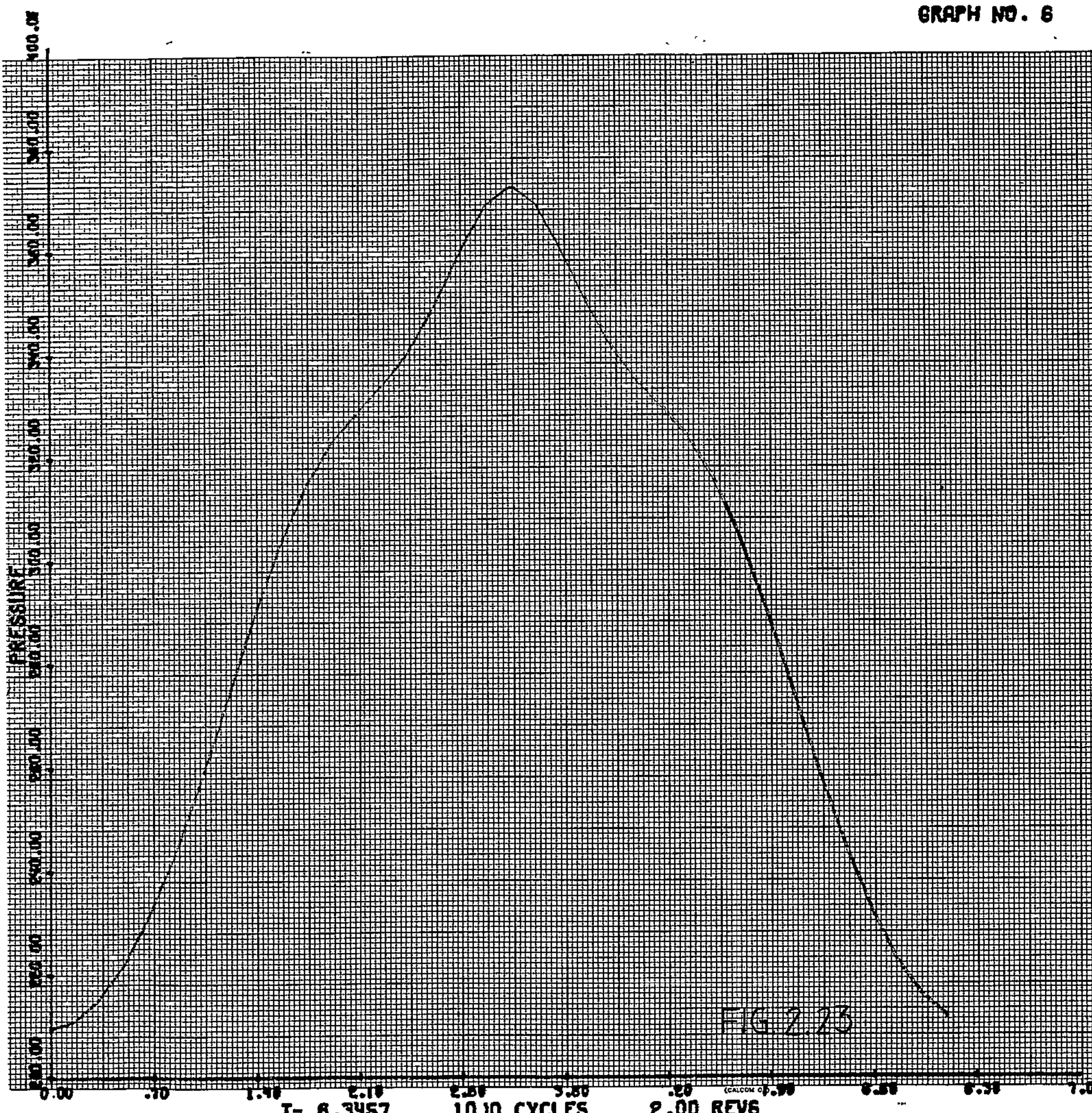

P OT OF PRESSURE
NUMERI AL SOLUTION
20 CONTOUR LEVELS
T= 8.3457
1000 CYCLES
2.00 REVS

FIG 2.21



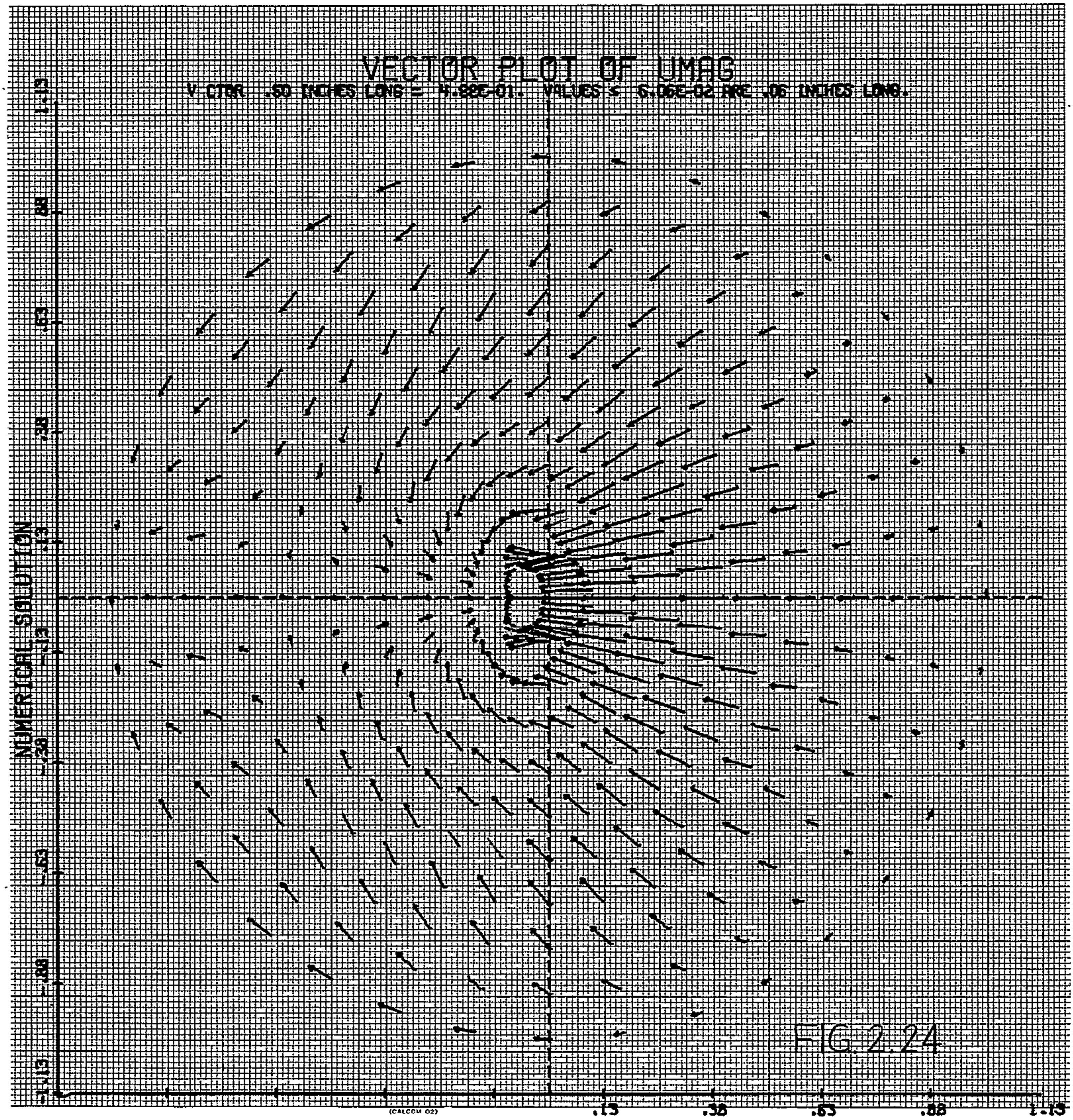
T= 8.3457 1000 CYCLES 2. 0 REVS

FIG. 2.22



GRAPH NO. 6
PRESSURE
0.00
.70
1.40
2.10
7.00
T= 6.3457
10)0 CYCLES
2.00 REVS

FIG.2.23


VECTOR PLOT OF UMAG
V CTOR .50 INCHES LONG = 4.88E-01. VALUES ≤ 6.06E-02 ARE .06 INCHES LONG.
NUMERICAL SOLUTION
1.13
.88
.63
.38
.13
-.13
-.38
-.63
-.88
-1.13
.13
.38
.63
.88
1.13
(CALCOMP 02)
FIG. 2.24

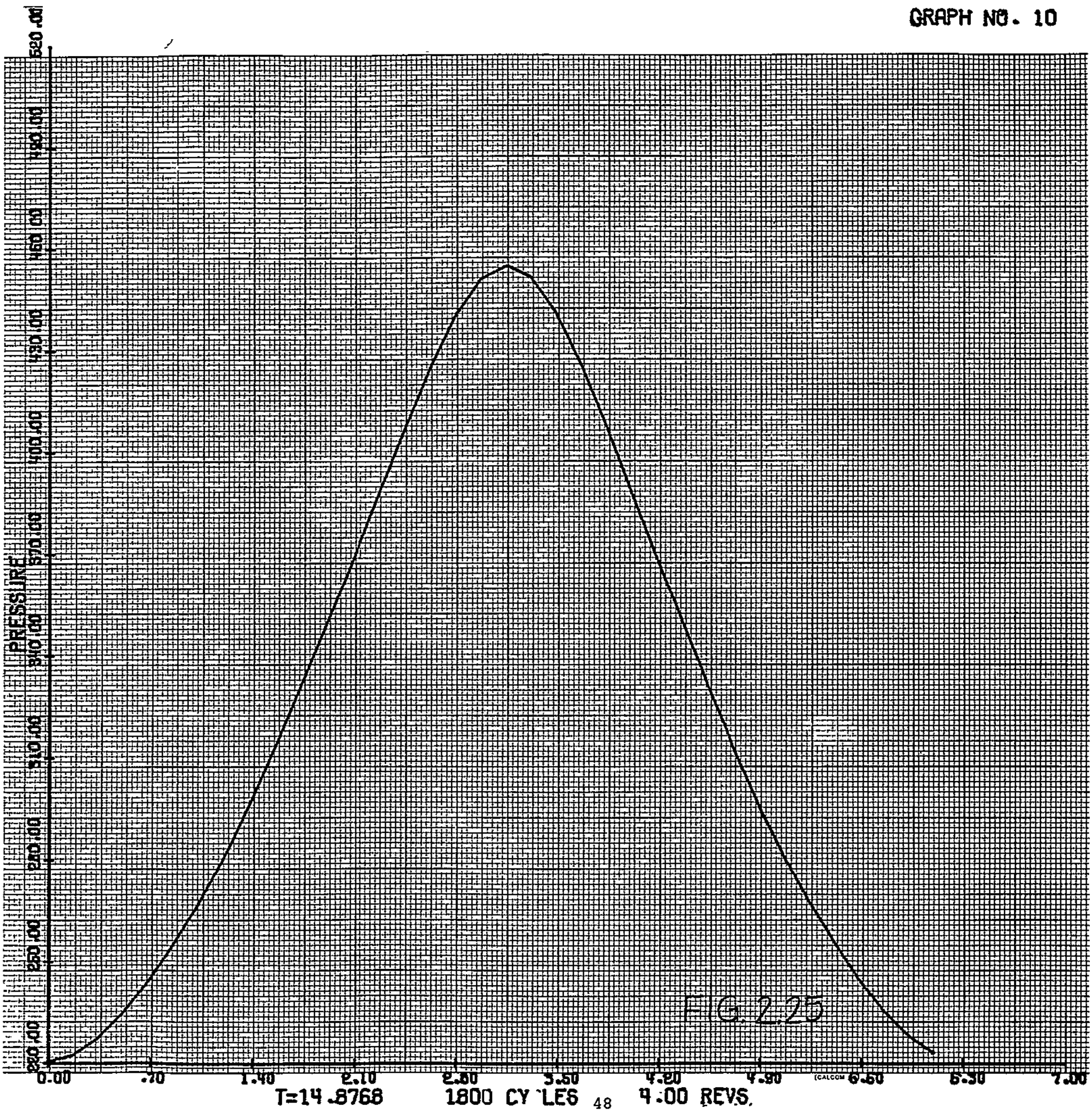

GRAPH NO. 10
PRESSURE
520.00
490.00
460.00
430.00
400.00
370.00
340.00
310.00
280.00
250.00
220.00
0.00
.70
1.40
2.10
2.80
3.50
4.20
4.90
5.60
6.30
7.00
T=14.8768 1800 CY LES 4.00 REVS.

FIG. 2.25

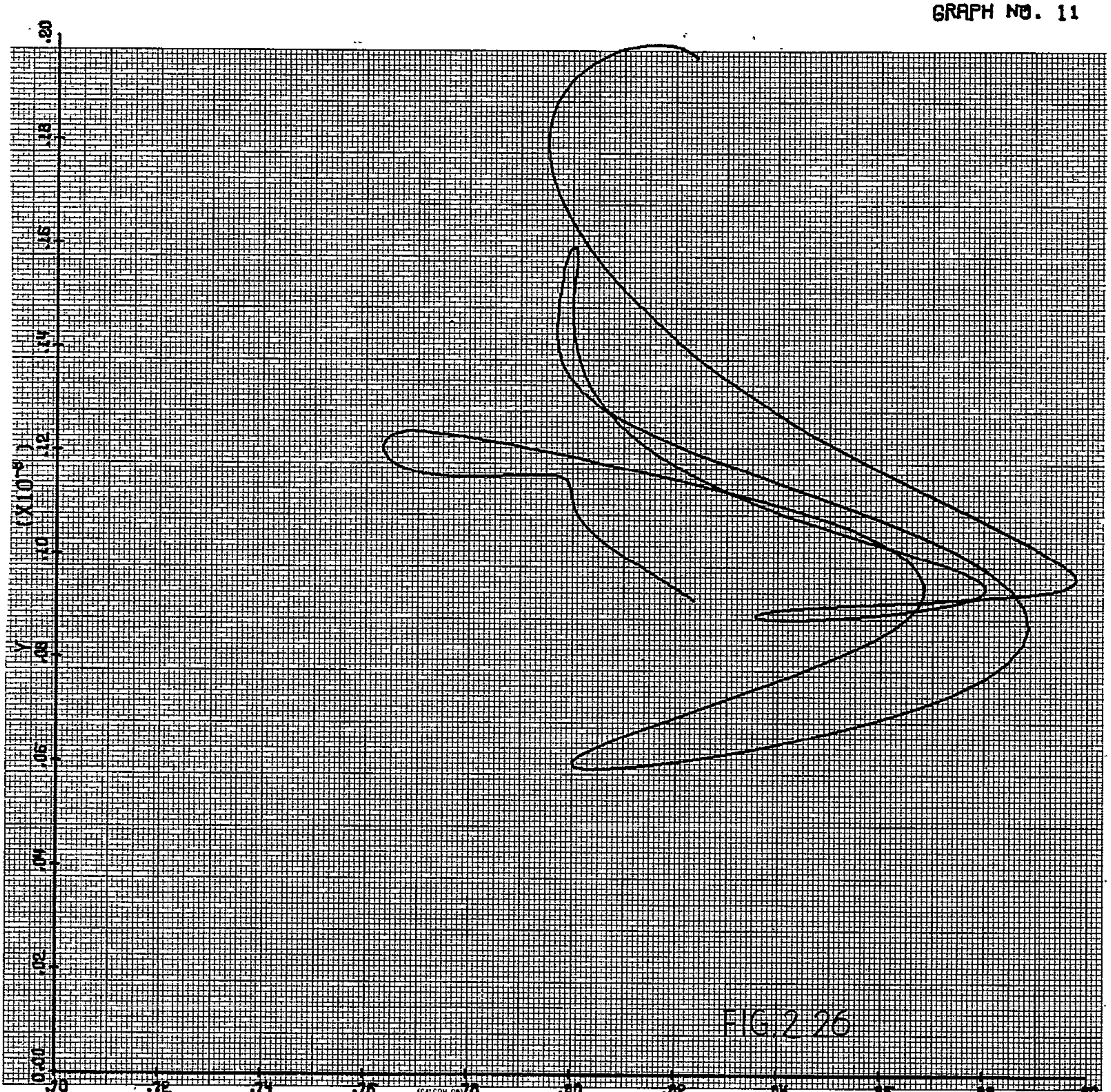

.20
.18
.16
.14
.12
.10
.08
.06
.04
.02
0.00
Y (X10⁻¹)
.70
.72
.74
.76
.78
.80
.82
.84
.86
.88
.90
X

FIG.2.26

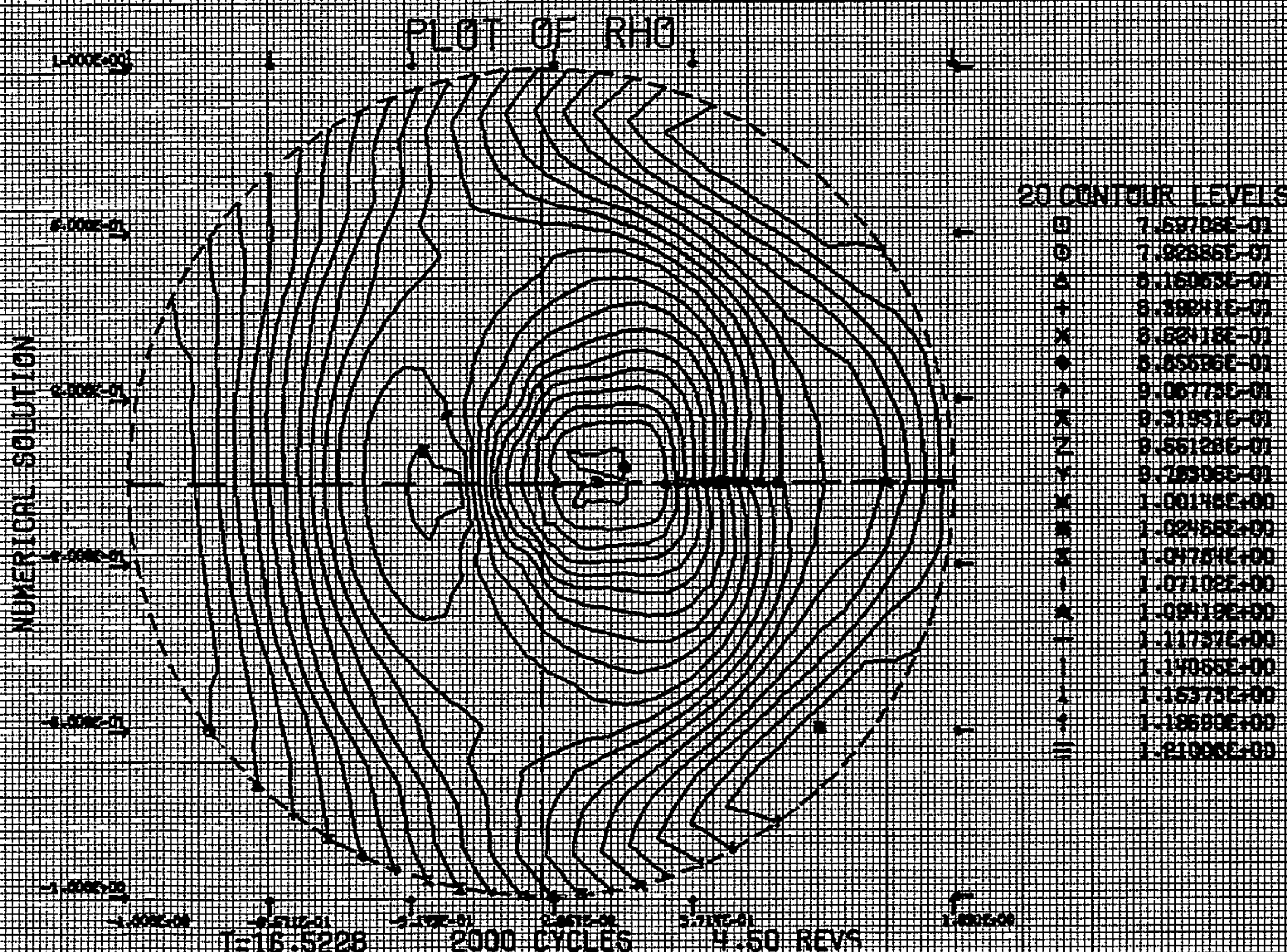

PLOT OF RHO
NUMERICAL SOLUTION
20 CONTOUR LEVELS
T=16.5228
2000 CYCLES
4.50 REVS

FIG.2.27

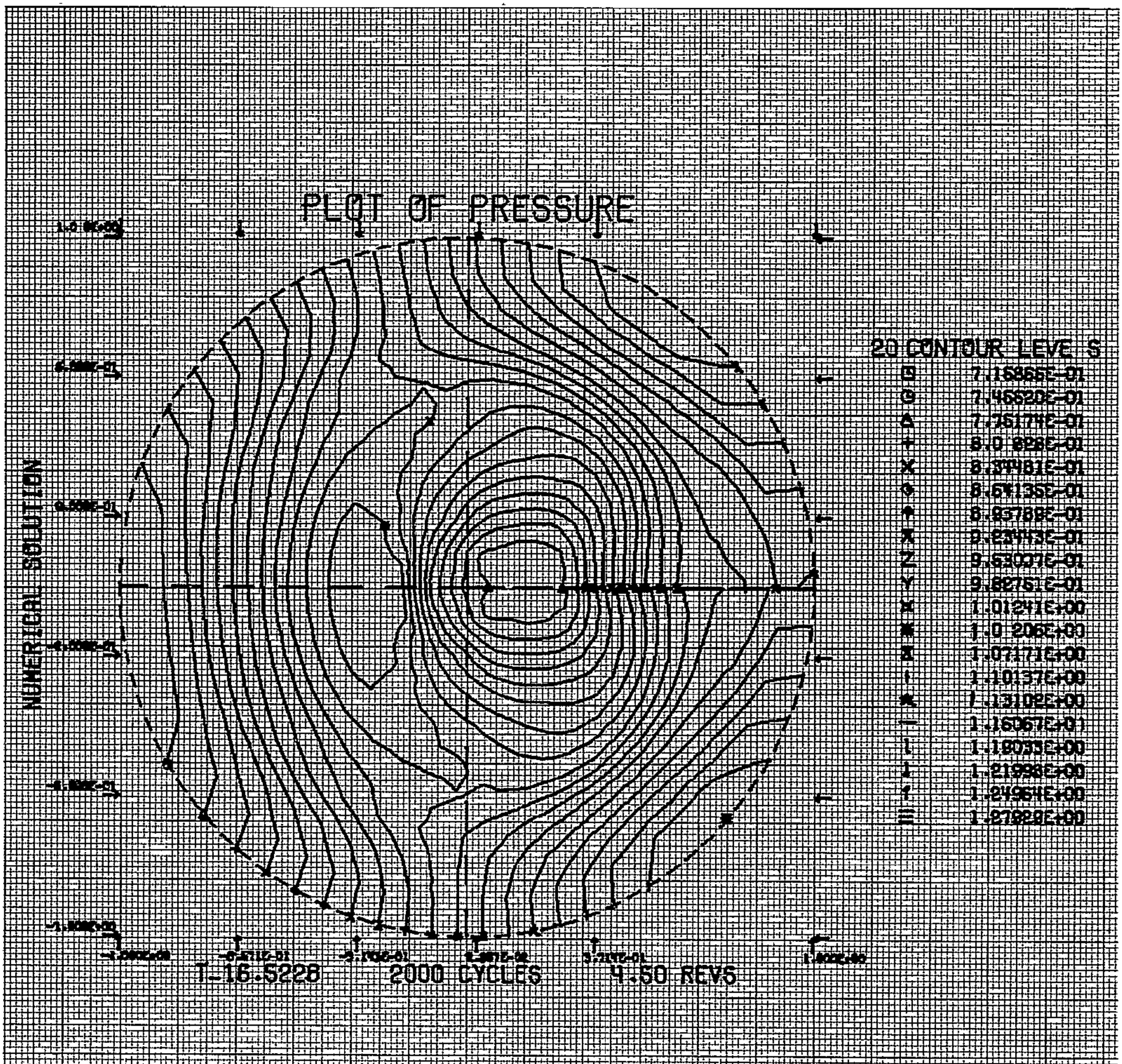

PLOT OF PRESSURE
NUMERICAL SOLUTION
20 CONTOUR LEVELS
7.16865E-01
7.46620E-01
7.76174E-01
8.0 928E-01
8.34481E-01
8.64135E-01
8.93789E-01
9.23443E-01
9.53097E-01
9.82751E-01
1.01241E+00
1.0 206E+00
1.07171E+00
1.10137E+00
1.13102E+00
1.16067E+01
1.19033E+00
1.21998E+00
1.24964E+00
1.27929E+00
T-16.5228
2000 CYCLES
4.50 REVS

FIG.2.28

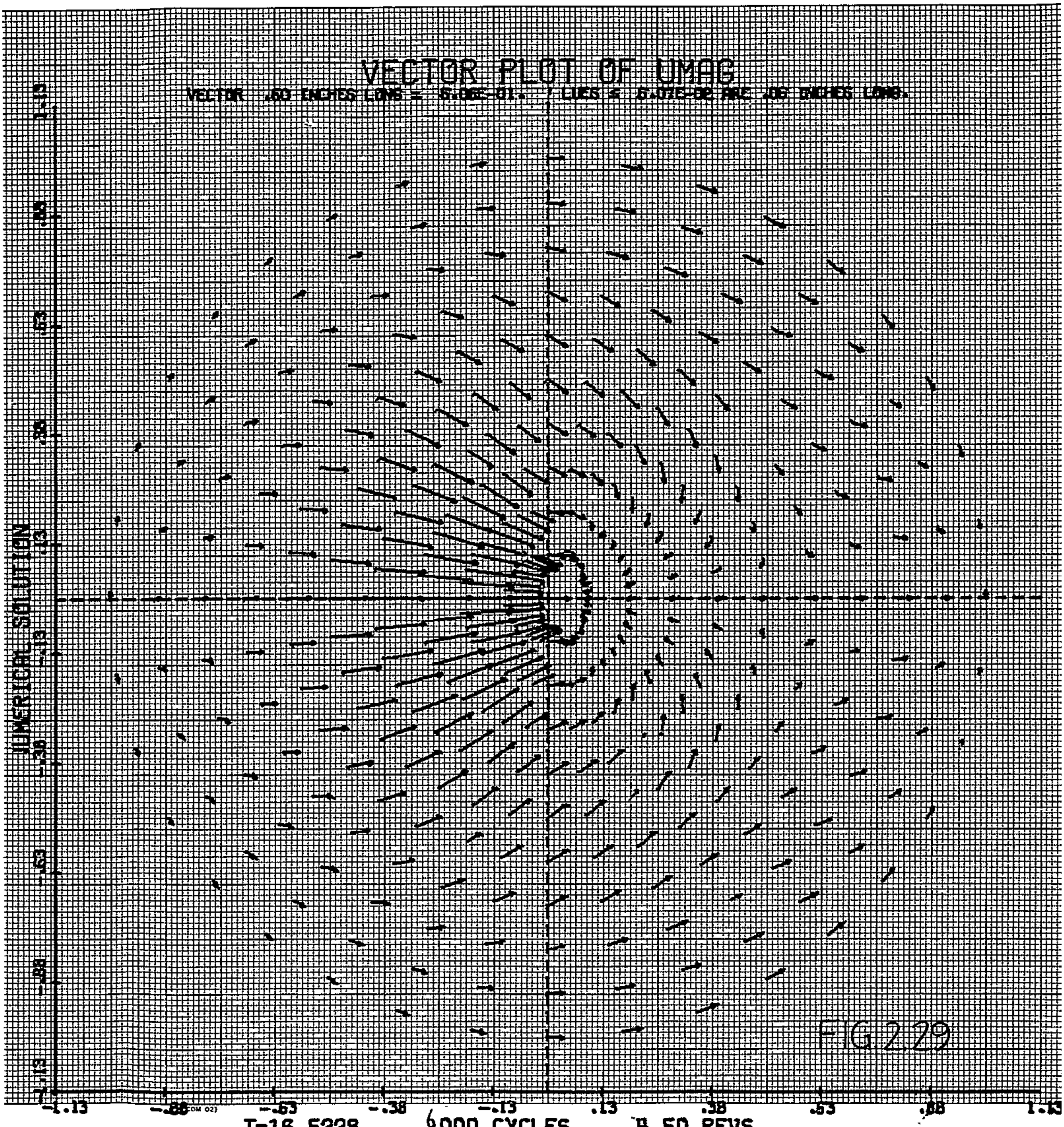

VECTOR PLOT OF UMAG
NUMERICAL SOLUTION
-1.13
-.88
-.63
-.38
-.13
.13
.38
.63
.88
1.13
T=16.5228
2000 CYCLES
4.50 REVS

FIG.2.29

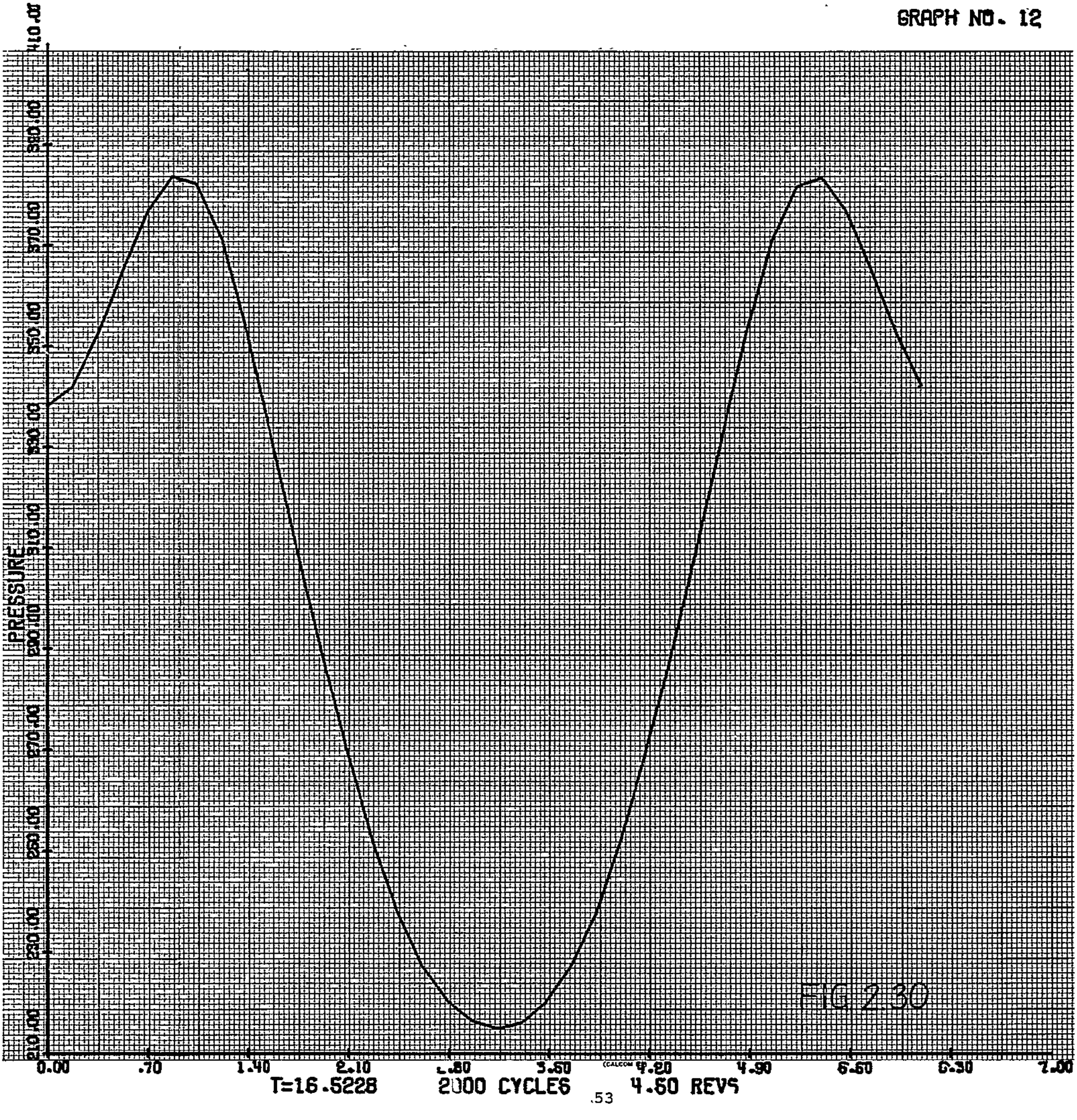

410.00
390.00
370.00
350.00
330.00
310.00
PRESSURE
290.00
270.00
250.00
230.00
210.00
0.00
.70
1.40
2.10
2.80
3.50
4.20
4.90
5.60
6.30
7.00
T=16.5228
2000 CYCLES
4.60 REVS

FIG. 2.30

The first tangential (sloshing) mode (with ε=0.5, γ=1.2) was recomputed using the differential equations (3.1), (3.2), (3.3) and (3.4) in which, however, the right hand side of the continuity and energy equation were modified by a mass source and an energy source. The source term itself is modified to include a simulation of axial outflow by the method outlined in Reference (2), page 25. The combustion model used was also presented in Reference (2) and appears in that report under Section E - The Reduced Godsave Analysis. The plots resulting from this calculation have not been included in this report since they are similar to Figures (2.1) through (2.30). However, the following table gives the comparison in peak pressure measured in the flowfield with and without combustion present at constant intevals of cycle count. As the table shows the maximum pressure levels with combustion are somewhat larger than those in which the flow field spins without interaction with the droplet spray.

TABLE I

First Tangential Sloshing

Peak Pressure (psia)

Cycle Count	No Energy	Energy Modified Godsave Model	Time Millisec
0	450	450	0.
200	430	435	.280
400	462	510	.526
600	566	561	.774
800	495	522	1.02
1000	404	405	1.26
⋮		⋮	
1600	380 (taken at 2000 cycles)	509	1.98

The calculation with energy input went unstable at 1600 cycles which corresponds to a physical time of almost 2 millisec. The instability is numerical and is related to the approximate method used in simulating a time dependent axial flux. It is necessary to include such a term in the calculations so that energy and, therefore, the pressure levels do not grow unbounded.

The initial conditions for the next calculation were a slight modification of the first transverse (sloshing) mode. In that calculation the tangential velocity component was $v(r,\theta)=0$ at $t=0$. In this calculation a small amount of solid body rotation is introduced at $t=0$. The tangential velocity component for pure solid body rotation is $v=\omega r$, where ω is the angular velocity. We compute the value of ω by

$$\omega^2 = k \sum_i \sum_j E_{ij} \Big/ \{\tfrac{1}{2} \sum_i \sum_j \rho_{ij} r_{ij}^2\}$$

The constant k is the fraction of total initial energy in the field, given by the numerator, that will appear as the initial rotational energy. For Figures (2.31) to (2.82), k=0.01.

The value of pressure $\bar{p}$ = 313.5 psia, chamber radius R=5.52 inches and $\varepsilon=2/3$.

As one views these Figures, the most striking difference that appears in the wave motion (as in Figure (2.39)) is the large pressure level ≈760 psia that appears during the initial transient and the persistence of a pressure wave which has a fairly constant tangential structure over an arc of about $\pi/2$ radians but a strong radial variations (as in Figure (2.52)). Here the pressure at the wall at a point $\theta=\pi$ plus about 1/3 of a quadrant is equal to 1.81*313.5=565 psia while the pressure two-thirds inward towards the center is equal to 0.65*313.5=203 psia. The velocity field is shown in Figure (2.53) and it is seen that a large induced flow is associated with the wave. Figure (2.54) is a plot of the pressure at the chamber wall as a function θ in radians.

2 ⊕

The calculation was carried out to 6000 cycles which corresponds to a physical time of 7.35 milliseconds. As is seen from the pressure plot of Figure (2.80) the spatial structure of the wave persists although the amplitude is somewhat reduced from the earlier time amplitudes.

From the plots of density or pressure in Figures (2.67) and (2.68) a persistent feature of the field is the existance of two well defined wave structures at $\theta=\pi$ and $\theta \sim 3\pi/2$. This is also seen in Figure (2.70). As seen from Figures (2.50) and (2.71) the streakline exhibits a circular motion over the entire chamber cross section in sharp contrast to the streakline motion given for the first transverse without the small rotational component added.

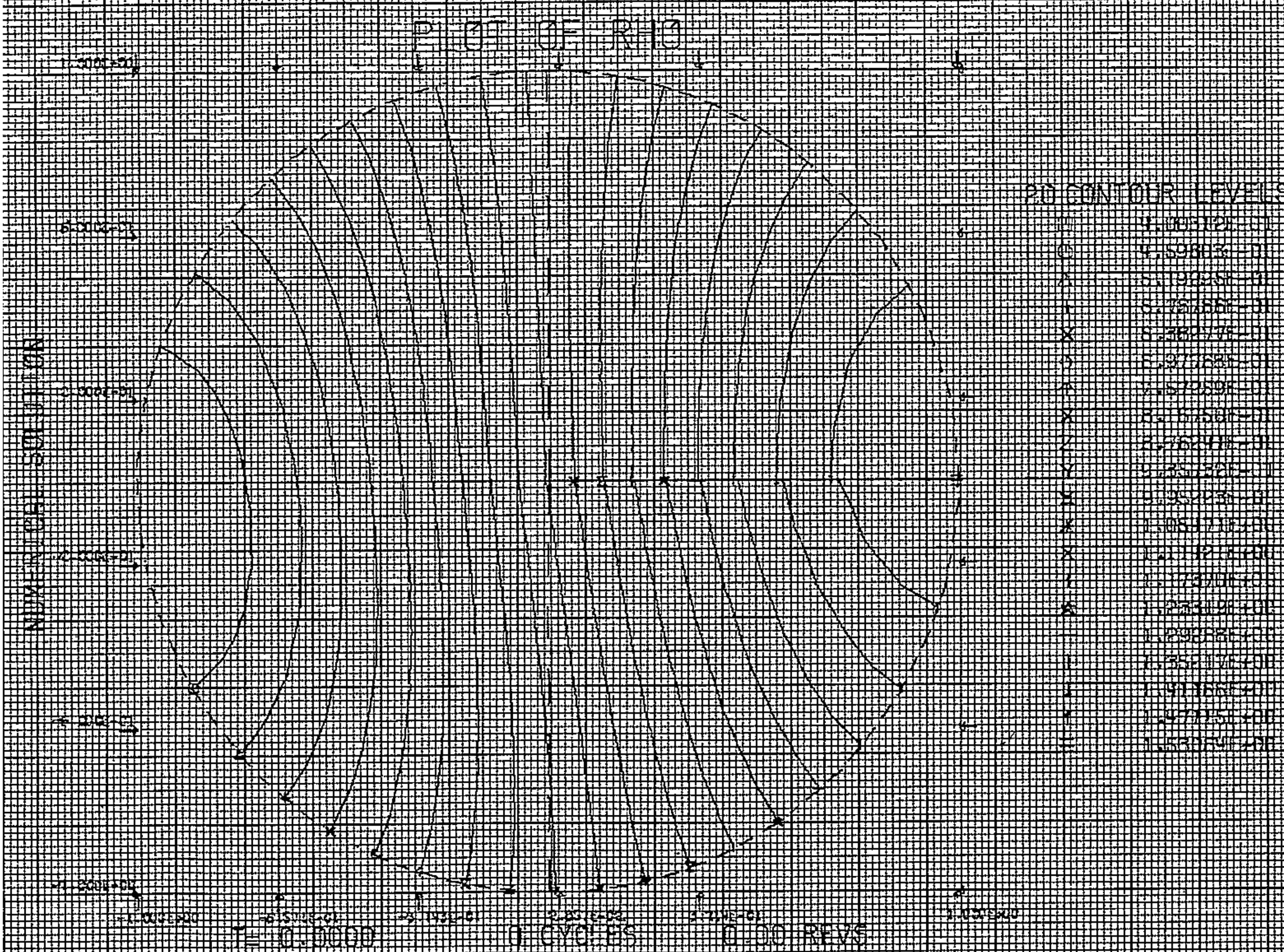


FIG 2.31

(CALCOM 02)

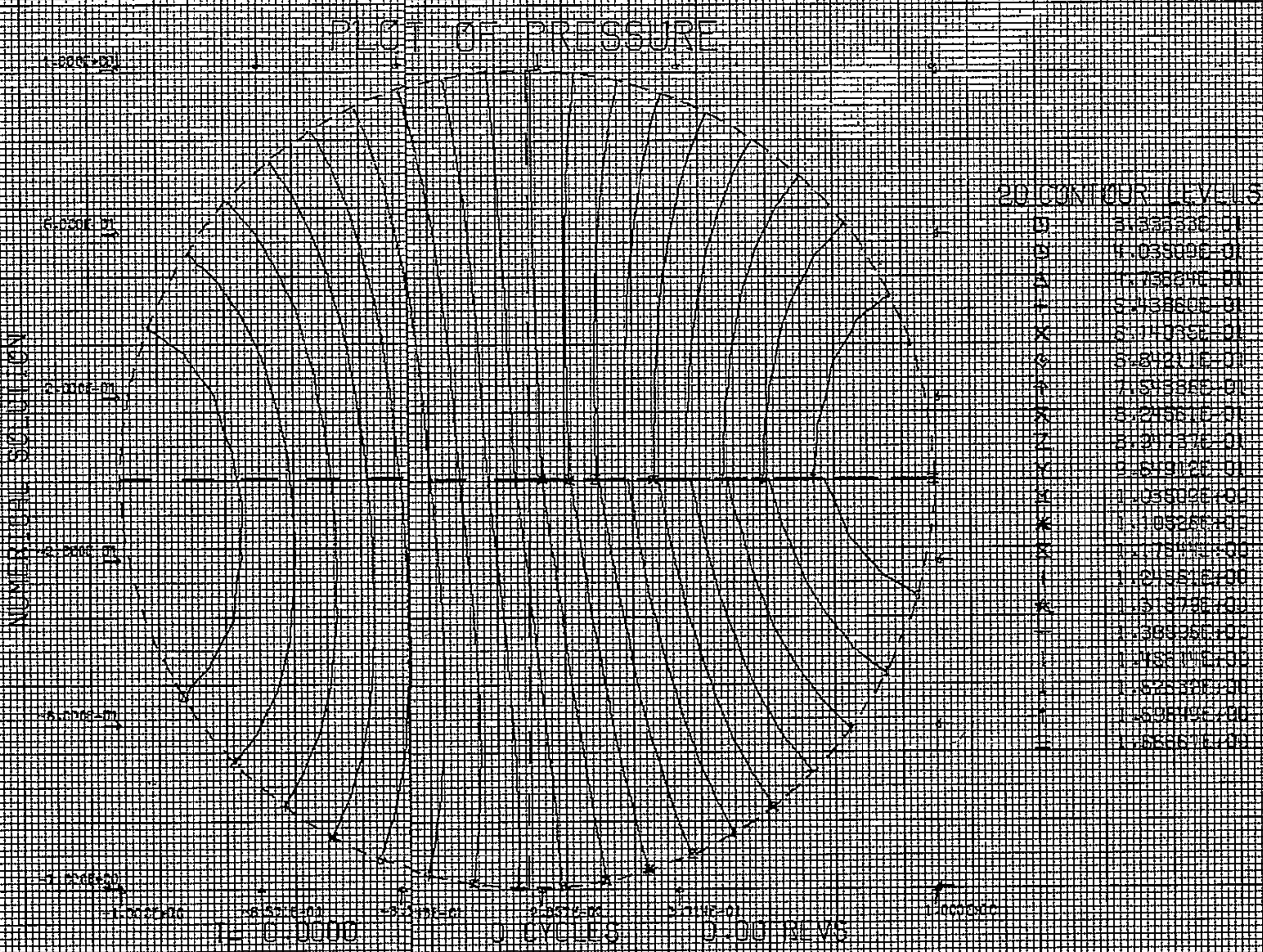

PLOT OF PRESSURE
NUMERICAL SOLUTION
20 CONTOUR LEVELS
T= 0.0000
0 CYCLES
0.00 REVS

FIG. 2.32

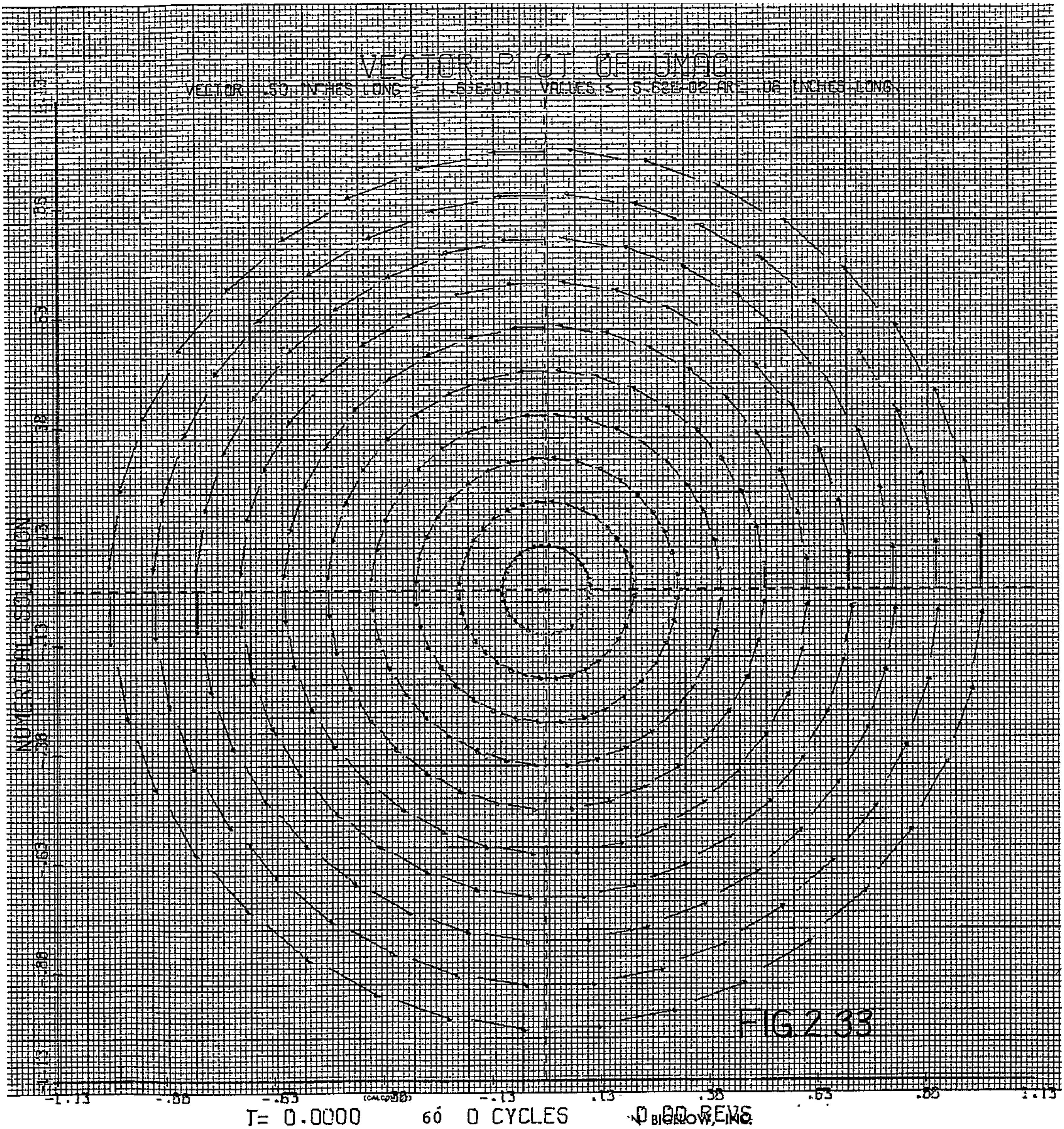

VECTOR PLOT OF UMAG
VECTOR .50 INCHES LONG = 1.63E-01. VALUES ≤ 5.62E-02 ARE .06 INCHES LONG
NUMERICAL SOLUTION
1.13
.88
.63
.38
.13
-.13
-.38
-.63
-.88
-1.13
-1.13
-.88
-.63
-.38
-.13
.13
.38
.63
.88
1.13
T= 0.0000
0 CYCLES
0.00 REVS

FIG.2.33

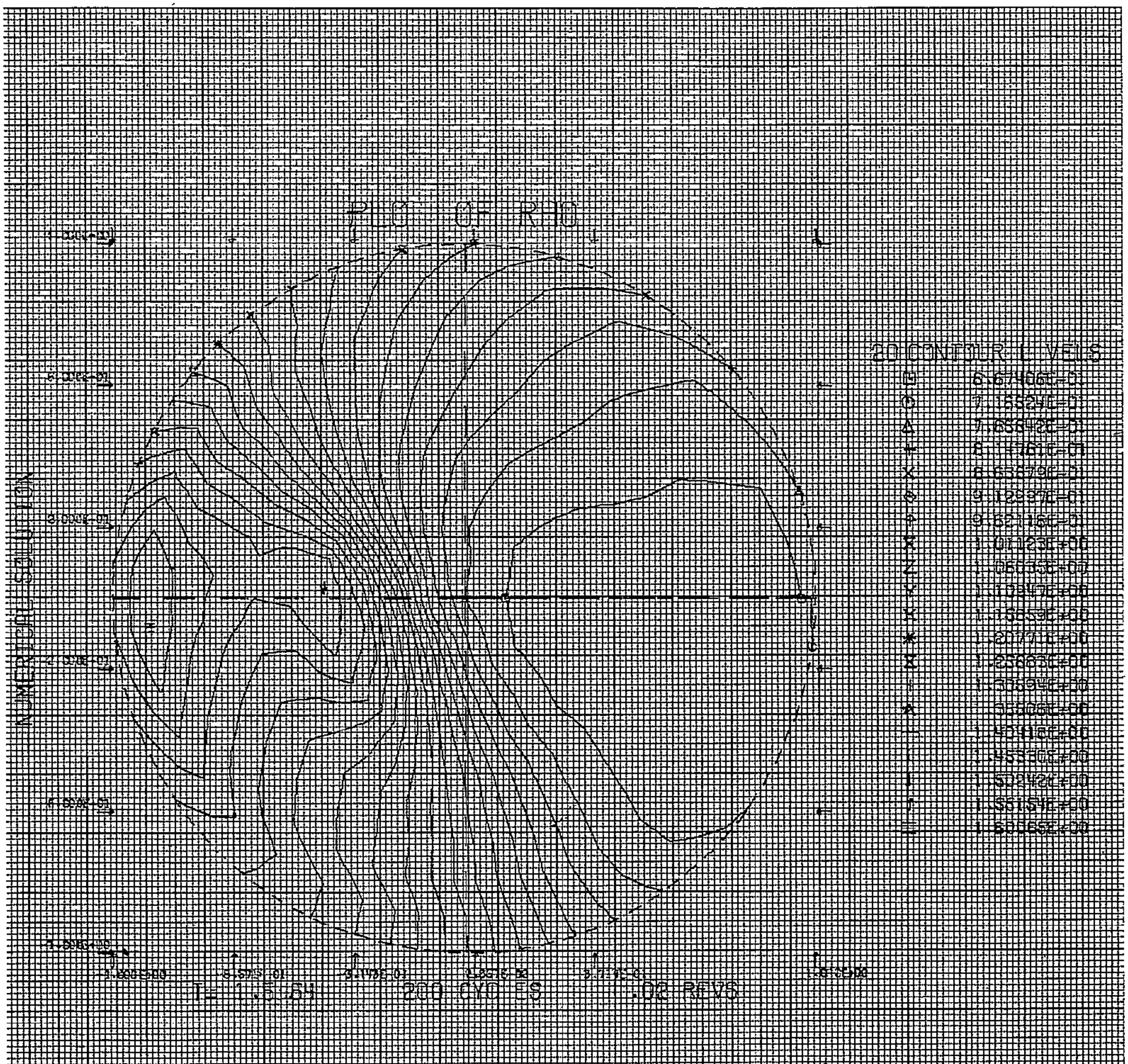

PLOT OF RHO
NUMERICAL SOLUTION
20 CONTOUR L VELS
6.67406E-01
7.16524E-01
7.65642E-01
8.14761E-01
8.63879E-01
9.12997E-01
9.62116E-01
1.01123E+00
1.06035E+00
1.10947E+00
1.15859E+00
1.20771E+00
1.25683E+00
1.30594E+00
1.40418E+00
1.45330E+00
1.50242E+00
1.55154E+00
1.60065E+00
200 CYCLES
.02 REVS

FIG 2.34

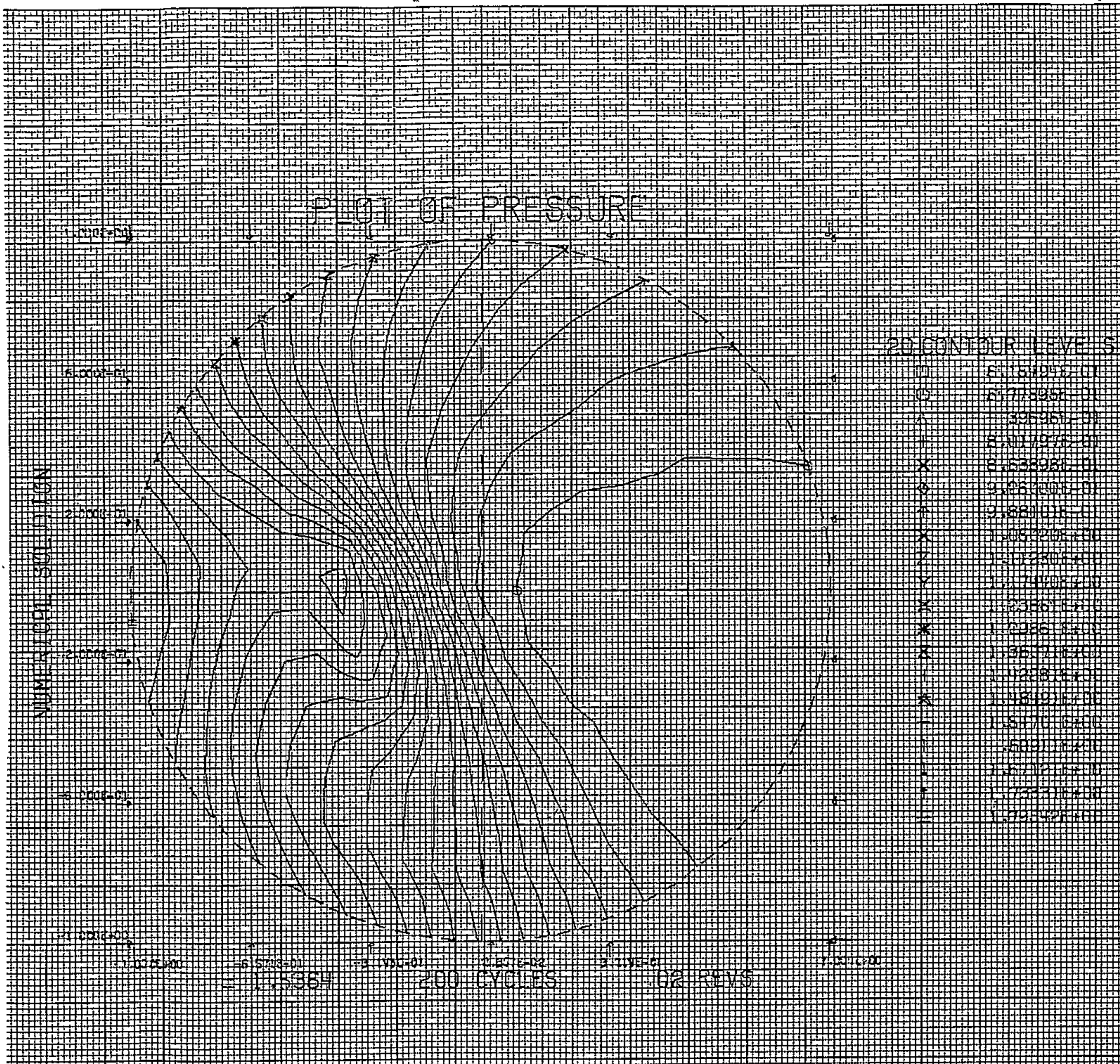

PLOT OF PRESSURE
20 CONTOUR LEVELS
NUMERICAL SOLUTION
200 CYCLES

FIG.2.35

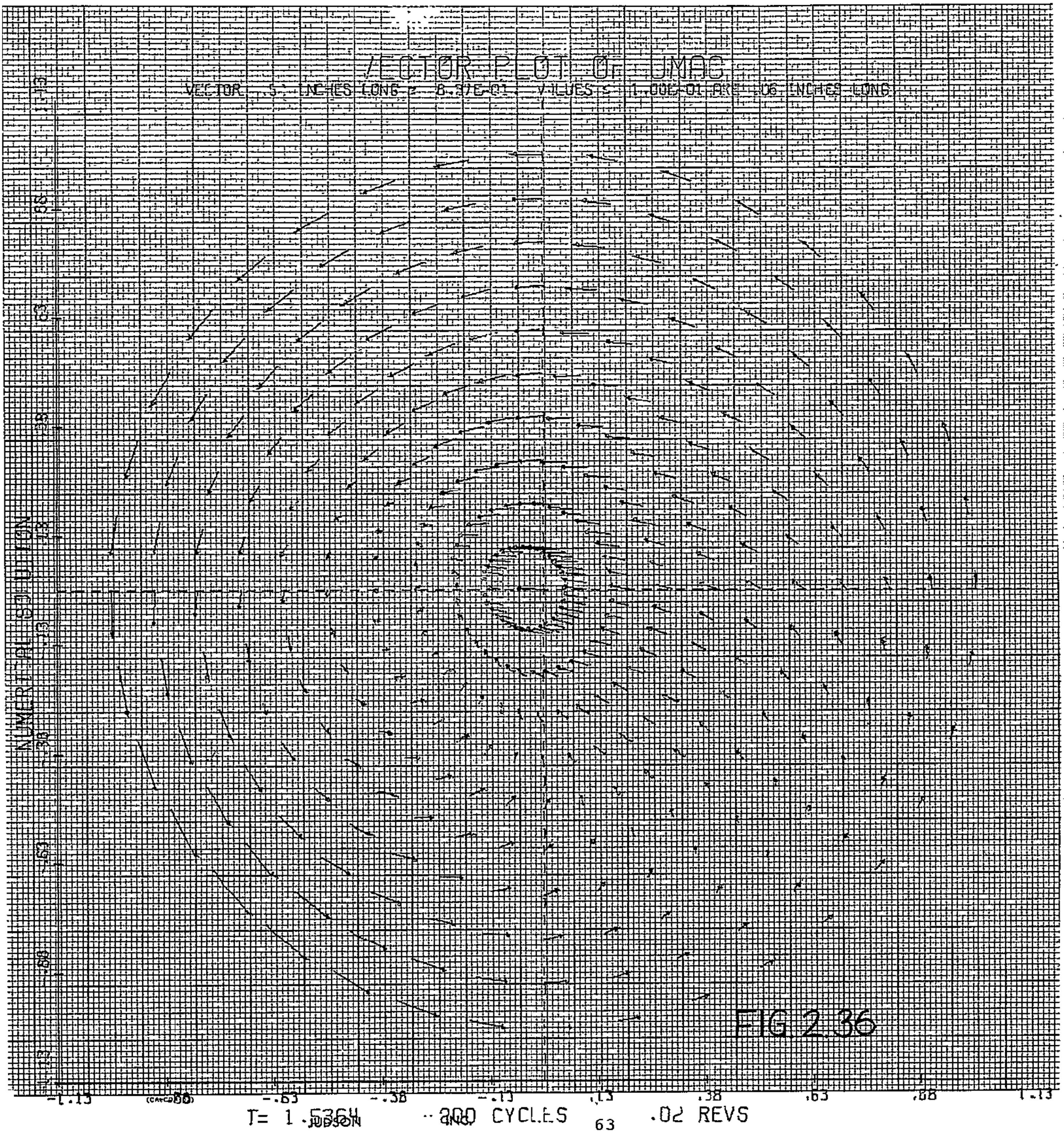

VECTOR PLOT OF UMAG
VECTOR .5 INCHES LONG = 8.37E-01, VALUES ≤ 1.00E-01 ARE .06 INCHES LONG
NUMERICAL SOLUTION
1.13
.88
.63
.38
.13
-.13
-.38
-.63
-.88
-1.13
-1.13
-.88
-.63
-.38
-.13
.13
.38
.63
.88
1.13
T= 1.5364 200 CYCLES .02 REVS

FIG.2.36

GRAPH NO. 1

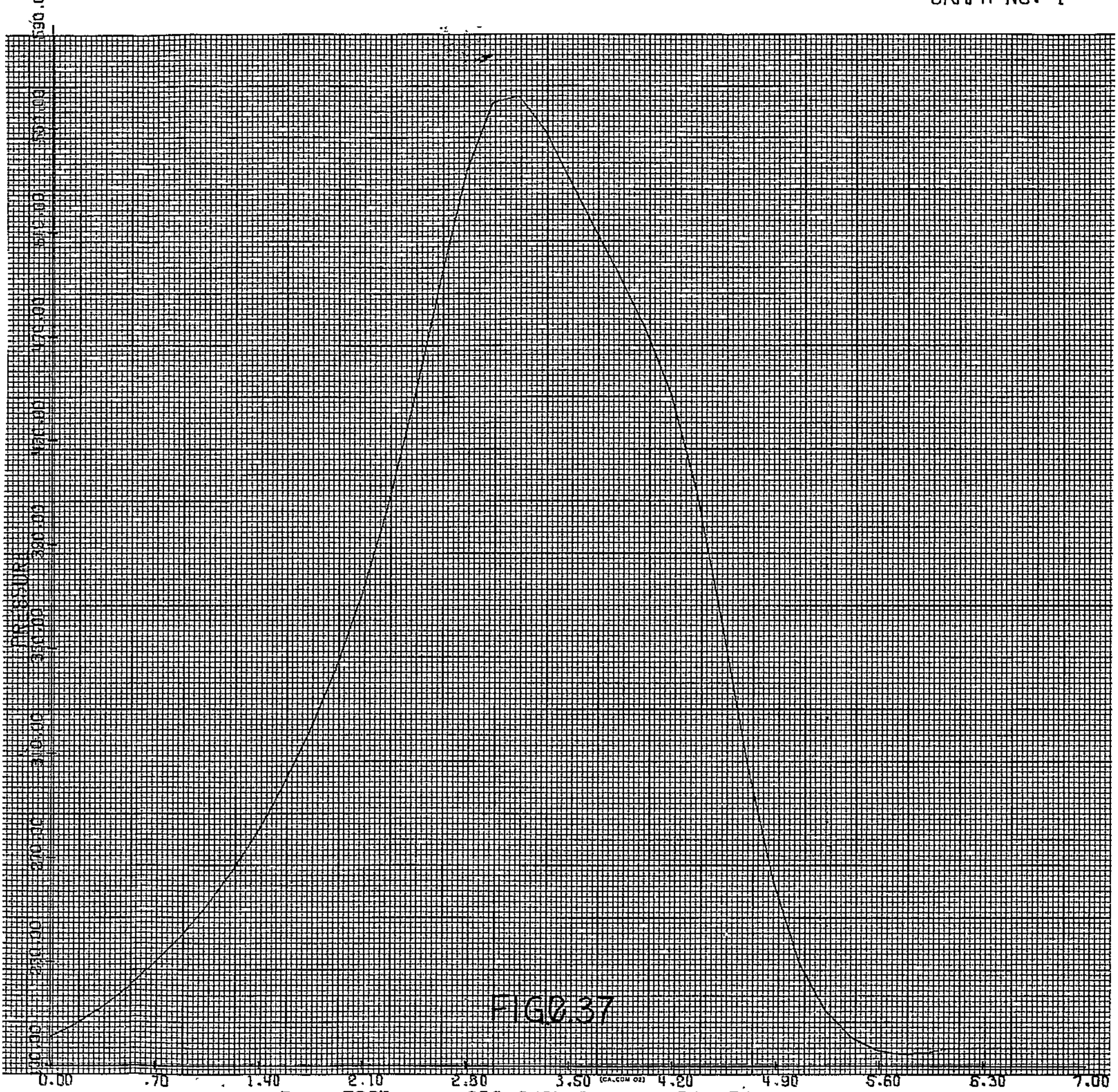


FIG 0.37

T= .5364 200 CYCLES .02 REVS

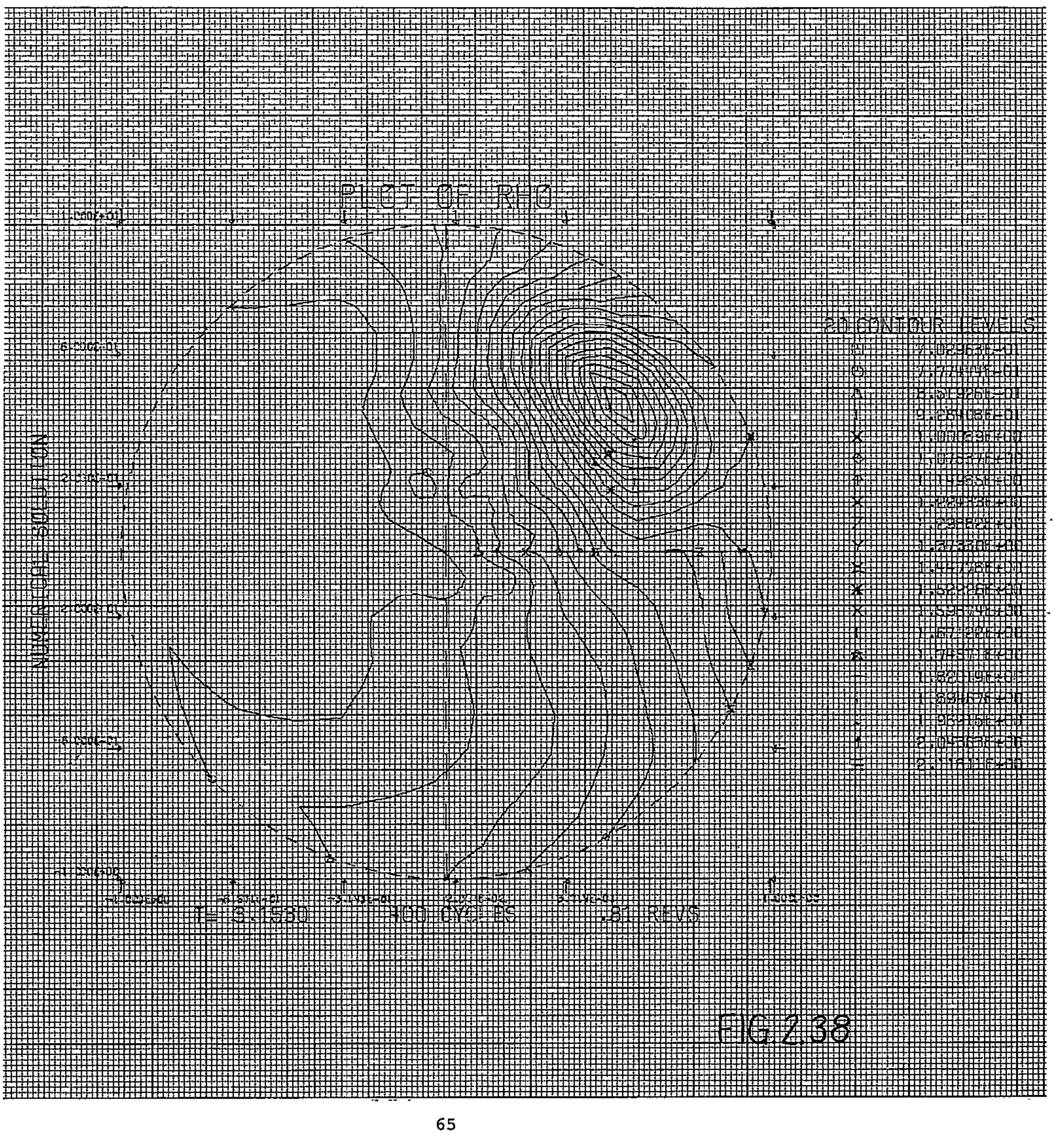

PLOT OF RHO
NUMERICAL SOLUTION
20 CONTOUR LEVELS
T= 3.1530
400 CYCLES
.81 REVS

FIG.2.38

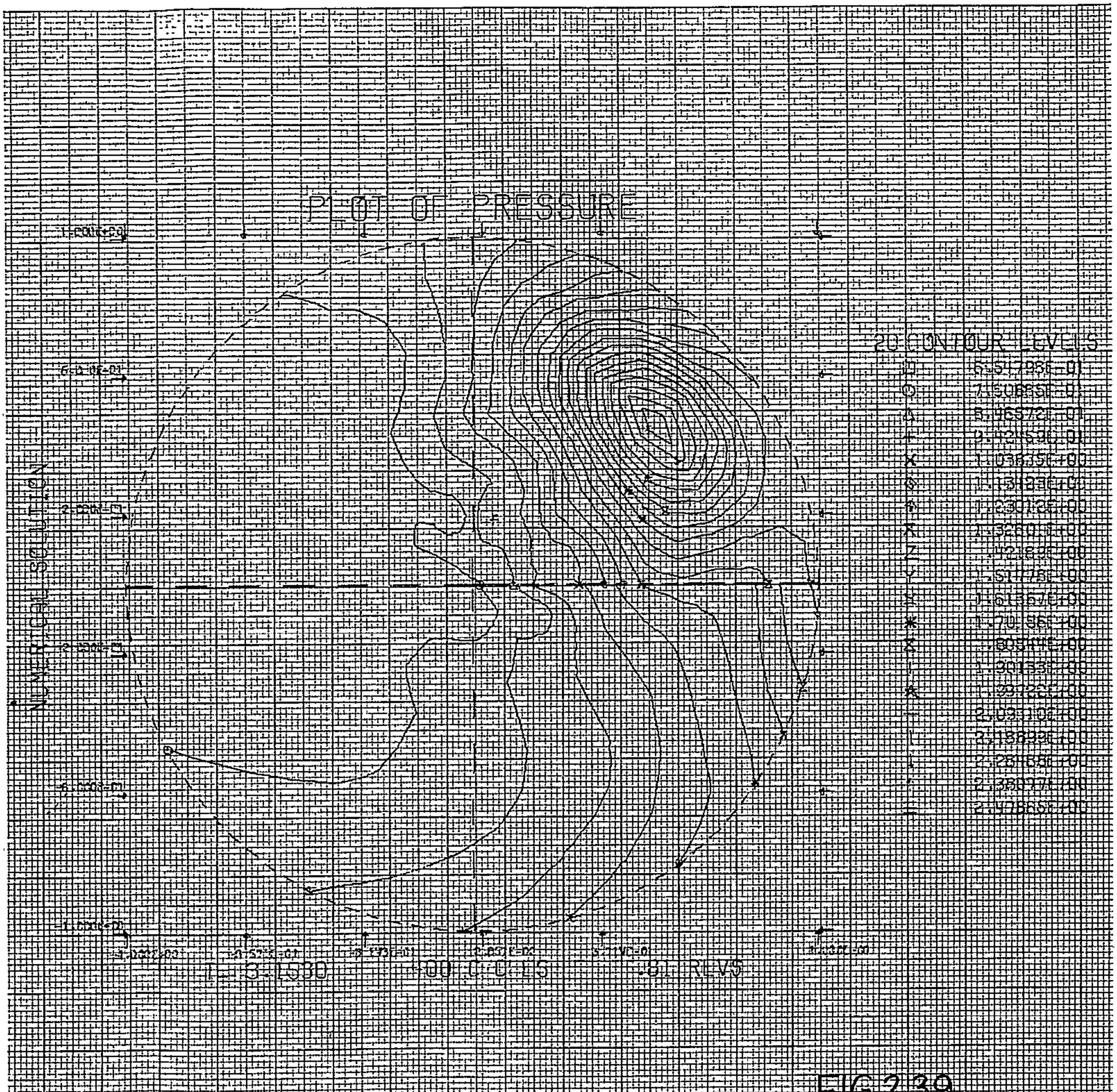

PLOT OF PRESSURE
NUMERICAL SOLUTION
20 CONTOUR LEVELS
.81 REVS

FIG 2.39

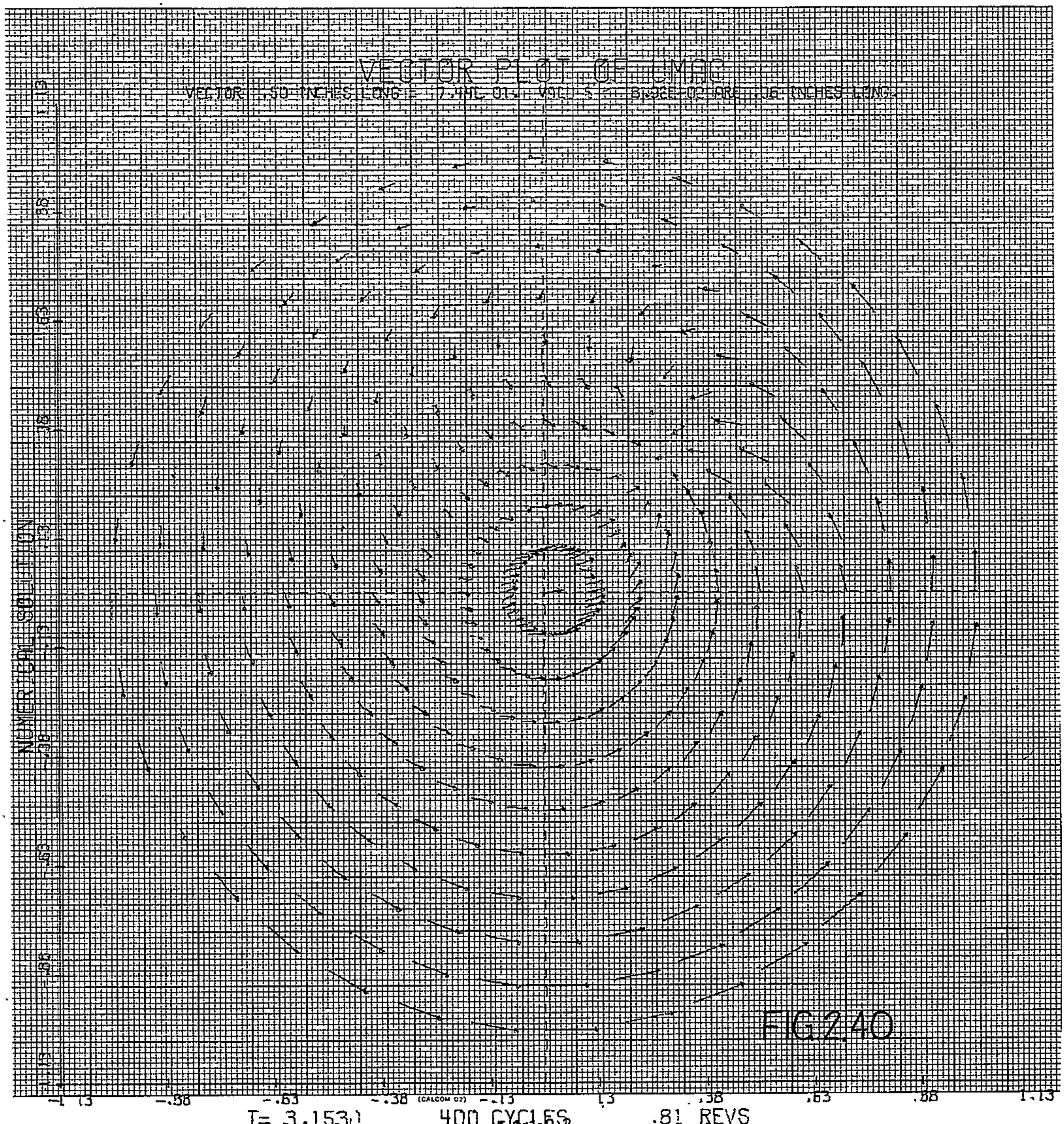
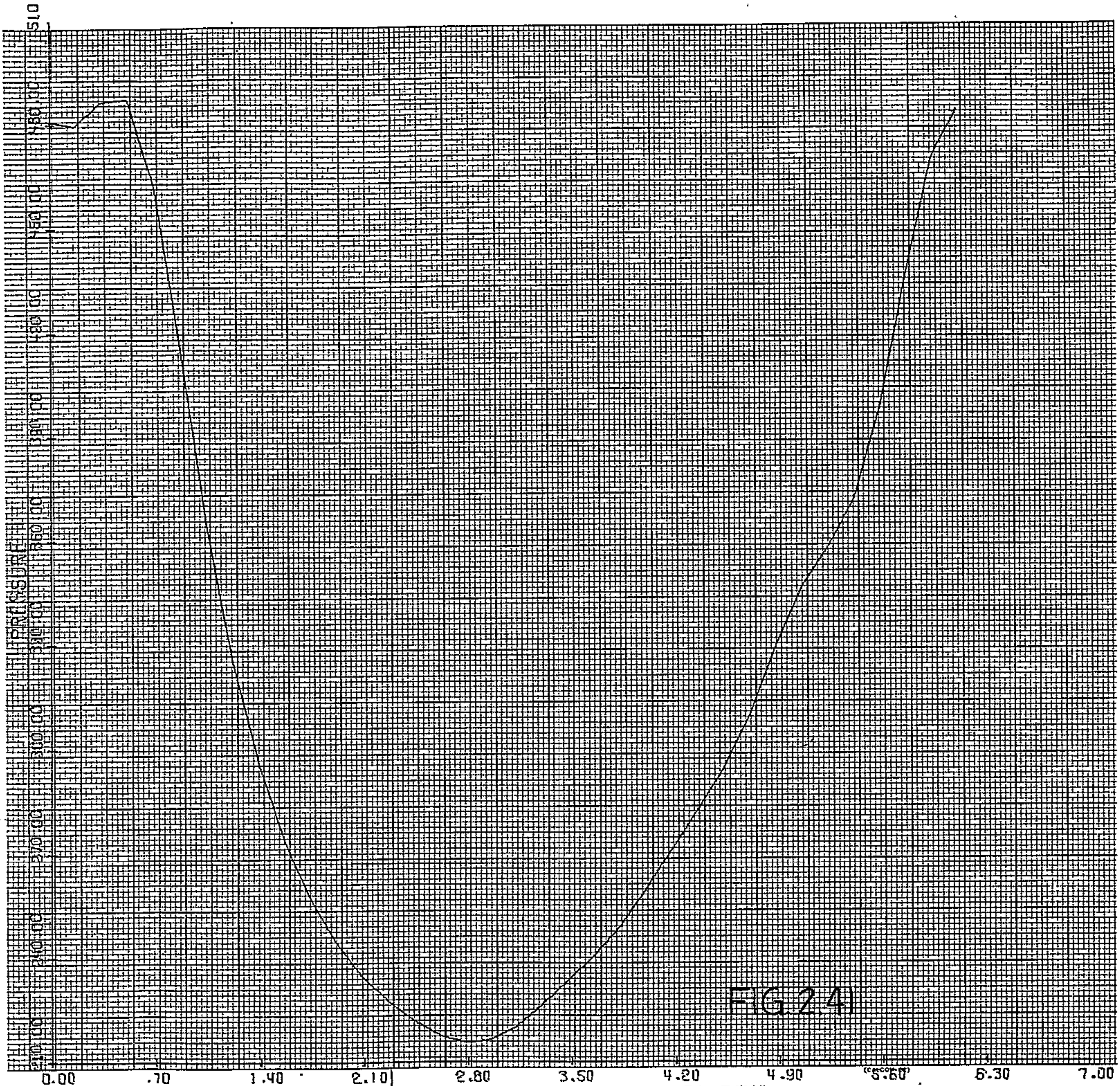

VECTOR PLOT OF UMAG
NUMERICAL SOLUTION
T= 3.1530
400 CYCLES
.81 REVS

FIG.2.40

GRAPH NO. 2



T= 13.1530 400 CYCLES .81 REVS

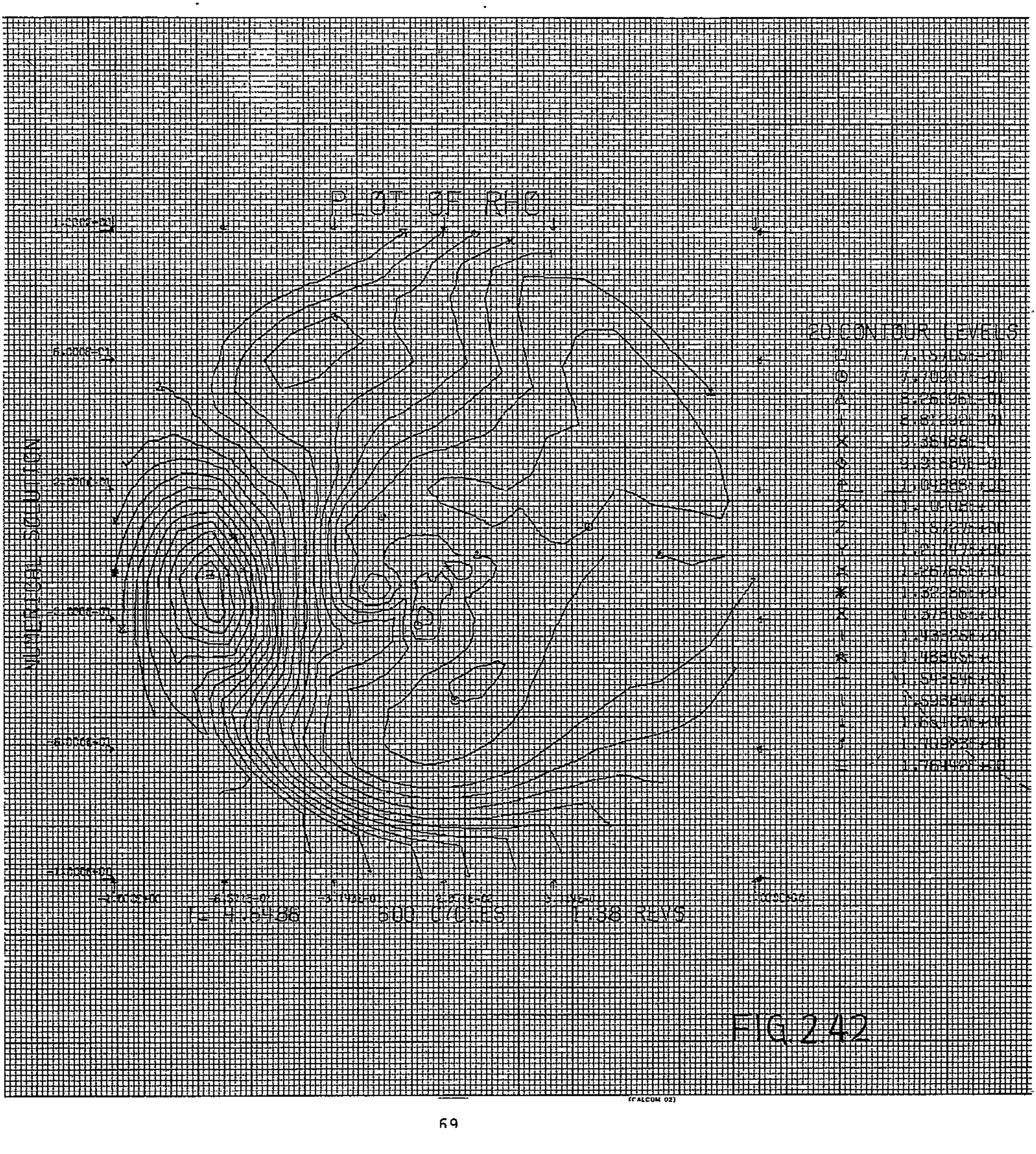

PLOT OF RHO
NUMERICAL SOLUTION
20 CONTOUR LEVELS
T= 4.6486
600 CYCLES
1.38 REVS

FIG.2.42

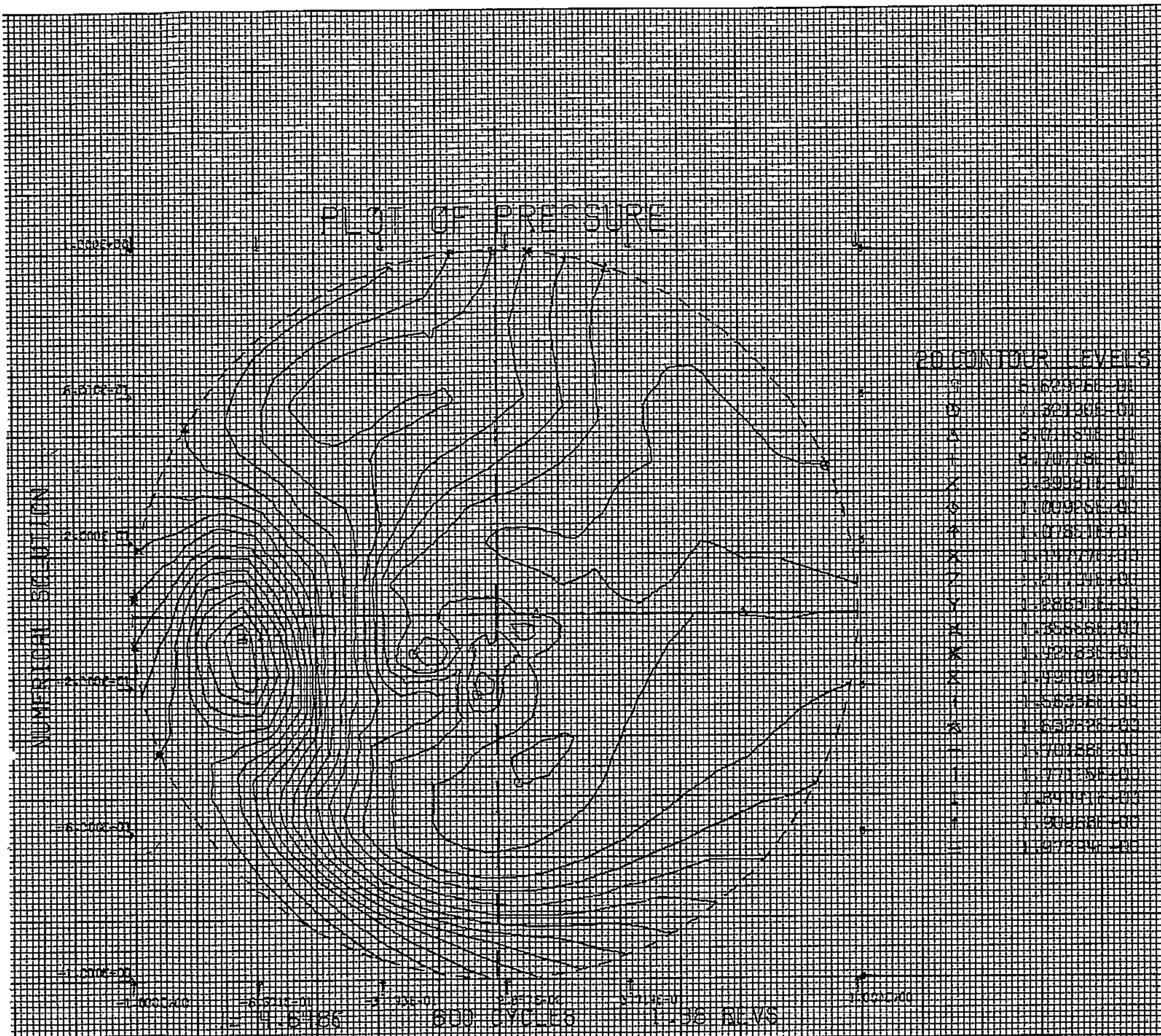

PLOT OF PRESSURE
NUMERICAL SOLUTION
20 CONTOUR LEVELS
600 CYCLES
REVS

FIG. 2.43

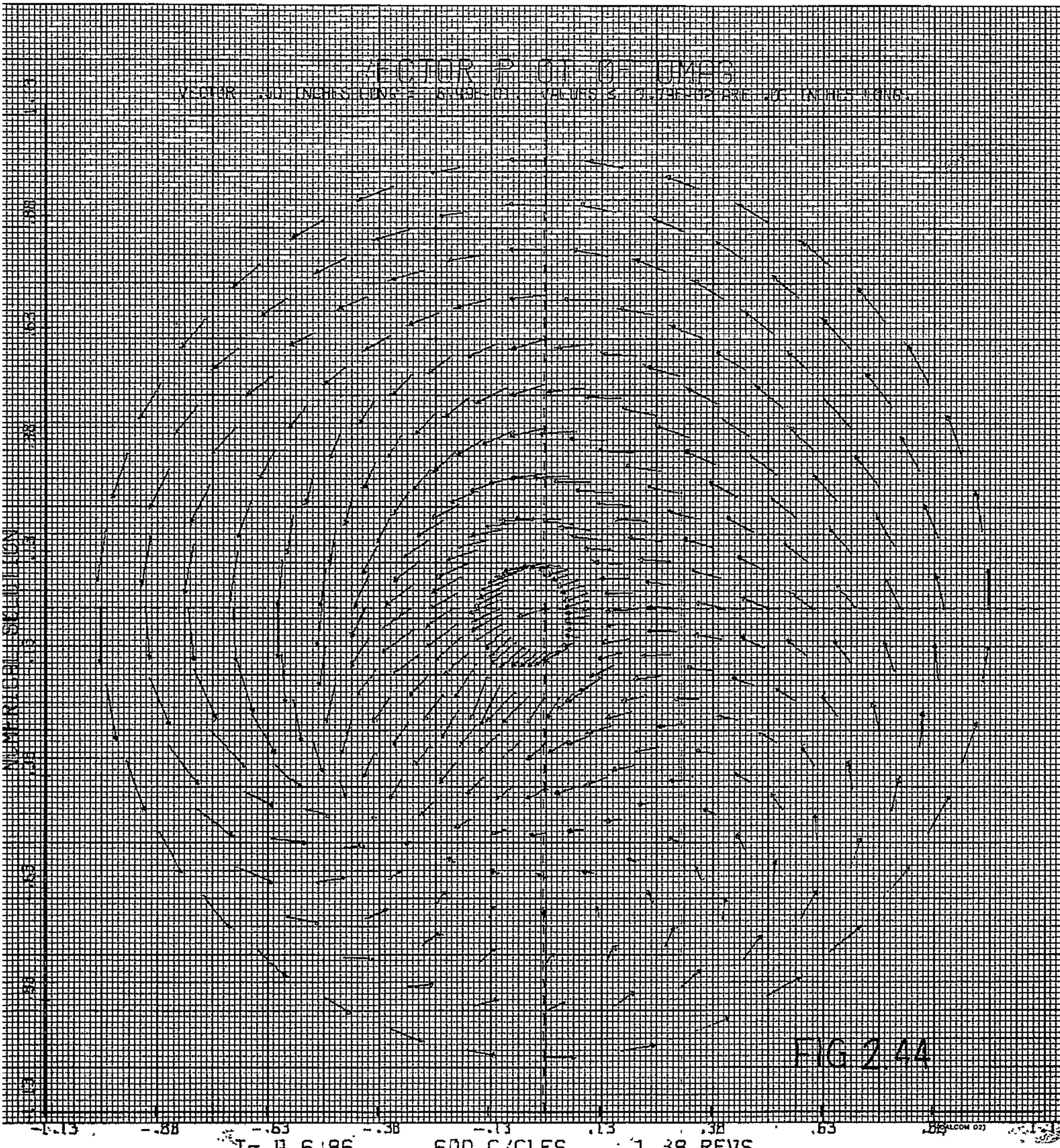


FIG 2.44

T= 4.6 86   600 CYCLES   1.38 REVS

GRAPH NO. 3

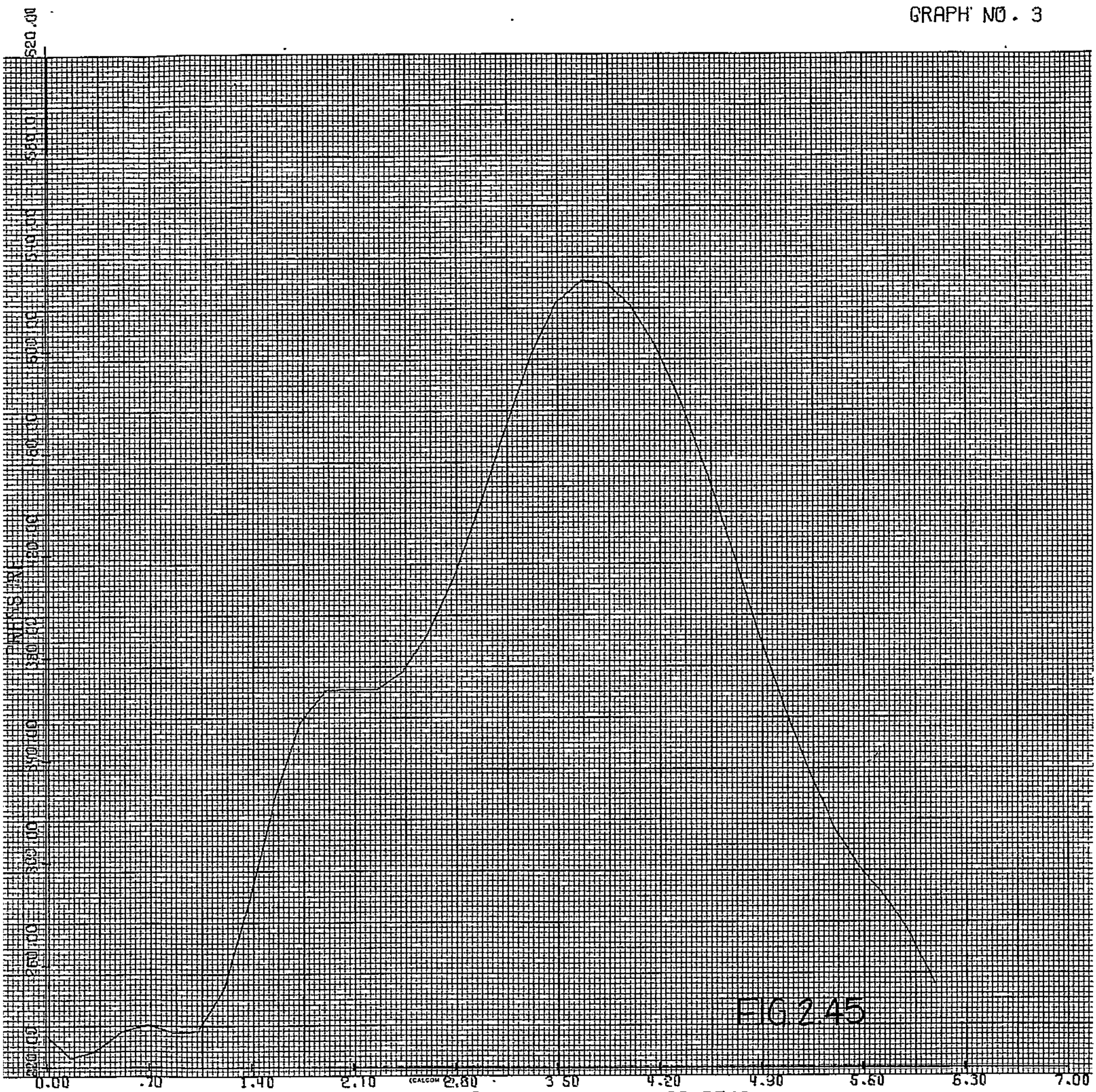
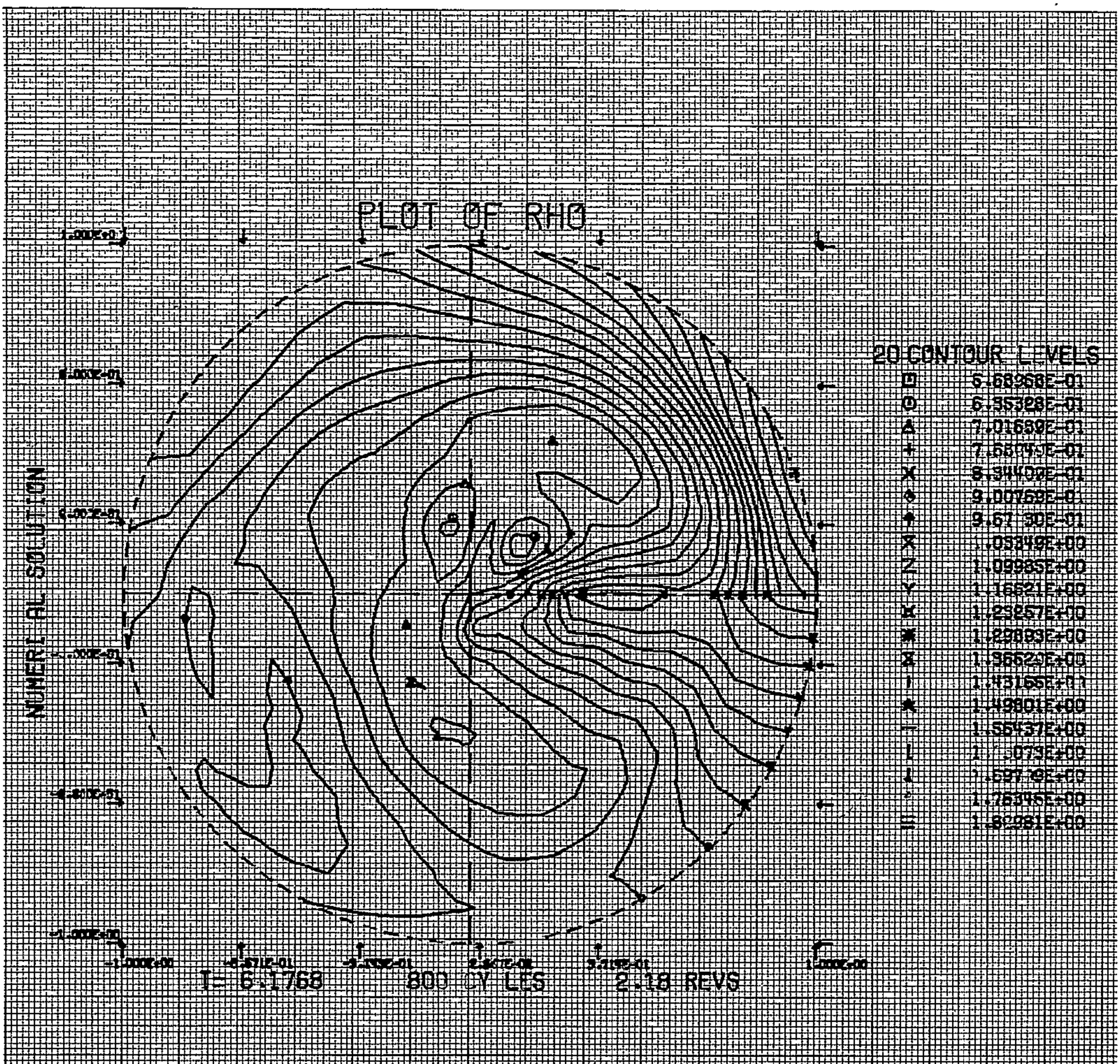


FIG.2.45

T= 4.64SEC   600 CYCLES   1.38 REVS



PLOT OF RHO
NUMERICAL SOLUTION
20 CONTOUR LEVELS
7.01689E-01
9.00769E-01
1.09985E+00
1.16621E+00
1.23257E+00
1.29893E+00
1.49801E+00
1.56437E+00
1.76345E+00
T= 6.1768
800 CYCLES
2.18 REVS

FIG. 2.46

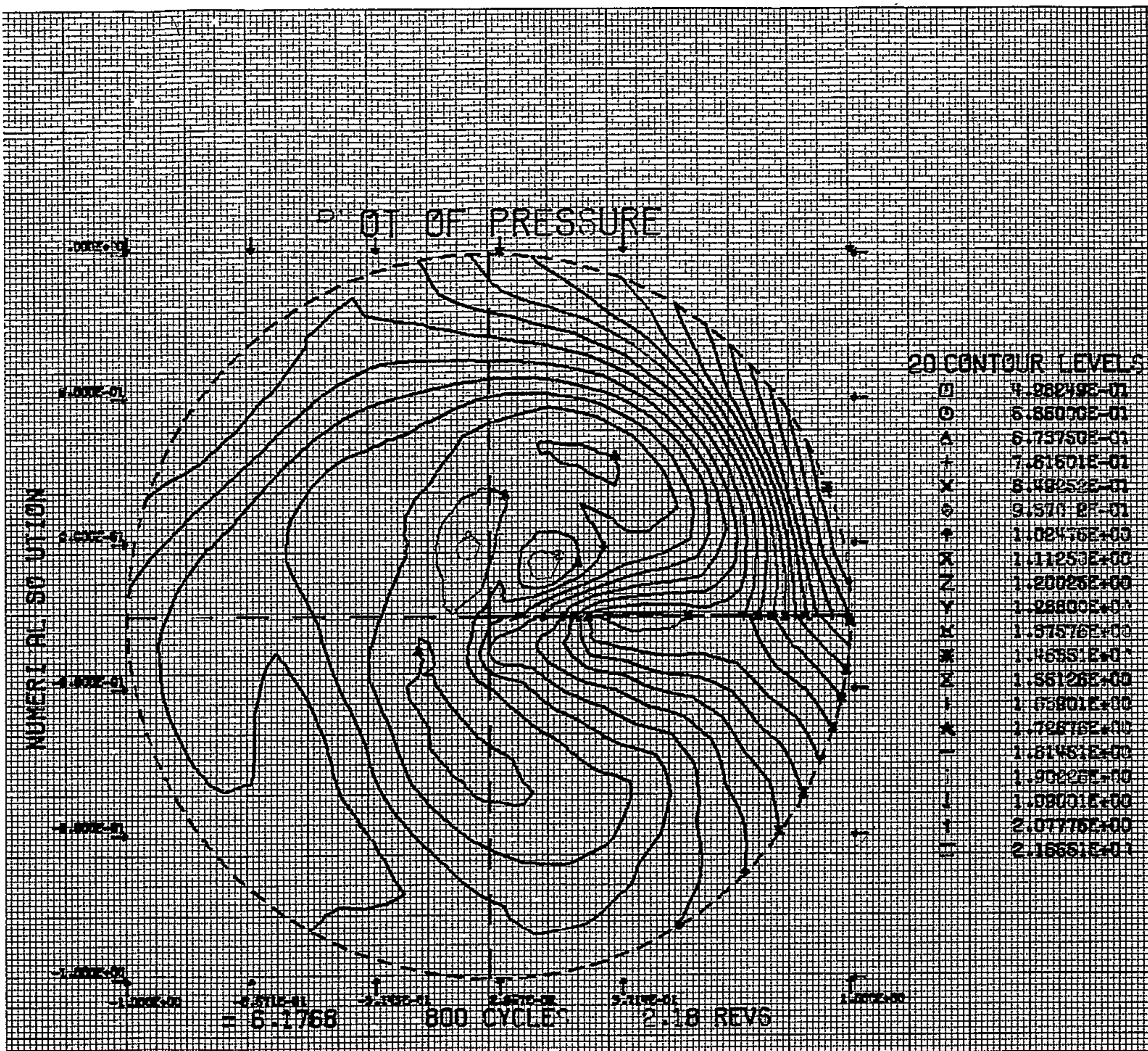
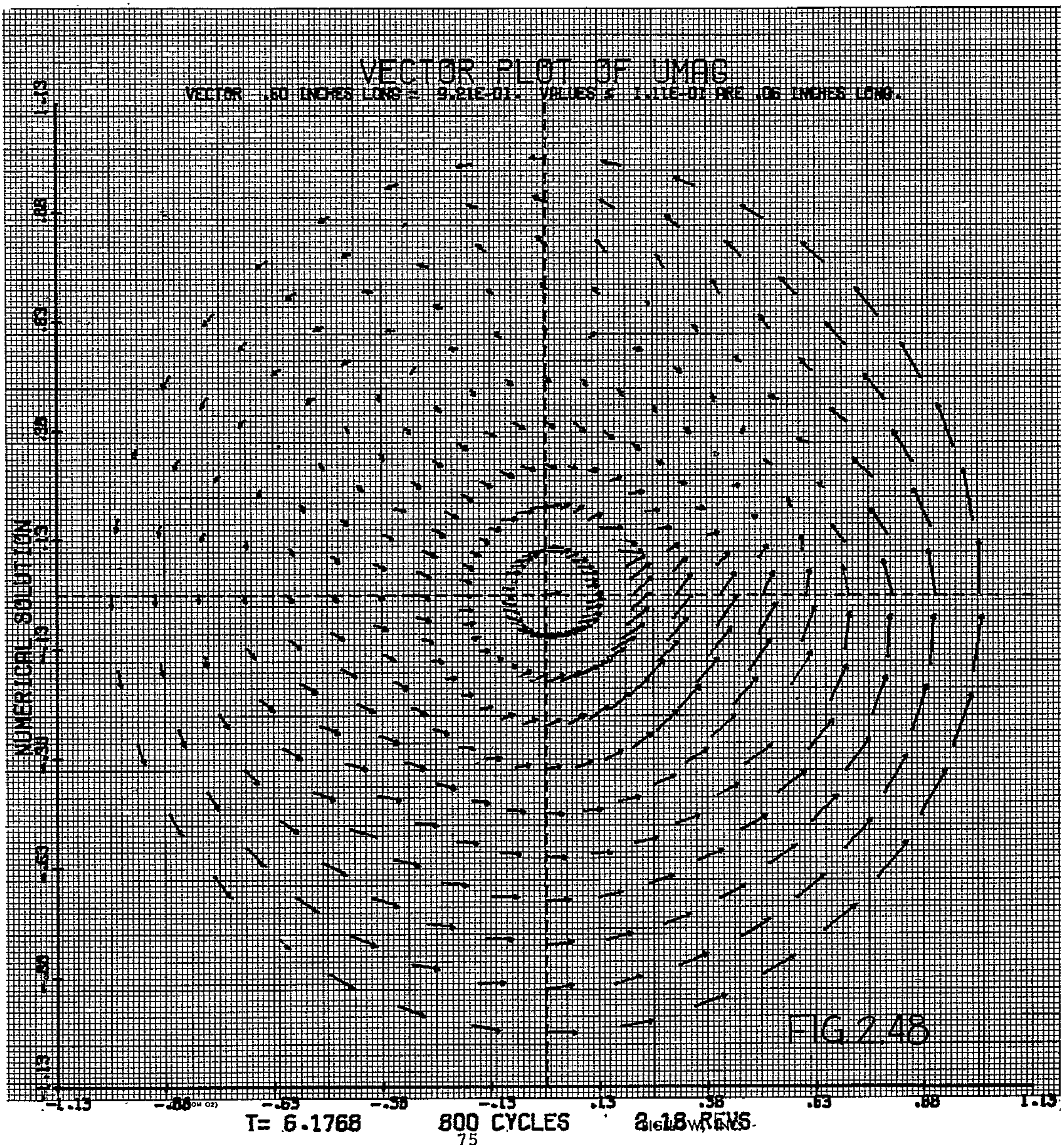

OT OF PRESSURE
20 CONTOUR LEVELS
1.02476E+00
1.20025E+00
2.07776E+00
6.1768
2.18 REVS

FIG. 2.47



VECTOR PLOT OF UMAG
VECTOR .60 INCHES LONG = 9.21E-01. VALUES ≤ 1.11E-01 ARE .06 INCHES LONG.
NUMERICAL SOLUTION
1.13
.88
.63
.38
.13
-.13
-.38
-.63
-.88
-1.13
-1.13
-.88
-.63
-.38
-.13
.13
.38
.63
.88
1.13
T= 6.1768
800 CYCLES
2.18 REVS

FIG.2.48

GRAPH NO. 1

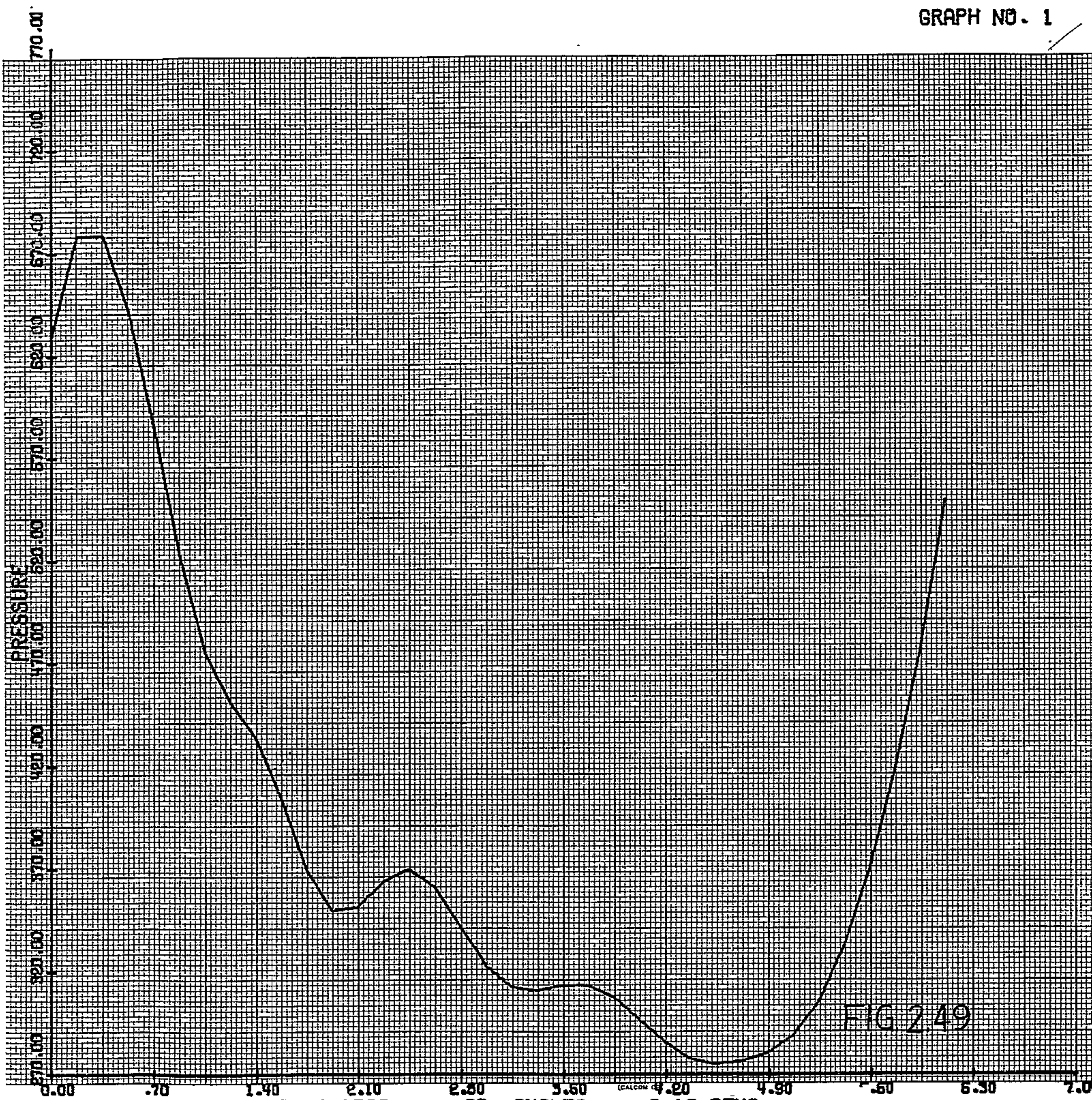


FIG 2.49

T= 6.1768 80, CYCLES 2.18 REVS

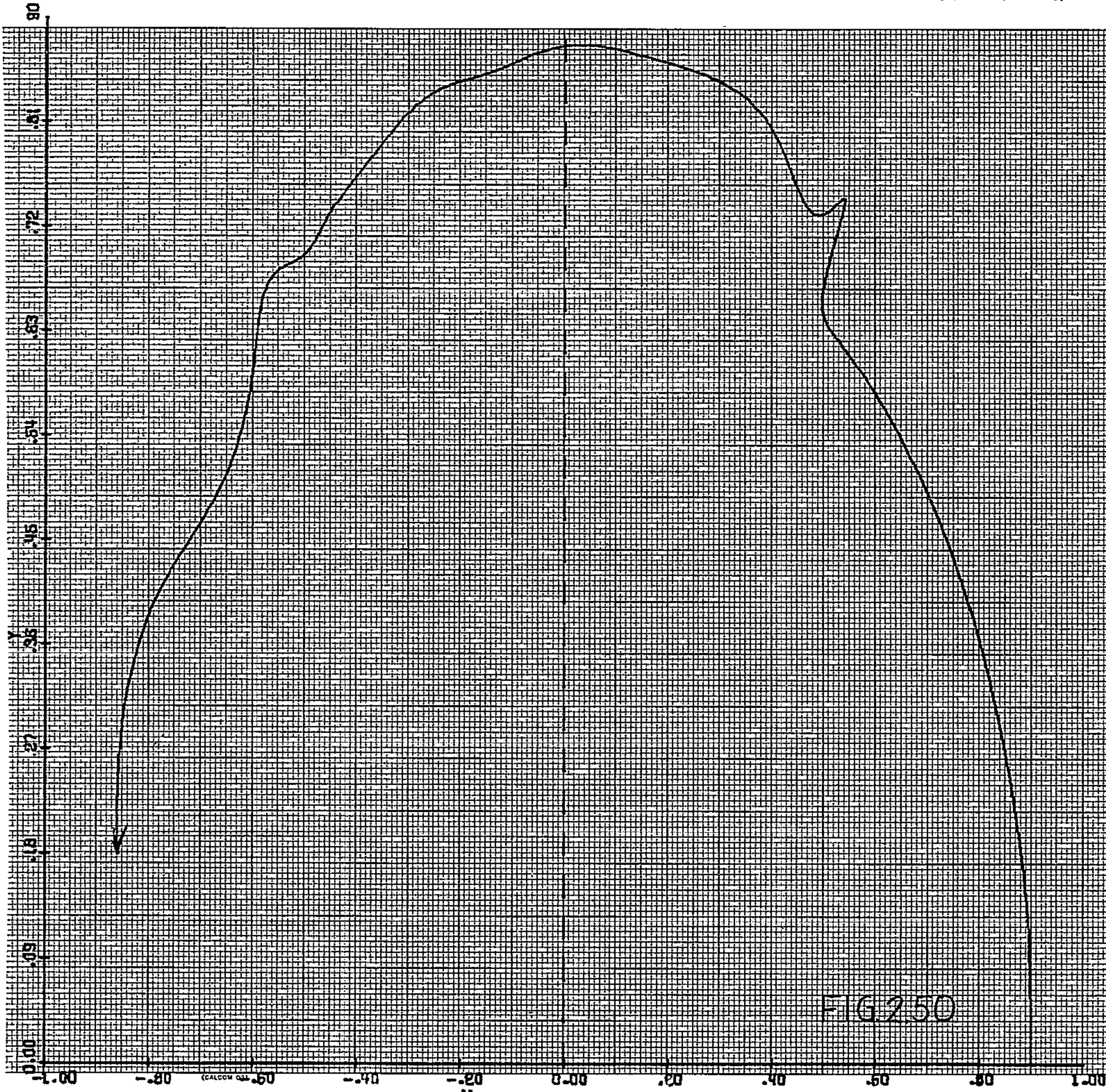


FIG.2.50
Y
X
-1.00
-.80
-.60
-.40
-.20
0.00
.20
.40
.60
.80
1.00
0.00
.09
.18
.27
.36
.45
.54
.63
.72
.81
.90

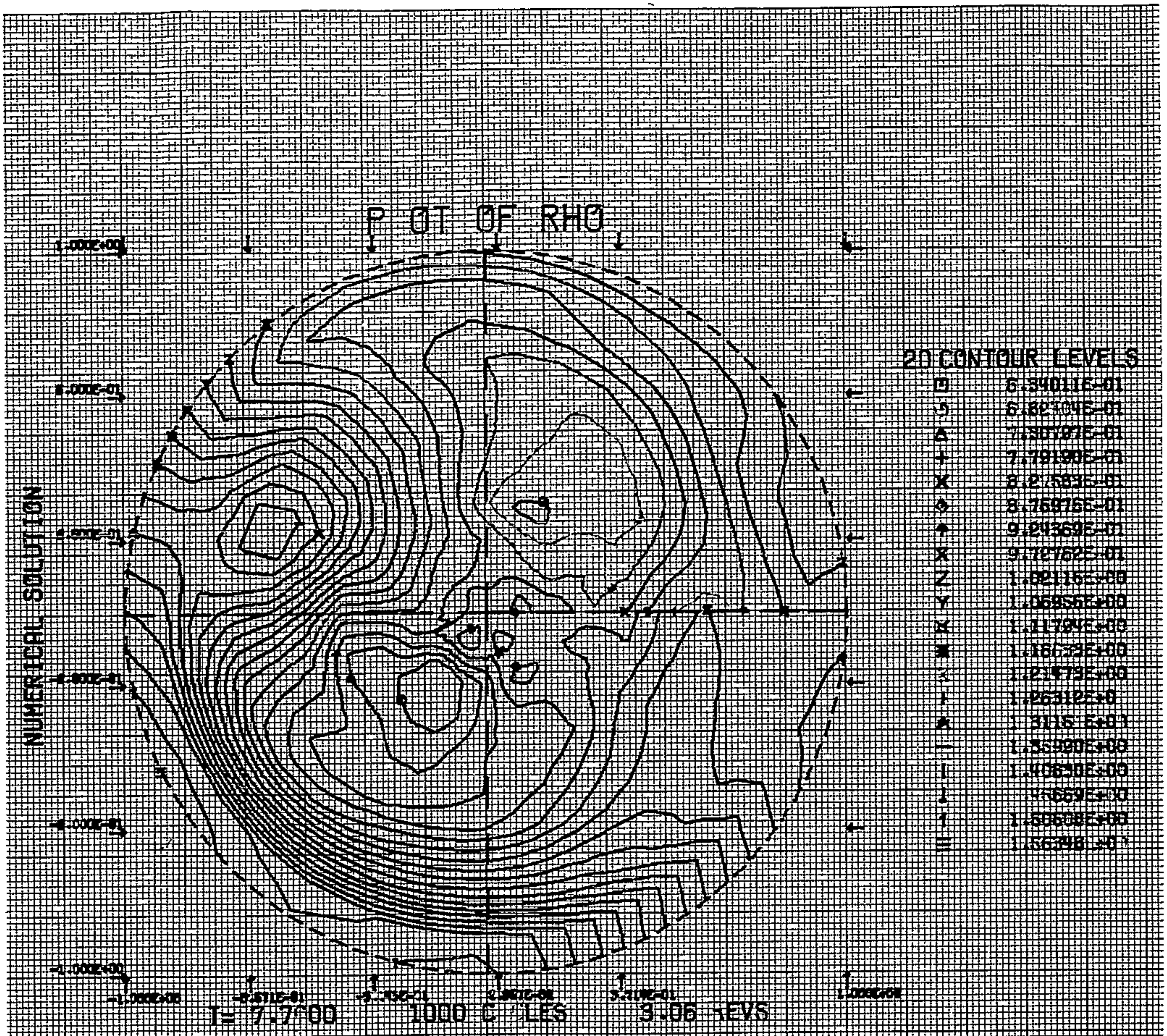

P OT OF RHO
NUMERICAL SOLUTION
20 CONTOUR LEVELS
6.34011E-01
7.79190E-01
8.76976E-01
9.24369E-01
1.02116E+00
1.06855E+00
1.11794E+00
1.21479E+00
1.35990E+00
1.40830E+00
1.50608E+00

FIG.2.51

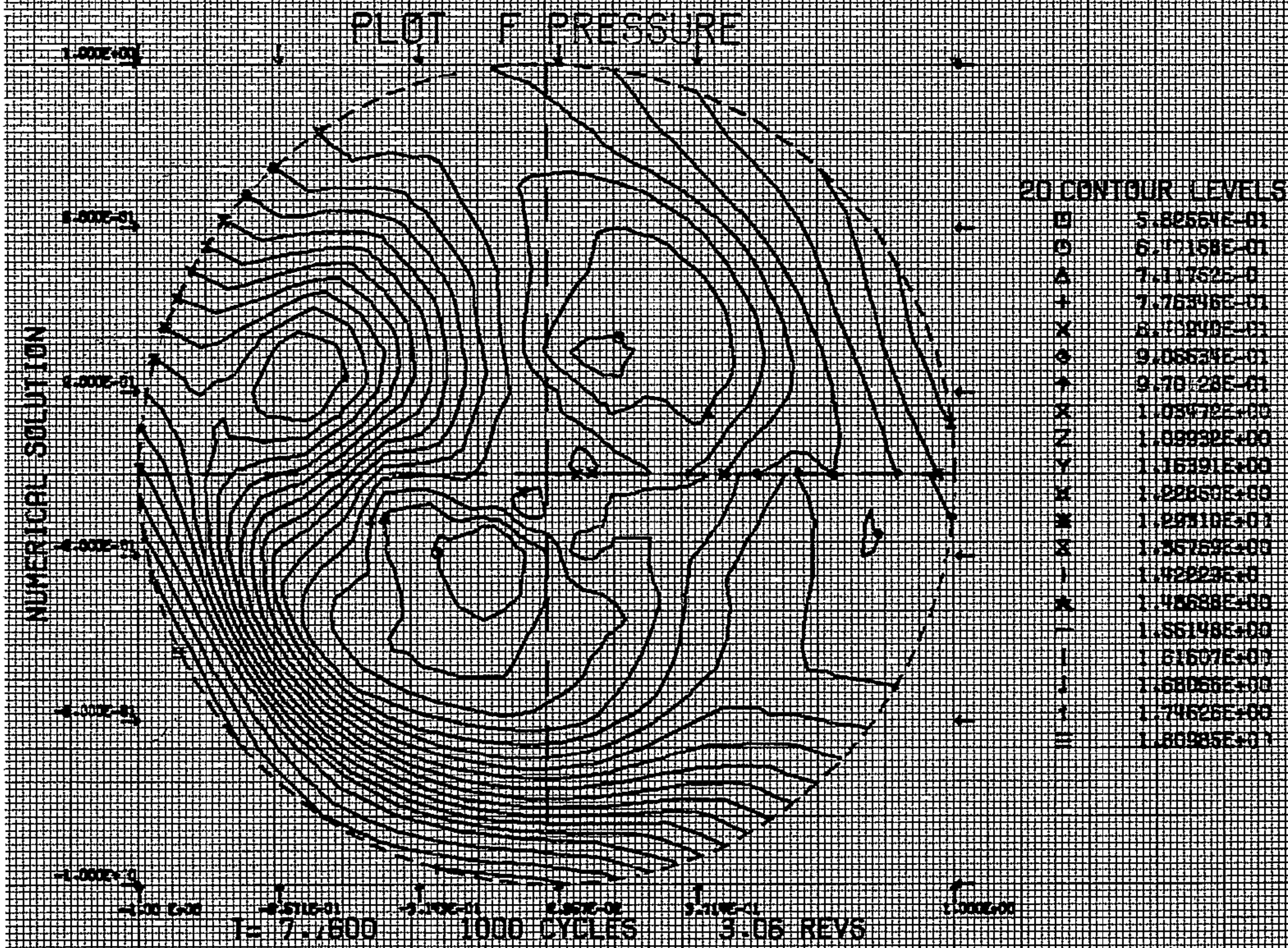

PLOT F PRESSURE
NUMERICAL SOLUTION
20 CONTOUR LEVELS
7.76346E-01
1.08932E+00
1.16391E+00
1.22850E+00
1.48688E+00
1.55148E+00
1.74626E+00
1000 CYCLES
3.06 REVS

FIG.2.52

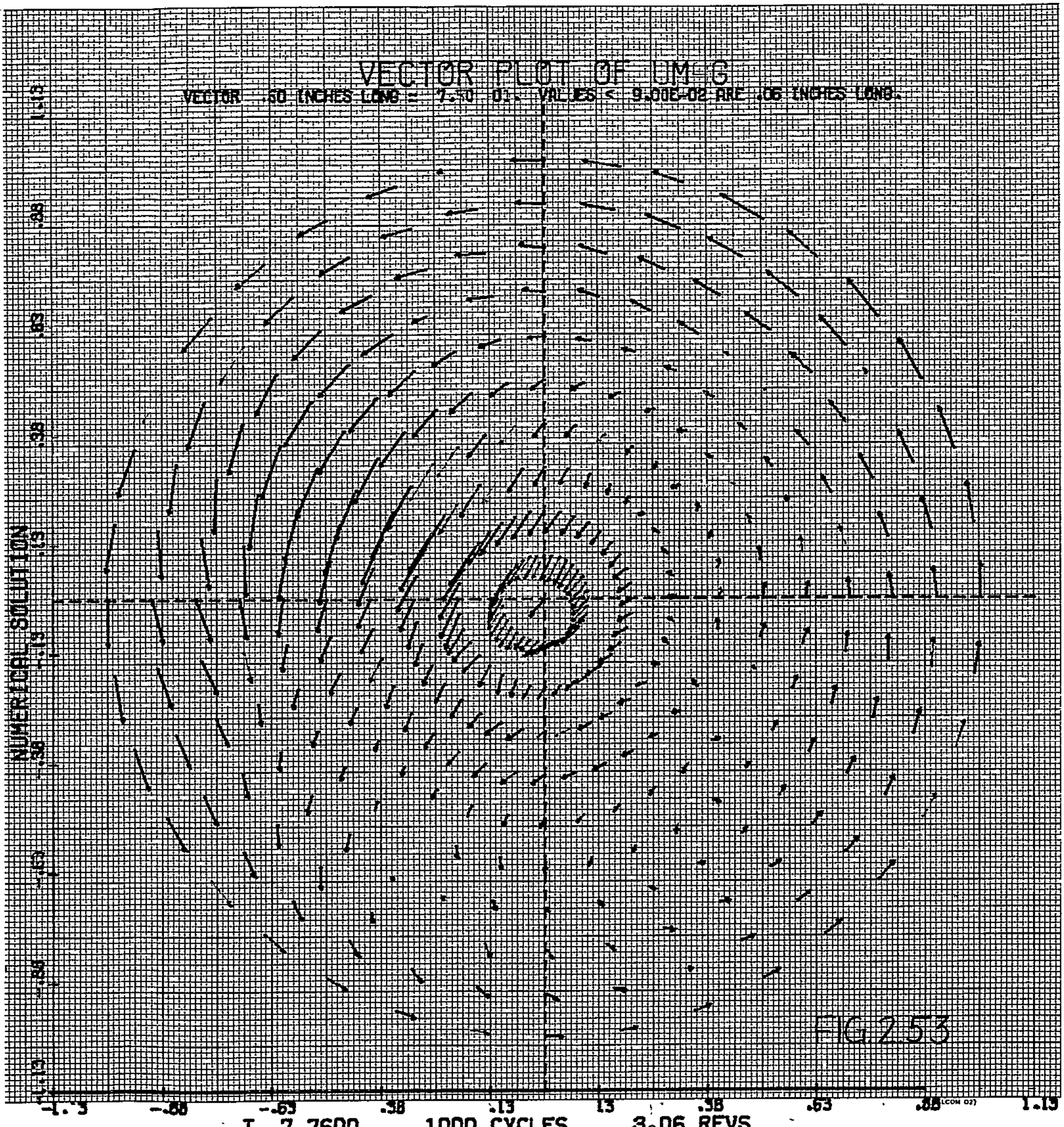

VECTOR PLOT OF UM-G
VECTOR .50 INCHES LONG = 7.50 01. VALUES < 9.00E-02 ARE .06 INCHES LONG.
NUMERICAL SOLUTION
1.13
.88
.63
.38
.13
-.13
-.38
-.63
-.88
-1.13
-.88
-.63
-.38
-.13
.13
.38
.63
.88
1.13
T- 7.7600
1000 CYCLES
3.06 REVS

FIG.2.53

GRAPH NO. 3

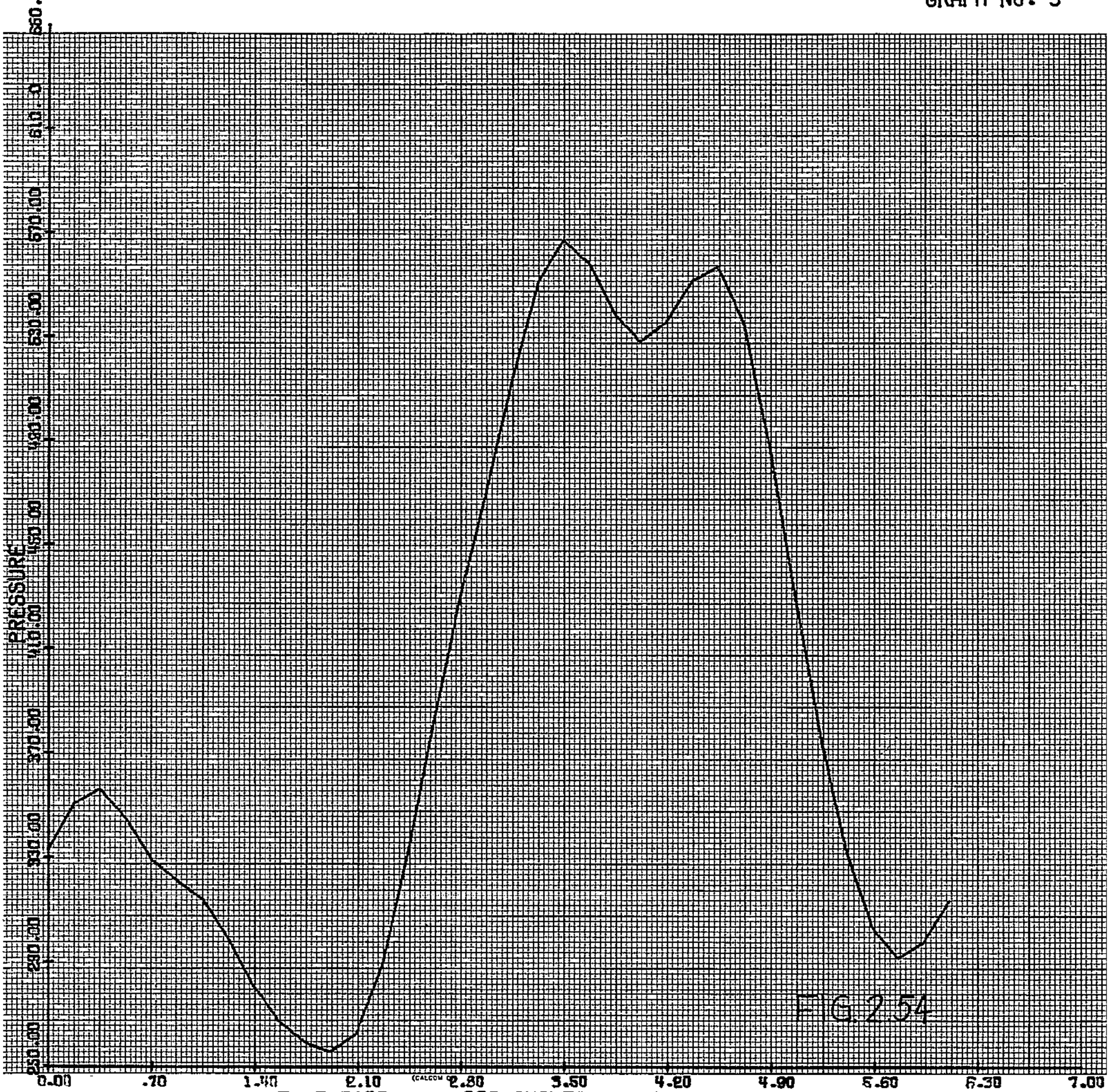
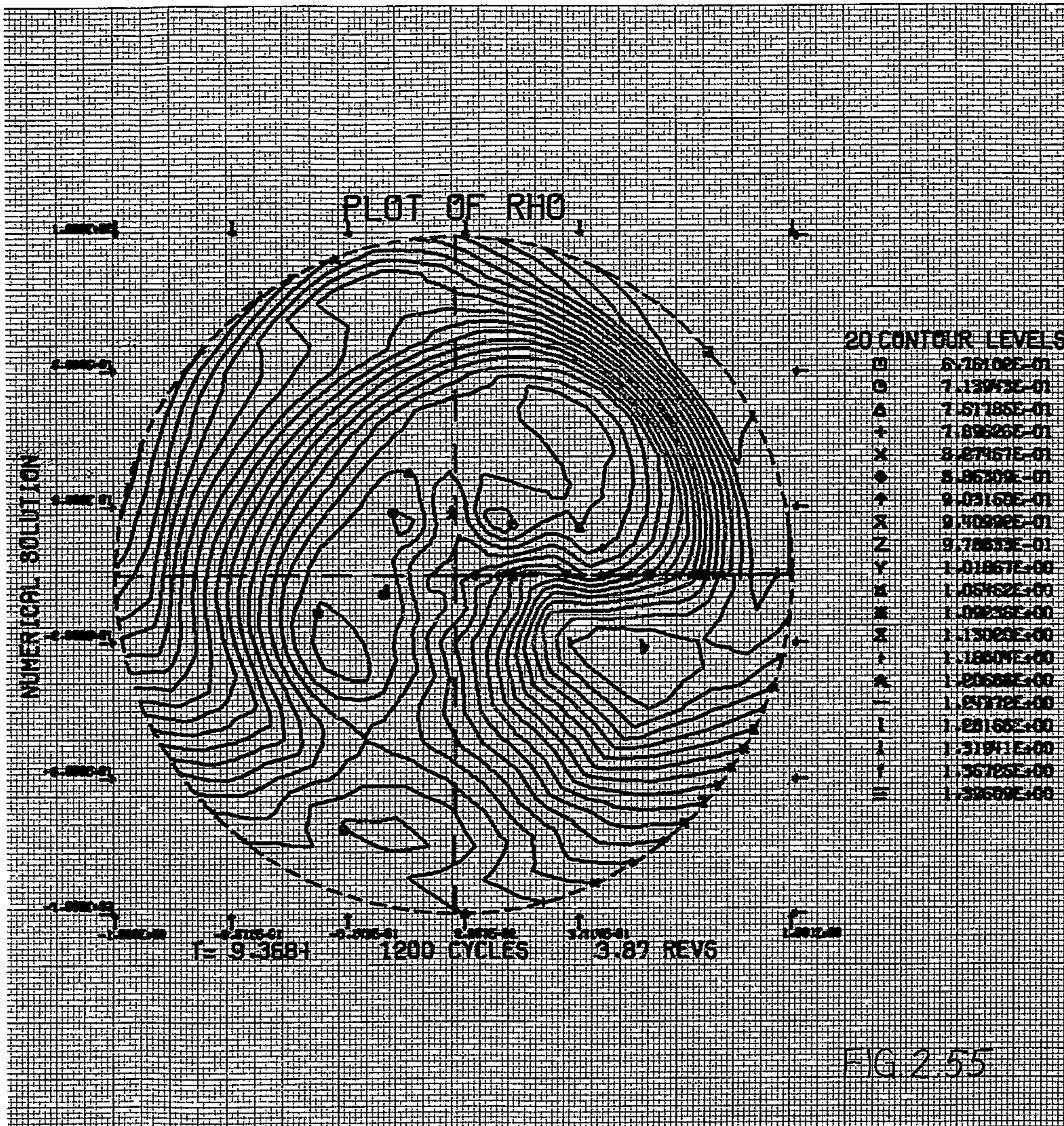


FIG.2.54

T= 7.7600 1000 CYCLES 3.06 REVS



PLOT OF RHO
NUMERICAL SOLUTION
20 CONTOUR LEVELS
T= 9.3684
1200 CYCLES
3.87 REVS

FIG. 2.55

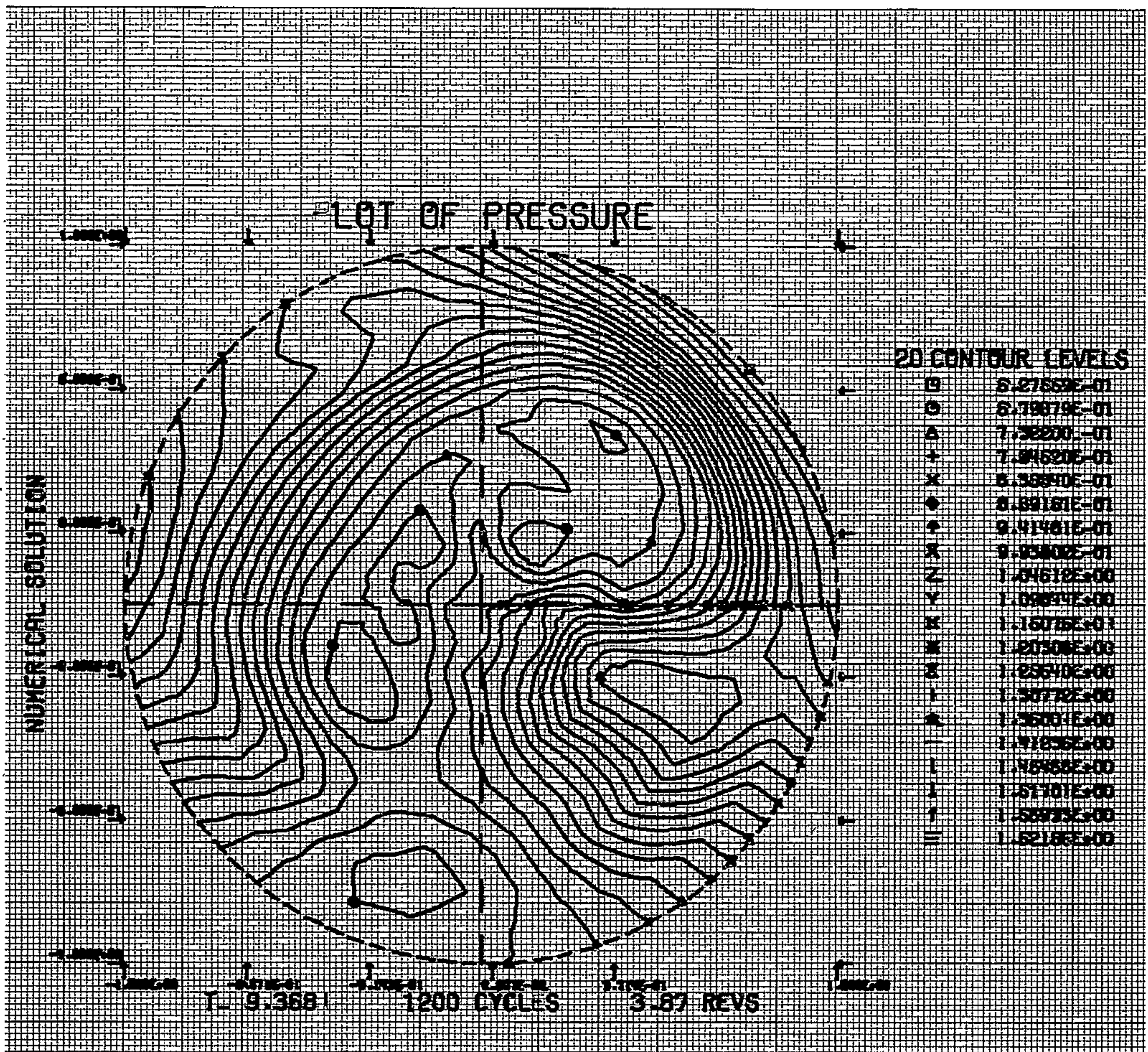

PLOT OF PRESSURE
NUMERICAL SOLUTION
20 CONTOUR LEVELS
T= 9.368
1200 CYCLES
3.87 REVS

FIG.2.56

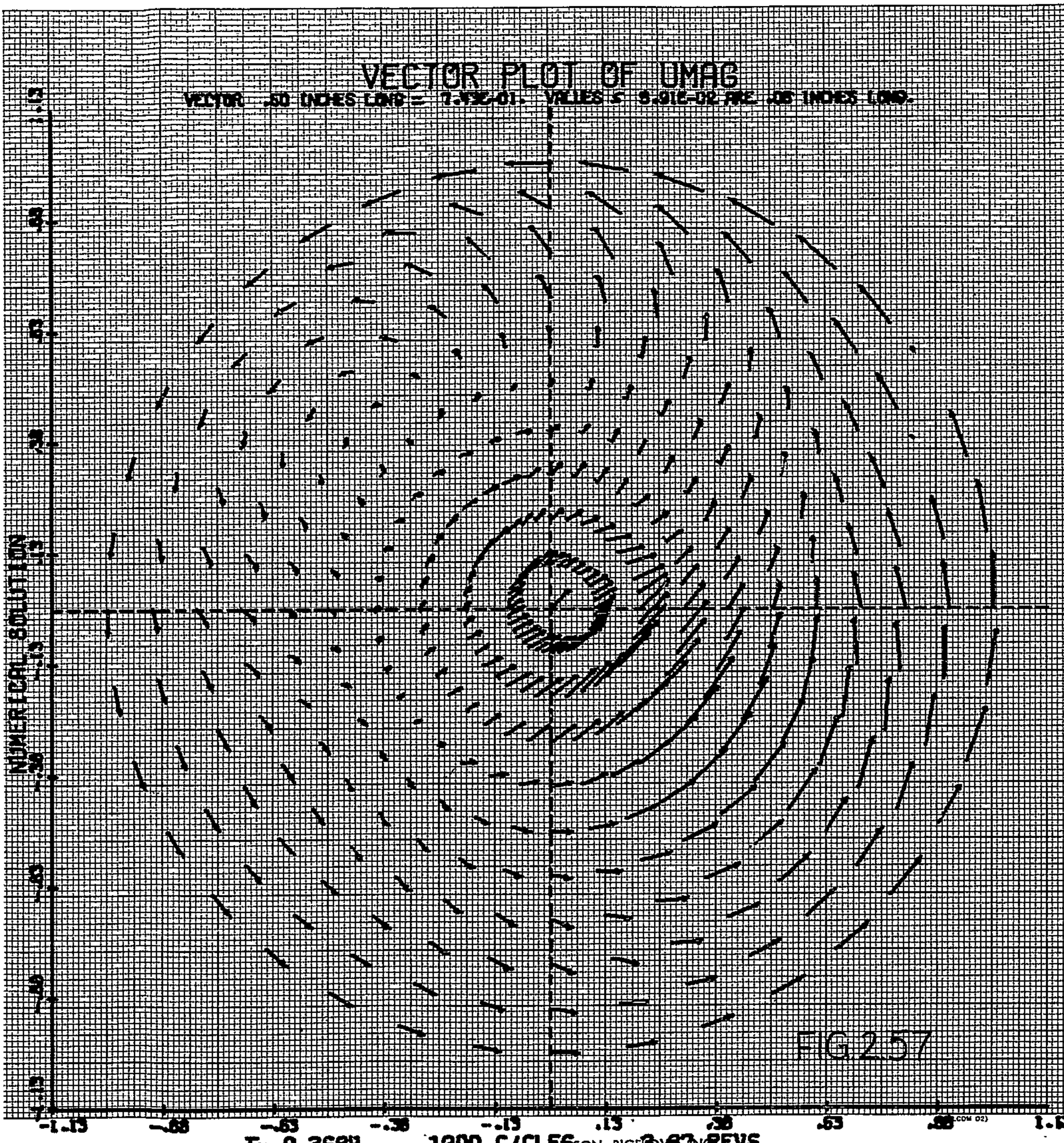

VECTOR PLOT OF UMAG
NUMERICAL SOLUTION
-1.13
-.88
-.63
-.38
-.13
.13
.38
.63
.88
1.13
T= 9.3684
1200 CYCLES
3.87 REVS

FIG.2.57

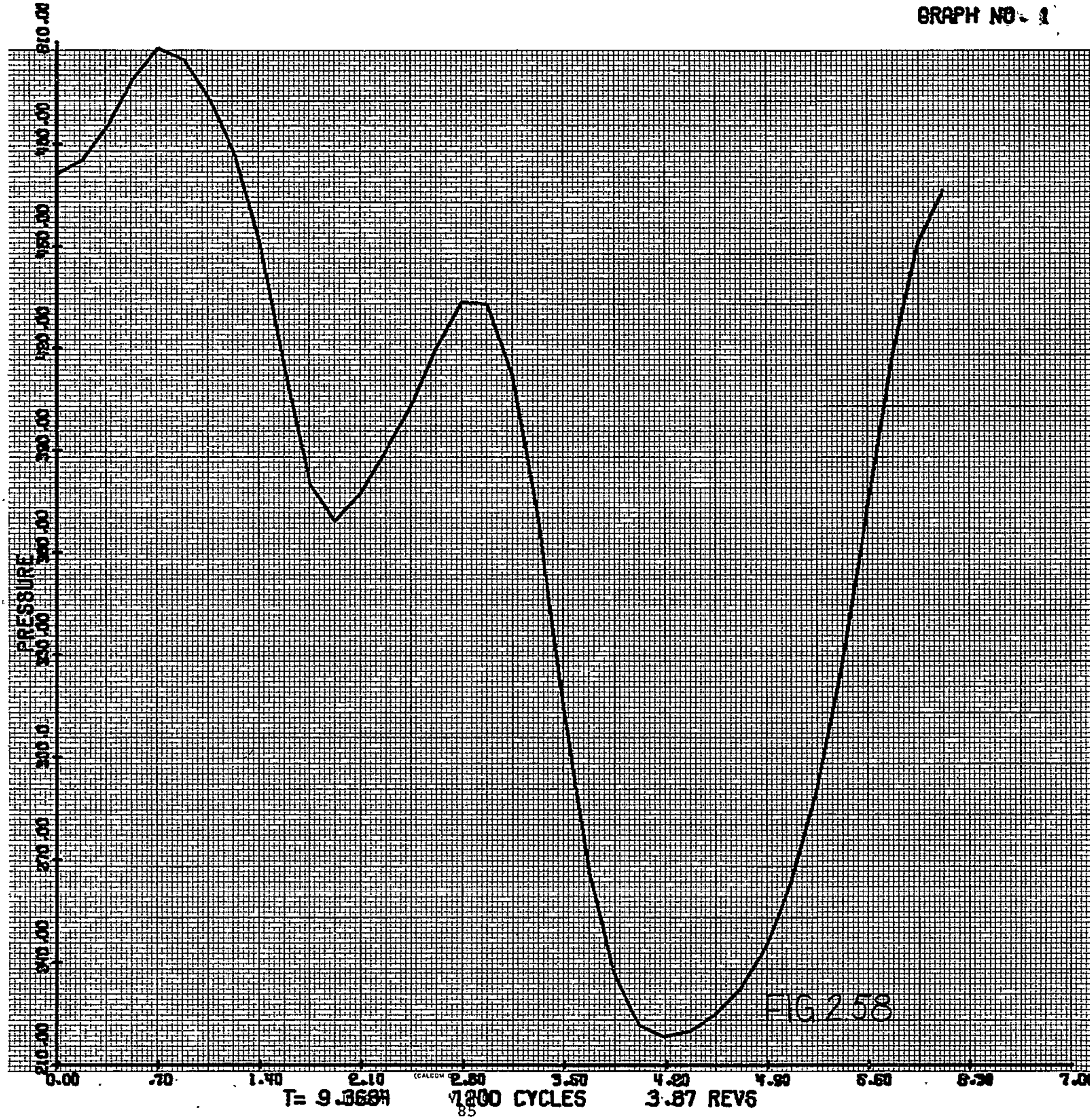

GRAPH NO. 1
PRESSURE
0.00
.70
1.40
2.10
2.80
3.50
4.20
4.90
5.60
6.30
7.00
T= 9.3684
1200 CYCLES
3.87 REVS

FIG 2.58

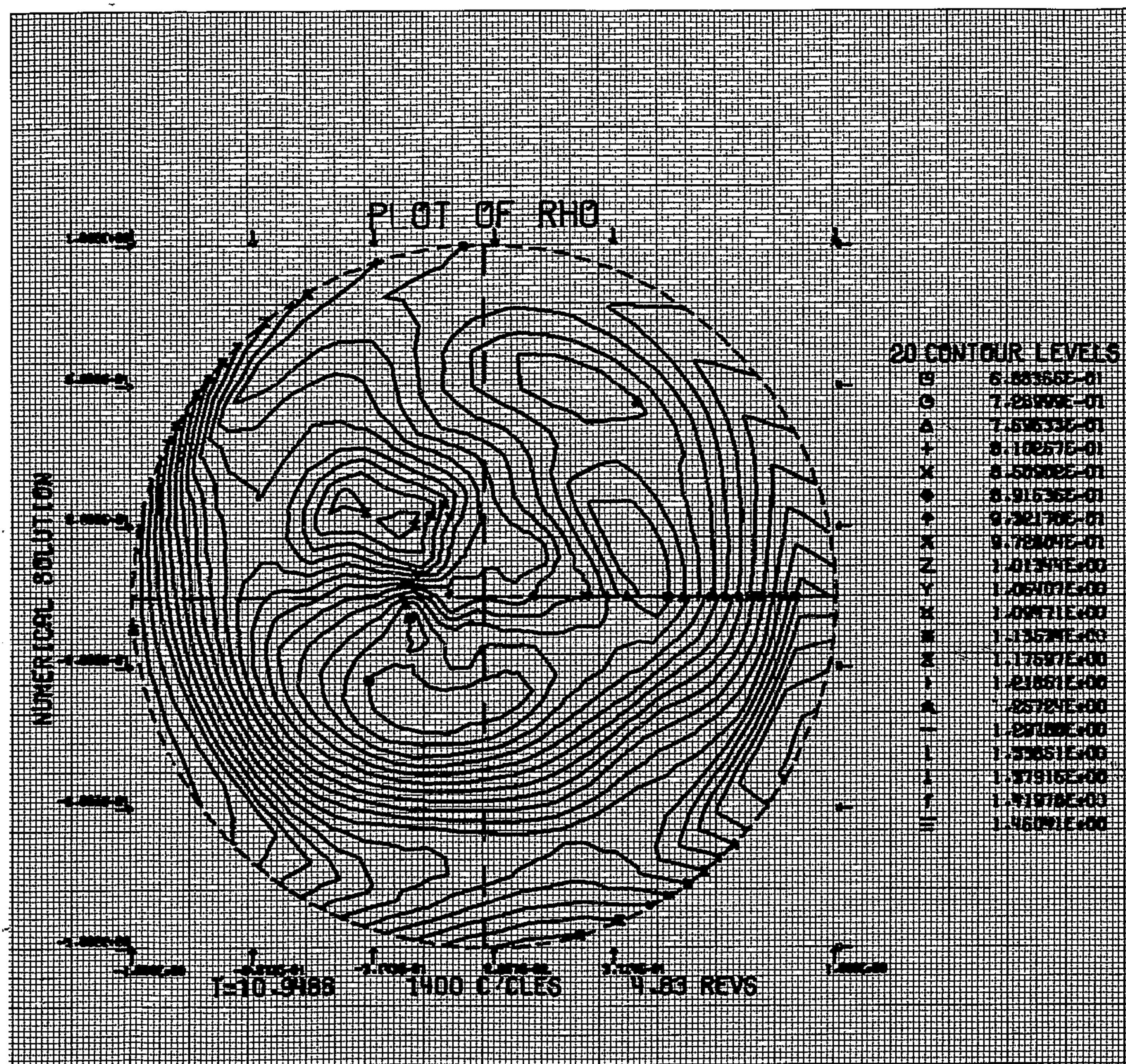

PLOT OF RHO
NUMERICAL SOLUTION
20 CONTOUR LEVELS
6.88365E-01
7.29999E-01
7.69633E-01
8.10257E-01
8.50902E-01
8.91536E-01
9.32170E-01
9.72804E-01
1.01344E+00
1.05407E+00
1.09471E+00
1.13534E+00
1.17597E+00
1.21661E+00
1.25724E+00
1.29788E+00
1.33851E+00
1.37915E+00
1.41978E+00
1.46042E+00
T=10.9488 1400 CYCLES 4.83 REVS

FIG.2.59

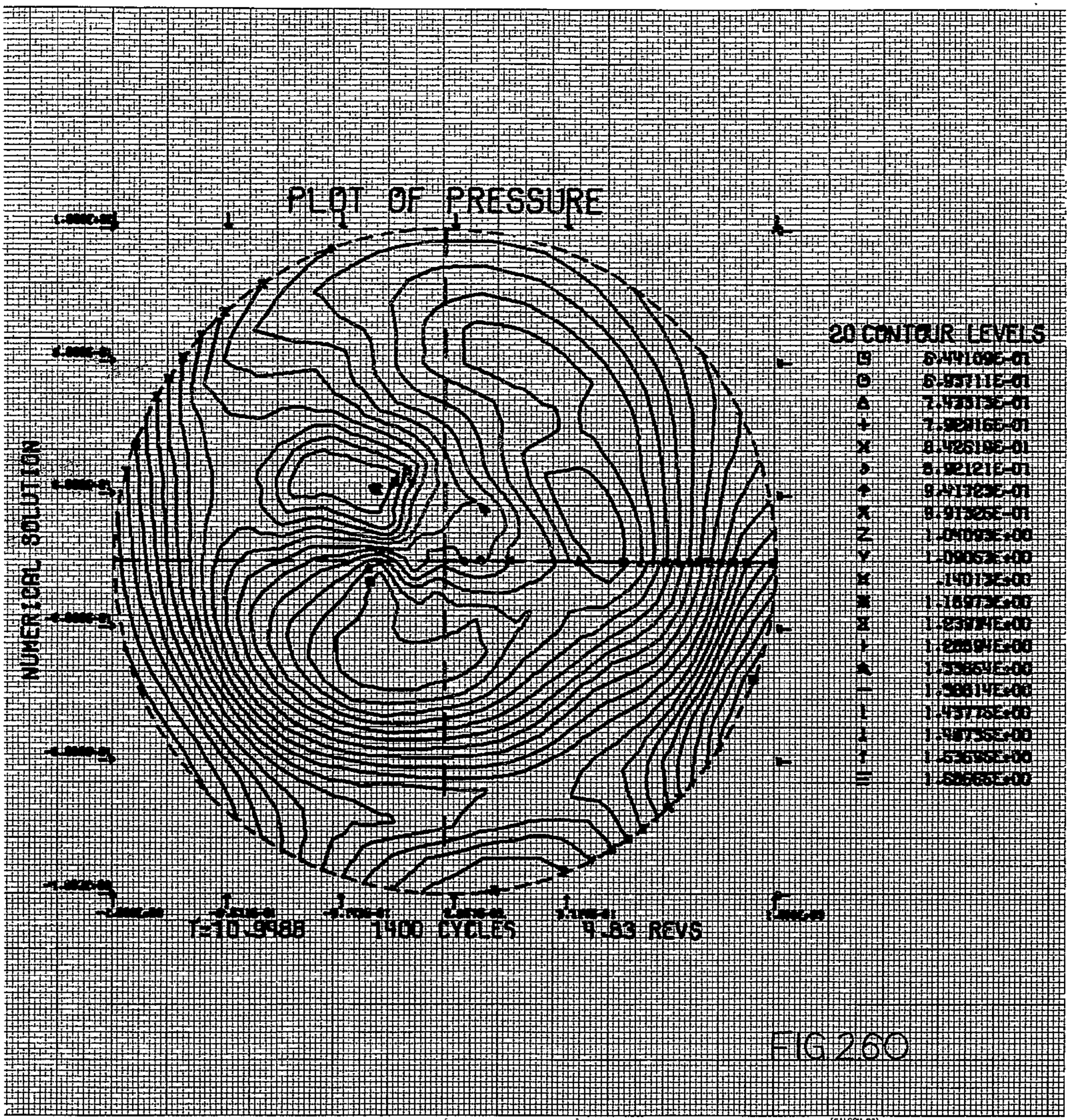
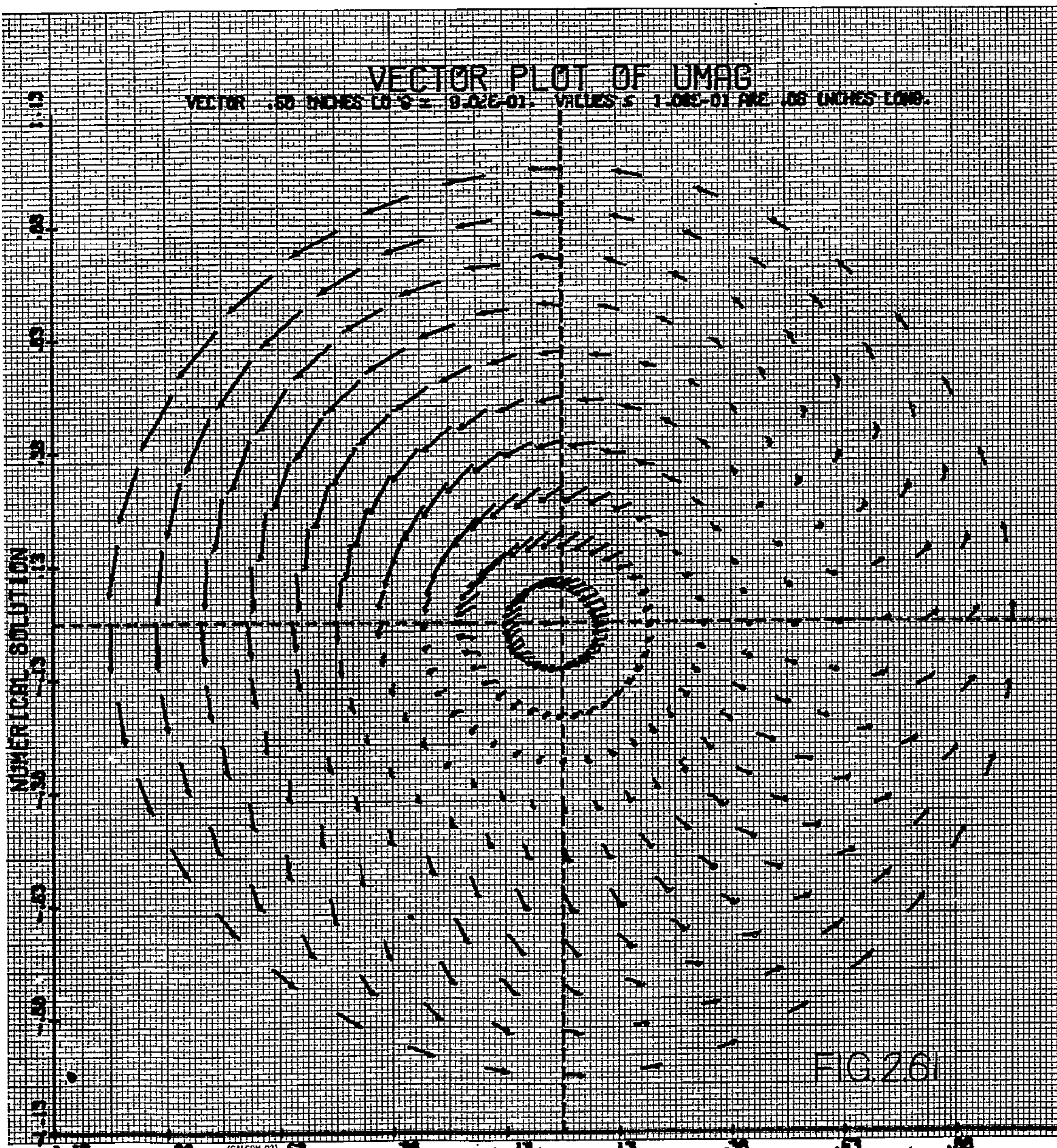

PLOT OF PRESSURE
20 CONTOUR LEVELS
NUMERICAL SOLUTION
1400 CYCLES
4.83 REVS

FIG. 2.60



T=10.9488 1400 CYCLES 4.83 REVS

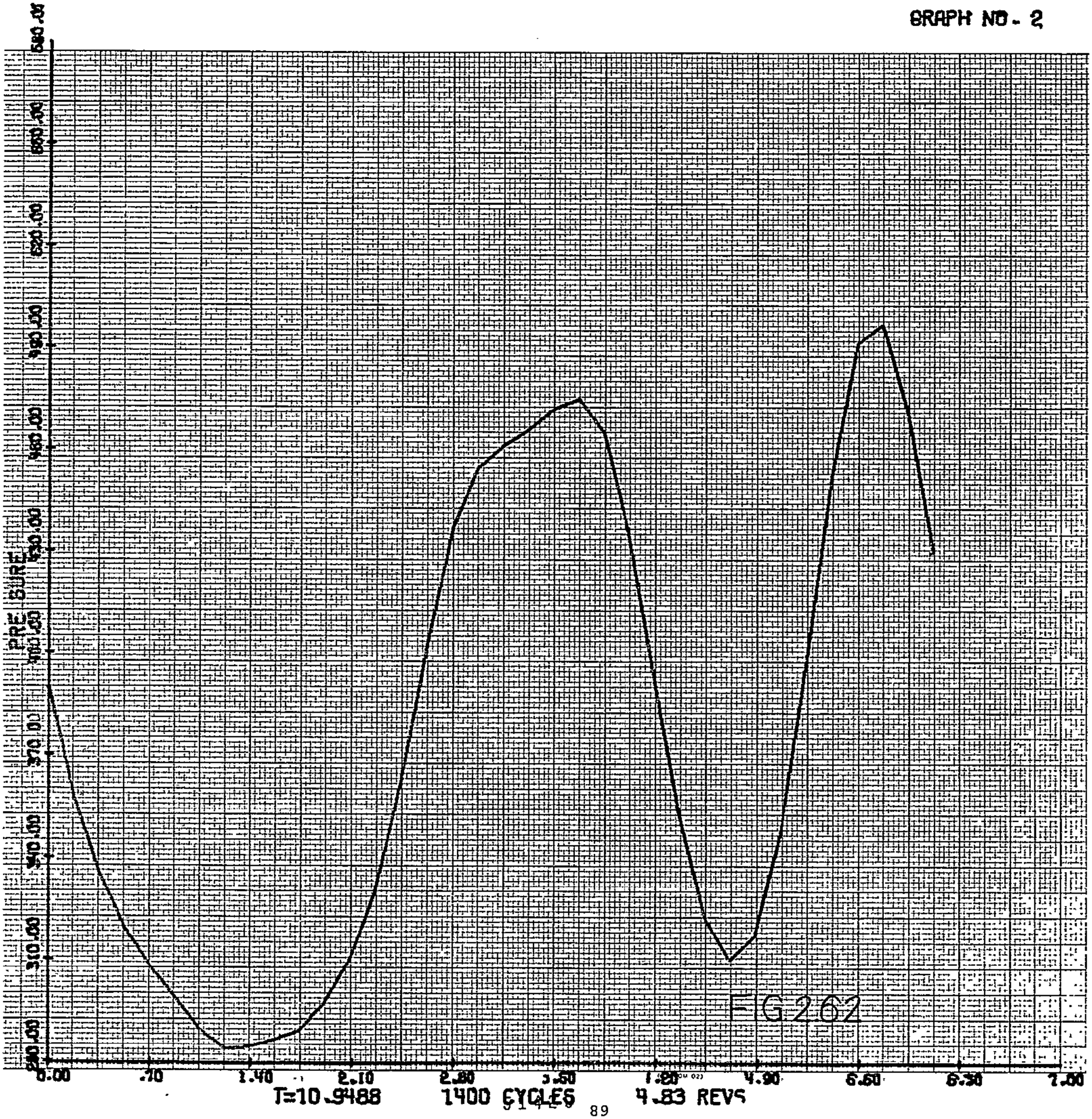

GRAPH NO. 2
PRE SURE
280.00
310.00
340.00
370.00
400.00
430.00
460.00
490.00
520.00
550.00
580.00
0.00
.70
1.40
2.10
2.80
3.50
4.20
4.90
5.60
6.30
7.00
T=10.9488 1400 CYCLES 4.83 REVS

FIG 2.62

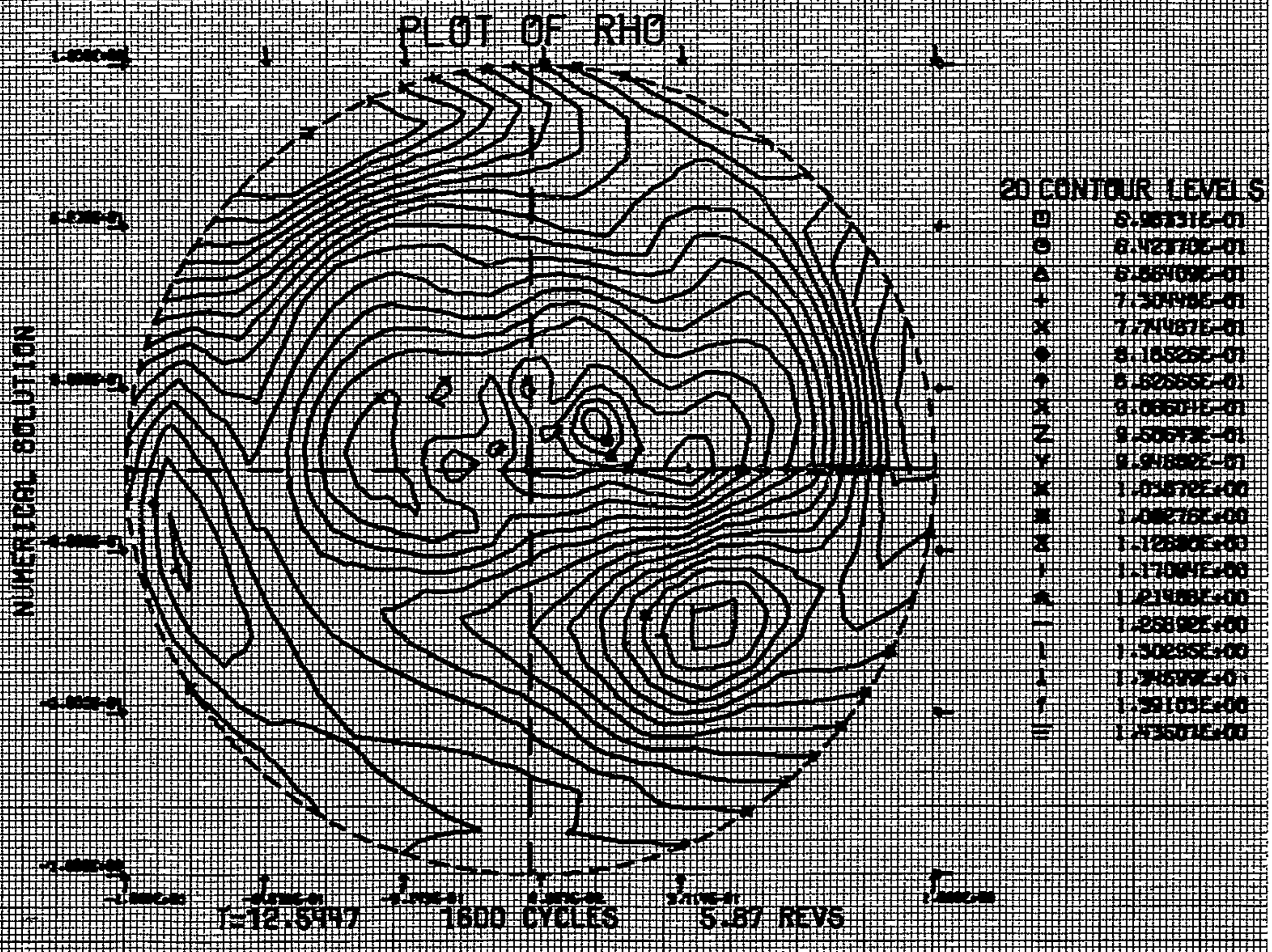

PLOT OF RHO
NUMERICAL SOLUTION
20 CONTOUR LEVELS
1600 CYCLES

FIG. 2.63

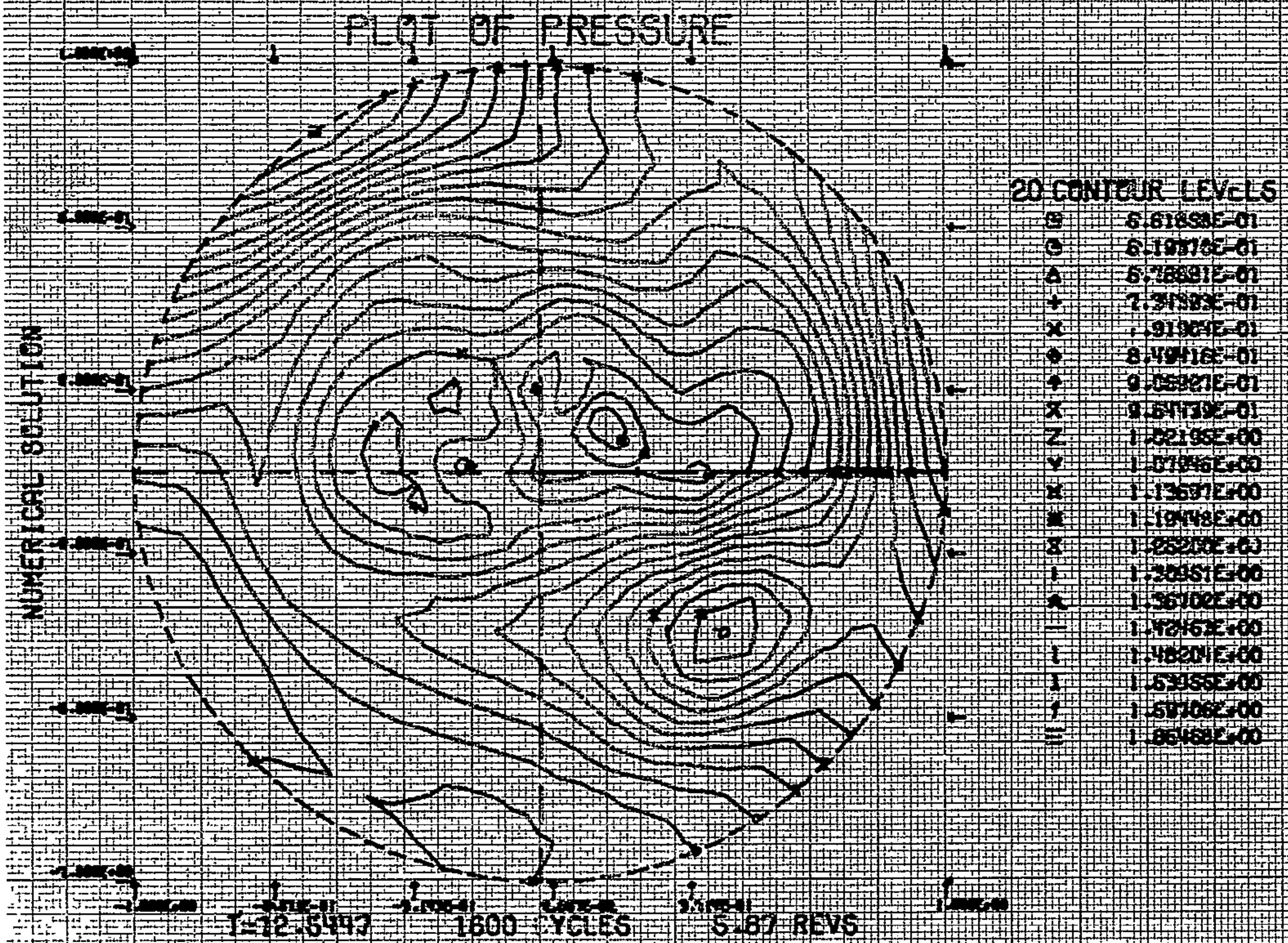

PLOT OF PRESSURE
NUMERICAL SOLUTION
20 CONTOUR LEVELS
T=12.5447 1600 CYCLES 5.87 REVS

FIG 2.64

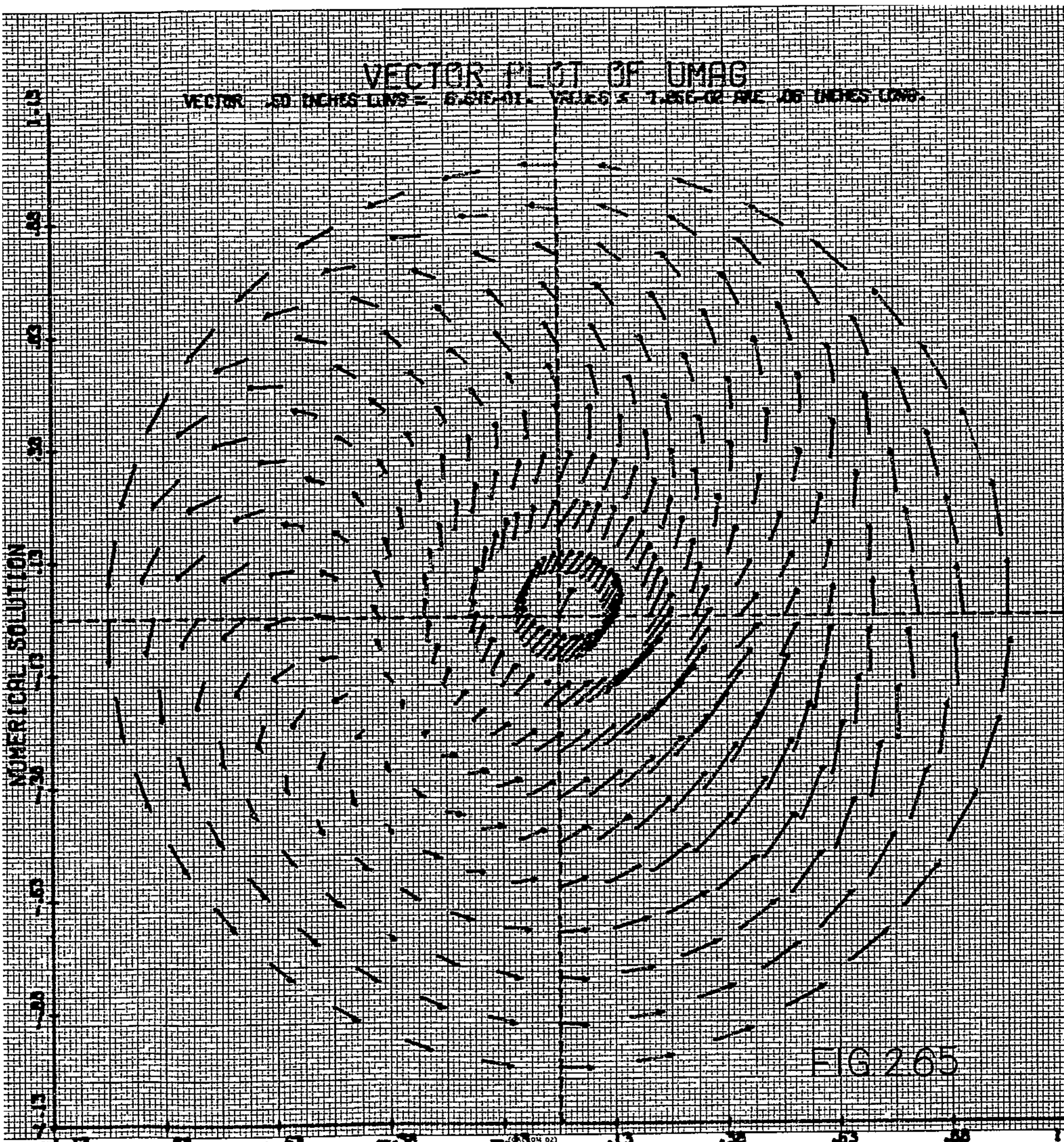


FIG 2.65

T=12.6447 1600 CYCLES 5.87 REVS

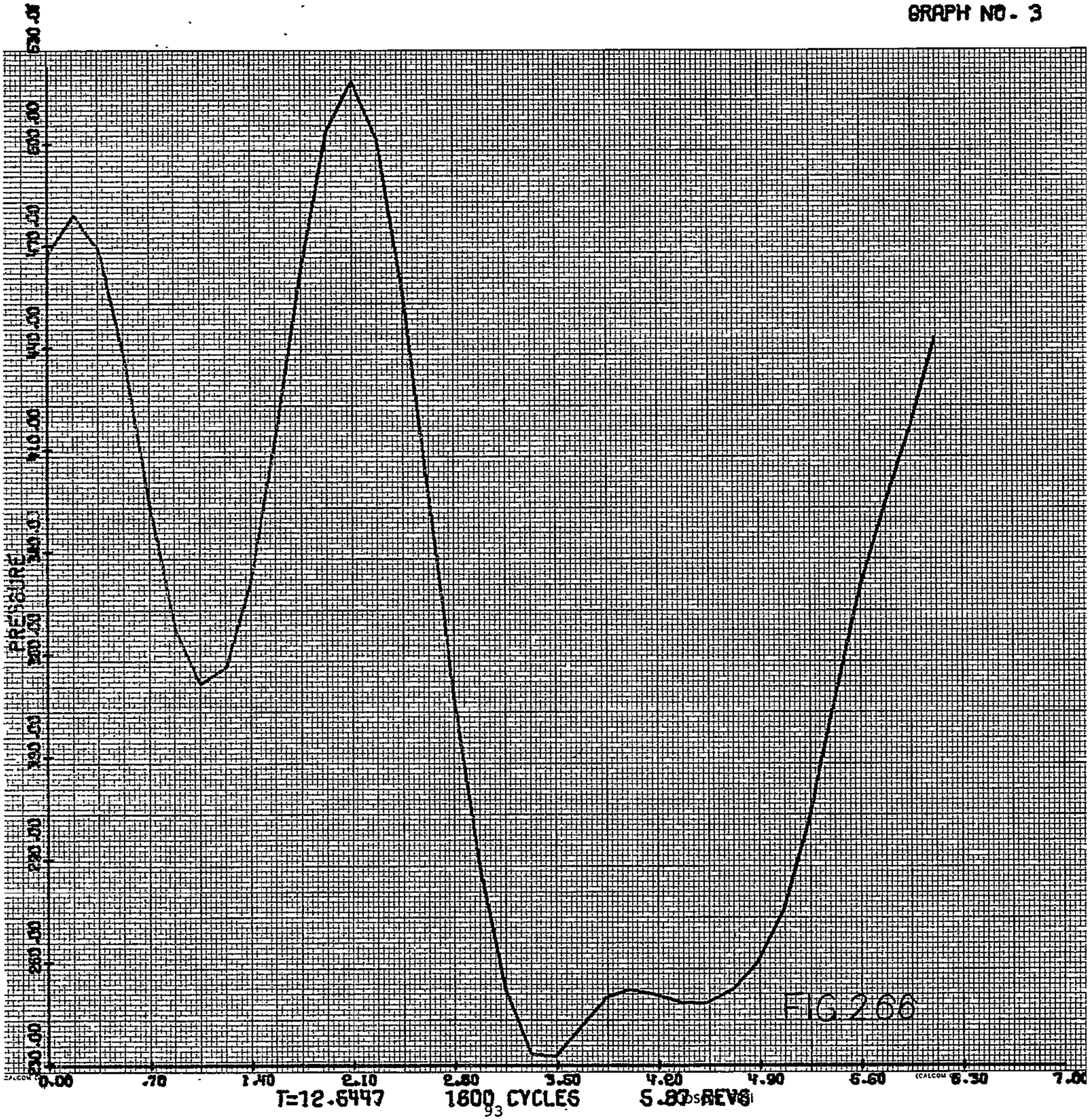


FIG.266

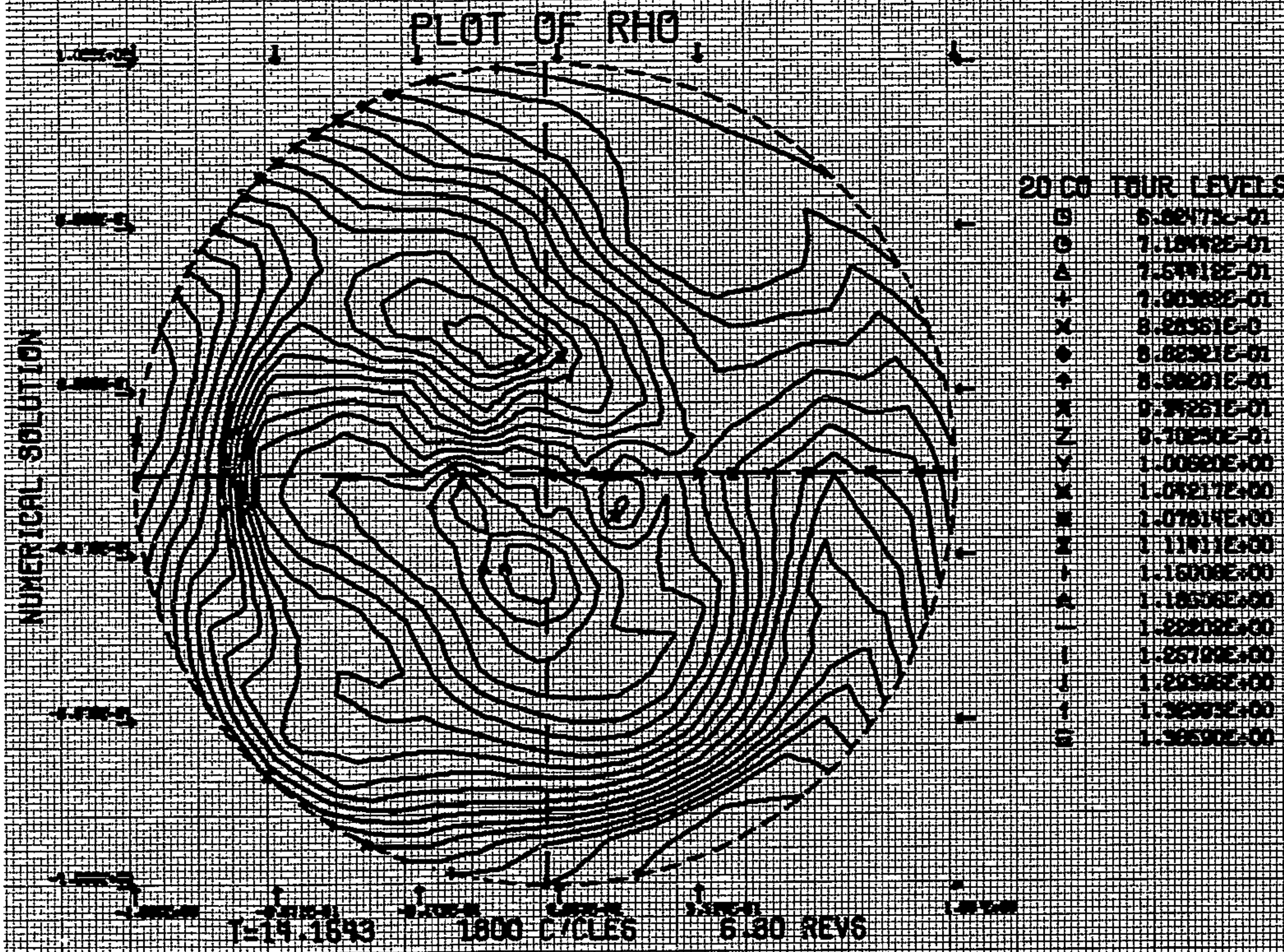

PLOT OF RHO
NUMERICAL SOLUTION
20 CONTOUR LEVELS
1800 CYCLES
6.80 REVS

FIG. 2.67

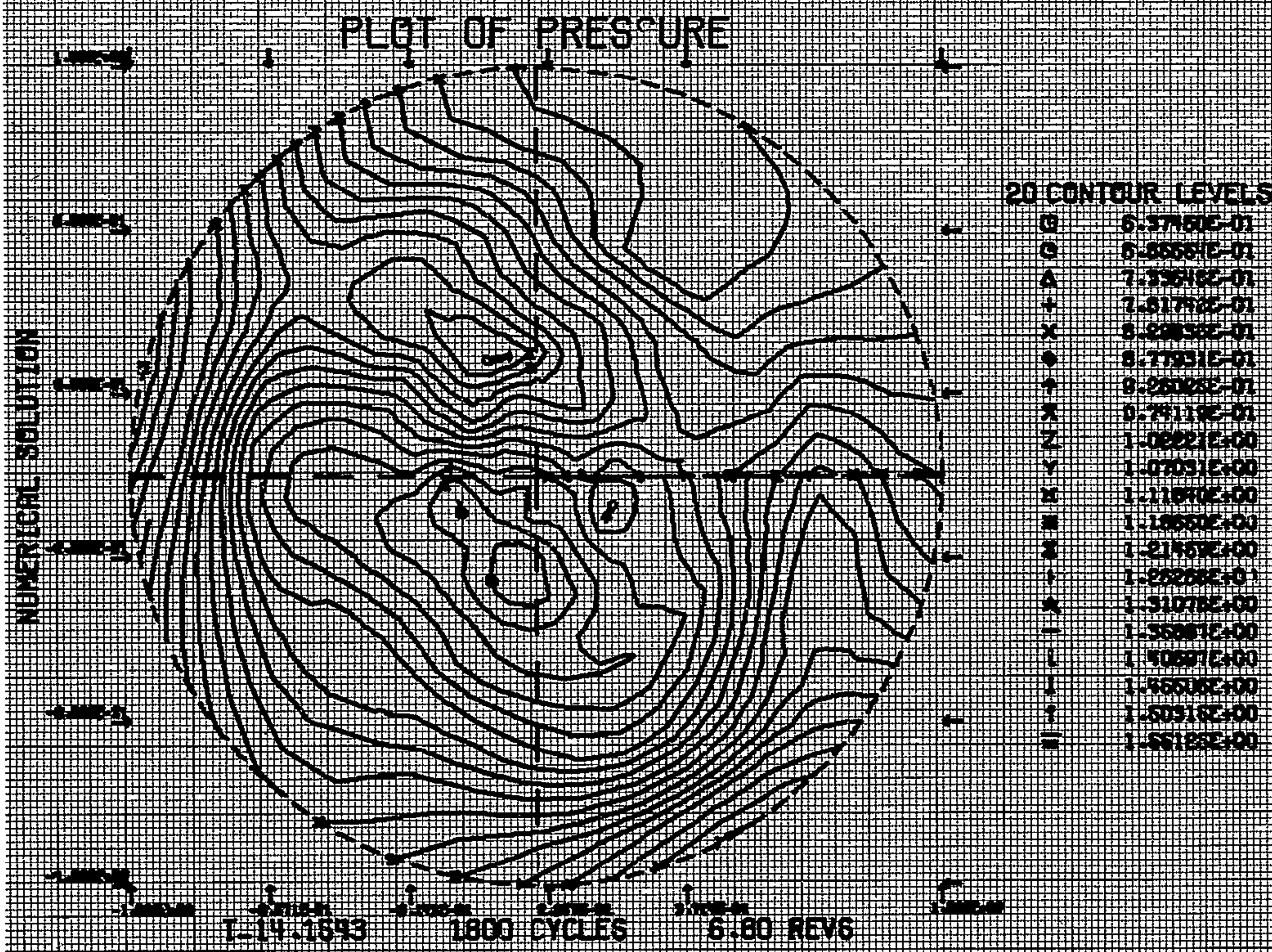

PLOT OF PRESSURE
NUMERICAL SOLUTION
20 CONTOUR LEVELS
1800 CYCLES
6.80 REVS

FIG 2.68

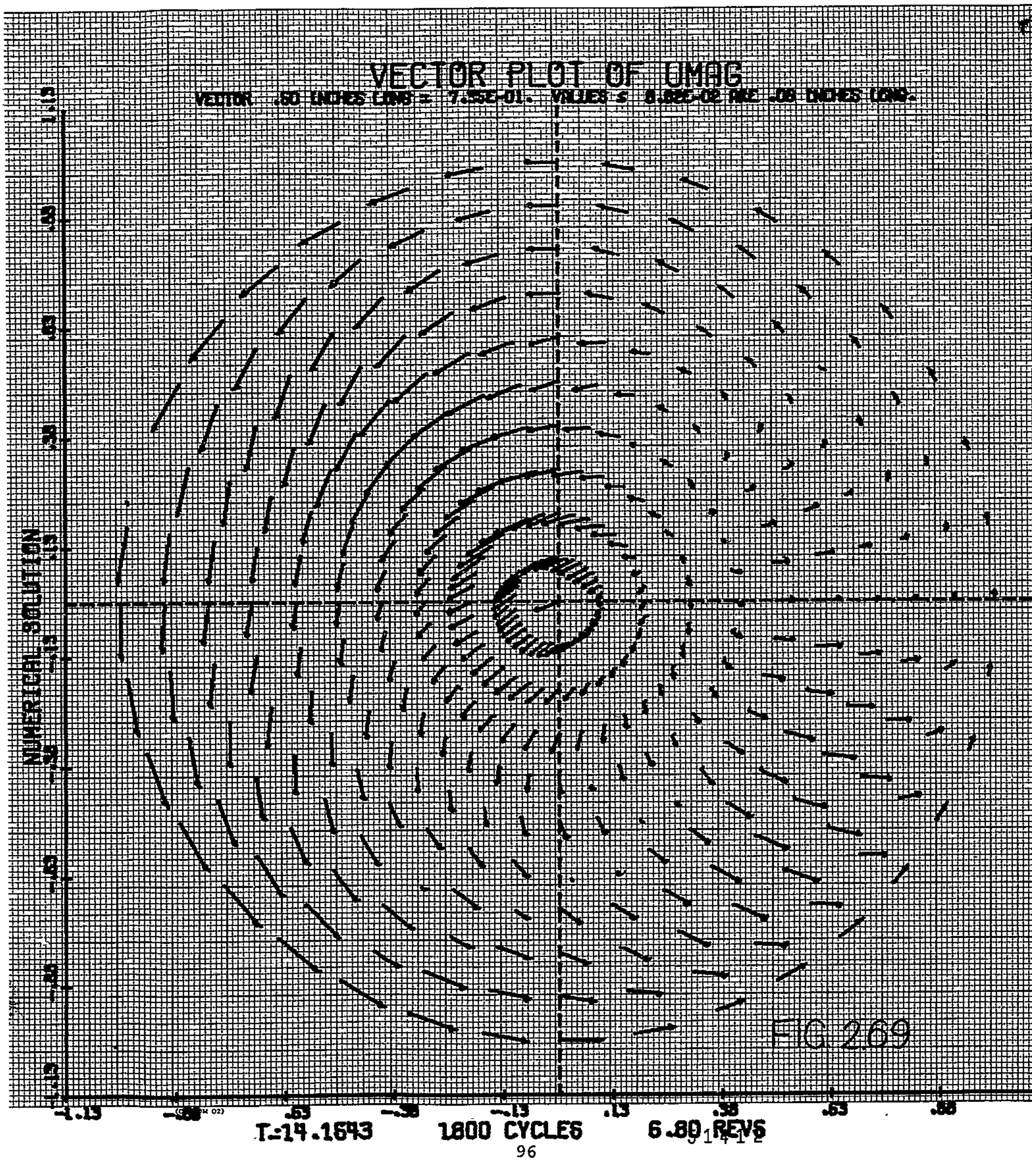

VECTOR PLOT OF UMAG
VECTOR .50 INCHES LONG = 7.35E-01. VALUES ≤ 8.82E-02 ARE .06 INCHES LONG.
NUMERICAL SOLUTION
T=14.1643 1800 CYCLES 6.80 REVS

FIG. 2.69

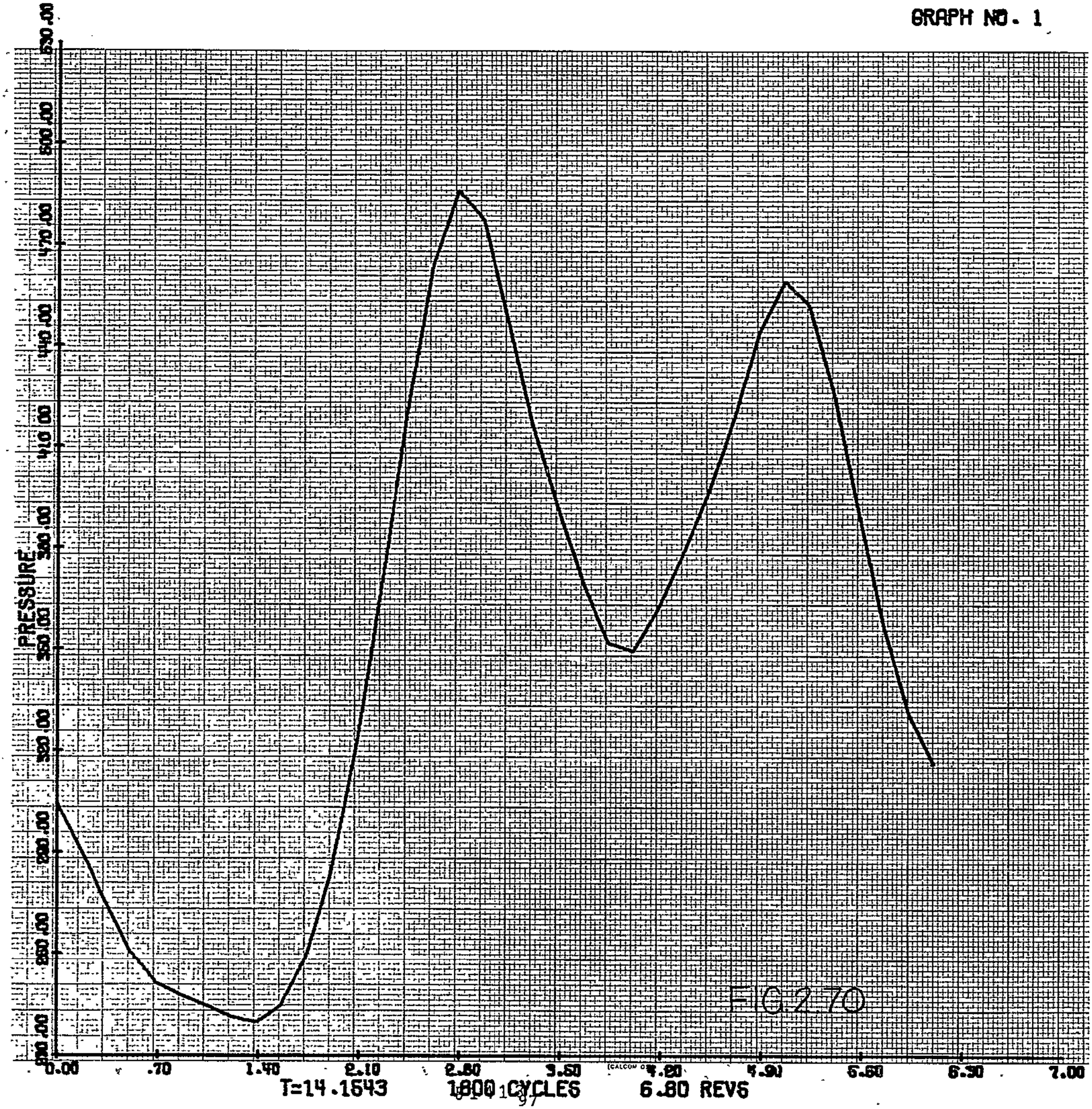
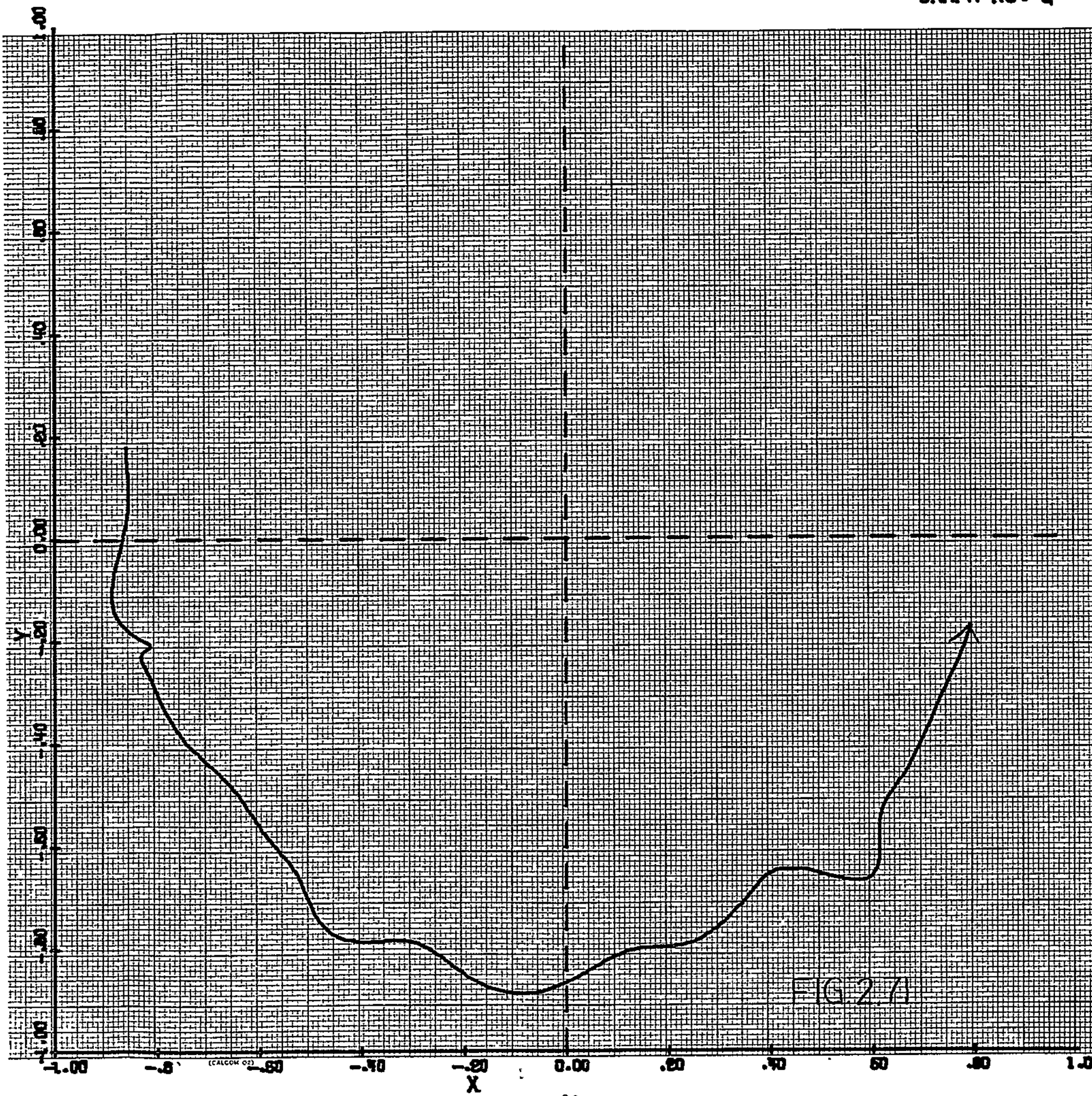

GRAPH NO. 1
PRESSURE
530.00
500.00
470.00
440.00
410.00
380.00
350.00
320.00
290.00
260.00
230.00
0.00
.70
1.40
2.10
2.80
3.50
4.20
4.90
5.60
6.30
7.00
T=14.1543
1800 CYCLES
6.80 REVS

FIG. 2.70

GRAPH NO. 2



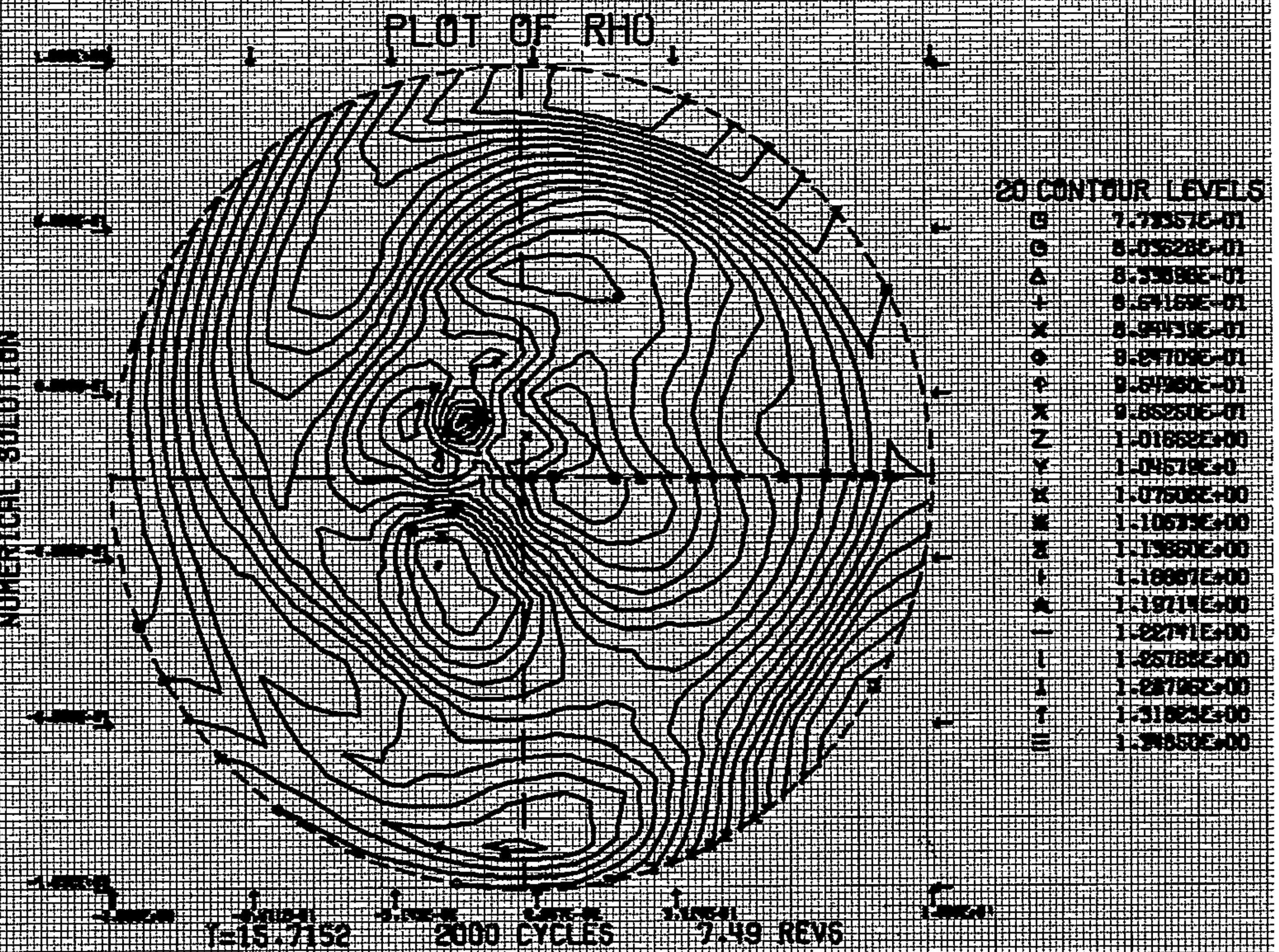

PLOT OF RHO
NUMERICAL SOLUTION
20 CONTOUR LEVELS
T=15.7152
2000 CYCLES
7.49 REVS

FIG. 2.72

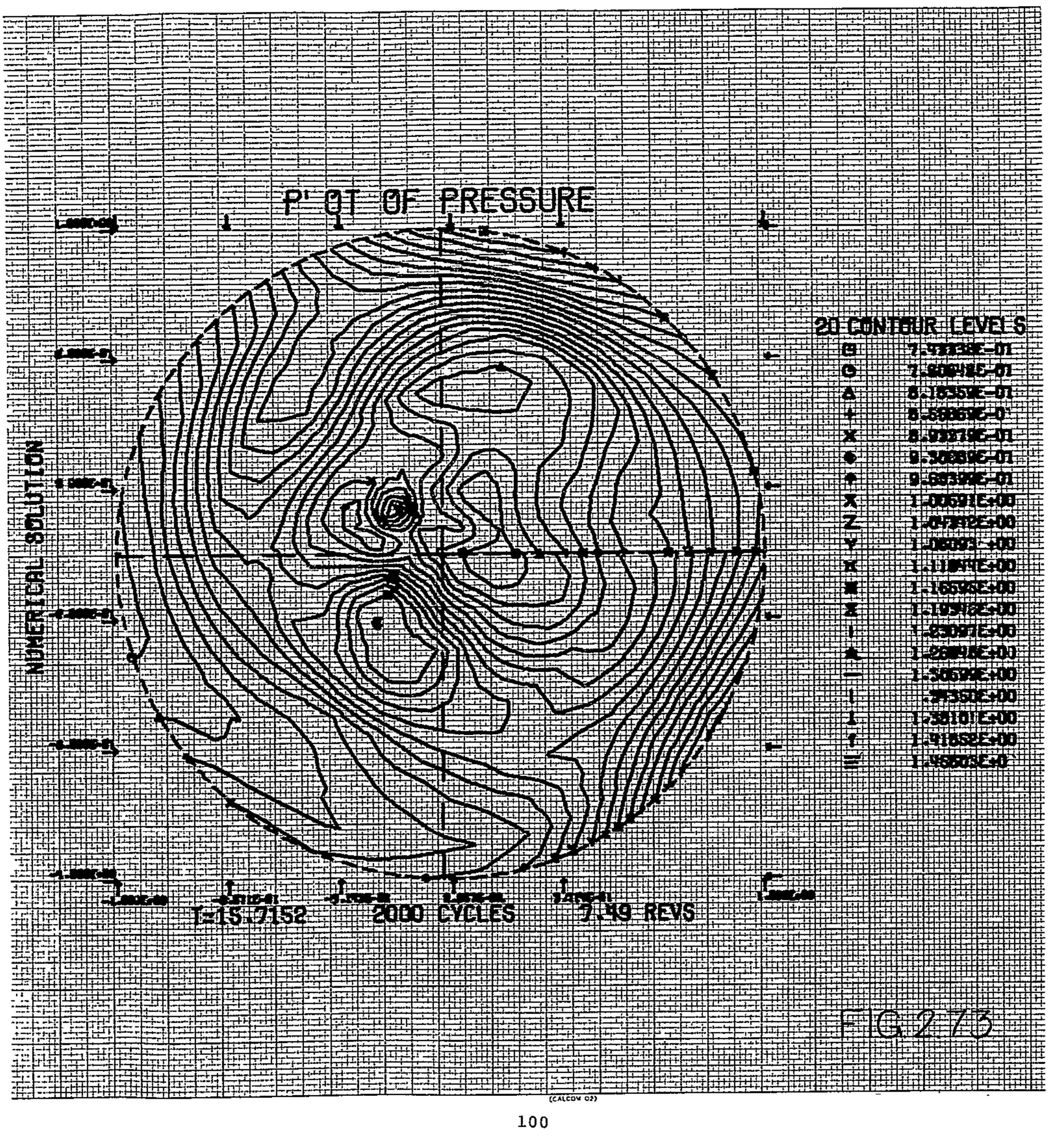

P' OT OF PRESSURE
NUMERICAL SOLUTION
20 CONTOUR LEVELS
T=15.7152
2000 CYCLES
7.49 REVS
(CALCOM 02)

FIG.273

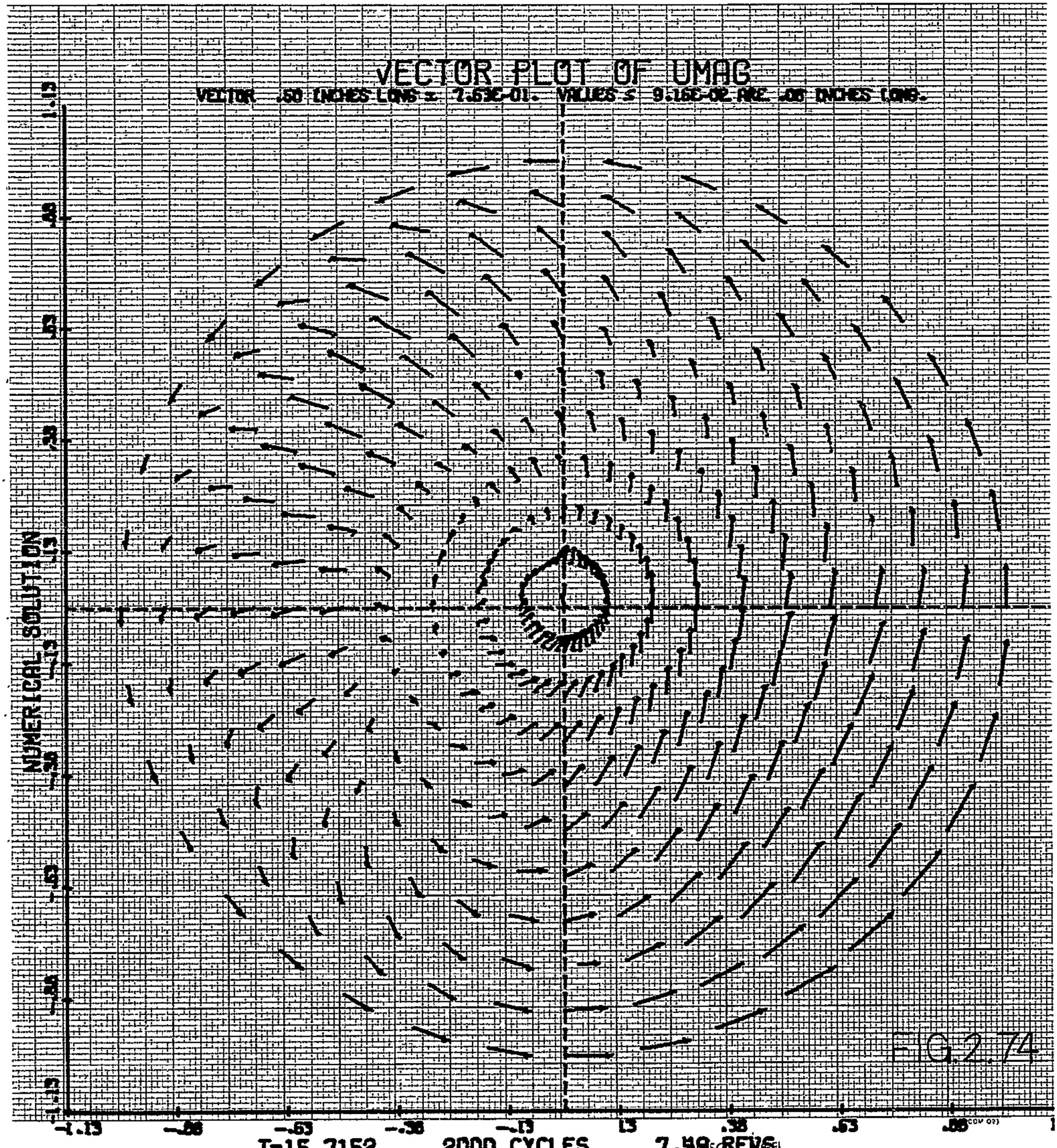
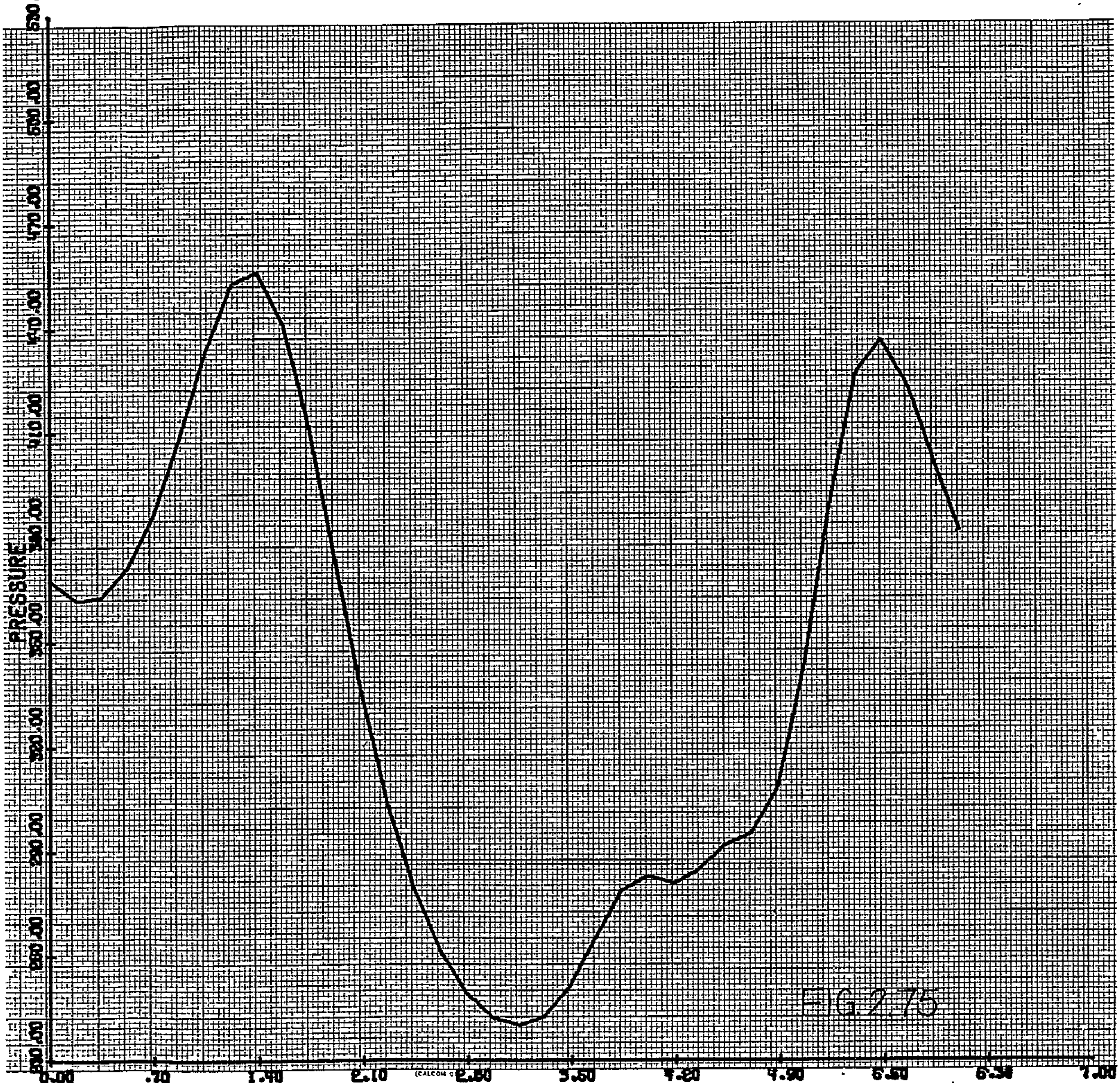

VECTOR PLOT OF UMAG
VECTOR .50 INCHES LONG = 7.63E-01. VALUES ≤ 9.16E-02 ARE .06 INCHES LONG.
NUMERICAL SOLUTION
1.13
.88
.63
.38
.13
-.13
-.38
-.63
-.88
-1.13
-1.13
-.88
-.63
-.38
-.13
.13
.38
.63
.88
T=15.7152
2000 CYCLES
7.49 REVS

FIG.2.74

GRAPH NO. 3



T=15.7152 2800 CYCLES 7.40 REVS

FIG.2.75

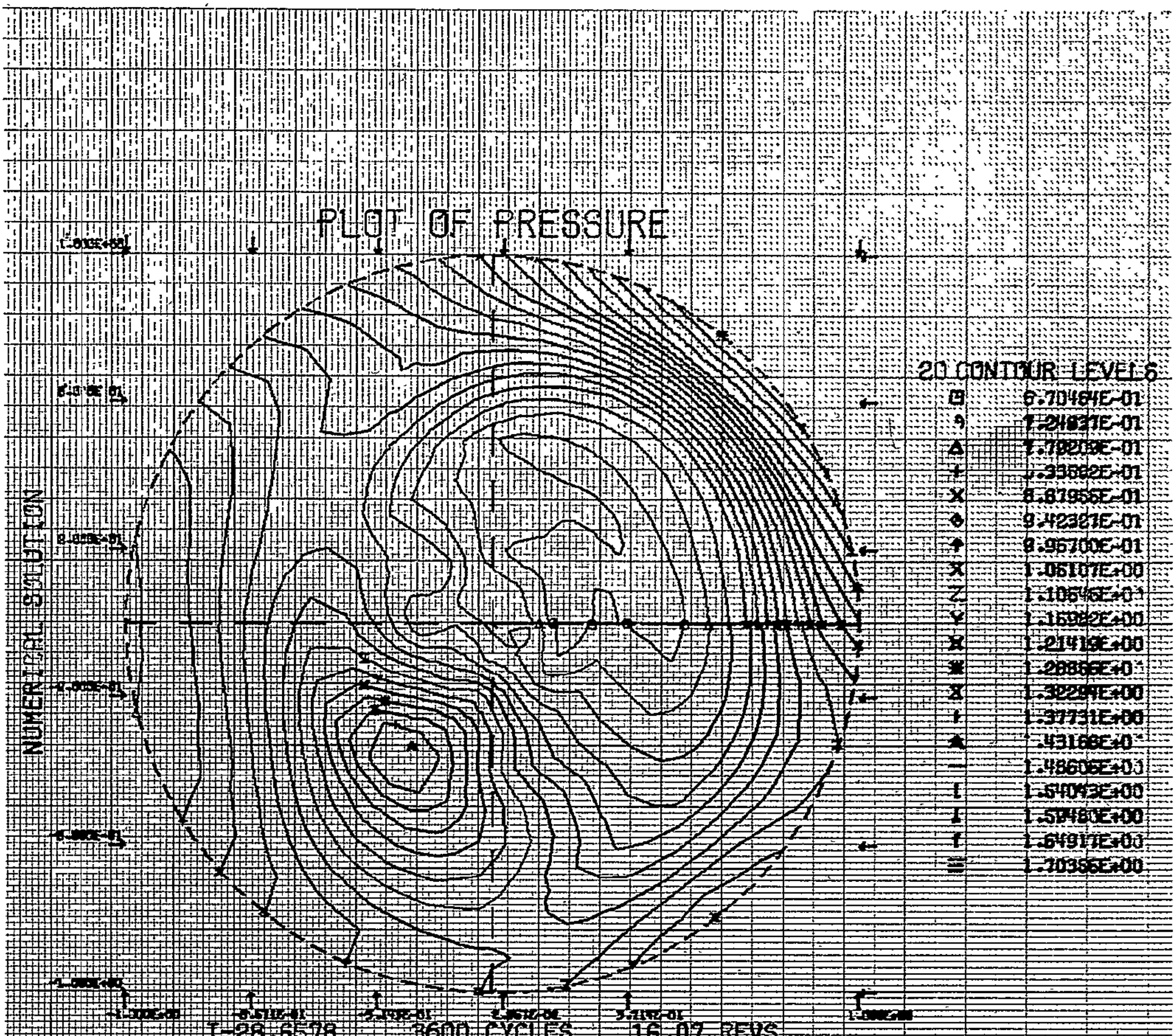

PLOT OF PRESSURE
NUMERICAL SOLUTION
20 CONTOUR LEVELS
6.70464E-01
7.24827E-01
7.79209E-01
9.87955E-01
9.42321E-01
1.05107E+00
1.21419E+00
1.32294E+00
1.37731E+00
1.48606E+00
1.59480E+00
1.64917E+00
1.70356E+00
T=28.6578 3600 CYCLES 16.07 REVS

FIG.2.76

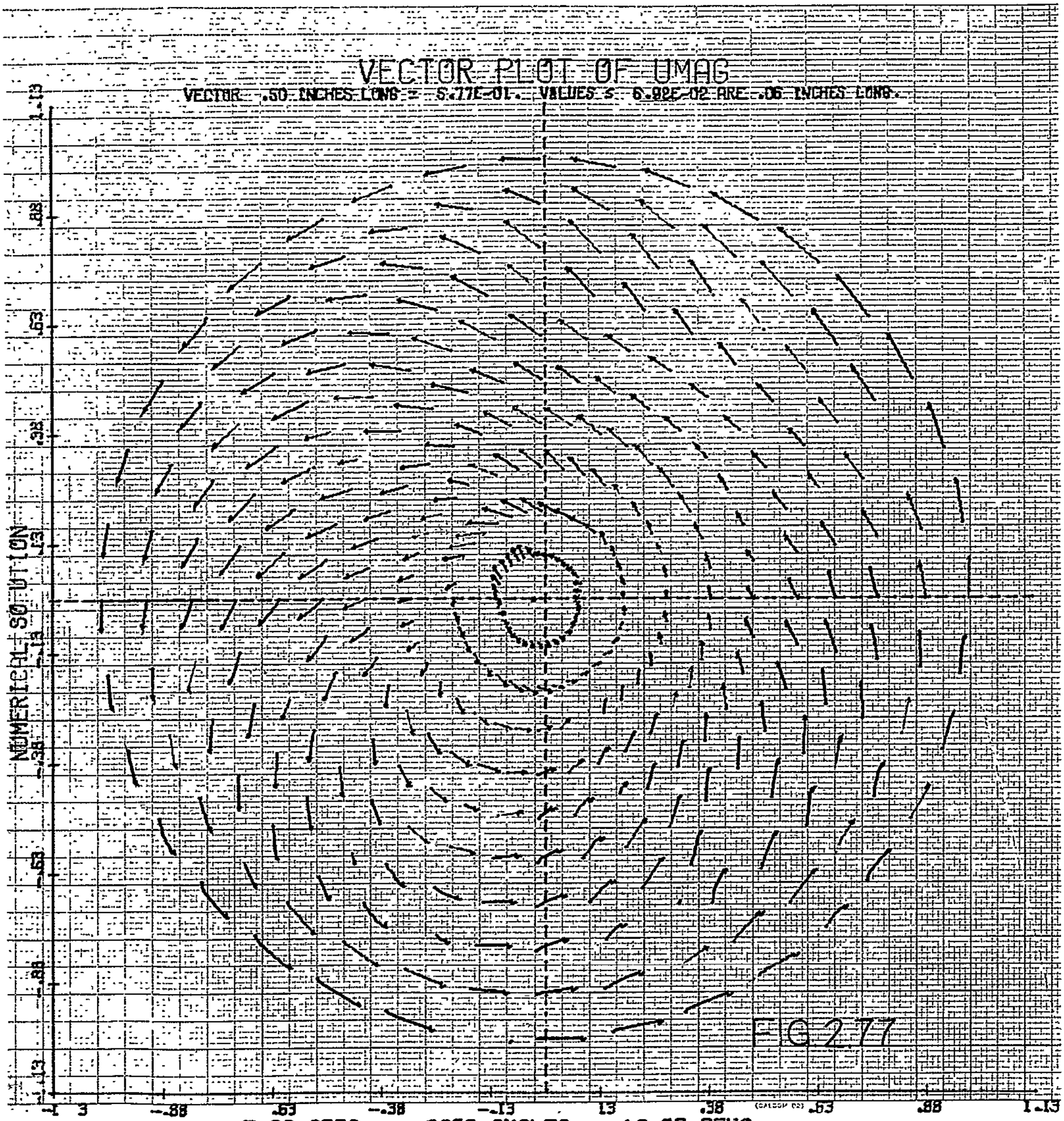

VECTOR PLOT OF UMAG
VECTOR .50 INCHES LONG = 5.77E-01. VALUES ≤ 6.92E-02 ARE .06 INCHES LONG.
NUMERICAL SOLUTION
T=23.6578 3600 CYCLES 16.07 REVS

FIG 2.77

GRAPH NO. 3

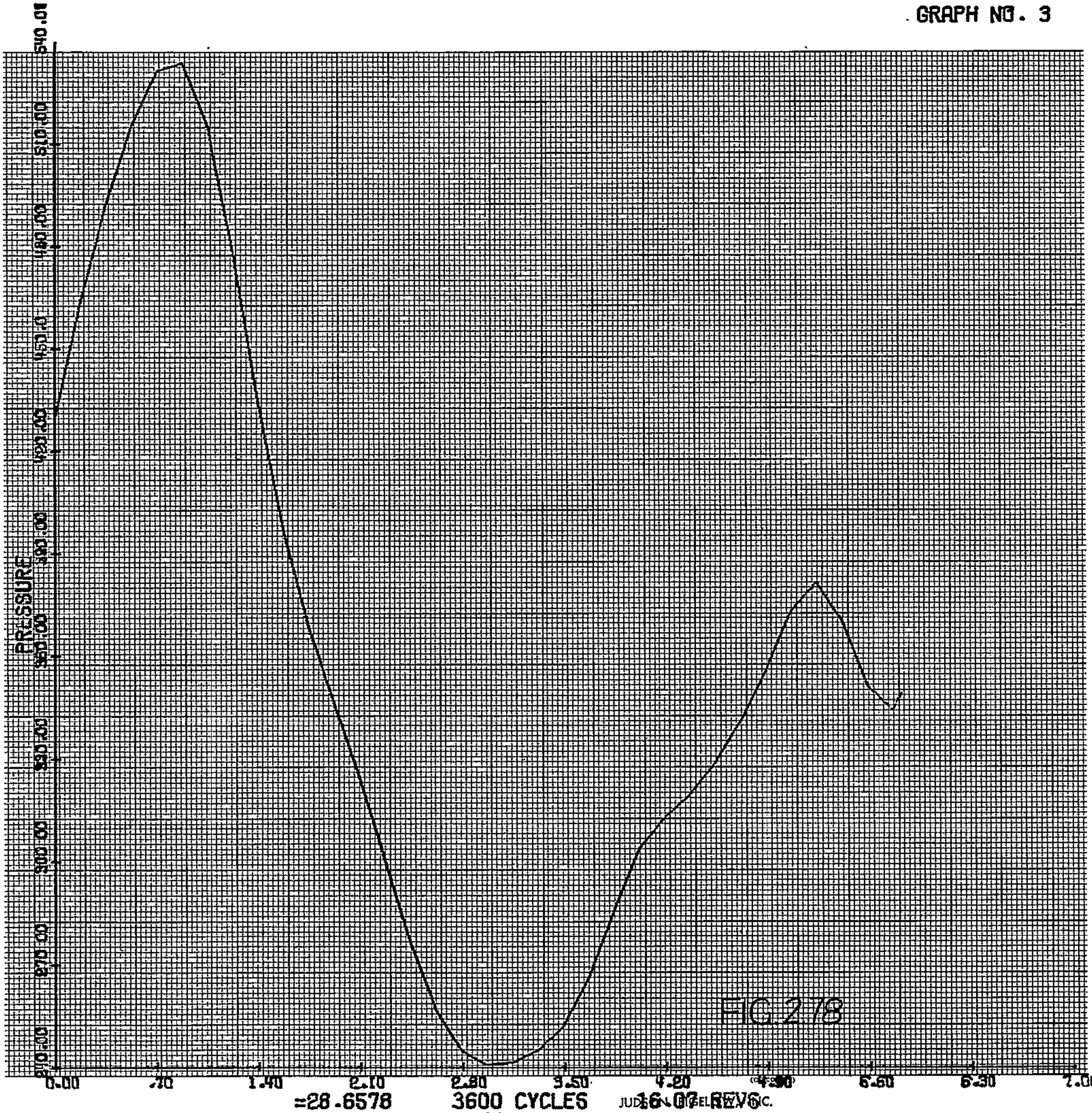


FIG.278

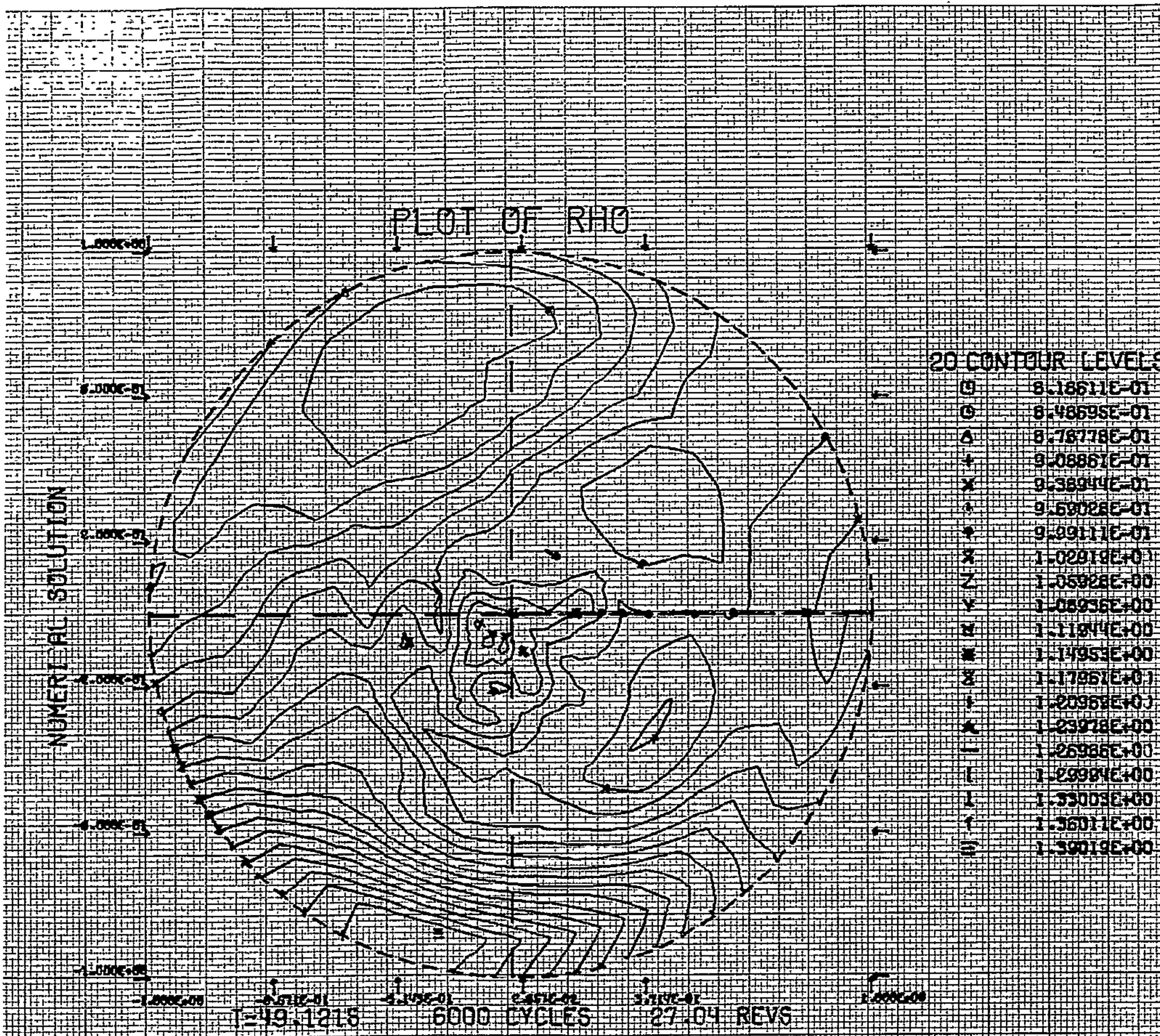
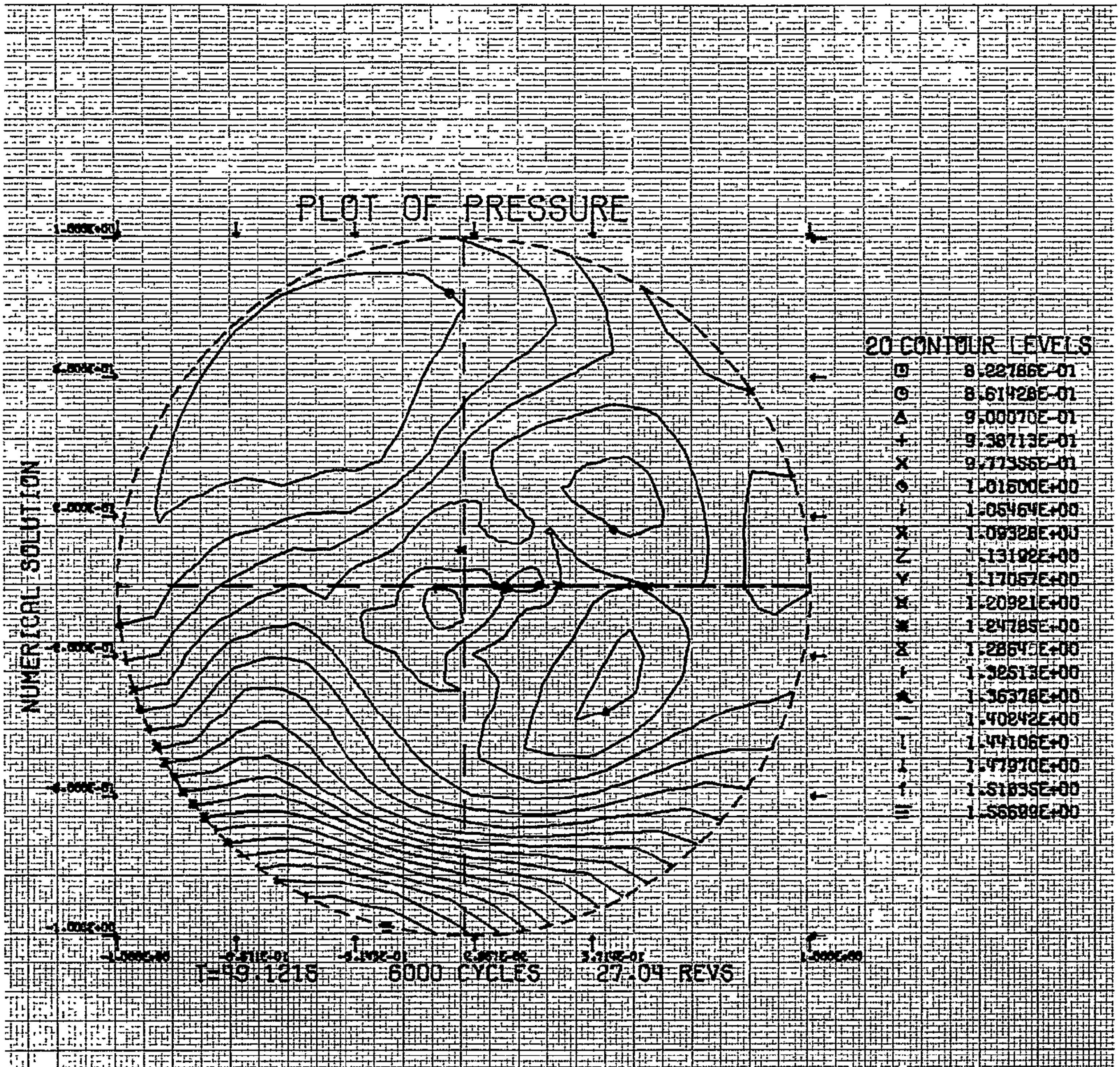

PLOT OF RHO
NUMERICAL SOLUTION
20 CONTOUR LEVELS
8.18611E-01
8.48695E-01
8.78778E-01
9.08861E-01
9.38944E-01
9.69028E-01
9.99111E-01
1.02919E+00
1.05928E+00
1.08936E+00
1.11944E+00
1.14953E+00
1.17961E+00
1.20969E+00
1.23978E+00
1.26986E+00
1.29994E+00
1.33003E+00
1.36011E+00
1.39019E+00
T=49.1215
6000 CYCLES
27.04 REVS

FIG 279



PLOT OF PRESSURE
NUMERICAL SOLUTION
20 CONTOUR LEVELS
8.22786E-01
8.61428E-01
9.00070E-01
9.38713E-01
9.77356E-01
1.01600E+00
1.05464E+00
1.09328E+00
1.13192E+00
1.17057E+00
1.20921E+00
1.24785E+00
1.28649E+00
1.32513E+00
1.36378E+00
1.40242E+00
1.44106E+00
1.47970E+00
1.51835E+00
1.55699E+00
T=99.1215 6000 CYCLES 27.04 REVS

FIG. 280

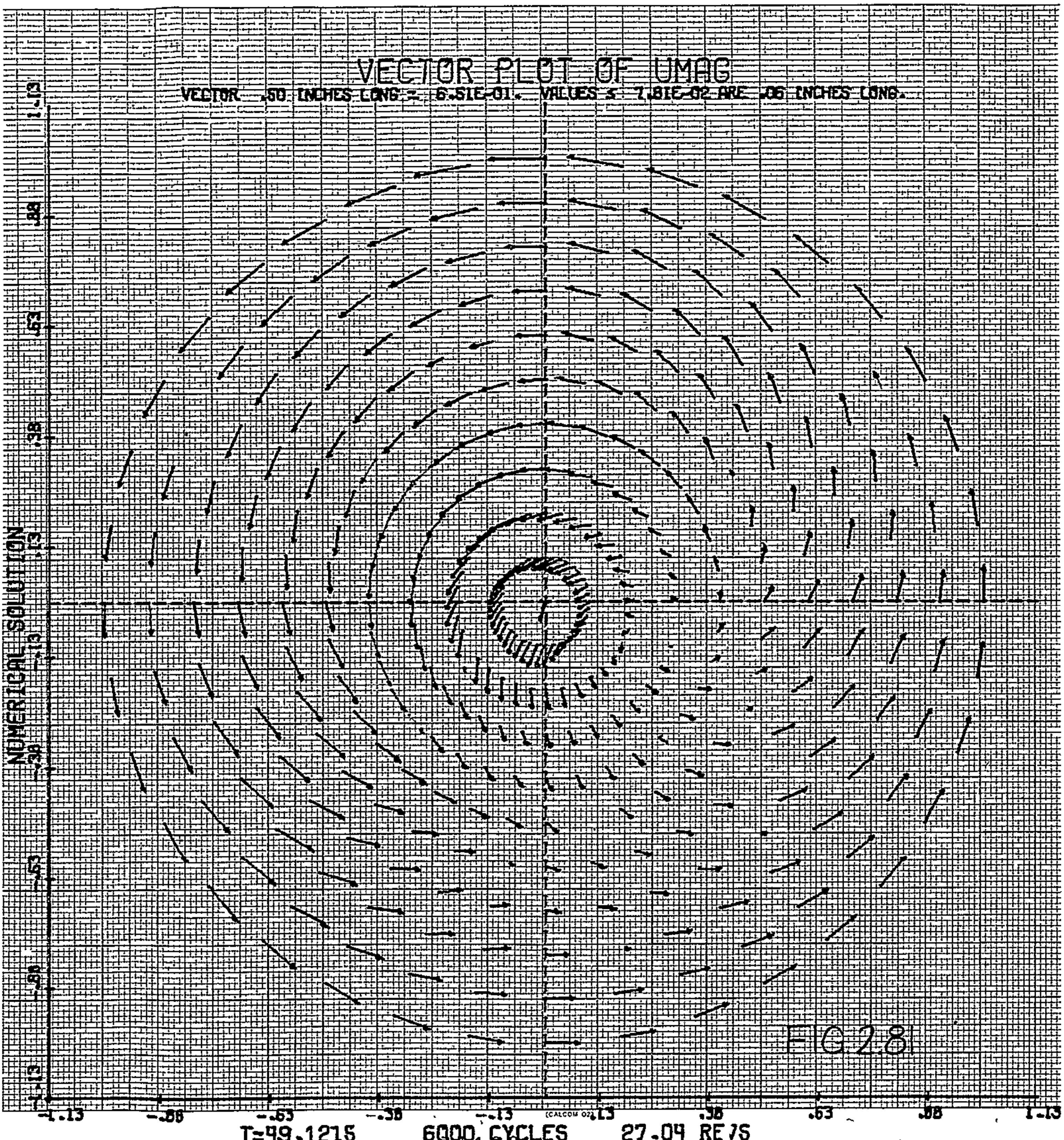

VECTOR PLOT OF UMAG
VECTOR .50 INCHES LONG = 6.51E-01. VALUES ≤ 7.81E-02 ARE .06 INCHES LONG.
NUMERICAL SOLUTION
1.13
.88
.63
.38
.13
-.13
-.38
-.63
-.88
-1.13
-1.13
-.88
-.63
-.38
-.13
.13
.38
.63
.88
1.13
T=49.1215
6000 CYCLES
27.04 REVS

FIG.2.81

GRAPH NO. 3

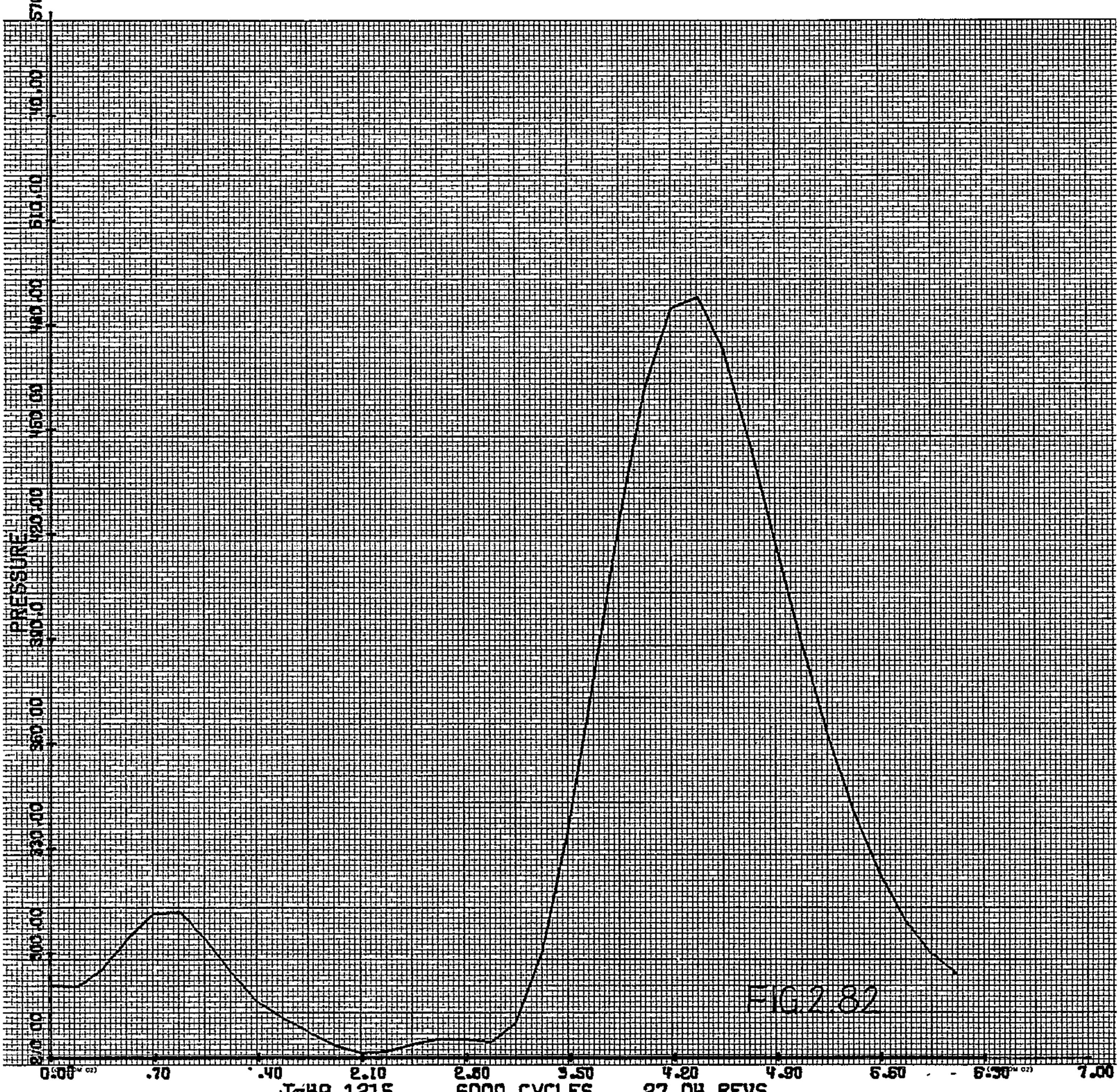


FIG.2.82

IV. ANNULAR MOTOR EXPERIMENTS

In Reference (1), the partial differential equations and the associated difference equations for the annular motor are presented. The geometry of the calculation is θ-z-t. This section describes time dependent calculations of the complete fluid dynamic annular model coupled to the combustion model (referred to as the modified Godsave analysis in Reference (1)). The first calculation has as initial conditions a solution to a wave equation analagous to Equation (2.15) modified for the annular model. This solution is superimposed on the steady state solution given in Reference (1). This requirement is needed since the classical solution is derived for the "organ" pipe geometry where there is no condition of net axial flow to be met. There have been various analyses for a classical solution with Mach number effect included. However, the condition at the injector assumes a discontinuous combustion model (concentrated combustion) to satisfy the boundary condition of zero velocity at the injector face and the linearizing field condition of constant Mach number.

The starting point for the linearized solution is Equation (3.7). The solution is assumed to be of the form given by System (3.8) with $\bar{u}\neq 0$ but $\bar{v}=0$. We start by noting

$$p = \rho^{\gamma} \tag{4.1}$$

for smooth flows.

Hence for a particle in such flows total differentiation yields

$$\frac{1}{p}\frac{Dp}{Dt} - \frac{\gamma}{\rho}\frac{D\rho}{Dt} = 0 \tag{4.2}$$

where in the annular coordinate system

$$\frac{D}{Dt} \equiv \frac{\partial}{\partial t} + u\frac{\partial}{\partial z} + \frac{v}{r}\frac{\partial}{\partial \theta}$$

Using the definition of sound speed under the assumption that the equation of state is the ideal gas law, Equation (4.2) is written as

$$\frac{D\rho}{Dt} = c^{-2}\frac{Dp}{Dt} \tag{4.3}$$

and is used in Equation (3.7) to give

$$\frac{Dp}{Dt} = c^{-2}\rho \quad \mathrm{div}\ \underset{\sim}{u} \tag{4.4}$$

The divergence operator is defined as

$$\mathrm{div} = \frac{\partial}{\partial z} + \overline{\frac{1}{r}\frac{\partial}{\partial \theta}}$$

and $\underset{\sim}{u}$ is given by

$$\underset{\sim}{u} = \begin{pmatrix} \overline{u} + u' \\ v' \end{pmatrix} \tag{4.5a}$$

In this analysis $\overline{u} = \overline{u}(z)$.

Then, if we assume that $\overline{p}=\overline{p}(z)$ and $\overline{\rho}=\overline{\rho}(z)$ only; also $p'=p'(z,$ and $\rho'=\rho'(z,\theta)$. Hence

$$p = \overline{p} + p'$$
$$\rho = \overline{\rho} + \rho' \tag{4.5b}$$

is substituted into Equation (4.4) also using the definition (4.5) one gets to first order in the perturbed properties

$$p'_t + \overline{u}\, p'_z = -\overline{c}^2\overline{\rho}\ \mathrm{div}\ \underset{\sim}{u} \tag{4.6}$$

Equations (3.5) and (3.6) can be written as

$$\underset{\sim}{u}_t + \overline{u}\ \ \underset{\sim}{u}_z = -\ \overline{\rho}^{-1}\ \mathrm{grad}\ p' \tag{4.7}$$

using relations (4.5b).

Again we let the potential ψ define the velocity field

$$\underset{\sim}{u} = \mathrm{grad}\ \psi \tag{4.8}$$

Hence, Equation (4.6) becomes

$$p'_t + \overline{u}\, p'_z = -\overline{c}^2\overline{\rho}\nabla^2\psi \tag{4.9}$$

Equation (4.7) is written in terms of the potential ψ:

$$\mathrm{grad}\ \psi_t + \overline{u}\ \mathrm{grad}\ \psi_z = -\overline{\rho}^{-1}\ \ \mathrm{grad}\ p' \tag{4.10}$$

or

$$\psi_{tz} + \overline{u}\ \psi_{zz} = -\overline{\rho}^{-1}\ \ p'_z \tag{4.11}$$

Equation (4.10), being an exact differential quantity, may be written

$$\psi_t + \overline{u}\ \ \psi_z = -\ \overline{\rho}^{-1}\ p' \tag{4.12}$$

If Equation (4.12) is differentiated with respect to the time t, one obtains

$$\psi_{tt} + \bar{u}\,\psi_{zz} = -\bar{\rho}^{-1}\,p'_t \tag{4.13}$$

Combine Equations (4.9), (4.11) and (4.13) to give

$$\psi_{tt} + \bar{u}\,\psi_{tz} + \bar{u}\,\psi_{zt} + \bar{u}^2\,\psi_{zz} = \bar{c}^2\nabla^2\psi = \bar{c}^2(\psi_{zz}+\frac{1}{r^2}\psi_{\theta\theta})$$

or

$$(\bar{c}^2-\bar{u}^2)\,\psi_{zz} + \bar{c}^2\,\psi_{\theta\theta} - 2\bar{u}\,\psi_{zt} - \psi_{tt} = 0 \tag{4.14}$$

which is the defining equation in the space $(\theta\text{-}z\text{-}t)$. We let $r=1$ in the present formulation.

One possible solution to (4.14) of the form (4.5b) is

$$\begin{aligned} p &= \bar{p}\,\{1-\gamma\varepsilon\cos(k\bar{c}t+\theta)\cos(\beta z)\} \\ u &= \bar{u} - \bar{c}\,\frac{\varepsilon}{k}\beta\sin(k\bar{c}t+\theta)\sin(\beta z) \\ v &= \bar{c}\,\frac{\varepsilon}{k}\cos(k\bar{c}t+\theta)\cos(\beta z) \end{aligned} \tag{4.15}$$

where the constants β, k and ε are defined by

$$\begin{aligned} \beta &= 2\pi/L \\ k &= -\bar{u}/\bar{c}\;\beta - \sqrt{\beta^2-1} \\ \varepsilon &= \frac{p'_{max}-\bar{p}}{\gamma\bar{p}} \end{aligned} \tag{4.16}$$

conditions (4.16) are obtained by requiring $u'(z) = 0$ for $z=0$ and $z=L$, L being the chamber length.

To define the initial conditions for the numerical calculation we choose a value of $0 \leq \varepsilon \leq 1$ which defines the value of the maximum perturbed pressure p'_{max}. Then

$$
\begin{aligned}
p(z,\theta,0) &= p(z)\ \{1+p'(z,\theta)\} \\
\rho(z,\theta,0) &= p(z,\theta,0)^{1/\gamma} \\
u(z,\theta,0) &= u(z)\ \{1+u'(z,\theta)\} \\
v(z,\theta,0) &= v'(z,\theta)
\end{aligned}
\tag{4.17}
$$

defines the initial conditions (at t=0) including the deviation from the one-dimensional steady state solution $p(z)$ and $u(z)$.

Figures (3.1) - (3.6) are the result of a calculation using the initial conditions derived from the wave equation. For this calculation no energy was introduced into the system. After 800 cycles or 2.2 milliseconds the maximum pressure had dropped by two orders of magnitude as the gas effectively evacuated from the chamber. When this same calculation was run with energy, a steady state was reached by 500 cycles or 2.7 milliseconds. These results are shown in Figures (3.7) - (3.14). Figure (3.14) shows essentially the same pressure profile at 6.5 milliseconds as observed at 2.7 milliseconds.

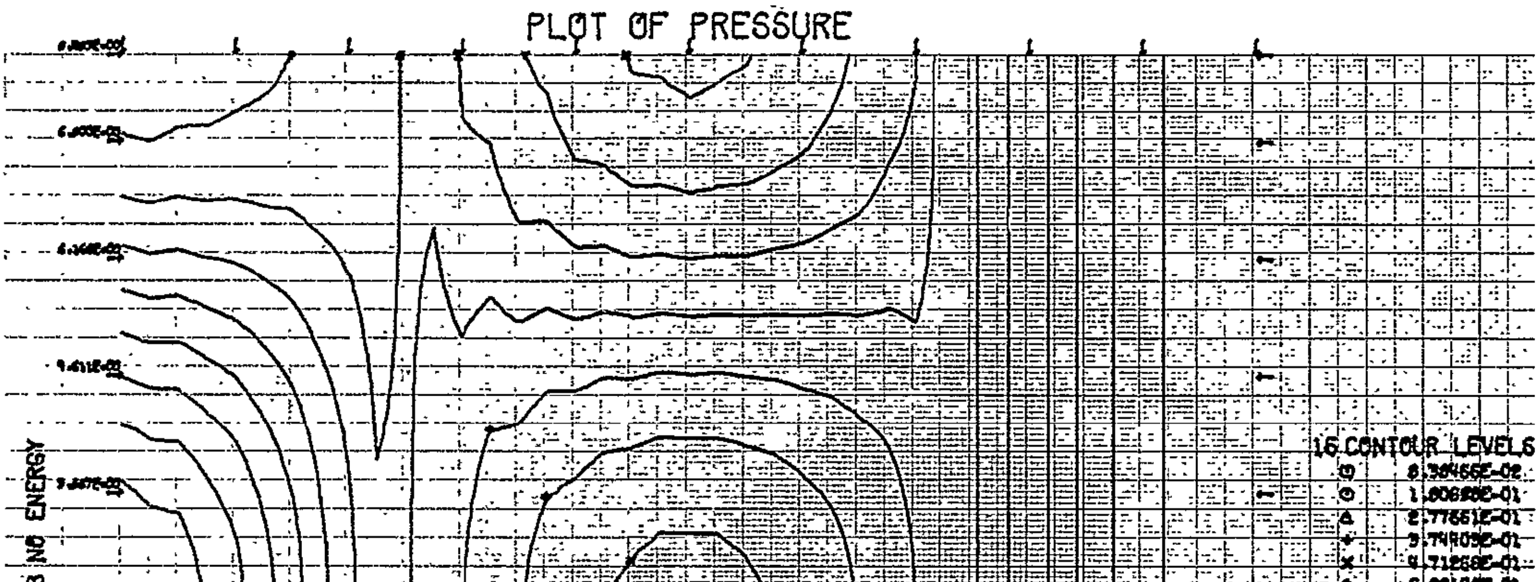

PLOT OF PRESSURE
CLASSICAL PERTURB NO ENERGY
16 CONTOUR LEVELS
T= 0.000
0 CYCLES

FIG. 3.1

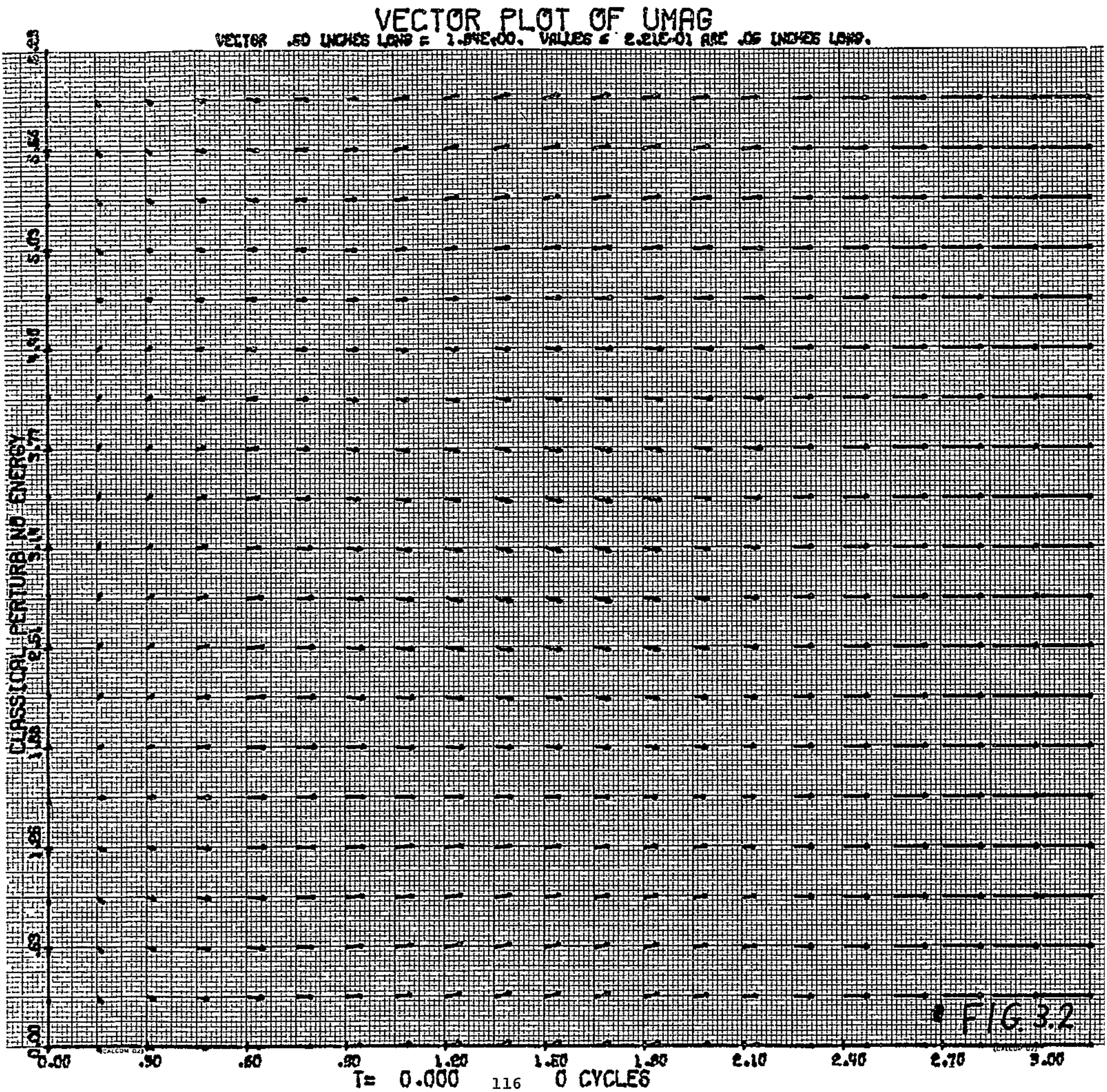
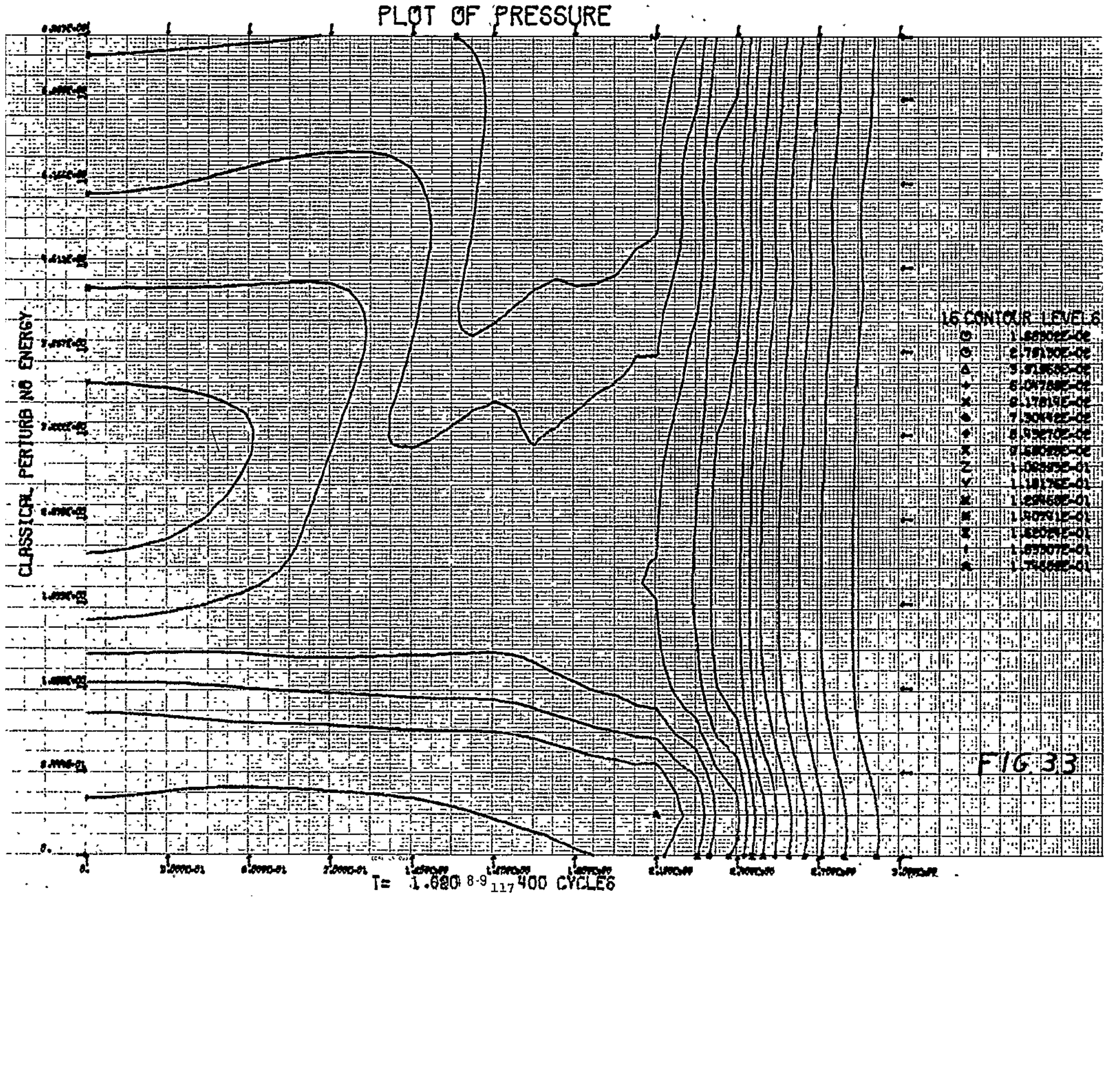
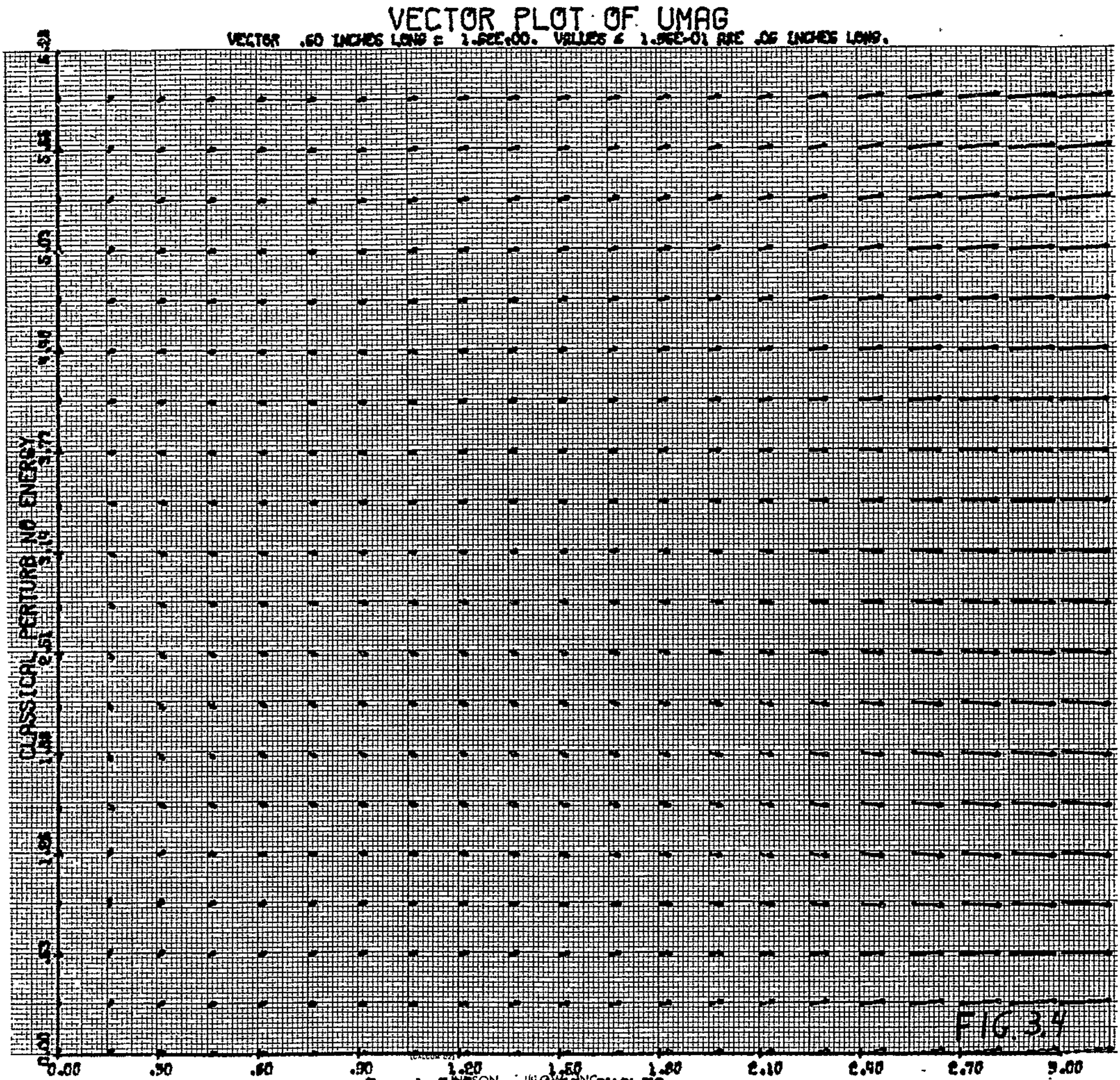

VECTOR PLOT OF UMAG
VECTOR .50 INCHES LONG = 1.84E+00. VALUES ≤ 2.21E-01 ARE .05 INCHES LONG.
CLASSICAL PERTURB NO ENERGY
0.00
.63
1.26
1.88
2.51
3.14
3.77
4.40
5.03
5.65
6.28
0.00
.30
.60
.90
1.20
1.50
1.80
2.10
2.40
2.70
3.00
T= 0.000 0 CYCLES

FIG. 3.2


PLOT OF PRESSURE
CLASSICAL PERTURB NO ENERGY
16 CONTOUR LEVELS
T= 1.680 117 400 CYCLES
FIG 33


VECTOR PLOT OF UMAG
CLASSICAL PERTURB NO ENERGY
0.00
.63
1.26
1.88
2.51
3.14
3.77
4.40
5.03
5.65
6.28
0.00
.30
.60
.90
1.20
1.50
1.80
2.10
2.40
2.70
3.00
T= 1.020 400 CYCLES
FIG. 3.4

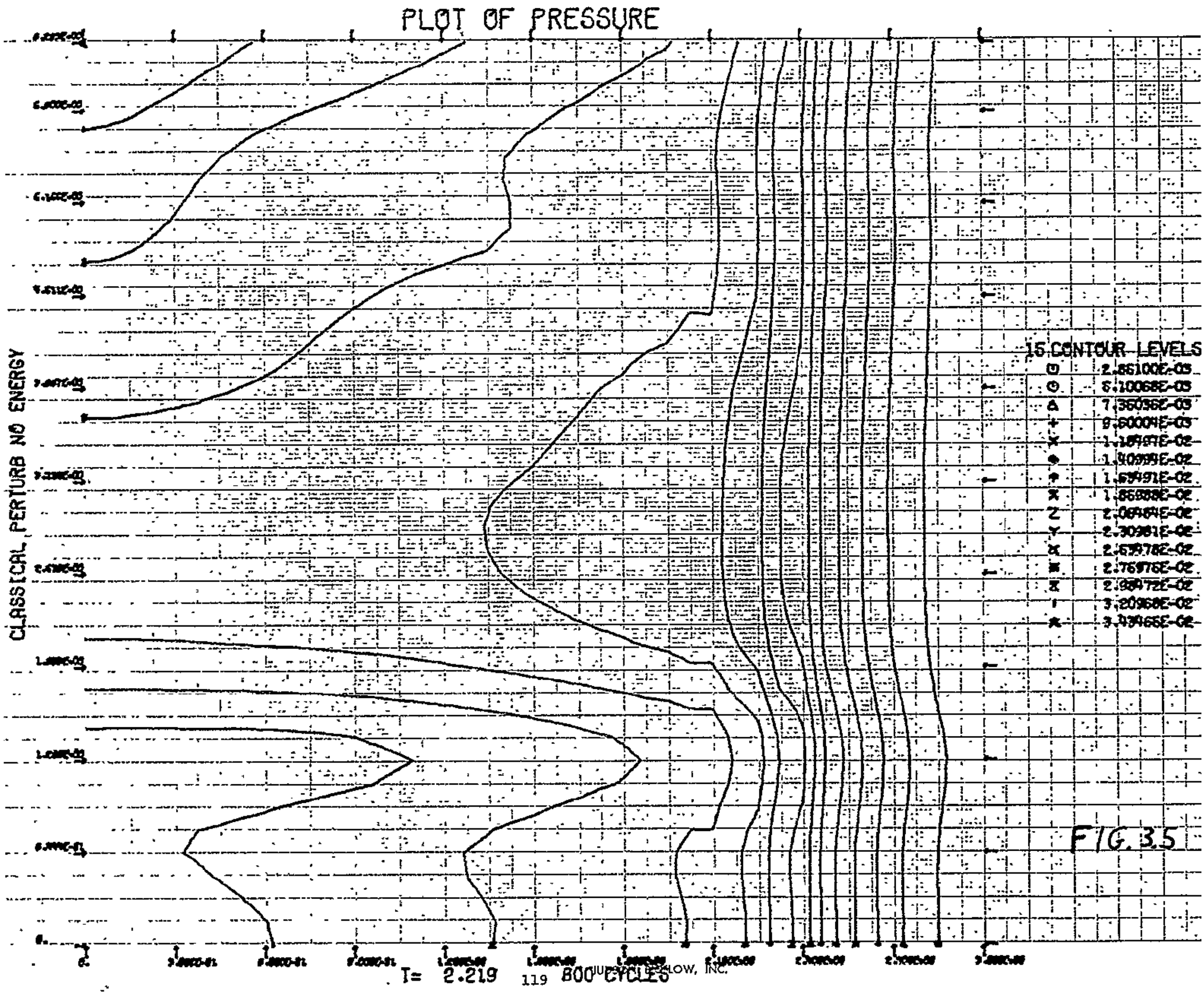
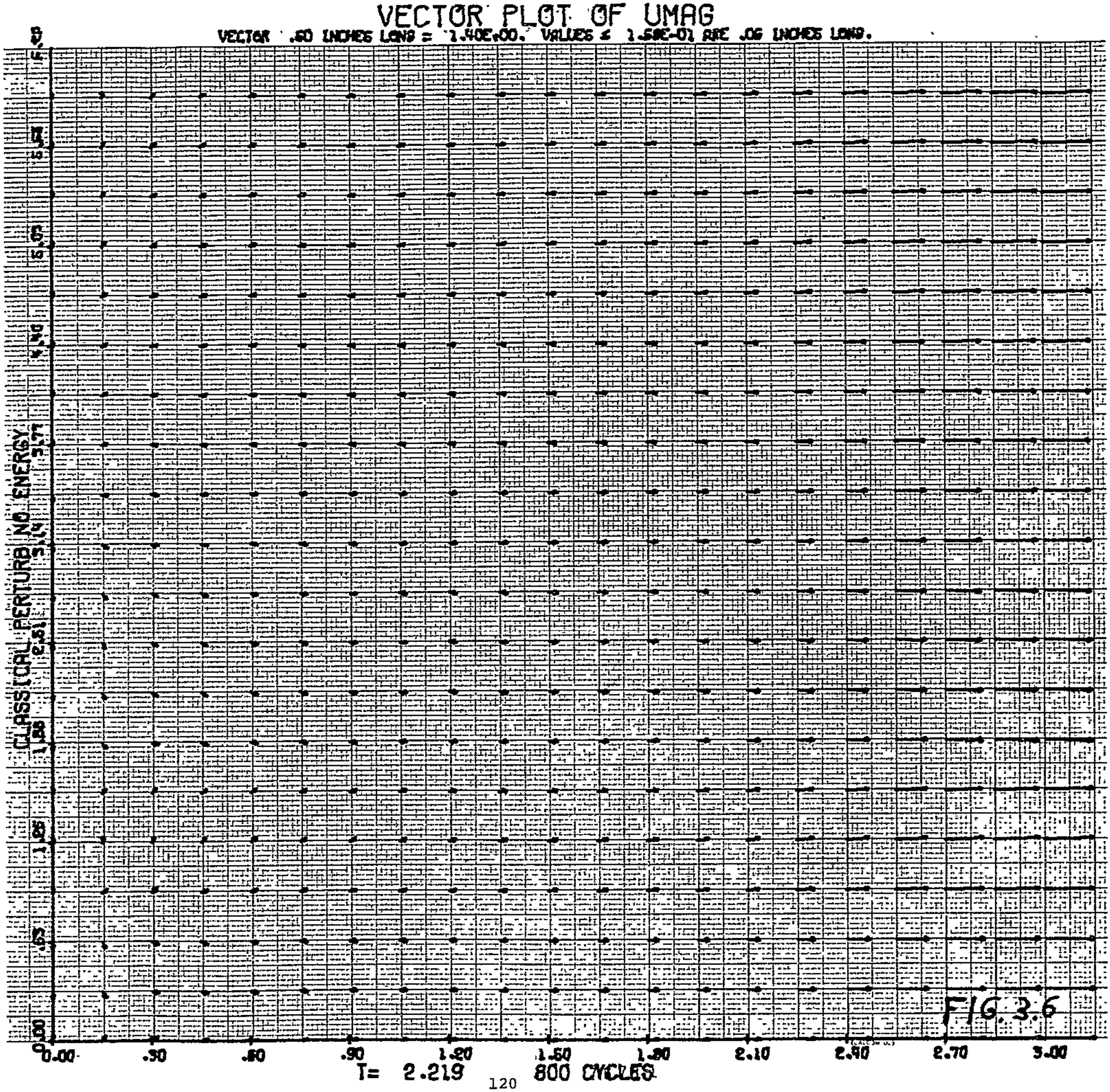

PLOT OF PRESSURE
CLASSICAL PERTURB NO ENERGY
15 CONTOUR LEVELS
T= 2.219 800 CYCLES

FIG. 3.5


VECTOR PLOT OF UMAG
VECTOR .60 INCHES LONG = 1.40E+00. VALUES ≤ 1.50E-01 ARE .06 INCHES LONG.
CLASSICAL PERTURB NO ENERGY
0.00
.30
.60
.90
1.20
1.50
1.80
2.10
2.40
2.70
3.00
T= 2.219
800 CYCLES
FIG. 3.6

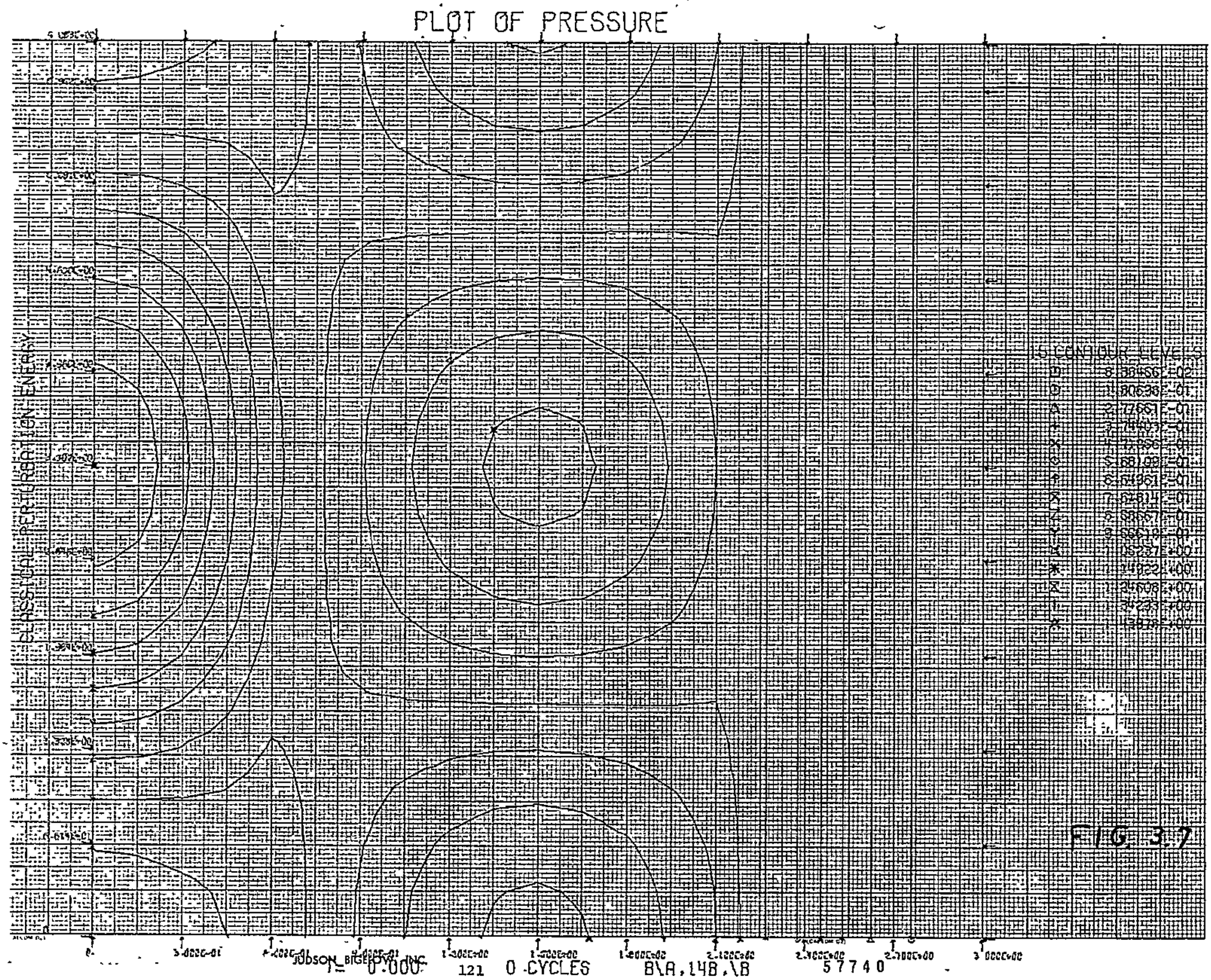

PLOT OF PRESSURE
CLASSICAL PERTURBATION ENERGY
15 CONTOUR LEVELS
T= 0.000 121 0 CYCLES B\A,14B,\B 57740

FIG. 3.7

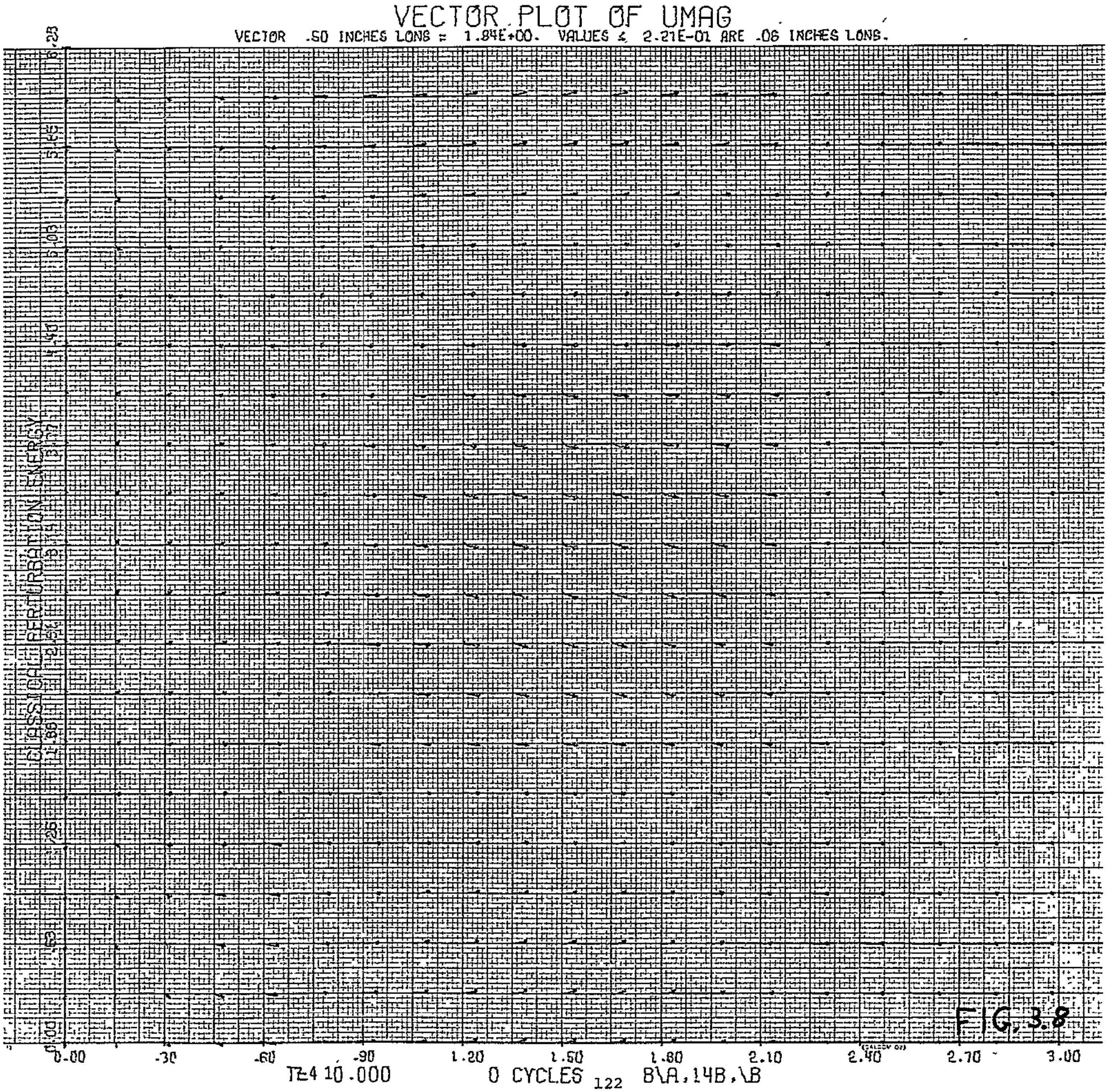
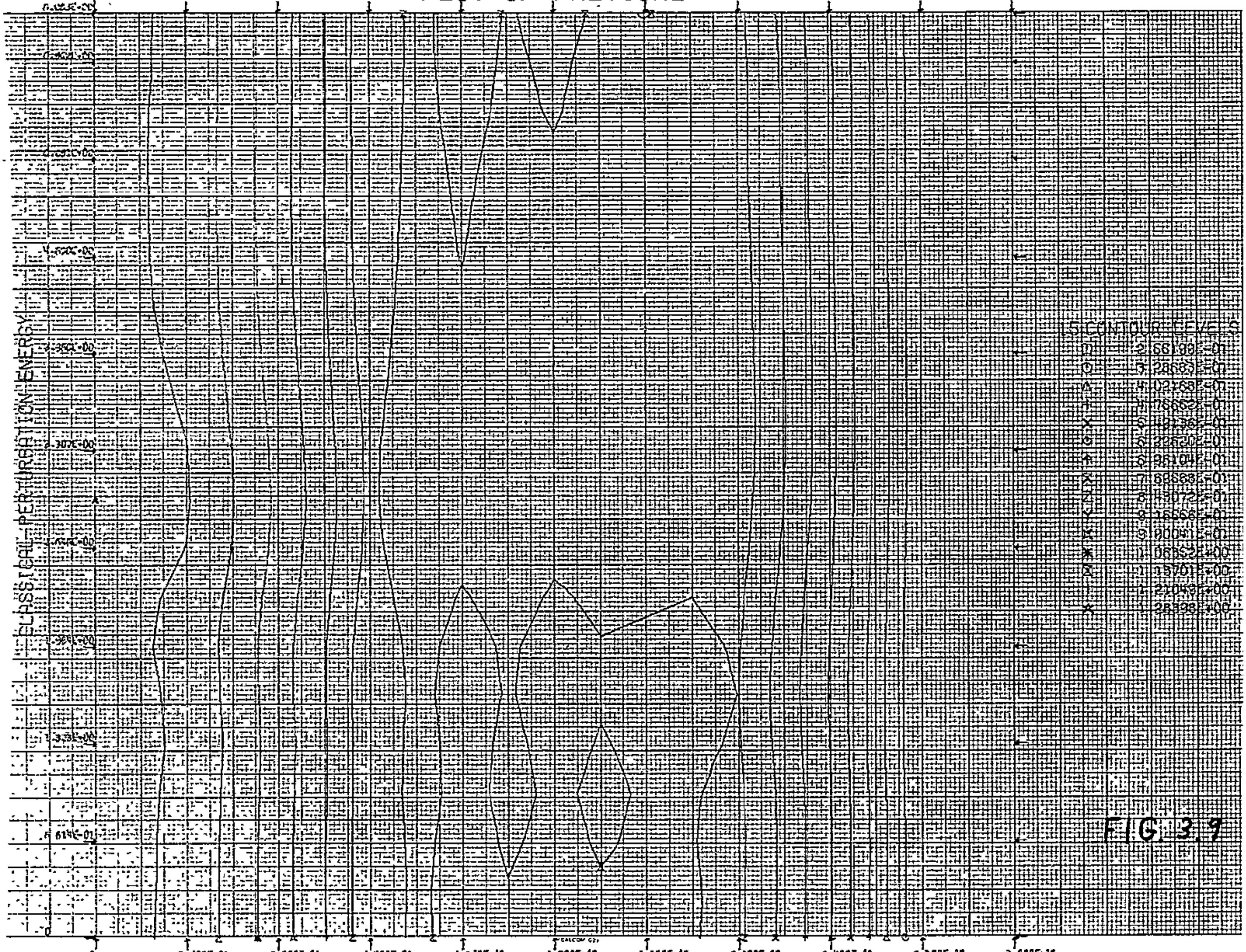

VECTOR PLOT OF UMAG
VECTOR .50 INCHES LONG = 1.84E+00. VALUES ≤ 2.21E-01 ARE .06 INCHES LONG.
CLASSICAL PERTURBATION ENERGY
0.00
.63
1.25
1.88
2.51
3.14
3.77
4.40
5.03
5.65
6.28
0.00
.30
.60
.90
1.20
1.50
1.80
2.10
2.40
2.70
3.00
10.000
0 CYCLES
B\A,14B,\B

FIG. 3.8

PLOT OF PRESSURE



T= 1.607 123 300 CYCLES B\A.148.\B

FIG. 3.9

VECTOR PLOT OF UMAG

VECTOR .50 INCHES LONG = 1.44E+00. VALUES < 1.73E-01 ARE .06 INCHES LONG.

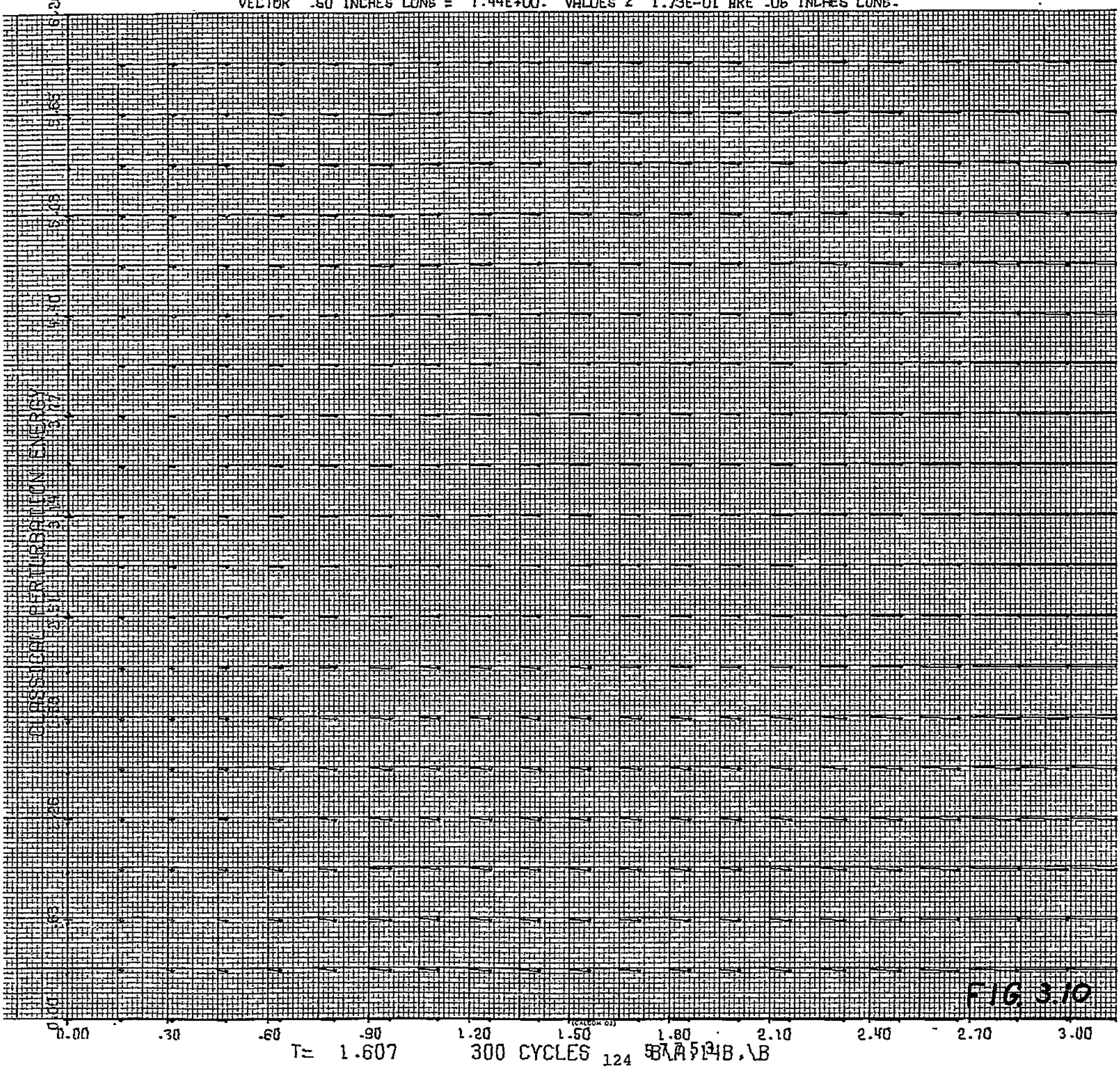


FIG. 3.10

T= 1.607 300 CYCLES

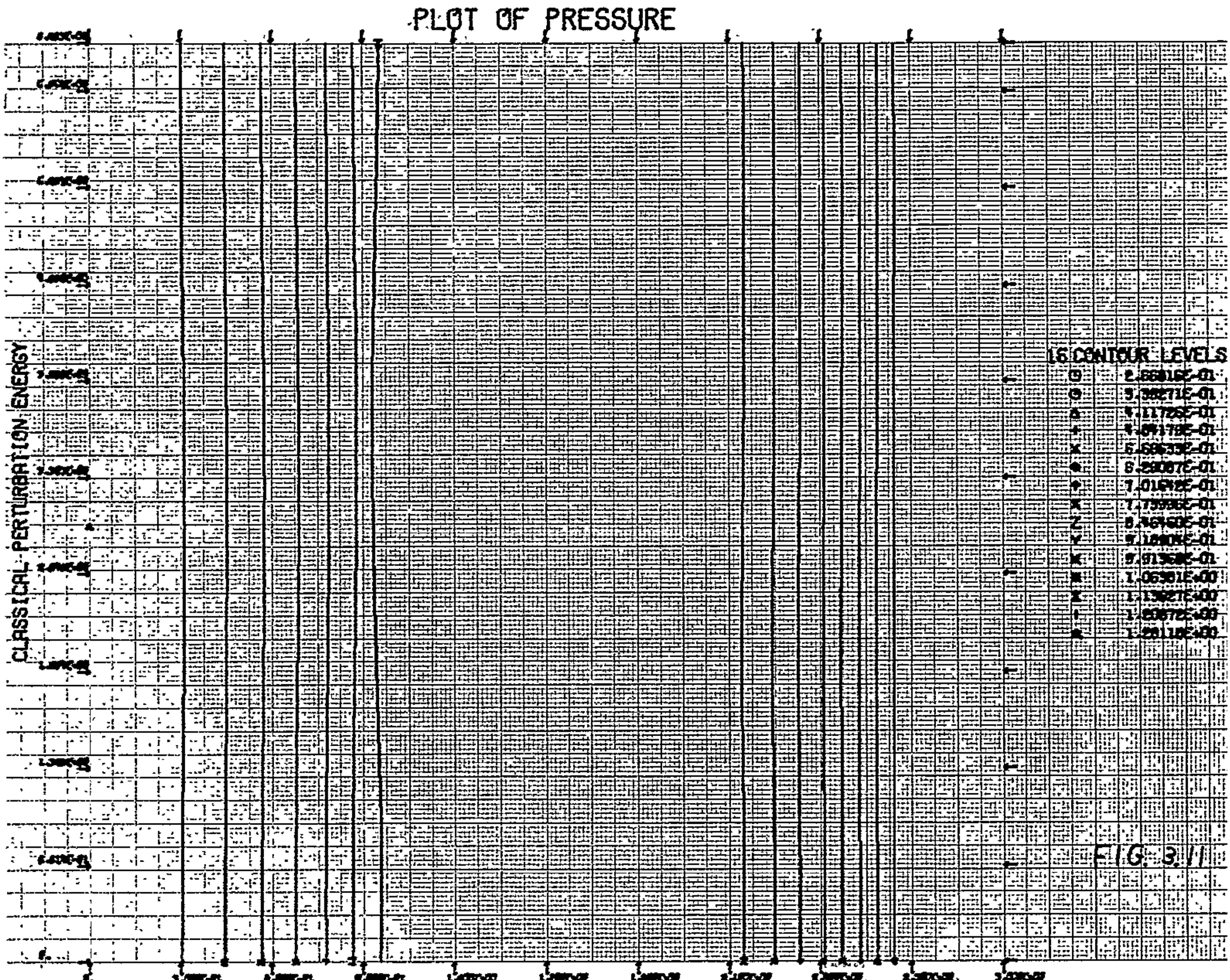


FIG. 3.11

TE 5 20.698 . 500 CYCLES 125 B\A.14B.\B

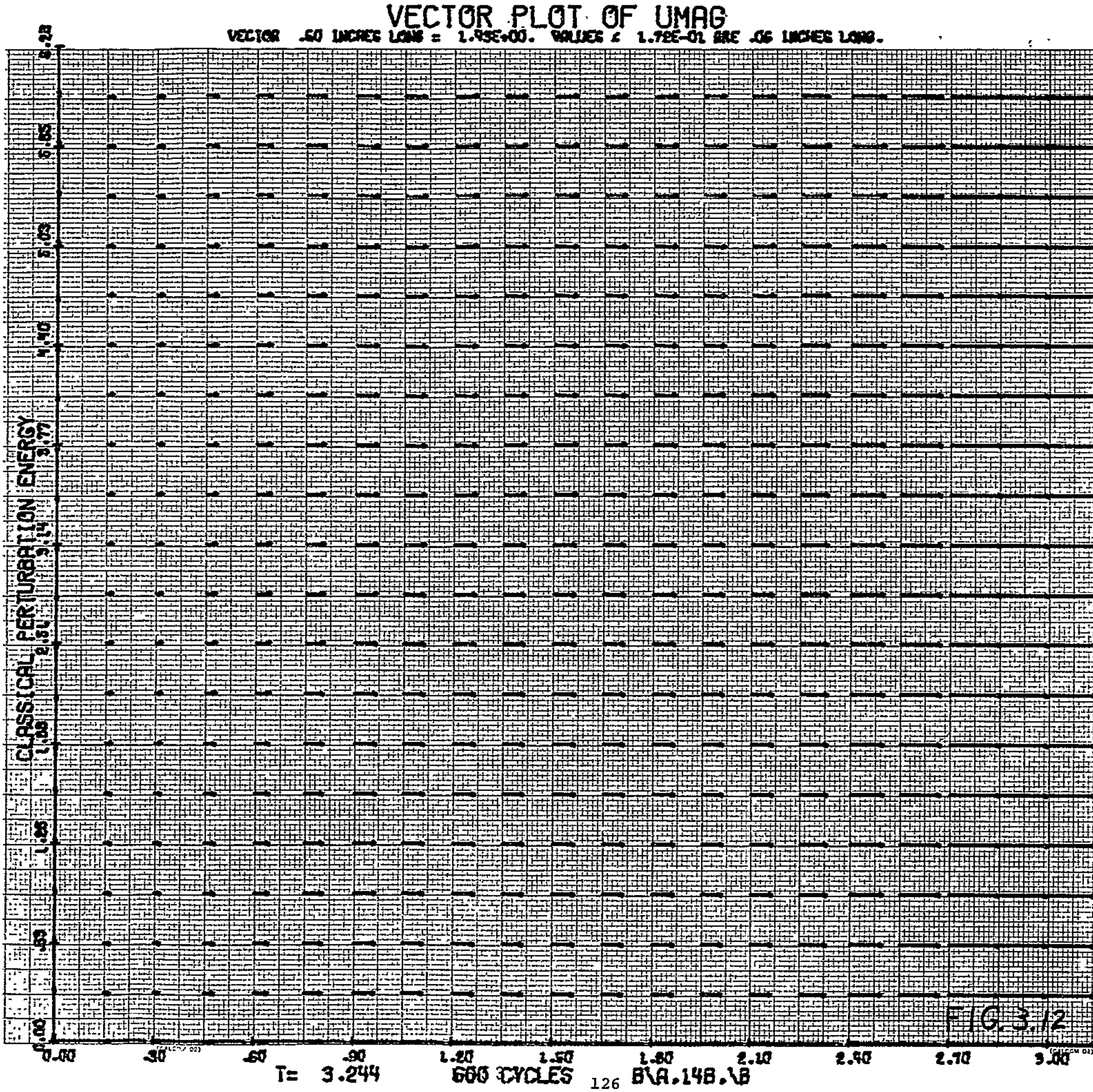
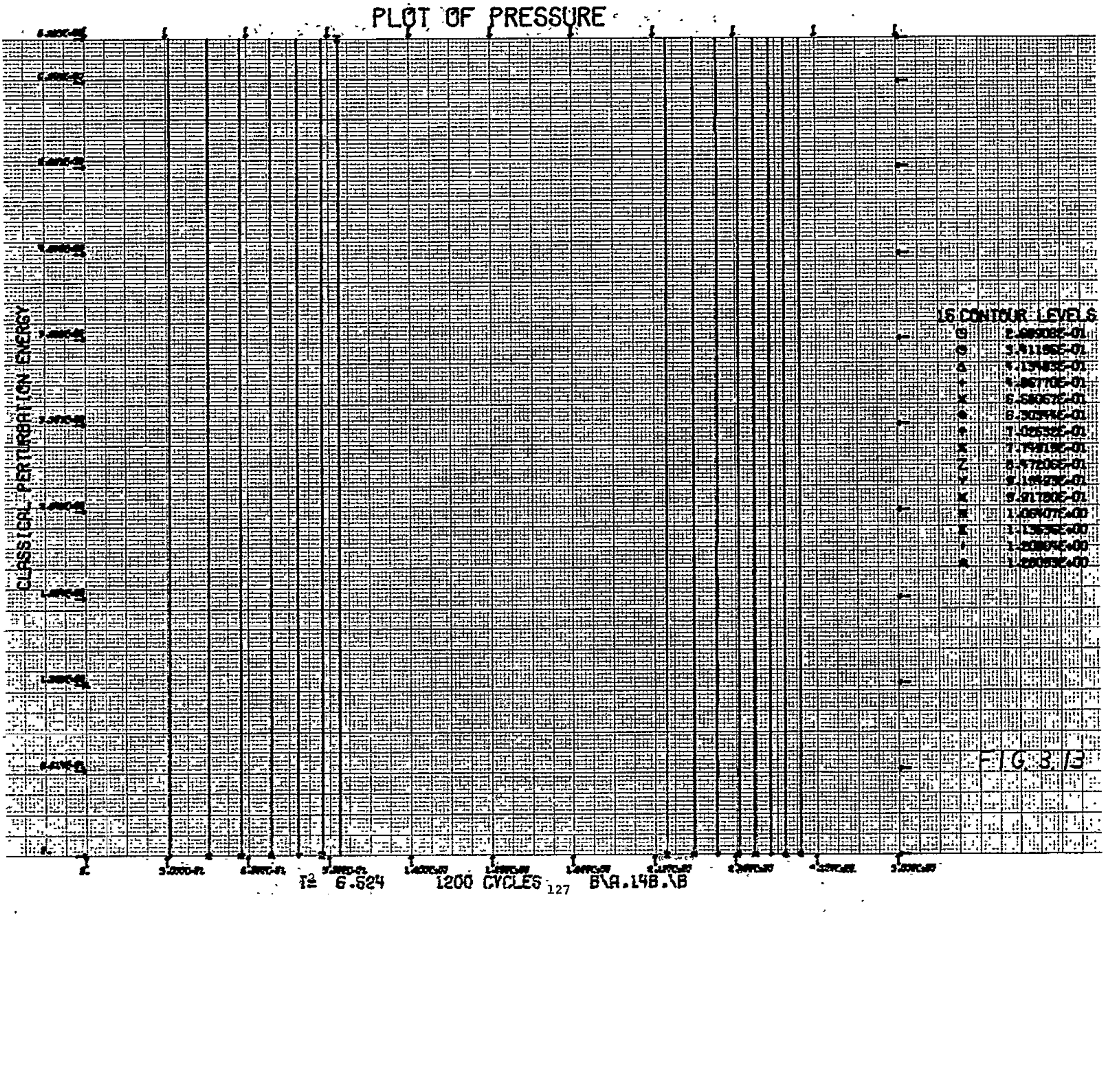

VECTOR PLOT OF UMAG
CLASSICAL PERTURBATION ENERGY
T= 3.244 600 CYCLES
B\A.148.\B

FIG. 3.12


PLOT OF PRESSURE
CLASSICAL PERTURBATION ENERGY
15 CONTOUR LEVELS
T= 6.524 1200 CYCLES B\A.148.\B
FIG 3.13

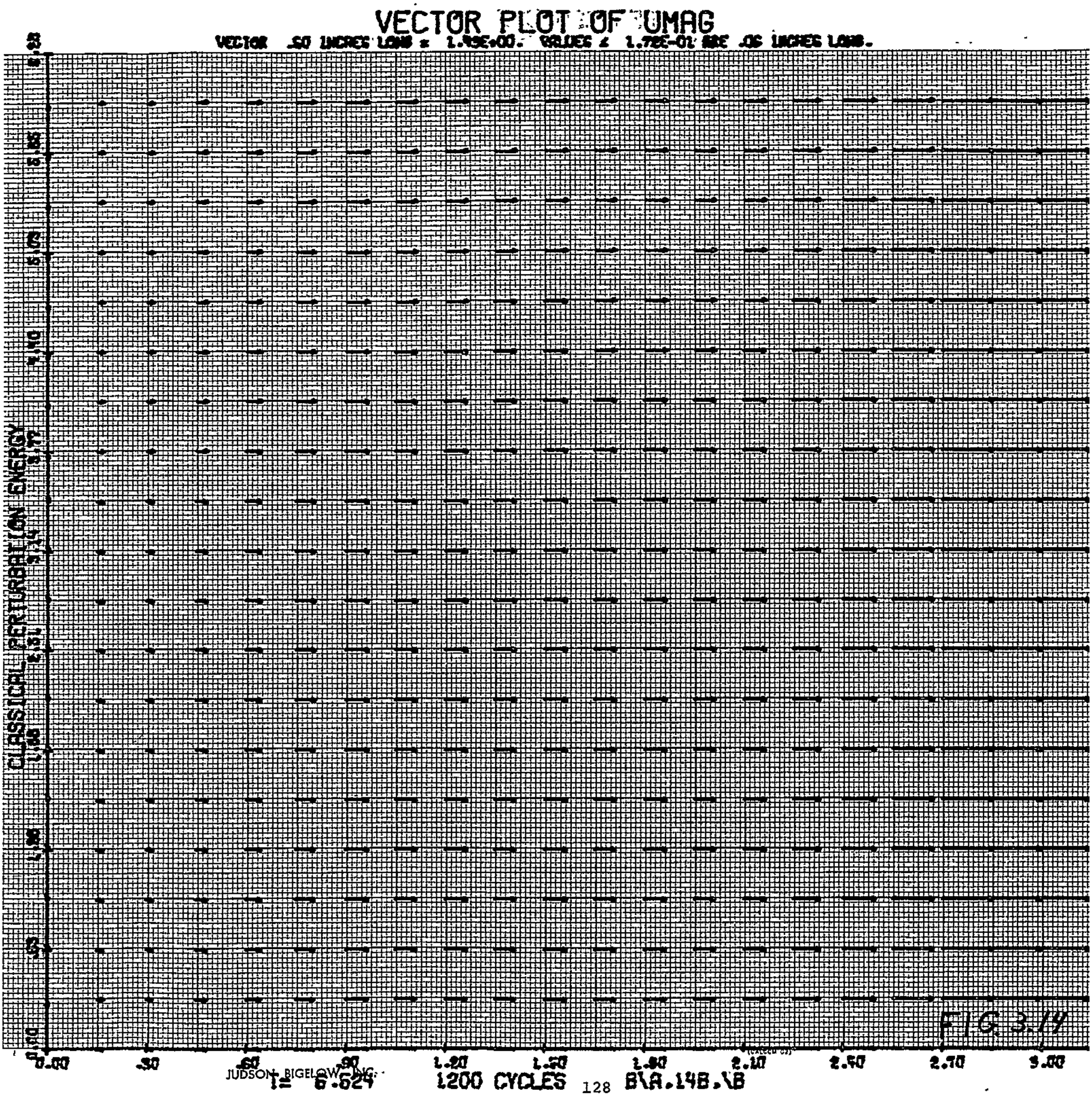

VECTOR PLOT OF UMAG
CLASSICAL PERTURBATION ENERGY
0.00
.30
.60
.90
1.20
1.50
1.80
2.10
2.40
2.70
3.00
T= 6.624
1200 CYCLES
B\A.14B.\B

FIG. 3.14

Initial conditions for a "pop" were simulated with an energy pulse equal to 4% of the energy in the chamber. The center of the pop was placed at $\theta = 3\pi/2$ and $z = 3\Delta z$. It extended for a distance of $3\Delta\theta$ on each side in the θ direction and for a distance of Δz on each side in the z direction. The pressure in the pop was obtained from the formula

$$p = \overline{p} + c\,A \sin\left(\frac{\pi}{6\Delta\theta}\,\theta\right) \qquad \frac{3\pi}{2} - 3\Delta\theta \leq \theta \leq \frac{3\pi}{2} + 3\Delta\theta$$

where $c = \frac{1}{2}$ for $z = 2\Delta z$ $\quad c = \frac{1}{4}$ for $z = \Delta z,\ 3\Delta z$.

A was obtained

$$E_{pop} = \frac{1}{\gamma-1}(cA)^{\frac{\gamma-1}{\gamma}} \int_{\frac{3\pi}{2} - 3\Delta\theta}^{\frac{3\pi}{2} + 3\Delta\theta} \sin\left(\frac{\pi}{6\Delta\theta}\right)^{\frac{\gamma-1}{\gamma}} d\theta \qquad (4.18)$$

which was obtained by integrating the $pd(\frac{1}{\rho})$ work over the above interval. The energy of the "pop" E_{pop} is specified to be 4% of the total chamber energy.

The calculation of "pops" or local high pressure regions introduced new difficulties in the annular calculation. As is indicated from Figure 3.15, the maximum non-dimensional pressure is 19.38 (or almost 6000 psia) at the center of the pop. The pop was localized to two increments in the axial direction near the injector-face; the tangential increment is specified from Equation (4.18). It was found that the difference scheme (as given in Reference 1) was unable to carry out the transient phase of the calculation given the above initial conditions. The calculation quickly became unstable after a few cycles of computation. A multistep smoothing operator was introduced into the calculation (after Lapidus, see Richtmyer and Morton, Academic Press). It is a third order operator so that in the smooth parts of the flow field, the order of accuracy of the calculation is preserved.

The smoothing operator is a two step operator in the same way that the difference equations are two step schemes. Let D_- denote the following backward difference operator

$$D_- w_{j+1,k} = w_{j+1,k} - w_{j,k}$$

Then define

$$\tilde{W}_{j,k} = W_{j,k} + \lambda_j \kappa \, D_- \, \{ |\underline{D} u_{j+1,k}| \underline{D} W_{j+1,k} \} \tag{4.19a}$$

where u is the velocity in the j direction and $\lambda_j = \Delta t/\Delta j$, Δj the space step in the j direction. The constant $\kappa = 0.5$. Then define

$$\tilde{\tilde{W}}_{j,k} = \tilde{W}_{j,k} + \lambda_k \kappa \, D_- \, \{ |D\tilde{v}_{j,k+1}| D_- \tilde{W}_{j,k+1} \} \tag{4.19b}$$

where $D_- W_{j,k} = W_{j,k} - W_{j,k-1}$ here and $\lambda_k = \Delta t/\Delta_k$, Δ_k the space step in the k direction. The velocity in the k direction is v. The quantities on the right hand side of Equation (4.19b) are obtained by sweeping through the mesh using Equation (4.19a). The constant

$\kappa = 0.5$ in Equation (4.19b). The value of $\tilde{\tilde{W}}$ is taken to be the final smoothed results to be used as the starting data in the next cycle of the computation. It is clear that Equations (4.19a) and (4.19b) are third order differences and will vanish quickly where the flow is smooth. Only where the flow undergoes rapid variation will this artificial viscosity make a contribution.

For the last two calculations described below, Equations (4.19a) and (4.19b) were used each cycle for the first fifty cycles.

The initial conditions corresponding to the "pop" calculation are given in Figures (3.15) and (3.16) while the transient resulting from the initial disturbance is given in one hundred cycle increments in Figures (3.17) to (3.22). Unfortunately the printouts are too far apart to follow the initial transient motion, however, since there is no energy addition one observes that the maximum chamber pressure undergoes a uniform decay out to about 2.1 milliseconds. It is anticipated that future runs will result in more closely spaced snapshots.

The same initial data, used in the above calculation, was the starting point for a reacting flow problem. The only difference between this case and the above is that mass and energy addition, as prescribed by the modified Godsave analysis, was turned on. As stated Figures (3.15) and (3.16) give the initial distribution of pressure and velocity. Figure (3.23) shows the pressure field after fifty cycles or .23 milliseconds. The calculation stopped at 99 cycles, just before the next snapshot. This was the first time this case was run so that comments will be reserved until more results are obtained.

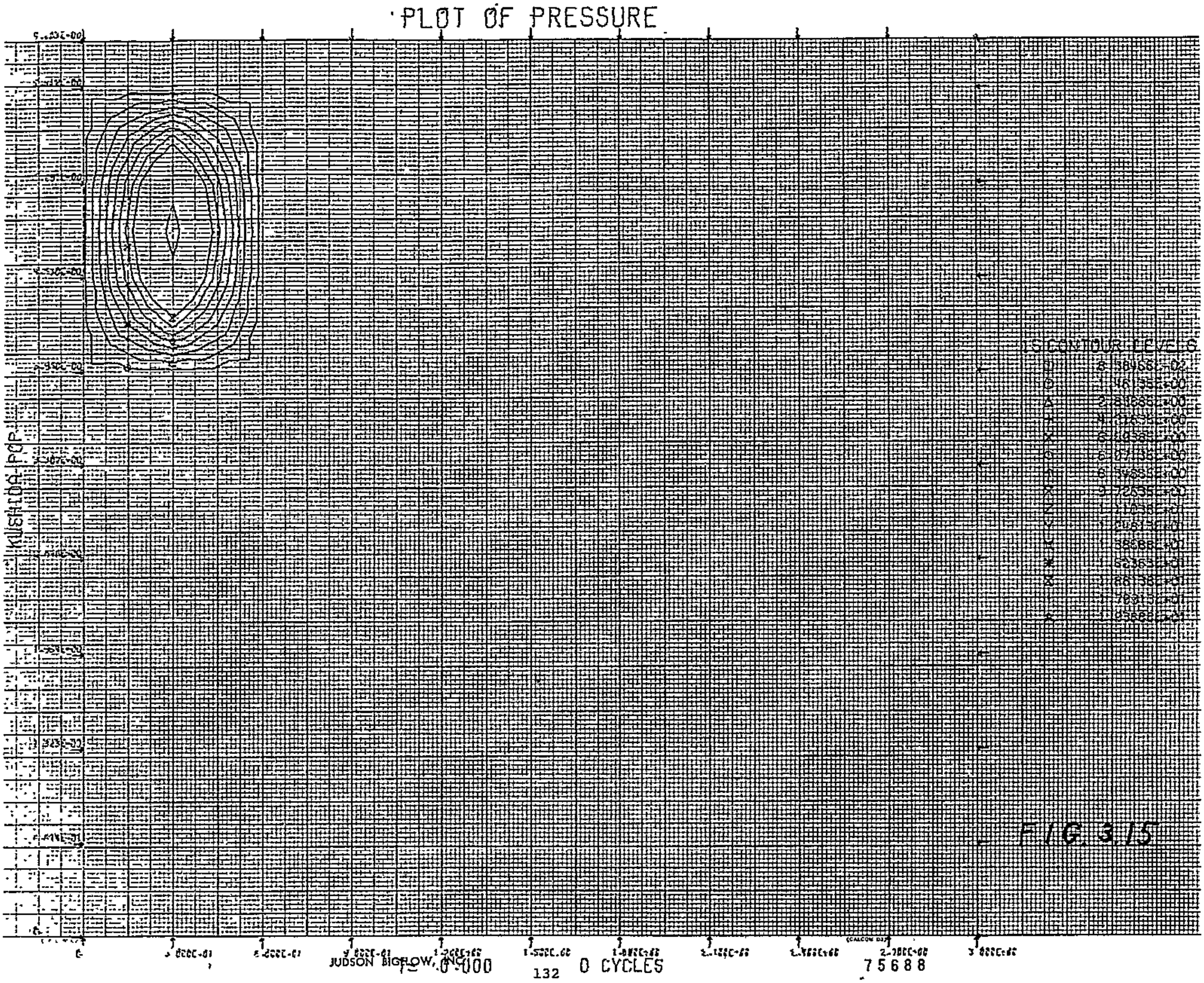


FIG. 3.15

VECTOR PLOT OF UMAG

VECTOR .50 INCHES LONG = 1.89E+00. VALUES ≤ 2.27E-01 ARE .06 INCHES LONG.

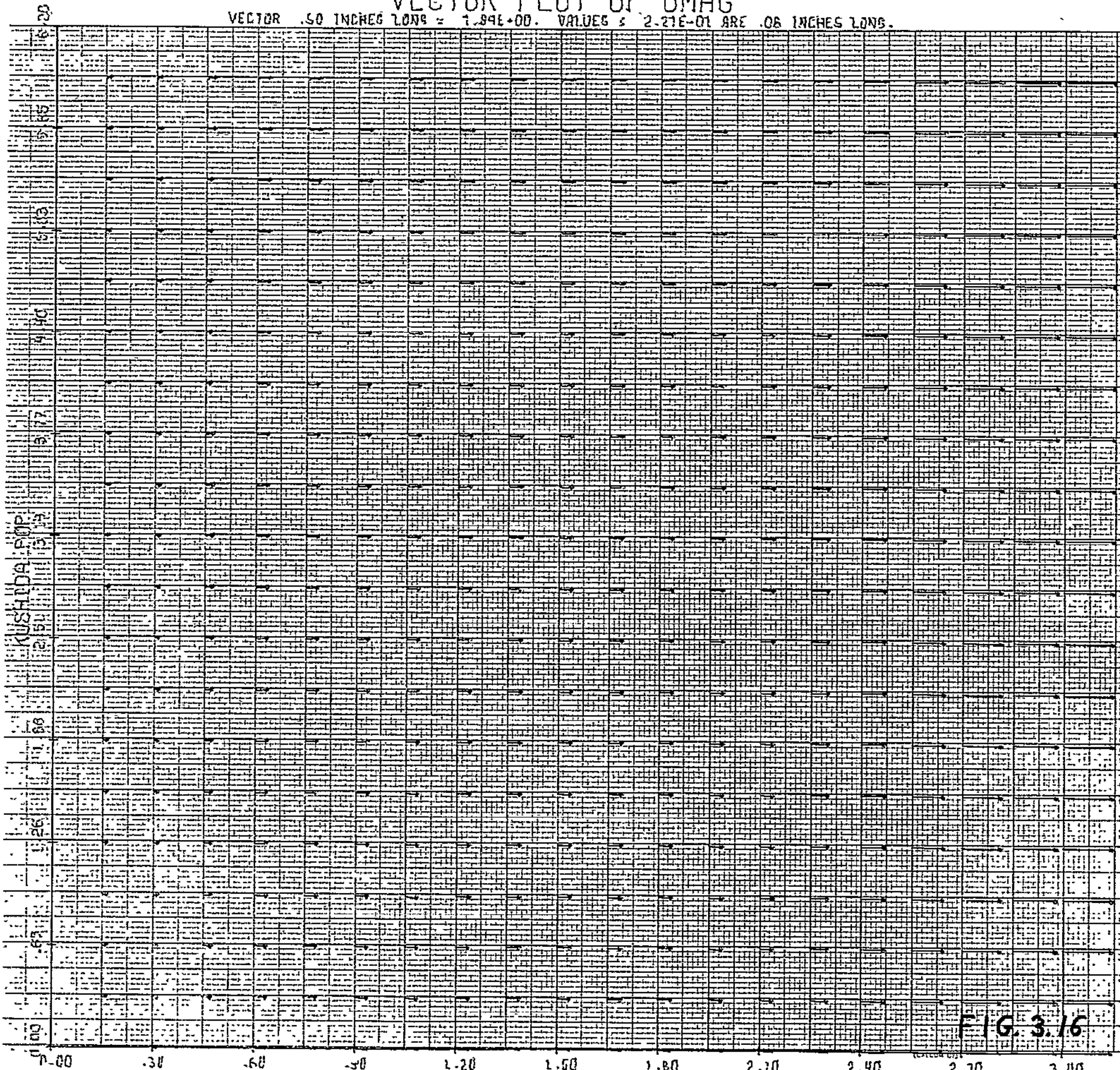
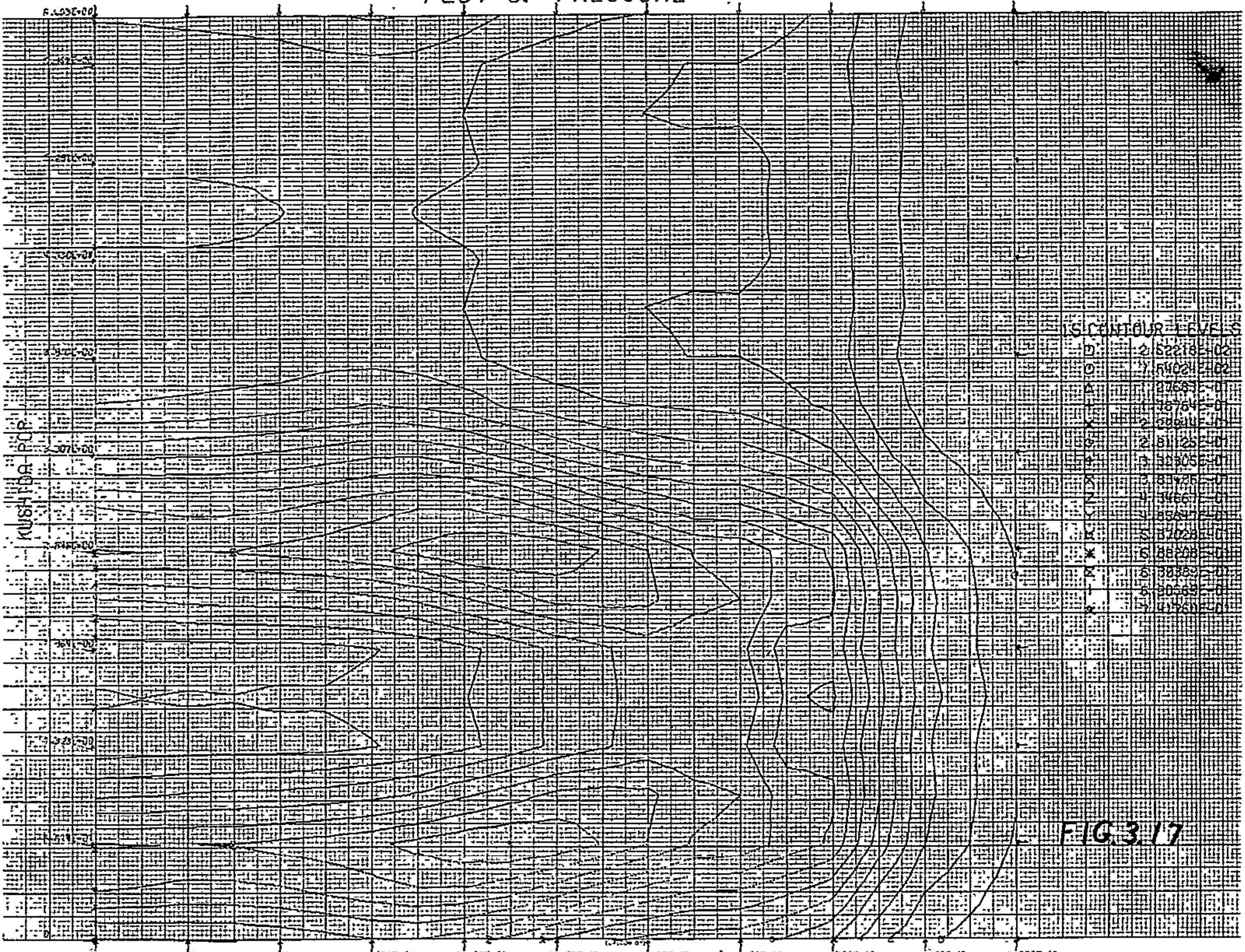


FIG. 3.16

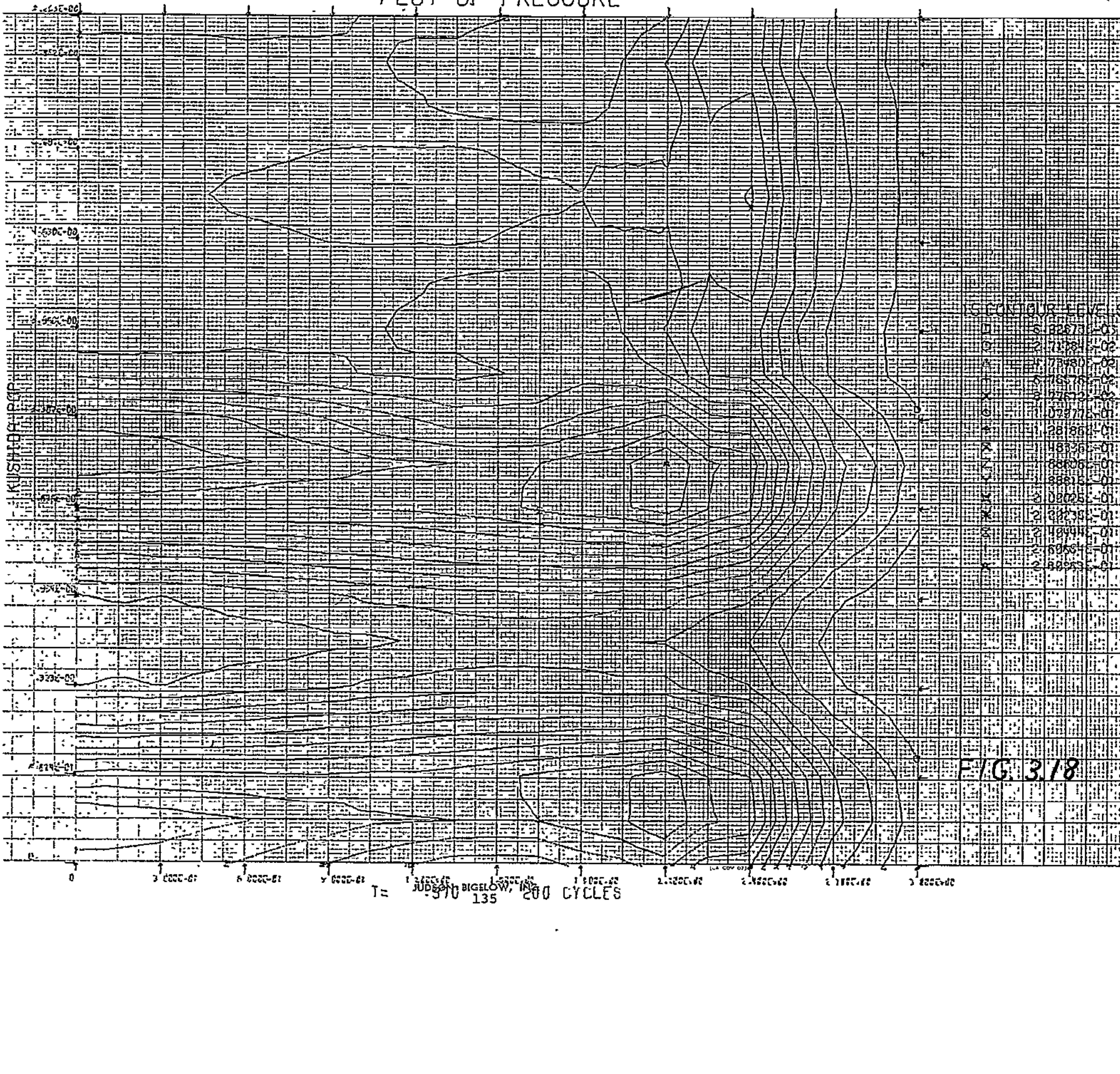
T= 0.000 0 CYCLES

PLOT OF PRESSURE



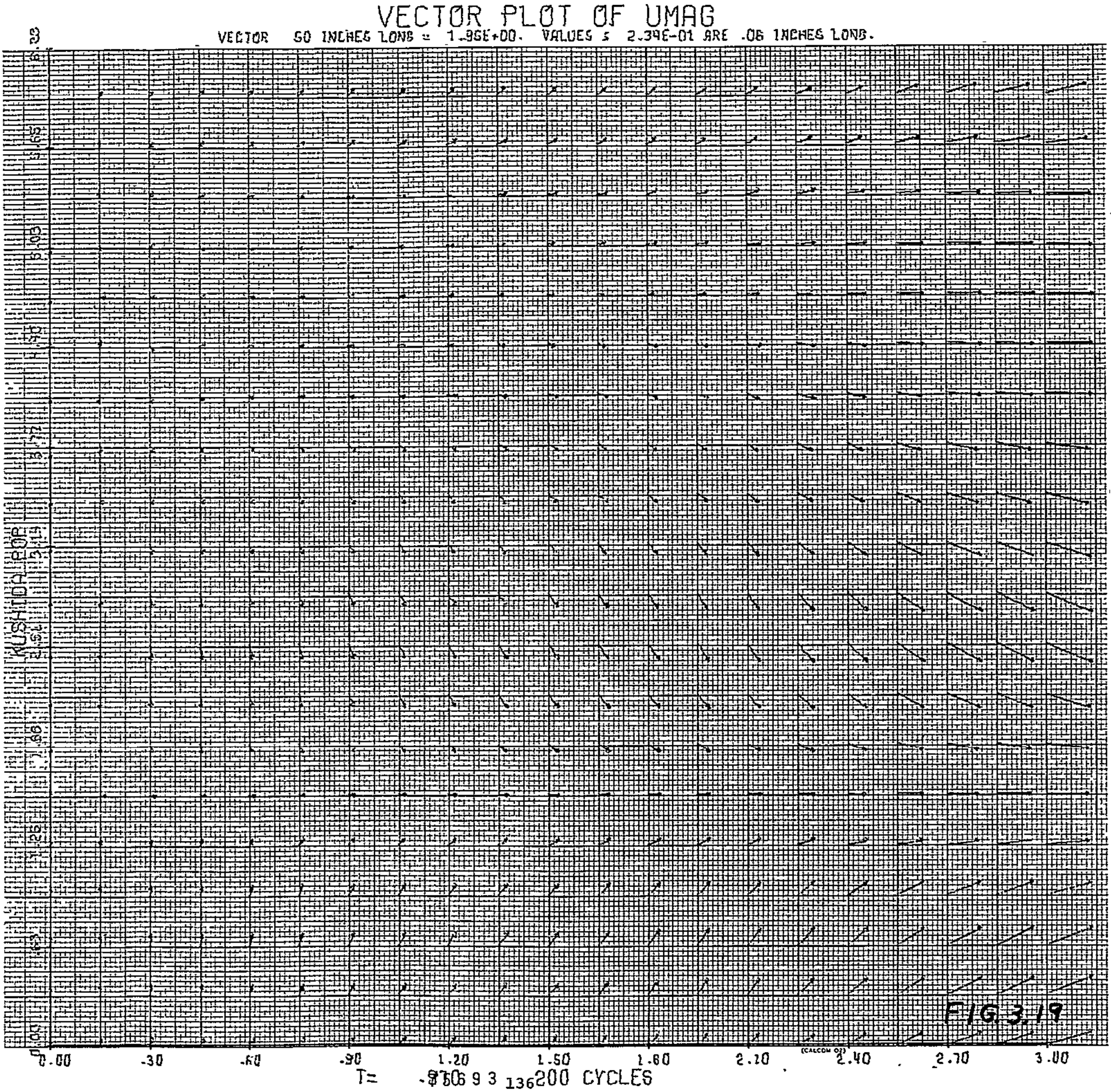
T= .468 100 CYCLES

PLOT OF PRESSURE



T= .970 200 CYCLES

JUDSON BIGELOW, INC.



VECTOR PLOT OF UMAG
VECTOR .50 INCHES LONG = 1.96E+00. VALUES ≤ 2.34E-01 ARE .06 INCHES LONG.
KUSHIDA POP
0.00
.30
.60
.90
1.20
1.50
1.80
2.10
2.40
2.70
3.00
T= 200 CYCLES

FIG. 3.19

PLOT OF PRESSURE

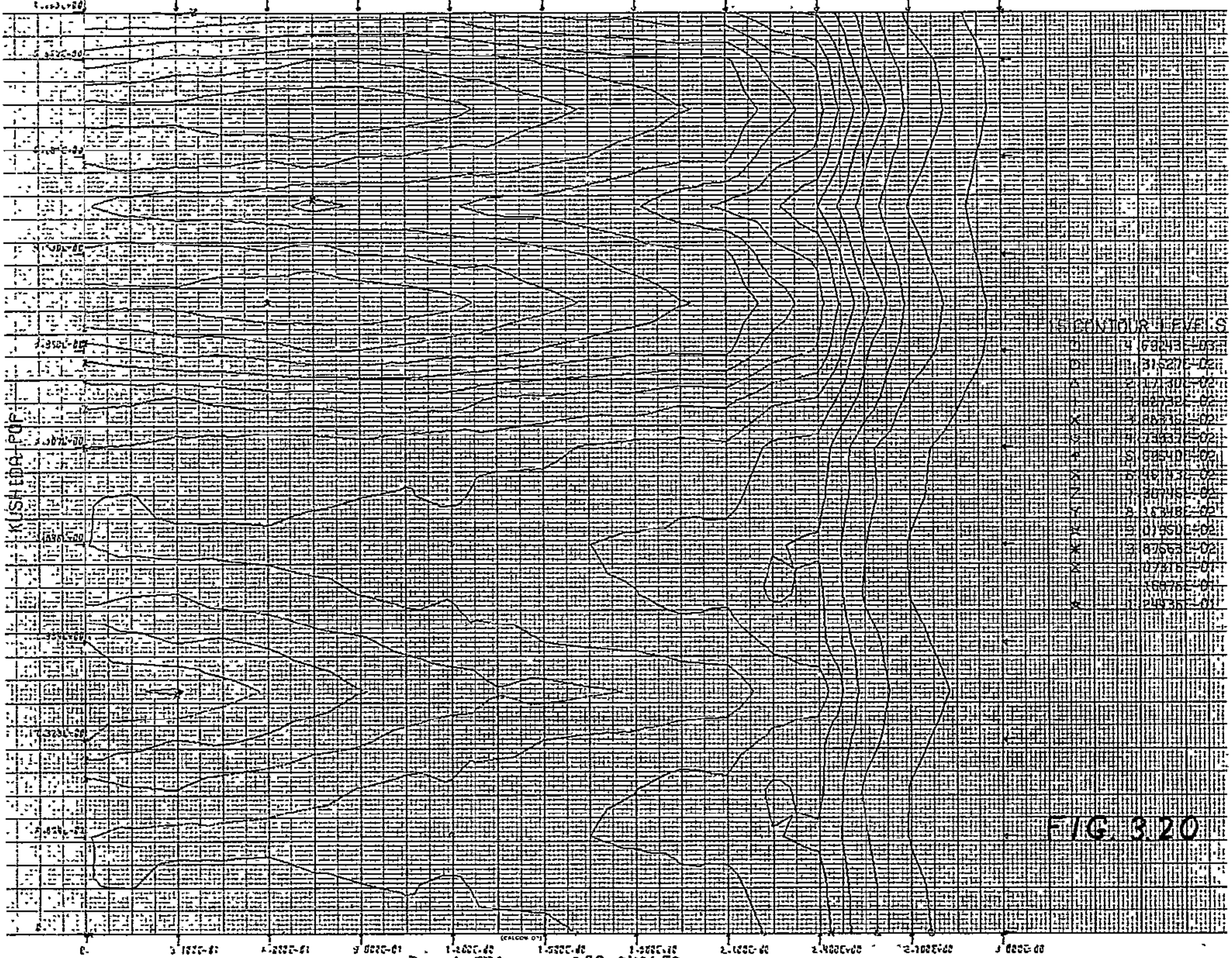


FIG. 3.20

T= 1.531 300 CYCLES

PLOT OF PRESSURE

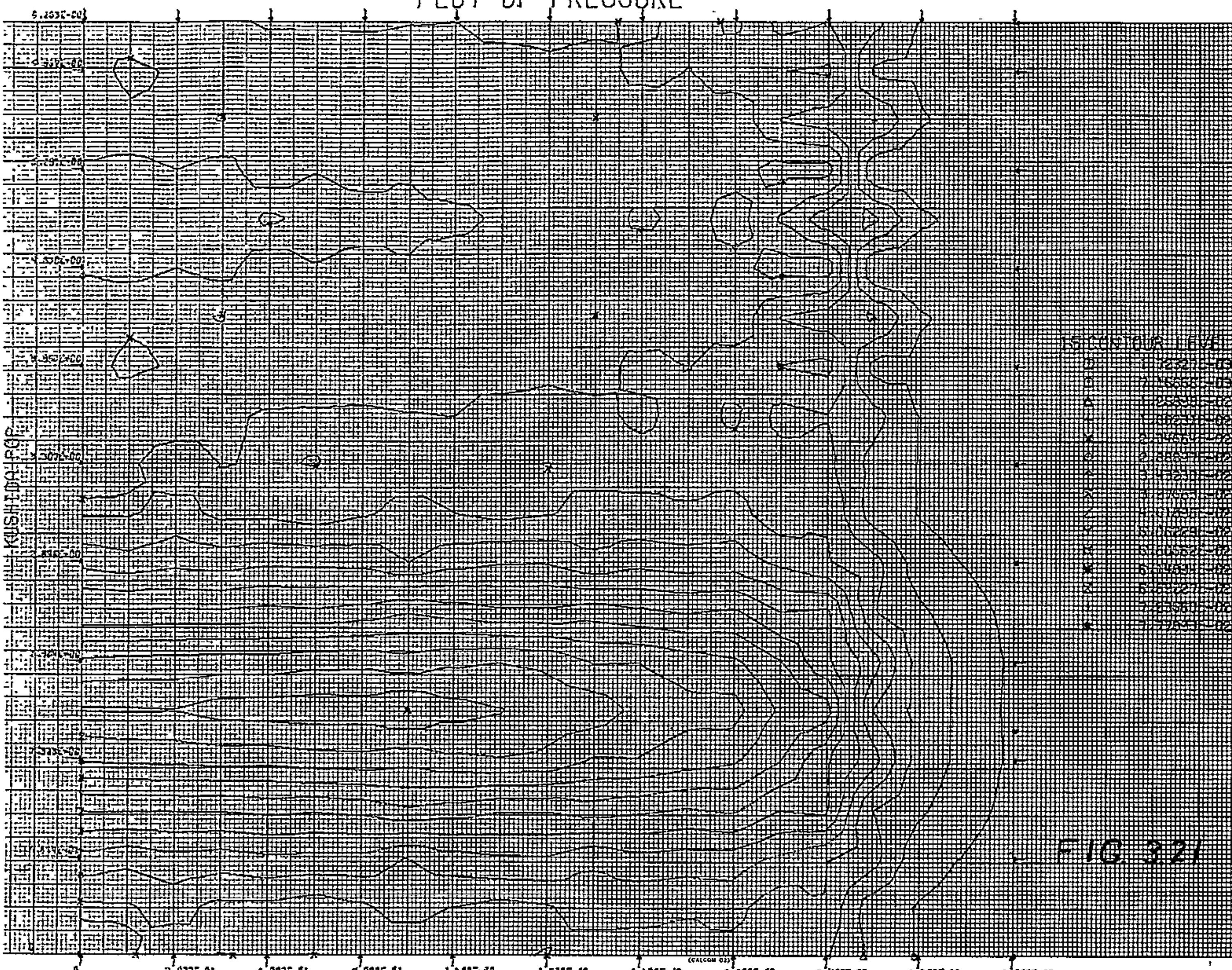


FIG. 321

T= 2.077 138

VECTOR PLOT OF UMAG

VECTOR .50 INCHES LONG = 2.08E+00. VALUES ≤ 2.51E-01 ARE .06 INCHES LONG.

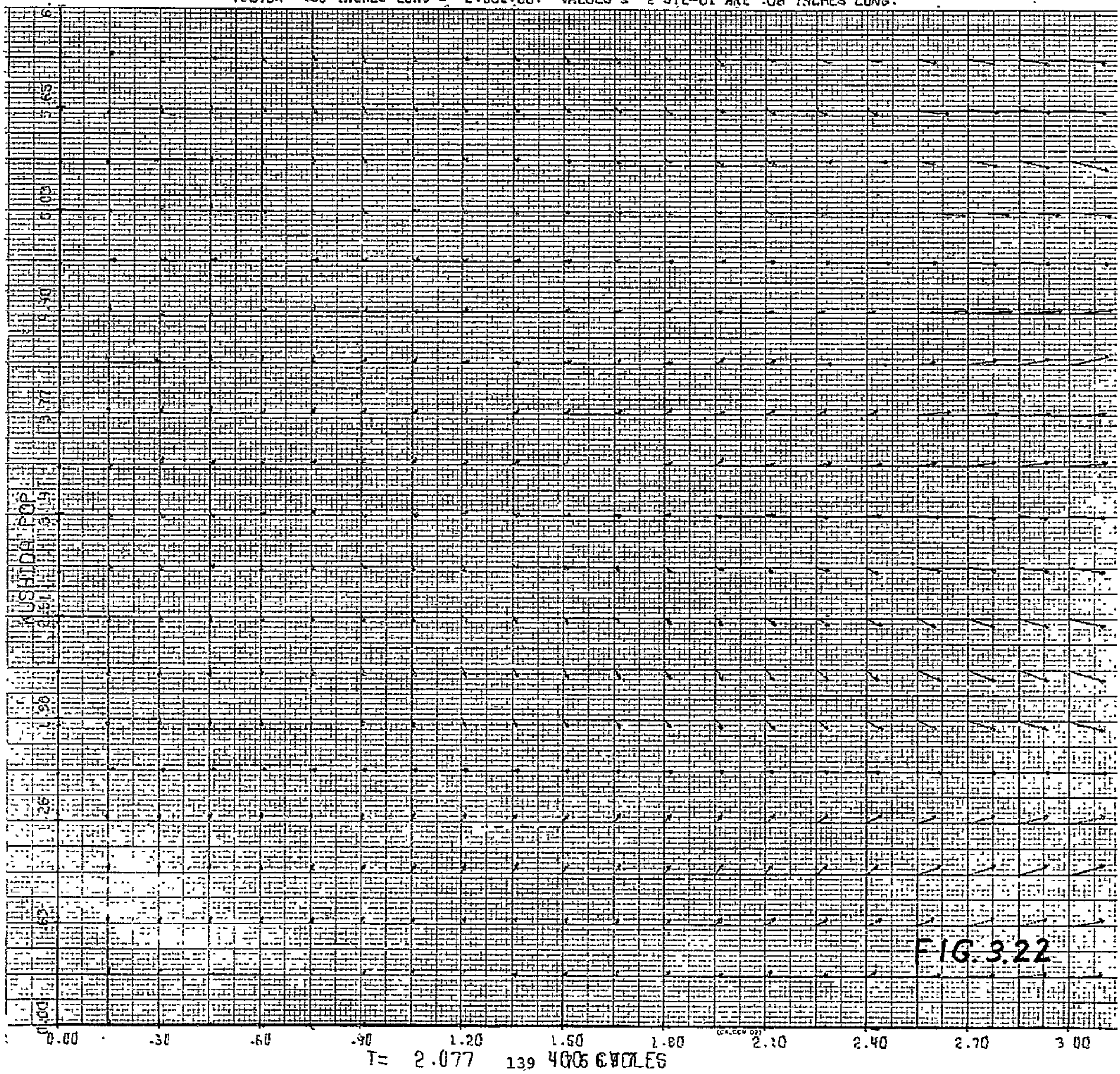
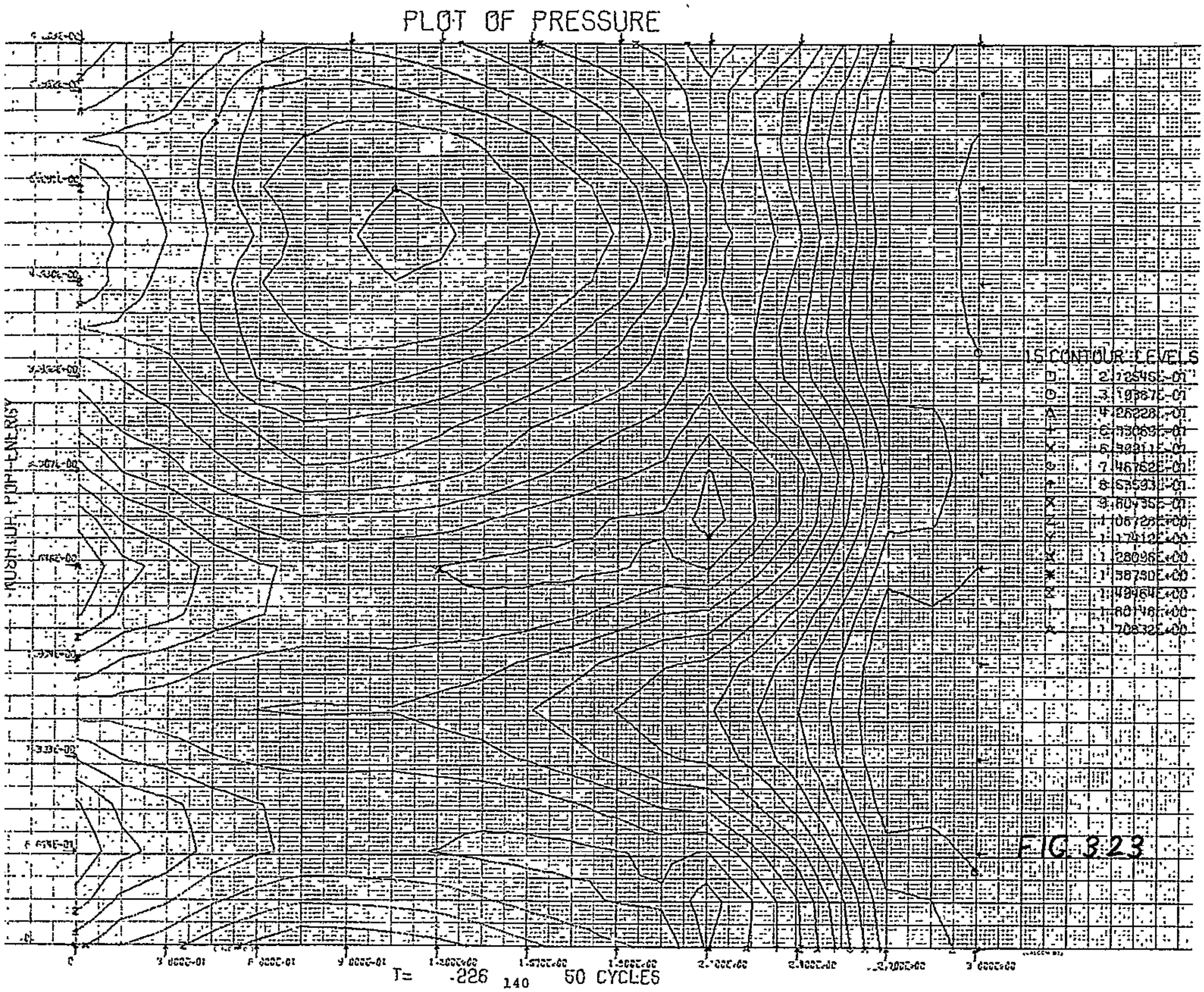


FIG.3.22


PLOT OF PRESSURE
MOMENTUM POT ENERGY
15 CONTOUR LEVELS
2.72545E-01
3.19367E-01
4.26228E-01
5.33069E-01
6.39911E-01
7.46752E-01
8.53593E-01
9.60435E-01
1.06728E+00
1.17412E+00
1.28096E+00
1.38780E+00
1.49464E+00
1.60148E+00
1.70832E+00
T= .226 140 50 CYCLES
FIG 3.23

REFERENCES

1. Burstein, Samuel Z., Chinitz, Wallace, and Schechter, Harold, Nonlinear Combustion Instability in Liquid-Propellant Rocket Motors, Final Report, Contract No. 951946 to Jet Propulsion Laboratory, June 1969, 191 pp.

2. Bonnell, John M., An Investigation of Spherical Blast Waves and Detonation Waves in a Rocket Combustion Chamber, Jet Propulsion Laboratory, T.R. 32-1286, August 15, 1968, 21 pp.

3. Goldstine, Herman H., and VonNeumann, John, Blast Wave Calculation, Comm. Pure and Applied Math., Vol. 8, 327-354 (1955).

APPENDIX A

Computer Program Descriptions

Pancake and Annular Motors

This section describes the information necessary to prepare input data cards for programs COMB and TRDL which are used to compute numerical solutions to the pancake model and annular model of combustion in a rocket motor. The equations for these models are described in the previous sections and in Reference (1). In addition, listings of both computer programs are included. A listing os a special version of COMB that was written for a bomb blast with center at an arbitrary point in the chamber is also included.

I. Description of the Computer Programs

The basic logic of the computer programs is described in Reference (1). This section describes the changes that were made in those programs. For the pancake motor the plotting routines were removed and made into a separate program COMPLT. This was done because higher priority is given to running programs using less core memory and thus more computer runs could be made per day. It is also now possible to edit the plotted output. The program puts information to be plotted on tape 2 which is then processed by program COMPLT. The main program was converted to a subroutine COMB and a new main program COMBM is used to call this subroutine. This allows the use of I mode binary decks which are easier to handle and require less compilation time. The main program cannot be in I mode binary. The subroutine which calculates initial conditions in terms of Bessel functions BESFCT was rewritten in a much simpler form to allow changes in these conditions to be more easily made.

In both the pancake and annular programs the term ΔH_R in the energy subroutine PHIDOTV was changed from the constant 12,400 Btu/lbm to a function of temperature using the following table.

$T(^{o}R)$	ΔH_R(Btu/lbm)
536.4	8150
1800	7540
2700	6970
3600	6300
4500	5620
5800	4900
6300	4160
7200	3380
8100	2580
9000	1860
9900	1010
10800	225

The annular motor programs dimension were changed to allow for a finer mesh and better accuracy. Also, the subroutine to evaluate $F(\tilde{w})$, $G(\tilde{w})$ and $S(\tilde{w})$ were rewritten as two subroutines, VECTR and TMPVCT, respectively. In this way many quantities which were calculated separately in two or all three subroutines are now calculated only once.

II. Input Preparation

There are basically two types of input data, integer and decimal. An integer is a number without a decimal point which must be right adjusted in its field. A decimal is a number with a decimal point which may be followed by an exponent of the form E±n. The ±n represents the power of 10 by which the number is to be multiplied and n may consist of up to three digits. The + may be omitted if n is positive and the E need not appear if either + or - is present. A description of the necessary input cards for both programs follows:

Card	Columns	Type	Description
1	1-5	Integer	<0 end of input deck.
			=0 find starting conditions on tape 1.
			>0 generate starting conditions.
	6-10	Integer	=0 do not save conditions on tape 3.
			=n>0 save every nth cycle on tape 3.
2	1-40		Any alphameric information to be printed at beginning of plots.
3	1-5	Integer	Total number of mesh points on each ray including center point and reflection point at boundary or total number of z mesh points.
	6-10	Integer	Total number of rays or θ mesh points

Card	Columns	Type	Description
4	1-5	Integer	Total number of cycles for case.
	6-10	Integer	Number of cycles between edited printouts.
	11-15	Integer	Number of cycles between plots of pressure, density and vector fields.
	16-20	Integer	=0 no printout of W. ≠0 printout of entire W vector whenever edited printout appears.
	21-25	Integer	Used in annular program =0 do not smooth solution ≠0 smooth solution

The following fields on card 4 are not used in the annular program.

	21-25	Integer	=0 no plot of linearized solution. ≠0 plot of solution of linearized equations whenever plots of non-linear equation obtained.
	26-30	Integer	=n>0 plot tangential pressure distribution every n cycles. =0 no plot.
	31-35	Integer	Value of radius at which tangential pressure plot to be made. This is given by mesh point number from 2 for first row to 1 less than the number on card 3 columns 1-5 for the boundary.

Card	Columns	Type	Description
4	36-40	Integer	Number of cycles between streakline plots of 'particle' moving in r-θ plane.
	41-45	Integer	Number of cycles between points retained for streakline plot.
5	1-10	Decimal	Safety factor for time stepsize.
	11-20	Decimal	Coefficient α in equation for nozzle shape in annular program. Not used in pancake program.

The following fields are not used in the annular program.

Card	Columns	Type	Description
	21-30	Decimal	≠0- factor to control magnitude of simple forcing function. =0 use droplet evaporation forcing function.
	31-40	Decimal	r coordinate of 'particle' to be tracked.
	41-50	Decimal	θ coordinate of 'particle' to be tracked.
	51-60	Decimal	Number of droplets per unit volume
6	1-14	Decimal	Specific heat ration cp/cv = γ
	15-28	Decimal	Pressure in cylinder in psi.
	29-42	Decimal	Maximum amplitude of a pressure disturbance in psi. Not used in annular program.

Card	Columns	Type	Description
6	43-56	Decimal	Duration of initial bomb blast in millisec. Not used in annular program.
	57-66	Decimal	Chamber radius in feet.
7	1-14	Decimal	Molecular weight of fluid in cylinder.
	15-28	Decimal	Stagnation temperature. Initial density in bomb blast version.
	29-42	Decimal	Zero of derivative of Bessel function J_1 used in initialization at pancake motor. Used in annular motor as option. =0 classical solution used for initial condition. $\neq$ "pop" used for initial condition.
	43-56	Decimal	Number of droplets per unit volume. Not used in pancake motor.

In the annular program a set of cards determining the steady state solution vector V are inserted here. There is one card for each z mesh point with the four components of the vector punched on the card in four decimal fields of 15 columns each.

The bomb blast version of the pancake motor has the punched output of the similarity solution program placed here.

The entire sequence of cards may be repeated in order to run several different cases. However, no more than one case should use an input tape or an output tape.

III. Description of Output

The printed output consists first of various input and calculated values that remain constant throughout the program. Properties of the flow field are printed along the rays $\theta=0$, π, $\pi/2$ and $3\pi/2$. These are the density, radial velocity, tangential velocity, internal energy, pressure and Mach number. If printout of the entire W vector is requested these values are printed in the order of row of constant radius.

In the contour plots, where lines of constant pressure and density are given, each contour is marked by a symbol. At the right of the plot the value of the pressure or density at each contour level is printed next to the appropriate symbol. For the annular motor the vertical axis represents θ and the horizontal axis represents values of z. The plots of the moving particle are in the (x-y) coordinate system. If the plot is given is sections over the entire run, the point where a plot ended is used as the starting point of the next plot so that a continuous plot is obtained. However, since the program that does the plotting automatically sets the scale according to the range of the data to be plotted, the various sections of the plot may have different scales.

Appendix B

```
      PROGRAM COMBM(INPUT,OUTPUT,TAPE1,TAPE3,TAPE5=INPUT,TAPE6=OUTPUT,
     1TAPE2)
      COMMON PROPTY,GAMMA3,GAMMA1,GAMMA,DELR,DELO,PO,PMAX,PR,ER,TAU,IMAX
     1,JMAX,T,DTDR,DTDO,DELT,P,AO,ROO,NCYCMX,R
      DIMENSION PROPTY(12,36,4)
      CALL COMB
      STOP
      END
      SUBROUTINE COMB
C                                                                       COMB0001
C***********************************************************************COMB0002
C          TIME DEPENDENT EQUATIONS OF HYDRODYNAMICS                    COMB0003
C***********************************************************************COMB0004
C***********************************************************************COMB0005
C       SOLUTION RESTRICTED TO THE TRANSVERSE PLANE IN CYLINDRICAL      COMB0006
C       COORDINATES                                                     COMB0007
C       FINITE DIFFERENCE APPROXIMATION ... LAX WENDROFF TWO STEP       COMB0008
C       BURSTEIN VARIATION ...                                          COMB0009
C***********************************************************************COMB0010
C***********************************************************************COMB0011
C                        ***********************************            COMB0012
C                        INPUT PARAMETERS IN MAIN ROUTINE               COMB0013
C                        -------------------------------                COMB0014
C       ISTART            IF .EQ. 11111   NEW CASE                      COMB0015
C                         IF .EQ. 00000   CONTINUATION OF PREVIOUS CALCULATION COMB0016
C       IMAX              MAX NUMBER OF MESH POINTS ON A RADIUS INCLUDING ONE COMB0017
C                         POINT FOR THE BOUNDARY CONDITION              COMB0018
C       JMAX              MAX NUMBER OF RAYS                            COMB0019
C       NCYCMX            TOTAL NUMBER OF TIME STEPS FOR THE PRESENT    COMB0020
C                         INTEGRATION                                   COMB0021
C       NNPCMX            NUMBER OF TIME INTEGRATIONS BETWEEN PICTURES  COMB0022
C       FUDG              SAFETY FACTOR FOR AUGMENTATION OF LINEAR      COMB0023
C                         STABILITY ANALYSIS                            COMB0024
C                        INPUT PARAMETERS IN OTHER ROUTINES             COMB0025
C                        ----------------------------------             COMB0026
C       GAMMA             RATIO OF SPECIFIC HEATS                       COMB0027
C       PO                UNPERTURBED PRESSURE                          COMB0028
C       PMAX              AMPLITUDE OF PRESSURE PERTURBATION            COMB0029
C       TAU               DURATION OF PRESSURE PERTURBATION             COMB0030
C       RMAX              RADIUS OF CYLINDER                            COMB0031
C       XM                MOLECULAR WEIGHT OF GAS                       COMB0032
C       TO                GAS STAGNATION TEMPERATURE                    COMB0033
C***********************************************************************COMB0034
C                        ----------------------------------------       COMB0035
C                             SCALING RULES...SEE SUB. INITIL           COMB0036
C                        ----------------------------------------       COMB0037
```

```
C*****************************************************************COMB0038
C                 IN PUT TAPES                                     COMB0039
C                 ------------                                     COMB0040
C      BINARY TAPE 1      INPUT TAPE CONTAINING INFORMATION FOR THE    COMB0041
C                         CONTINUATION OF THE CALCULATION          COMB0042
C                OUT PUT TAPES                                     COMB0043
C                -------------                                     COMB0044
C      BINARY TAPE 3      OUTPUT TAPE CONTAINS INFORMATION FOR THE     COMB004
C                         CONTINUATION OF COMPUTATION              COMB0048
C*****************************************************************COMB0049
C ***************************************************************  COMB0050
C  ************************************************************    COMB0051
C    *******************************************************       COMB0052
C        ********************************************              COMB0053
C               ********************************                   COMB0054
C                             *************                        COMB0055
C                                                                  COMB0056
C                                                                  COMB005
C                                                                  COMB0058
C      MAIN                                                        COMB0059
C*********  EXECUTIVE PROGRAM WHICH CONTROLS INPUT,OUTPUT AND      COMB0060
C*********  LOGICAL FLOW FOR THE COMPUTATION   ********************COMB0061
      COMMON PROPTY,GAMMA3,GAMMA1,GAMMA,DELR,DELO,PO,PMAX,PR,ER,TAU,IMAXCOMB0062
     1,JMAX,T,DTDR,DTDO,DELT,P,AO,ROO,NCYCMX,R
      DIMENSION PROPTY(12,36,4)
      COMMON/LE/ FJMAX,IMAXM1,RAD(12),OVRLAY(12,36,12),OVRPRM(16,36,12)
      COMMON/NCPLOT/NTCYCL,ISTART,RC,OC
      COMMON/MIDDLE/VMAG,VDIR
      COMMON/PRTOPT/INTPNT
      COMMON/TLDR/TTLDP
      COMMON/EDOT/EKF
      COMMON/PHIP/DLN,EN,V,EKFCT,DMF,DMFT,ELFZ,CPR3,FPHIFS,DMPINF,DMRINF
     1,CRE,CAP
      COMMON/QQXM/XM,TO,QKR
      COMMON/JPPT/JP,PTCRD
      DIMENSION PTCRD(501,2)
      DATA ZERO,ONE/0.0,1.0/
      REWIND 2
    5 CONTINUE
      NCLCM=0
      NTCYCL=0
      NCYCLE=0
      NPICTR=0
      NSAVE=0
      NPNTC=0
      JP=1
```

```
      TTLDR=0.0
      T=0.0
      NPRPLT=1
      DLN=2.0E-4
      PHI=1.0
      V=120.0
      FS=.695652174
      READ(5,666)ISTART,ISAVE
      IF(ISTART.LT.0) GO TO 186
      READ(5,668)RAD(2),RAD(3),RAD(4)
  668 FORMAT(4A10)
      RAD(1)=10H1
      RAD(5)=0.0
      WRITE(6,668) RAD(1),RAD(2),RAD(3),RAD(4)
      RAD(1)=10H554901
      WRITE(2) RAD(1),RAD(2),RAD(3),RAD(4)
      READ(5,666)IMAX,JMAX
      FJMAX=JMAX
      IMAXM1=IMAX-1                                                     COMB0101
      READ(5,666)NCYCMX,NNPCMX,KALKOM,INTPNT,IBSPLT,IPRPLT,JPR,IPTPLT,L
  666 FORMAT(9I5)
      WRITE(6,6066)NCYCMX,NNPCMX                                        COMB0076
 6066 FORMAT(39H0 TOTAL NUMBER OF CYCLES FOR THIS CASE=I4,4X,34HNUMBER O
     2F CYCLES BETWEEN PICTURES=I4///)                                  COMB0079
      READ(5,667)FUDG,D,EKFCT,RC,OC,EN
      PTCRD(1,1)=RC
      PTCRD(1,2)=OC
  667 FORMAT(7E10.7)
C       INITIALIZE MESH AND FIND DELT FOR CONVERG. AND STABILITY        COMB0083
      CALL INITIL                                                       COMB0084
      IF(ISTART.NE.0) CALL CHARGE
      CALL CONVRG(FUDG)                                                 COMB0086
      EKFS=EKFCT*EKF/DELT
      IF(EKFCT.EQ.0.0) EKFS=1.0
      DMFT=EN*R/(ROO*AO)
      DMF=DMFT*778.0*32.17/(AO*AO)
      PHITFS=PHI*FS
      ELFZ=540.0+178.2/PHITFS
      CPR3=0.276*0.591
      FPHIFS=1.0+1.0/PHITFS
      DMPINF=PO*GAMMA*GAMMA1
      DMRINF=144.0*XM/1545.0
      CRE=DLN*AO*ROO*23.66432/2.5E-5
      CAP=(1.0/300.0)**.0108
      NPNTC=MOD(NTCYCL,IPTPLT)
  757 CONTINUE
```

```
C       PRINT INITIAL FIELD                                              COMB008
      CALL SECOND(RTM)
      CALL PRTOUT(RTM)
      IF(ISTART)70,80,70
   70 CONTINUE
      WRITE(2) ZERO,T,NTCYCL,AO,R,JPR
      WRITE(2) VMAG,VDIR,TTLDR,GAMMA,PO,ROO,IMAX,JMAX,NPRPLT,
     1(((PROPTY(I,J,K),K=1,4),J=1,JMAX),I=1,IMAX)
   80 NCYCLE=NCYCLE+1
      NTCYCL=NTCYCL+1
      NPNTC=NPNTC+1
      T=T+DELT                                                          COMB0105
      CALL MVPNT(RC,OC)
      IF(MOD(NPNTC,L).NE.0) GO TO 801
      JP=JP+1
      PTCRD(JP,1)=RC
      PTCRD(JP,2)=OC
  801 CONTINUE
      EKF=EKFS
      CALL FRSTRO
      DO 500 I=2,IMAX
      DO 500 J=1,JMAX
      CALL VECTFR(I,J,OVRLAY(1,J,I))
      CALL VECTRG(I,J,OVRLAY(5,J,I))
      CALL VECTRS(I,J,OVRLAY(9,J,I))
  500 CONTINUE
C     COMPUTE W(T+DELT) FOR FIRST ROW
      DO 501 I=2,IMAXM1
      RD=RAD(I)+0.5*DELR
      I1=I+1
      DO 501 J=1,JMAX
      CALL QUAD1(I,J,OVRPRM(1,J,I1),I)
      CALL TEMPFW(OVRPRM(1,J,I1),OVRPRM(5,J,I1),I,J,1)
      CALL TEMPGW(OVRPRM(1,J,I1),OVRPRM(9,J,I1))
      CALL TEMPSW(OVRPRM(1,J,I1),OVRPRM(13,J,I1))
      OVRPRM(13,J,I1)=RD*OVRPRM(13,J,I1)
      OVRPRM(16,J,I1)=RD*OVRPRM(16,J,I1)
  501 CONTINUE
C  COMPUTE W(T+DELT)AT INTERIOR POINTS
      CALL GENPT(3,IMAXM1)
C       COMPLETE MESH CALC AT I=1,J=1                                   COMB0150
      CALL CENTER
      CALL CHARGE                                                       COMB0162
      NCLCM=NCLCM+1
      NPICTR=NPICTR+1
      NSAVE=NSAVE+1
```

```
C       NOISE SUPPRESSOP                                                    COMB0163
      DO 50 I=1,IMAX                                                        COMB0164
      DO 50 J=1,JMAX                                                        COMB0165
      IF(I.EQ.1.AND.J.GT.1) GO TO 50                                        COMB0166
C       TEST FOR NEGATIVE DENSITY                                           COMB016
      IF(PROPTY(I,J,1).LE.0.) GO TO 181                                     COMB0168
      DO 51 K=2,3                                                           COMB0169
C       ELIMINATE SPURIOUS VELOCITIES                                       COMB0170
      IF(ABS (PROPTY(I,J,K)).LE.1.E-14) PROPTY(I,J,K)=0.
   51 CONTINUE                                                              COMB0172
   50 CONTINUE
C       FIND NEW STABILITY NUMBER                                           COMB0158
      CALL CONVRG(FUDG)                                                     COMB0159
   81 CONTINUE
      IF(NPNTC-IPTPLT)820,810,810
  810 CONTINUE
      WRITE(2) ONE,T,NTCYCL,AO,R,JP
      WRITE(2)(PTCRD(I,1),PTCRD(I,2),I=1,JP)
      NPNTC=0
      JP=1
      PTCRD(1,1)=RC
      PTCRD(1,2)=OC
  820 CONTINUE
      IF(NCLCM-KALKOM)89,88,88
   88 CONTINUE
      WRITE(6,606)T,NTCYCL
  606 FORMAT(3X11HTOTAL TIME=E14.7,3X17HNUMBER OF CYCLES=I4)                COMB0186
      NCLCM=0
      IF(IPRPLT.NE.0) NPRPLT=MOD(NCYCLE,IPRPLT)
      WRITE(2) ZERO,T,NTCYCL,AO,R,JPR
      WRITE(2) VMAG,VDIR,TTLDR,GAMMA,PO,ROO,IMAX,JMAX,NPRPLT,
     1(((PROPTY(I,J,K),K=1,4),J=1,JMAX),I=1,IMAX)
   89 CONTINUE
C       TEST TO SEE IF FINISHED                                             COMB0174
      IF(NCYCLE-NCYCMX)890,91,91
  890 CONTINUE
C       TEST TO SEE IF PRINT TIME AFFIRMATIVE                               COMB0176
      IF(NPICTR-NNPCMX)901,90,90
   90 CONTINUE
C       PRINT DISPLAY OF FLOW FIELD ALONG THE RAYS 0,PI/2,PI,3*PI/2         COMB0180
      CALL SECOND(RTM)
      CALL PRTOUT(RTM)
C       RESTART PRINT CYCLE                                                 COMB018
      NPICTR=0
  901 IF(ISAVE.EQ.0.OR.NSAVE.LT.ISAVE) GO TO 80
  902 CONTINUE
```

```
      REWIND 3
      WRITE(3)(((PROPTY(I,J,K),K=1,4),J=1,JMAX),I=1,IMAX)
      WRITE(3)T,NTCYCL,VMAG,VDIR,TTLDR,RC,OC
      WRITE(3) JP,(PTCRD(I,1),PTCRD(I,2),I=1,JP)
      NSAVE=0
      GO TO 80                                                          COMB0189
   91 CONTINUE
      CALL SECOND(RTM)
      CALL PRTOUT(RTM)
      IF(ISAVE)182,5,182
  181 WRITE(6,6666)                                                     COMB019
 6666 FORMAT(1H1,20X,63H********** ERROR ABORT  -  NEG. DENSITY IN THE FCOMB0198
     1IELD **********/21X,63H*******************************************COMB0199
     2********************)                                             COMB0200
      WRITE(6,707)(I,J,(PROPTY(I,J,K),K=1,4),I=1,IMAX)
  707 FORMAT(1X,2I4,4E23.7)
      GO TO 184                                                         COMB0201
  182 CONTINUE
      REWIND 3
      WRITE(3)    (((PROPTY(I,J,K),K=1,4),J=1,JMAX),I=1,IMAX)
      WRITE(3)T,NTCYCL,VMAG,VDIR,TTLDR,RC,OC
      WRITE(3) JP,(PTCRD(I,1),PTCRD(I,2),I=1,JP)
  184 GO TO 5
  186 CONTINUE
      ENDFILE 2
      END FILE 3
      STOP
      END                                                               COMB0209
      SUBROUTINE INITIL                                                 COMB0465
C********* SETS UP THE INITIAL DATA IN THE ARRAY PROPTY AFTER           COMB0466
C*********  COMPLETING THE NORMALIZATION FROM INPUT DATA  **************COMB046
C       ********,*******************************************************COMB0468
C       ****************************************************************COMB0469
C       ****************************************************************COMB0470
C       * * * * * * * * * * * * SCALING RULES * * * * * * * * * * * * * *COMB0471
C        * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * COMB0472
C       EQ1*(TO/RHOO),,EQ2*(TO/(RHOO*AO)),,EQ3*(TO/(RHOO*AO)),,         COMB0473
C       EQ4*(TO/(RHOO*AO**2)),,,,TO=(RMAX/AO)                           COMB0474
C        * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * COMB0475
C       **********   TIME IS SCALED BY ...THE REF TIME SCALE   **********COMB0476
C       **********   DISTANCE IS SCALED BY...COMBUSTION CHAMBER RADIUS***COMB047
C       **********   VELOCITY IS SCALED BY...REF SOUND SPEED   **********COMB0478
C       **********   DENSITY IS SCALED BY...REFERENCE DENSITY************COMB0479
C       **********   PRESSURE IS SCALED BY...REF SOUND SPEED**2*REF     COMB0480
C       **********   DENSITY*****************************************COMB0481
C       ****************************************************************COMB0482
```

```
C      ****************************************************************COMB0483
C      ****************************************************************COMB0484
      COMMON PROPTY,GAMMA3,GAMMA1,GAMMA,DELR,DELO,PO,PMAX,PR,ER,TAU,IMAXCOMB0485
     1,JMAX,T,DTDR,DTDO,DT,P,AO,ROO,NCYCMX,RMAX
      DIMENSION PROPTY(12,36,4)
      COMMON/LE/ FJMAX,IMAXM1,RAD(12),OVRLAY(12,36,12),OVRPRM(16,36,12)
      COMMON/CNTRE/COSTH(36),SINTH(36),THTA(36)
      COMMON/MIDDLE/VMAG,VDIR
      COMMON/NCPLOT/NTCYCL,ISTART,RC,OC
      COMMON/EDOT/EKF
      COMMON/TLDR/TTLDR
      COMMON/QQXM/XM,TO,QKR
      COMMON/JPPT/JP,PTCRD
      DIMENSION PTCRD(501,2)
      DATA PI,TWOPI/3.14159265,6.28318531/
      READ(5,5)          GAMMA,PO,PMAX,TAU,RMAX                         COMB0490
      WRITE(6,65)                                                       COMB0491
   65 FORMAT(4X,5HGAMMA,5X,16HCHAMBER PRESSURE,2X,14HPULSE PRESSURE,4X,1COMB0492
     1                                                               5H COMB0493
     2PULSE DURATION,4X,                                                COMB0494
     3              14HCHAMBER RADIUS/20X,3HPSI,15X,3HPSI,15X,3HSEC,15XCOMB0495
     2,2HFT//)                                                          COMB0496
C       * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *COMB049
C       * * * * * *  BOMB PULSE PRESSURE AND TIME DURATION ARE ONLY     COMB0498
C       * * * * * *  TO BE REFERENCED WHEN INITIAL CONDITIONS ARE       COMB0499
C       * * * * * *  FOR A PULSE TYPE CALCULATION  * * * * * * * * * * COMB0500
C       * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *COMB0501
      WRITE(6,4)          GAMMA,PO,PMAX,TAU,RMAX                        COMB0502
    4 FORMAT(1X,2E14.7,E18.7,E19.7,F12.4///)                            COMB0503
      READ(5,5) XM,TO,QKR
      WRITE(6,68)                                                       COMB0505
   68 FORMAT(3X,10HMOL WEIGHT,3X,19HCHAMBER TEMPERATURE//)              COMB0506
      WRITE(6,5)XM,TO                                                   COMB050
      WRITE(6,76)                                                       COMB0508
   76 FORMAT(1H ///)                                                    COMB0509
C       GAMMA  SPECIFIC HEAT RATIO                                      COMB0510
C       PO  UNPERTURBED PRESSURE                                        COMB0511
C       PMAX  AMPLITUDE OF PRESSURE PERTURBATION                        COMB0512
C       TAU IS DURATION OF PERT.                                        COMB0513
C       IMAX,JMAX MESH DIMENSIONS
C       XM  MOLECULAR WEIGHT, TO  CHAMBER TEMPERATURE                   COMB0515
    7 FORMAT(2I4)                                                       COMB0516
    5 FORMAT(4E14.7,2F10.4)                                             COMB051
      PO=PO                                                             COMB0519
      ROO=(PO*144.)/((1545./XM)*TO)                                     COMB0520
      AO=SQRT (GAMMA*32.17*PO*144./ROO)                                 COMB0521
```

```
C        RHO=RO/ROO=1.                                                  COMB0522
        RHO=1.                                                          COMB0523
        TO=RMAX/AO                                                      COMB0524
        XTO=TO*1000.                                                    COMB0525
        EQ2=ROO*AO/TO                                                   COMB0526
        P=PO*144.*32.17/(RMAX*EQ2)                                      COMB052
        DELR=1.0/FLOAT(IMAX-2)
        DELO=2.*PI/FJMAX
        GAMMA3=GAMMA-3.                                                 COMB0530
        GAMMA1=GAMMA-1.                                                 COMB0531
        PR=PMAX/PO                                                      COMB0532
        ER=PR/GAMMA1                                                    COMB0533
        WRITE(6,67)                                                     COMB0534
   67 FORMAT(1X,16HREDUCED PRESSURE,2X,15HREF SOUND SPEED,2X,11HREF DENSCOMB0535
     1ITY,6X,7HDELTA R,7X,11HDELTA THETA,4X,43HPRESSURE AND INTERNAL ENECOMB0536
     2RGY RATIO OF PULSE//)                                             COMB053
        WRITE(6,6)          P,     AO,ROO,DELR,DELO,PR,ER,XTO           COMB0538
    6 FORMAT(7E16.7///16H REF TIME SCALE=F15.5/3X,8HMILLISEC///)        COMB0539
        WRITE(6,566)                                                    COMB0540
  566 FORMAT(1X,117H************************************** END OF INITCOMB0541
     1ILIZATION PHASE ******************************************/COMB0542
     2/////)                                                            COMB0543
        ELN=TWOPI/FJMAX
        RAD(1)=0.0
        DO 501 I=2,IMAX
  501   RAD(I)=RAD(I-1)+DELR
        DO 510 J=1,JMAX
        TH=J-1
        TH=TH*DELO
        THTA(J)=TH
        COSTH(J)=COS(TH)
  510   SINTH(J)=SIN(TH)
        EKF=P/GAMMA1
        IF(ISTART)506,505,506
  505   CONTINUE
        READ(1)      (((PROPTY(I,J,K),K=1,4),J=1,JMAX),I=1,IMAX)
        READ(1) T,NTCYCL,VMAG,VDIR,TTLDR,RC,OC
        READ(1)  JP,(PTCRD(I,1),PTCRD(I,2),I=1,JP)
        RETURN
  506   CONTINUE
        CALL BESFCT
  504   RETURN                                                          COMB0562
        END
        SUBROUTINE BESFCT
        COMMON PROPTY,GAMMA3,GAMMA1,GAMMA,DELR,DELO,PO,PMAX,PR,ER,TAU,IMAX
     1,JMAX,T,DTDR,DTDO,DELT,P,AO,ROO,NCYCMX,R
```

```
      DIMENSION PROPTY(12,36,4)
      COMMON/LE/ JMAX,IMAXM1,RAD(12),OVRLAY(12,36,12),OVRPRM(16,36,12)
      COMMON/MIDDLE/VMAG,VDIR
      COMMON/QQXM/XM,T0,QKR
      COMMON/CNTRE/COSTH(36),SINTH(36),THTA(36)
      DIMENSION NBSFN(3),BSFN(3)
      NBSFN(1)=1
      AA=1.84118378
      CALL BESSEL(1,AA,1,NBSFN,BSFN)
      EPS=(PMAX-PO)/(PO*GAMMA*BSFN(1))
      HEPS=0.5*EPS
      VMAG=ABS(HEPS)
      VDIR=1.57079633
      GAMINV=1.0/GAMMA
      PDM=GAMINV/GAMMA1
      PROPTY(1,1,1)=1.0
      PROPTY(1,1,2)=0.0
      PROPTY(1,1,3)=VMAG
      PROPTY(1,1,4)=PDM+0.5*VMAG*VMAG
      NBSFN(1)=2
      NBSFN(2)=1
      NBSFN(3)=0
      DO 20 I=2,IMAXM1
      BB=QKR*RAD(I)
      CALL BESSEL(2,BB,3,NBSFN,BSFN)
      PP=EPS*BSFN(2)
      VV=PP/BB
      PP=GAMMA*PP
      UU=HEPS*(BSFN(3)-BSFN(1))
      DO 20 J=1,JMAX
      PRS=1.0-PP*COSTH(J)
      PROPTY(I,J,1)=PRS**GAMINV
      PROPTY(I,J,2)=PROPTY(I,J,1)*UU*SINTH(J)
      PROPTY(I,J,3)=PROPTY(I,J,1)*VV*COSTH(J)
      PROPTY(I,J,4)=PDM*PRS+0.5*(PROPTY(I,J,2)*PROPTY(I,J,2)+
     1PROPTY(I,J,3)*PROPTY(I,J,3))/PROPTY(I,J,1)
   20 CONTINUE
      RETURN
      END
      SUBROUTINE BESSEL(N,Z,NANS,NBSFN,BSFN)
      DIMENSION NBSFN (1),F(200),X(200),P(200),BSFN(1)
      COMMON/LE/QQ(12110)
      EQUIVALENCE (F,QQ(600)),(X,QQ(800)),(P,QQ(1000))
      Y=Z/2.
      IF (Y)11,271,11
  271 IF (N)273,272,273
```

```
  272 BSFN(1)=1.
      GO TO 274
  273 BSFN(1)=0.
  274 IF(NANS-1)90,90,275
  275 DO 181 I=2,NANS
      J=NBSFN(I)
      IF(J)276,277,276
  276 BSFN(I)=0.
      GO TO 181
  277 BSFN(I)=1.
  181 CONTINUE
      GO TO 90
   11 M=.75*ABS (Z)+6.
      NABS=IABS (N)
      NN=MAX0(NABS,2*M)
      DO 10 I=1,NN
   10 X(I)=Y/FLOAT (I)
      A0=0.
      LL=NN
      A1=Y*Y/FLOAT (LL*LL+LL)
      IF(A1-1.E-08)1,2,2
    1 F(NN)=A1
      GO TO 30
    2 B1=1.
      LL=LL+1
      C=Y*Y/FLOAT(LL*LL+LL)
      B2 = B1+C
      FRAC=ABS (C/B2)*A1
      IF(FRAC-1.E-08)1,3,3
    3 LL=LL+1
    6 C=Y*Y/FLOAT (LL*LL+LL)
      A2=A1+C*A0
      AB=C*B1
      B0=B1
      B1=B2
      B2=B1+C*B0
      FRAC=ABS (AB/B2)*FRAC
      IF(FRAC-1.E-08)4,5,5
    4 F(NN)=A2/B1
      GO TO 30
    5 LL=LL+1
      A0=A1
      A1=A2
      GO TO 6
   30 NNM=NN-1
      DO 40 NTIMES=1,NNM
```

```
      KK=NN-NTIMES
   40 F(KK)=X(KK)*X(KK+1)/(1.-F(KK+1))
      IF(NABS)34,34,35
   34 PZERO=1.
      P(1)=PZERO*X(1)/(1.-F(1))
      NP=2
      GO TO 59
   35 P(NABS)=1.
      NM=NABS-1
      DO 50 NTIMES=1, NM
      KK=NABS-NTIMES
   50 P(KK)=P(KK+1)*(1.-F(KK+1))/X(KK+1)
      PZERO=P(1)*(1.-F(1))/X(1)
   51 NP=NABS+1
   59 DO 60 KK=NP,NN
   60 P(KK)=P(KK-1)*X(KK)/(1.-F(KK))
      SUM=0.
      DO 70 I=2,NN,2
   70 SUM=P(I)+SUM
   71 BSFN(1)=1./(2.*SUM+PZERO)
   73 IF(NANS-1)90,90,75
   75 DO 81 I=2,NANS
      J=IABS (NBSFN(I))
      IF(J) 80,76,80
   76 BSFN(I)=PZERO*BSFN(1)
      GO TO 81
   80 BSFN(I)=P(J)*BSFN(1)
   81 CONTINUE
      DO 100 I=1,NANS
      IF (NBSFN(I))101,100,100
  101 IF (MOD(NBSFN(I),2))102,100,102
  102 BSFN(I)=-BSFN(I)
  100 CONTINUE
   90 RETURN
      END
      SUBROUTINE CONVRG(FACTOR)                                         COMB0355
C*********  COMPUTES A TIME INCREMENT TO BE USED IN DIFFERENCE EQUATION COMB0356
C*********  BASED ON C.F.L. CONDITION          **********************COMB035
      COMMON PROPTY,GAMMA3,GAMMA1,GAMMA,DELR,DELO,PO,PMAX,PR,ER,TAU,IMAXCOMB0358
     1,JMAX,T,DTDR,DTDO,DT,P,AO,ROO,NCYCMX,R
      DIMENSION PROPTY(12,36,4)
      COMMON/LE/ FJMAX,IMAXM1,RAD(12),OVRLAY(12,36,12),OVRPRM(16,36,12)
      DT=0.3/(16.*.75/3.9445)                                           COMB0364
      DT=0.6/(16.*.75/3.9445)                                           COMB0365
      DT=DT*FACTOR                                                      COMB0366
      SQ=1./SQRT (2.)                                                   COMB036
```

```
      DO 1 I=2,IMAXM1
      DO 1 J=1,JMAX                                                     COMB0369
      IF(I.EQ.1.AND.J.GE.2)GO TO 1                                      COMB0370
      VELSQR=(PROPTY(I,J,2)**2+PROPTY(I,J,3)**2)/PROPTY(I,J,1)**2       COMB0371
      IF(VELSQR) 8,8,2                                                  COMB0372
    2 BG=PROPTY(I,J,4)/PROPTY(I,J,1)-0.5*VELSQR                         COMB0373
      IF(BG) 8,7,7                                                      COMB0374
    7 DENOM=SQRT (GAMMA*GAMMA1*BG)                                      COMB0375
      TERM=SQ/(DENOM+SQRT (VELSQR))
      TERMO=TERM*FACTOR*DELO *RAD(I)
      TERMR=TERM*FACTOR*DELR
      DTE=DT
      DT=AMIN1(DT,TERMR,TERMO)                                          COMB0380
      IF(DT-DTE) 150,8,8
  150 IT=I                                                              COMB0383
      JT=J                                                              COMB0384
    8 CONTINUE                                                          COMB0385
    1 CONTINUE                                                          COMB0386
      DTDR=DT/DELR                                                      COMB038
      DTDO=DT/DELO                                                      COMB0388
      RETURN                                                            COMB0392
      END                                                               COMB0393
      SUBROUTINE MVPNT(RC,OC)
      COMMON PROPTY,GAMMA3,GAMMA1,GAMMA,DELR,DELO,PO,PMAX,PR,ER,TAU,IMAX
     1,JMAX,T,DTDR,DTDO,DT,P,AO,ROO   ,NCY
      DIMENSION PROPTY(12,36,4)
      COMMON/LE/ FJMAX,IMAXM1,RAD(12),OVRLAY(12,36,12),OVRPRM(16,36,12)
      COMMON/CNTRE/COSTH(36),SINTH(36),THTA(36)
      DATA PI,TWOPI/3.14159265,6.28318531/
      I=INT(RC/DELR)+1
      IF(I.EQ.1) I=2
      J=INT(OC/DELO)+1
      IF(J.GT.JMAX) J=JMAX
      JP1=J+1
      IF(JP1.GT.JMAX) JP1=1
      U1=PROPTY(I,J,2)/PROPTY(I,J,1)
      U2=PROPTY(I+1,J,2)/PROPTY(I+1,J,1)
      U3=PROPTY(I,JP1,2)/PROPTY(I,JP1,1)
      U4=PROPTY(I+1,JP1,2)/PROPTY(I+1,JP1,1)
      V1=PROPTY(I,J,3)/PROPTY(I,J,1)
      V2=PROPTY(I+1,J,3)/PROPTY(I+1,J,1)
      V3=PROPTY(I,JP1,3)/PROPTY(I,JP1,1)
      V4=PROPTY(I+1,JP1,3)/PROPTY(I+1,JP1,1)
      A=(RC-RAD(I))/DELR
      B=(OC-THTA(J))/DELO
      C=A*B
```

```
      U=U1+A*(U2-U1)+B*(U3-U1)+C* (U4-U3-U2+U1)
      V=V1+A*(V2-V1)+B*(V3-V1)+C*(V4-V3-V2+V1)
      OC=OC+V*DT/RC
      RC=RC+U*DT
      IF(RC.GT. 1. ) RC=1.
      IF(RC.GE.0.) GO TO 10
      RC=-RC
      OC=OC+PI
   10 CONTINUE
      IF(OC.LT.0.0) OC=OC+TWOPI
      IF(OC.GE.TWOPI) OC=OC-TWOPI
      RETURN
      END
      SUBROUTINE FRSTRO
C*********  COMPUTES VALUE OF W(T+DELT) ON THE CIRCLE R=DELR           COMB0781
C*********  THE FIRST ROW OF NET POINTS CONTIGUOUS TO THE CENTER  ******COMB0782
      COMMON PROPTY,GAMMA3,GAMMA1,GAMMA,DELR,DELO,PO,PMAX,PR,ER,TAU,IMAXCOMB0783
     1,JMAX,T,DTDR,DTDO,DT,P,AO,ROO,NCYCMX,R
      DIMENSION PROPTY(12,36,4)
      COMMON/LE/ FJMAX,IMAXM1,RAD(12),OVRLAY(12,36,12),OVRPRM(16,36,12)
      COMMON/CNTRE/COSTH(36),SINTH(36),THTA(36)
      COMMON/MIDDLE/VMAG,VDIR
      RHO=PROPTY(1,1,1)
      E=PROPTY(1,1,4)
      RHOU=RHO*VMAG
      DO 20 J=1,JMAX
      DO 15 I=1,3
      DO 10 K=1,4
   10 OVRPRM(K,J,I+3)=PROPTY(I,J,K)
   15 CONTINUE
      TH=VDIR-THTA(J)
      PROPTY(1,J,1)=RHO
      PROPTY(1,J,2)=RHOU*COS(TH)
      PROPTY(1,J,3)=RHOU*SIN(TH)
      PROPTY(1,J,4)=E
      DO 20 K=1,4
      PROPTY(1,J,K)=0.5*(PROPTY(1,J,K)+PROPTY(2,J,K))
   20 PROPTY(3,J,K)=0.5*(PROPTY(2,J,K)+PROPTY(3,J,K))
      DR=DELR
      DTR=DTDR
      DELR=0.5*DELR
      DTDR=2.0*DTDR
      RAD(1)=DELR
      RAD(2)=DR
      RAD(3)=DELR+DR
      DO 500 I=1,3
```

```
      DO 500 J=1,JMAX
      CALL VECTFR(I,J,OVRLAY(1,J,I))
      CALL VECTRG(I,J,OVRLAY(5,J,I))
      CALL VECTRS(I,J,OVRLAY(9,J,I))
  500 CONTINUE
      DO 501 I=1,2
      RD=RAD(I)+0.5*DELR
      I1=I+1
      DO 501 J=1,JMAX
      CALL QUAD1(I,J,OVRPRM(1,J,I1),I)
      CALL TEMPFW(OVRPRM(1,J,I1),OVRPRM(5,J,I1),I1,J,1)
      CALL TEMPGW(OVRPRM(1,J,I1),OVRPRM(9,J,I1))
      CALL TEMPSW(OVRPRM(1,J,I1),OVRPRM(13,J,I1))
      OVRPRM(13,J,I1)=RD*OVPPRM(13,J,I1)
      OVRPRM(16,J,I1)=RD*OVRPRM(16,J,I1)
  501 CONTINUE
      CALL GENPT(2,2)
      DELR=DR
      DTDR=DTR
      RAD(1)=0.0
      RAD(2)=DELR
      RAD(3)=DELR+DELR
      DO 40 J=1,JMAX
      DO 40 K=1,4
      OVRPRM(K,J,1)=PROPTY(2,J,K)
      DO 40 I=1,3
      PROPTY(I,J,K)=OVRPRM(K,J,I+3)
   40 CONTINUE
      RETURN
      END
      SUBROUTINE VECTFR(II,JJ,R)                                        COMB0395
C*********  COMPUTES THE VALUE OF THE VECTOR FUNCTUON F(W) WHERE W     COMB0396
C*********  IS THE VECTOR OF CONSERVATION VARIABLES  *******************COMB039
      COMMON PROPTY,GAMMA3,GAMMA1,GAMMA,DELR,DELO,PO,PMAX,PR,ER,TAU,IMAXCOMB0398
     1,JMAX,T,DTDR,DTDO,DT,P,AO,ROO,NCYCMX,RR
      DIMENSION PROPTY(12,36,4)
      COMMON/LE/ FJMAX,IMAXM1,RAD(12),OVRLAY(12,36,12),OVRPRM(16,36,12)
      DIMENSION R(4)                                                    COMB0404
      I=II                                                              COMB0405
      J=JJ                                                              COMB0406
      IF(J.LE.JMAX) GO TO 7                                             COMB040
      J=1                                                               COMB0408
    7 IF(J.EQ.0) J=JMAX                                                 COMB0409
      X=PROPTY(I,J,2)/PROPTY(I,J,1)                                     COMB0411
      Y=PROPTY(I,J,3)/PROPTY(I,J,1)                                     COMB0412
C      Z IS DISTANCE FROM ORIGIN                                        COMB0413
```

```
      Z=RAD(I)
      R(1)=-PROPTY(I,J,2)*Z                                             COMB0416
      R(2)=PROPTY(I,J,2)*GAMMA3*.5*X-GAMMA1*(PROPTY(I,J,4)-.5*Y*PROPTY  COMB041
     1(I,J,3))                                                          COMB0418
      R(2)=R(2)*Z                                                       COMB0419
      R(3)=-X*PROPTY(I,J,3)*Z                                           COMB0420
      R(4)=-(X*PROPTY(I,J,4)*GAMMA-GAMMA1*0.5*PROPTY(I,J,2)*(X**2+Y**2))COMB0421
     1*Z                                                                COMB0422
      RETURN                                                            COMB0423
      END                                                               COMB0424
      SUBROUTINE VECTRG(II,JJ,R)                                        COMB0426
C*********  COMPUTES THE VALUE OF THE VECTOR FUNCTUON G(W) WHERE W      COMB042
C*********  IS THE VECTOR OF CONSERVATION VARIABLES ******************COMB0428
      COMMON PROPTY,GAMMA3,GAMMA1,GAMMA,DELR,DELO,PO,PMAX,PR,ER,TAU,IMAXCOMB0429
     1,JMAX,T,DTDR,DTDO,DT,P,AO,ROO,NCYCMX,RR
      DIMENSION PROPTY(12,36,4)
      COMMON/LE/ FJMAX,IMAXM1,RAD(12),OVRLAY(12,36,12),OVRPRM(16,36,12)
      DIMENSION R(4)                                                    COMB0435
      I=II                                                              COMB0436
      J=JJ                                                              COMB043
      IF(J.LE.JMAX) GO TO 19                                            COMB0438
      J=1                                                               COMB0439
   19 IF(J.EQ.0) J=JMAX                                                 COMB0440
    9 X=PROPTY(I,J,3)/PROPTY(I,J,1)                                     COMB0443
    2 Y=PROPTY(I,J,2)/PROPTY(I,J,1)                                     COMB0446
    5 R(1)=-PROPTY(I,J,3)                                               COMB0458
      R(2)=R(1)*Y                                                       COMB0459
      R(3)=-0.5*GAMMA3*R(1)*X-GAMMA1*(PROPTY(I,J,4)-0.5*Y*PROPTY(I,J,2))COMB0460
      R(4)=-(GAMMA*X*PROPTY(I,J,4)-GAMMA1*0.5*PROPTY(I,J,3)*(X**2+Y**2))COMB0461
      RETURN                                                            COMB0462
      END                                                               COMB0463
      SUBROUTINE VECTRS(II,JJ,S)                                        COMB0583
C*********  COMPUTES THE VALUE OF THE VECTOR FUNCTUON S(W) WHERE W      COMB0584
C*********  IS THE VECTOR OF CONSERVATION VARIABLES ******************COMB0585
      COMMON PROPTY,GAMMA3,GAMMA1,GAMMA,DELR,DELO,PO,PMAX,PR,ER,TAU,IMAXCOMB0586
     1,JMAX,T,DTDR,DTDO,DT,P,AO,ROO,NCYCMX,R
      DIMENSION PROPTY(12,36,4)
      COMMON/LE/ FJMAX,IMAXM1,RAD(12),OVRLAY(12,36,12),OVRPRM(16,36,12)
      COMMON/EDOT/EKF
      COMMON/NCPLOT/NTCYCL,ISTART
      DIMENSION S(4),W(4)
      I=II                                                              COMB0593
      J=JJ                                                              COMB0594
      IF(J.LE.JMAX) GO TO 7                                             COMB0595
      J=1                                                               COMB0596
    7 IF(J.EQ.0) J=JMAX                                                 COMB059
```

```
      DO 10 K=1,4
      W(K)=PROPTY(I,J,K)
   10 CONTINUE
      X=PROPTY(I,J,3)/PROPTY(I,J,1)                                     COMB0598
      Y=PROPTY(I,J,2)/PROPTY(I,J,1)                                     COMB0599
      PRS=GAMMA1*(PROPTY(I,J,4)-0.5*PROPTY(I,J,1)*(Y**2+X**2))
      IF(PRS.LT.0.0) GO TO 50
      CALL PHIDOTV(W,S)
      S(1)=RAD(I)*S(1)*EKF
      S(2)=PROPTY(I,J,1)*X**2+PRS
      S(3)=-PROPTY(I,J,1)*X*Y                                           COMB0603
      S(4)=RAD(I)*S(4)*EKF
      RETURN                                                            COMB0605
   50 CONTINUE
      WRITE(6,707)((M,L,(PROPTY(L,M,K),K=1,4 ),M=1,JMAX),L=1,IMAX)
     2,J,I,(PROPTY(I,J,K),K=1,4),NTCYCL
  707 FORMAT(1H0//(2I4,4E23.7))
      STOP
      END                                                               COMB0606
      SUBROUTINE QUAD1(II,JJ,TEMPW1,LL)                                 COMB0650
C*********  COMPUTES W..SQUIGGLE(T+DELT), THE TEMPORARY VALUE          COMB0651
C*********  OF THE CONSERVATION VECTOR   *****************************COMB0652
      COMMON PROPTY,GAMMA3,GAMMA1,GAMMA,DELR,DELO,PO,PMAX,PR,ER,TAU,IMAXCOMB0653
     1,JMAX,T,DTDR,DTDO,DT,P,AO,ROO,NCYCMX,R
      DIMENSION PROPTY(12,36,4)
      COMMON/LE/ FJMAX,IMAXM1,RAD(12),OVRLAY(12,36,12),OVRPRM(16,36,12)
      DIMENSION AVEW(4),TEMPW1(4),AVES(4)
      L=LL                                                              COMB0660
      I=II                                                              COMB0661
      J=JJ                                                              COMB0662
      IF(L.EQ.IMAXM1) GO TO 10
      HDELR=0.5*DELR
      IF(J.LE.JMAX) GO TO 7                                             COMB0663
      J=1                                                               COMB0664
    7 IF(J.EQ.0) J=JMAX                                                 COMB0665
      JP1=J+1                                                           COMB0666
      IF(JP1.LE.JMAX) GO TO 8                                           COMB066
      JP1=1                                                             COMB0668
    8 IF(JP1.EQ.0)JP1=JMAX                                              COMB0669
      DO 1 K=1,4                                                        COMB0670
      AVES(K)=0.0
      IF(K.EQ.2.OR.K.EQ.3) AVES(K)=0.25*(OVRLAY(K+8,J,L)+OVRLAY(K+8,JP1,
     1L)+OVRLAY(K+8,JP1,L+1)+OVPLAY(K+8,J,L+1))*DT
    1 AVEW(K)=0.25*(PROPTY(I,J,K)+PROPTY(I+1,J,K)+PROPTY(I+1,JP1,K)+PROPCOMB0671
     XTY(I,JP1,K))                                                      COMB0672
      DO 2 K=1,4                                                        COMB0673
```

```
  2 TEMPW1(K)=AVEW(K)+((DTDR*(OVRLAY(K,J,L+1)-OVRLAY(K,J,L)+OVRLAY(K, COMB0674
   1JP1,L+1)-OVRLAY(K,JP1,L))      +DTDO*(OVRLAY(K+4,JP1,L)-OVRLAY(K+4,JCOMB0675
   2,L)+OVRLAY(K+4,JP1,L+1)-OVRLAY(K+4,J,L+1)))*.5+AVES(K))/(RAD(I)+
   3HDELR)
    GO TO 13
 10 RADE=RAD(I)+DELR/2.
    RADI=RADE-DELR
    TEMPW1(1)=OVRPRM(1,J,I)*RADI/RADE
    TEMPW1(2)=-TEMPW1(1)*(OVRPRM(2,J,I)/OVRPRM(1,J,I))
    TEMPW1(3)=OVRPRM(3,J,I)
    EMSQ=OVRPRM(2,J,I)**2
    ENSQ=OVRPRM(3,J,I)**2
    EMNR=(EMSQ+ENSQ)/OVRPRM(1,J,I)
    PINTS=GAMMA1*(OVRPRM(4,J,I)-.5*EMNR)
    PEXTS=PINTS+DELR/RADI*ENSQ/OVRPRM(1,J,I)
    TEMPW1(4)=PEXTS/GAMMA1+.5*(TEMPW1(2)**2+TEMPW1(3)**2)/TEMPW1(1)
 13 RETURN
    END                                                                 COMB0680
    SUBROUTINE TEMPFW(WTEMP,FOFW,II,JJ,IINDEX)
C*********  COMPUTES F(W..SQUIGGLE) WHERE W..SQUIGGLE IS THE TEMPORARY  COMB0609
C*********  VALUE OF THE CONSERVATION VECTOR      **********************COMB0610
    COMMON PROPTY,GAMMA3,GAMMA1,GAMMA,DELR,DELO,PO,PMAX,PR,ER,TAU,IMAXCOMB0611
   1,JMAX,T,DTDR,DTDO,DT,P,AO,ROO,NCYCMX,R
    DIMENSION PROPTY(12,36,4)
    COMMON/LE/ FJMAX,IMAXM1,RAD(12),OVRLAY(12,36,12),OVRPRM(16,36,12)
    DIMENSION WTEMP(4),FOFW(4)                                          COMB061
    JJJ=II-1
    Z=FLOAT(JJJ)+FLOAT(IINDEX)/2.
    Z=Z*DELR
    X=WTEMP(2)/WTEMP(1)                                                 COMB0621
    Y=WTEMP(3)/WTEMP(1)                                                 COMB0622
    FOFW(1)=-WTEMP(2)*Z                                                 COMB0623
    FOFW(2)=WTEMP(2)*GAMMA3*0.5*X-GAMMA1*(WTEMP(4)-0.5*Y*WTEMP(3))      COMB0624
    FOFW(2)=FOFW(2)*Z                                                   COMB0625
    FOFW(3)=-X*WTEMP(3)*Z                                               COMB0626
    FOFW(4)=-(X*WTEMP(4)*GAMMA-GAMMA1*0.5*WTEMP(2)*(X**2+Y**2)).*Z      COMB062
    RETURN                                                              COMB0628
    END                                                                 COMB0629
    SUBROUTINE TEMPGW(WTEMP,GOFW)                                       COMB0631
C*********  COMPUTES G(W..SQUIGGLE) WHERE W..SQUIGGLE IS THE TEMPORARY  COMB0632
C*********  VALUE OF THE CONSERVATION VECTOR      **********************COMB0633
    COMMON PROPTY,GAMMA3,GAMMA1,GAMMA,DELR,DELO,PO,PMAX,PR,ER,TAU,IMAXCOMB0634
   1,JMAX,T,DTDR,DTDO,DT,P,AO,ROO,NCYCMX,R
    DIMENSION PROPTY(12,36,4)
    COMMON/LE/ FJMAX,IMAXM1,RAD(12),OVRLAY(12,36,12),OVRPRM(16,36,12)
    DIMENSION WTEMP(4),GOFW(4)                                          COMB0640
```

```
      X=WTEMP(3)/WTEMP(1)                                               COMB0641
      Y=WTEMP(2)/WTEMP(1)                                               COMB0642
      GOFW(1)=-WTEMP(3)                                                 COMB0643
      GOFW(2)=GOFW(1)*Y                                                 COMB0644
      GOFW(3)=-0.5*GAMMA3*GOFW(1)*X-GAMMA1*(WTEMP(4)-0.5*Y*WTEMP(2))    COMB0645
      GOFW(4)=-(GAMMA*X*WTEMP(4)-GAMMA1*0.5*WTEMP(3)*(X**2+Y**2))       COMB0646
      RETURN                                                            COMB064
      END                                                               COMB0648
      SUBROUTINE TEMPSW(WTEMP,SOFW)                                     COMB0566
C*********  COMPUTES S(W..SQUIGGLE) WHERE W..SQUIGGLE IS THE TEMPORARY  COMB0564
C*********  VALUE OF THE CONSERVATION VECTOR      **********************COMB0565
      COMMON PROPTY,GAMMA3,GAMMA1,GAMMA,DELR,DELO,PO,PMAX,PR,ER,TAU,IMAXCOMB056
     1,JMAX,T,DTDR,DTDO,DT,P,AO,ROO,NCYCMX,R
      DIMENSION PROPTY(12,36,4)
      COMMON/LE/ FJMAX,IMAXM1,RAD(12),OVRLAY(12,36,12),OVRPRM(16,36,12)
      COMMON/EDOT/EKF
      COMMON/NCPLOT/NTCYCL,ISTART
      DIMENSION WTEMP(4),SOFW(4)                                        COMB0573
      X=WTEMP(3)/WTEMP(1)                                               COMB0574
      Y=WTEMP(2)/WTEMP(1)                                               COMB0575
      PRS=GAMMA1*(WTEMP(4)-0.5*WTEMP(1)*(Y**2+X**2))
      IF(PRS.LT.0.0) GO TO 50
      CALL PHIDOTV(WTEMP,SOFW)
      SOFW(1)=SOFW(1)*EKF
      SOFW(2)=WTEMP(1)*X**2+PRS
      SOFW(3)=-WTEMP(1)*X*Y                                             COMB0578
      SOFW(4)=SOFW(4)*EKF
      RETURN                                                            COMB0580
   50 CONTINUE
      WRITE(6,707)((J,I,(OVRPRM(K,J,I),K=1,4 ),J=1,JMAX),I=1,IMAX)
     2,I,I,WTEMP,NTCYCL
  707 FORMAT(1H0//(2I4,4E23.7))
      STOP
      END                                                               COMB0581
      SUBROUTINE GENPT(IF,IL)
C*********  COMPUTES THE VALUE OF W(T+DELT) AT REGULAR MESH POINTS      COMB0724
C*********  AND AT BOUNDARY POINTS      ********************************COMB0725
      COMMON PROPTY,GAMMA3,GAMMA1,GAMMA,DELR,DELO,PO,PMAX,PR,ER,TAU,IMAXCOMB0726
     1,JMAX,T,DTDR,DTDO,DT,P,AO,ROO,NCYCMX,R
      DIMENSION PROPTY(12,36,4)
      COMMON/LE/ FJMAX,IMAXM1,RAD(12),OVRLAY(12,36,12),OVRPRM(16,36,12)
      DIMENSION FDRVBR(4),GDRVBR(4),SAVEBR(4)
      DO 5 I=IF,IL
      DO 10 J=1,JMAX
   50 JP1=J+1                                                           COMB0761
      JM1=J-1                                                           COMB0762
```

```
      IF(JP1.LE.JMAX) GO TO 7                                           COMB0763
      JP1=1                                                             COMB0764
    7 IF(JM1.EQ.0)JM1=JMAX                                              COMB0765
      IM=I-1
      IP=I+1
      DO 4 K=1,4                                                        COMB0766
      K4=K+4
    1 FDRVBR(K)=(OVRPRM(K4,J,IP)-OVRPRM(K4,J,I)+OVRPRM(K4,JM1,IP)-
     1OVRPRM(K4,JM1,I))
      K8=K+8
    2 GDRVBR(K)=(OVRPRM(K8,J,IP)-OVRPRM(K8,JM1,IP)+OVRPRM(K8,J,I)-
     1OVRPRM(K8,JM1,I))
      K12=K+12
    3 SAVEBR(K)=.25*(OVRPRM(K12,J,IP)+OVRPRM(K12,J,I)+OVRPRM(K12,JM1,IP)
     1+OVRPRM(K12,JM1,I))
      OVRKJI=-OVRLAY(K8,J,I)
      IF(K.EQ.2.OR.K.EQ.3) OVRKJI=0.5*(OVRLAY(K8,J,IP)+OVRLAY(K8,J,IM))
    4 PROPTY(I,J,K)=PROPTY(I,J,K)+((DTDR*((OVRLAY(K,J,IP)-OVRLAY(K,J,IM)
     1)+FDRVBR(K))*.25+DTDO*((OVRLAY(K4,JP1,I)-OVRLAY(K4,JM1,I))+GDRVBR
     2(K))*.25+DT*(OVRKJI+SAVEBR(K))*.5)/RAD(I))
   10 CONTINUE
    5 CONTINUE
      RETURN                                                            COMB077
      END                                                               COMB0778
      SUBROUTINE PHIDOTV(PHIIN,PHIOUT)
      COMMON PROPTY,GAMMA3,GAMMA1,GAMMA,DELR,DELO,PO,PMAX,PR,ER,TAU,IMAX
     1,JMAX,T,DTDR,DTDO,DT,P,AO,ROO,NCYCMX,R
      DIMENSION PROPTY(12,36,4)
      COMMON/LE/ FJMAX,IMAXM1,RAD(12),OVRLAY(12,36,12),OVRPRM(16,36,12)
      COMMON/PHIP/DLN,EN,V,EKFCT,DMF,DMFT,ELFZ,CPR3,FPHIFS,DMPINF,DMRINF
     1,CRE,CAP
      COMMON/QQXM/XM,TO,QKR
      DIMENSION PHIIN(4),PHIOUT(4),TT(12),DHRT(12)
      DATA TWOPI,PI,RZ/6.28318531,3.14159265,1544./
      DATA G,CPSINF/32.17,.3539658/
      DATA TT,DHRT/10800.0,9900.0,9000.0,8100.0,7200.0,6300.0,5400.0,
     X4500.0,3600.0,2700.0,1800.0,536.4,225.0,1010.0,1860.0,2580.0,
     X3380.0,4160.0,4900.0,5620.0,6300.0,6970.0,7540.0,8150.0/
      PHIOUT(2)=0.0
      PHIOUT(3)=0.0
      IF(EKFCT)300,10,200
   10 CONTINUE
      USQ=PHIIN(2)*PHIIN(2)+PHIIN(3)*PHIIN(3)
      PINF=DMPINF*(PHIIN(4)-0.5*USQ/PHIIN(1))
      RHOINF=PHIIN(1)*ROO
      TINF=DMRINF*PINF/RHOINF
```

```
      A300=3.73E-5*TINF+1.855
      ALP=ALOG(PINF)
      AP=CAP*A300*EXP(.0108*ALP)
      TL=8700./(15.0622979-ALP)
      IF(PINF.GT.2131.) TL=1176.
      SK=.691716667E-4-2.31388889E-8*TL
      CP=.137857+TL*(.52715E-3-TL*.119907E-6)
      EMFZ=TWOPI*DLN*SK/CP*AP
      RE=CRE*SQRT(USQ/TINF)
      EMFD=EMFZ*(1.0+CPR3*SQRT(RE))
      PSI=EMFD*FPHIFS
      CALL SEEK(TINF,TT,11,IJ)
   60 FORMAT(11H SEEK ERROR,2E20.10)
      IF(IJ)80,70,80
   70 PRINT 60,TINF,TT(1)
      STOP
   80 CONTINUE
      DHR=DHRT(IJ)+(DHRT(IJ+1)-DHRT(IJ))*(TINF-TT(IJ))/(TT(IJ+1)-TT(IJ))
      XIS=EMFD*(DHR-ELFZ)
      PHIOUT(1)=PSI*DMFT
      PHIOUT(4)=XIS*DMF
  100 CONTINUE
      RETURN
  200 CONTINUE
      X=PHIIN(3)/PHIIN(1)
      Y=PHIIN(2)/PHIIN(1)
      PRS=GAMMA1*(PHIIN(4)-0.5*PHIIN(1)*(Y**2+X**2))
      PHIOUT(1)=0.0
      PPRM=PRS-1.0
      IF(PPRM.LT.0.0) PPRM=SQRT(-PPRM)
      PHIOUT(4)=PPRM
      RETURN
  300 CONTINUE
      PHIOUT(1)=0.0
      PHIOUT(4)=0.0
      RETURN
      END
      ASCENTF SUBROUTINE SEEK(E,EOUT,NOUT,I)
            BSS 6
ENTRY       BSS 1
            SA1 B1
            SA2 B2
            SB7 1
            FX3 X2-X1
            NZ X3 SK1
            SX6 B7
```

```
         SA6 B4
         ZR B0 ENTRY
 SK1     PL X3 SK2
         SX6 B0
         SA6 B4
         ZR B0 ENTRY
 SK2     SA4 B3
         SB2 B2-B7
         SB5 B7
         SB6 X4+B7
 SK3     SB1 B5+B7
         NE B1 B6 SK4
         SX6 B5
         SA6 B4
         ZR B0 ENTRY
 SK4     SX5 B5+B6
         AX7 B7 X5
         SA2 X7+B2
         IX3 X2-X1
         ZR X3 SK6
         PL X3 SK5
         SB6 X7
         ZR B0 SK3
 SK5     SB5 X7
         ZR B0 SK3
 SK6     SA0 X7
         SX6 A0-B7
         SA6 B4
         ZR B0 ENTRY
         END
      SUBROUTINE CENTER
C*********  COMPUTES THE VALUE OF W(T+DELT) AT THE CENTER OF THE         COMB0683
C*********  CYLINDER BY SCALER AND VECTOR AVERAGING   *****************COMB0684
      COMMON PROPTY,GAMMA3,GAMMA1,GAMMA,DELR,DELO,PO,PMAX,PR,ER,TAU,IMAXCOMB0685
     1,JMAX,T,DTDR,DTDO,DT,P,AO,ROO,NCYCMX,R
      DIMENSION PROPTY(12,36,4)
      COMMON/LE/ FJMAX,IMAXM1,RAD(12),OVRLAY(12,36,12),OVRPRM(16,36,12)
      COMMON/CNTRE/COSTH(36),SINTH(36),THTA(36)
      COMMON/MIDDLE/VMAG,VDIR
      COMMON/TLDR/TTLDR
      COMMON/EDOT/EKF
      DIMENSION FNUM(4),GNUM(4),WXY(4,36,2),FXY(4,36,2),GXY(4,36,2)
     1,X(36),Y(36)
      EQUIVALENCE (WXY,OVRLAY),(FXY,OVRLAY(289)),(GXY,OVRLAY(577))
     1,(X,OVRLAY(865)),(Y,OVRLAY(901))
      DATA PI,TWOPI/3.1415926,6.2831852/
```

```
      DR2=.5*DELR
      DO 10 J=1,JMAX
      X(J)=DELR*COSTH(J)
      Y(J)=DELR*SINTH(J)
      WXY(1,J,1)=PROPTY(2,J,1)
      WXY(4,J,1)=PROPTY(2,J,4)
      WXY(2,J,1)=PROPTY(2,J,2)*COSTH(J)-PROPTY(2,J,3)*SINTH(J)
      WXY(3,J,1)=PROPTY(2,J,2)*SINTH(J)+PROPTY(2,J,3)*COSTH(J)
      WXY(1,J,2)=OVRPRM(1,J,1)
      WXY(4,J,2)=OVRPRM(4,J,1)
      WXY(2,J,2)=OVRPRM(2,J,1)*COSTH(J)-OVRPRM(3,J,1)*SINTH(J)
      WXY(3,J,2)=OVRPRM(2,J,1)*SINTH(J)+OVRPRM(3,J,1)*COSTH(J)
      DO 20 K=1,4
      PROPTY(2,J,K)=OVRPRM(K,J,1)
   20 CONTINUE
      DO 25 I=1,2
      CALL TEMPFW(WXY(1,J,I),FXY(1,J,I),1,J,1)
      CALL TEMPGW(WXY(1,J,I),GXY(1,J,I) )
      DO 25 K=1,4
   25 FXY(K,J,I)=FXY(K,J,I)/DR2
   10 CONTINUE
      DO 35 K=1,4
      DO 30 J=1,JMAX
      FXY(K,J,1)=0.5*(FXY(K,J,1)+FXY(K,J,2))
      GXY(K,J,1)=0.5*(GXY(K,J,1)+GXY(K,J,2))
   30 CONTINUE
      FNUM(K)=0.
      GNUM(K)=0.
   35 CONTINUE
      DENOM=0.
      DO 40 J=1,JMAX
      JP1=J+1
      IF(J.EQ.JMAX)JP1=1
      YDIF=Y(JP1)-Y(J)
      DENOM=(X(JP1)+X(J))*YDIF+DENOM
      XDIF=X(JP1)-X(J)
      DO 50 K=1,4
      FNUM(K)=FNUM(K)+(FXY(K,J,1)+FXY(K,JP1,1))*YDIF
   50 GNUM(K)=GNUM(K)+(GXY(K,J,1)+GXY(K,JP1,1))*XDIF
   40 CONTINUE
      DO 55 K=1,4
      FNUM(K)=FNUM(K)/DENOM
      GNUM(K)=GNUM(K)/DENOM
      PROPTY(1,1,K)=PROPTY(1,1,K)+DT*(FNUM(K)-GNUM(K))
   55 CONTINUE
      QA=VDIR
```

```
      VMAG=SQRT(PROPTY(1,1,2)**2+PROPTY(1,1,3)**2)/PROPTY(1,1,1)
      VDIR=ATAN2(PROPTY(1,1,3),PROPTY(1,1,2))
      QA=ABS(VDIR-QA)
      IF(QA.GT.PI) QA=TWOPI-QA
      TTLDR=TTLDR+QA
      RETURN                                                            COMB0720
      END                                                               COMB0721
      SUBROUTINE CHARGE
C*********  COMPUTES BOUNDARY CONDITIONS AT THE PERIPHERY
C*********  OF THE CYLINDRICAL SURFACE     *****************************COMB0918
      COMMON PROPTY,GAMMA3,GAMMA1,GAMMA,DELR,DELO,PO,PMAX,PR,ER,TAU,IMAX
     1,JMAX,T,DTDR,DTDO,DT,P,AO,ROO,NCYCMX,R
      DIMENSION PROPTY(12,36,4)
      COMMON/LE/ FJMAX,IMAXM1,RAD(12),OVRLAY(12,36,12),OVRPRM(16,36,12)
      IMAXM2=IMAX-2
      RINT=RAD(IMAX-2)
      REXT=RAD(IMAX)
      RQUOT=RINT/REXT
      DO 10 J=1,JMAX
      PROPTY(IMAX,J,1)=PROPTY(IMAXM2,J,1)*RQUOT
      PROPTY(IMAX,J,2)=-PROPTY(IMAX,J,1)*(PROPTY(IMAXM2,J,2)/PROPTY(IMAX
     X M2,J,1))
      PROPTY(IMAX,J,3)=PROPTY(IMAXM2,J,3)
      RUSQ=(PROPTY(IMAXM2,J,2)**2)/PROPTY(IMAXM2,J,1)
      RVSQ=(PROPTY(IMAXM2,J,3)**2)/PROPTY(IMAXM2,J,1)
      PINT=GAMMA1*(PROPTY(IMAXM2,J,4)-.5*(RUSQ+RVSQ))
      PEXT=PINT+2.*DELR/RINT*RVSQ
      PROPTY(IMAX,J,4)=PEXT/GAMMA1+.5*(PROPTY(IMAX,J,2)**2+PROPTY(IMAX,
     *J,3)**2)/PROPTY(IMAX,J,1)
   10 CONTINUE
      RETURN
      END
      SUBROUTINE PRTOUT(RTM)
C*********  PRINTS A TERSE VIEW OF THE FLOW FIELD ALONG THE             COMB1421
C***Q*****  RAYS 0,PI/2,PI,3*PI/2        *******************************COM21422
      COMMON PROPTY,GAMMA3,GAMMA1,GAMMA,DELR,DELO,PO,PMAX,PR,ER,TAU,IMAXCOMB1423
     1,JMAX,T,DTDR,DTDO,DELT,P,AO,ROO,NCYCMX,R
      DIMENSION PROPTY(12,36,4)
      COMMON/NCPLOT/NTCYCL,ISTART,RC,OC
      COMMON/MIDDLE/VMAG,VDIR
      COMMON/PRTOPT/INTPNT
      COMMON/TLDR/TTLDR
      EQUIVALENCE(THTBAR,VDIR)
   75 LF=0
      L=1                                                               COMB1429
      ENREV=TTLDR/6.2831852
```

```
      WRITE(6,606) T,NTCYCL,ENREV,DELT,RTM
      IF(INTPNT.EQ.0) GO TO 1818
      J=1
      I=1
      WRITE(6,707) J,I,VMAG,VDIR
      WRITE(6,707)((J,I,(PROPTY(I,J,K),K=1,4),J=1,JMAX),I=1,IMAX)
  707 FORMAT(1X,2I4,4E23.7)
 1818 CONTINUE
      WRITE(6,6062)                                                     COMB1431
  606 FORMAT(12H1TOTAL TIME=E12.5,3X,17HNUMBER OF CYCLES=I4,3X,15HNUMBER
     1 OF REVS=F5.2,3X,10HTIME STEP=E12.5,3X,10HREAL TIME=F9.3///)
 6165 FORMAT(8X,5H RHO ,14X,3H U ,14X,3H V ,19X,10HINT ENERGY,8X,10H PRECOMB1434
     1SSURE ,10X,9H MACH NO.///)                                        COMB1435
 6061 FORMAT(1X,6E19.5)                                                 COMB1479
 6062 FORMAT(1H0,20X,32HSOLUTION ALONG THE RAY THETA = 0///)            COMB1436
 6063 FORMAT(1H0,20X,33HSOLUTION ALONG THE RAY THETA = PI///)           COMB143
 6064 FORMAT(1H0,20X,35HSOLUTION ALONG THE RAY THETA = PI/2///)         COMB1438
 6065 FORMAT(1H0,20X,37HSOLUTION ALONG THE RAY THETA = 3*PI/2///)       COMB1439
  402 CONTINUE
      WRITE(6,6165)                                                     COMB1432
      DO 1012 I=1,IMAX
      IF(L.NE.1.AND.I.EQ.1) GO TO 1013                                  COMB1441
      RHOX=PROPTY(I,L,1)                                                COMB1442
      UX=PROPTY(I,L,2)/RHOX                                             COMB1443
      VX=PROPTY(I,L,3)/RHOX                                             COMB1444
      EX=(PROPTY(I,L,4)/RHOX)-0.5*(UX**2+VX**2)                         COMB1445
      PX=GAMMA1*RHOX*EX                                                 COMB1446
      XMX=SQRT((UX**2+VX**2)/(GAMMA*PX/RHOX))                           COMB144
      IF(L.EQ.1.AND.I.EQ.1) GO TO 1014                                  COMB1448
      GO TO 1011                                                        COMB1449
 1014 TMPRO=RHOX                                                        COMB1450
      TMPEX=EX                                                          COMB1458
      TMPPX=PX                                                          COMB1459
      TMPXMX=XMX                                                        COMB1460
 1013 CONTINUE
      XT=L-1
      THETSR=XT*DELO
      DIFFT=THTBAR-THETSR                                               COMB1455
      TMPUX=VMAG*COS(DIFFT)
      TMPVX=VMAG*SIN(DIFFT)
      WRITE(6,6061)TMPRO,TMPUX,TMPVX,TMPEX,TMPPX,TMPXMX
      GO TO 1012                                                        COMB1464
 1011 WRITE(6,6061)             RHOX,UX,VX,EX,PX,XMX                    COMB1465
 1012 CONTINUE                                                          COMB1466
      LF=LF+1
      GO TO(404,504,403,1001),LF
```

```
  404 CONTINUE                                                    
      L=JMAX/2+1                                                  COMB1469
      WRITE(6,6063)                                               COMB1470
      GO TO 402                                                   COMB1472
  504 CONTINUE                                                    
      L=JMAX/4+1                                                  COMB1475
      WRITE(6,6064)                                               COMB1476
      GO TO 402                                                   COMB1478
  403 CONTINUE                                                    
      L=3*(JMAX/4)+1                                              COMB1482
      WRITE(6,6065)                                               
      GO TO 402                                                   
 1001 CONTINUE                                                    COMB1499
      RETURN                                                      COMB1500
      END                                                         COMB1501
```

Appendix C

```
      PROGRAM COMBM(INPUT,OUTPUT,TAPE1,TAPE3,TAPE5=INPUT,TAPE6=OUTPUT,
     1TAPE2)
      COMMON PROPTY,GAMMA3,GAMMA1,GAMMA,DELR,DELO,PO,PMAX,PR,ER,TAU,IMAX
     1,JMAX,T,DTDR,DTDO,DELT,P,AO,ROO,NCYCMX,R
      DIMENSION PROPTY(32,108,4)
      CALL COMB
      STOP
      END
```

```
      SUBROUTINE COMB                                                   999999
C                                                                       COMB0001
C***********************************************************************COMB0002
C       TIME DEPENDENT EQUATIONS OF HYDRODYNAMICS                       COM+0003
C***********************************************************************COM20004
C***********************************************************************COMB0005
C     SOLUTION RESTRICTED TO THE TRANSVERSE PLANE IN CYLINDRICAL        COMB0006
C     COORDINATES                                                       COMB000
C     FINITE DIFFERENCE APPROXIMATION ... LAX WENDROFF TWO STEP         COMB0008
C     BURSTEIN VARIATION ...                                            COMB0009
C***********************************************************************COMB0010
C***********************************************************************COM20011
C                    ********************************                   COMB0012
C                    INPUT PARAMETERS IN MAIN ROUTINE                   COMB0013
C                    --------------------------------                   COMB0014
C     ISTART          IF .EQ. 11111   NEW CASE                          COMB0015
C                     IF .EQ. 00000   CONTINUATION OF PREVIOUS CALCULATION COMB0016
C     IMAX            MAX NUMBER OF MESH POINTS ON A RADIUS INCLUDING ONE COMB001
C                     POINT FOR THE BOUNDARY CONDITION                  COMB0018
C     JMAX            MAX NUMBER OF RAYS                                COMB0019
C     NCYCMX          TOTAL NUMBER OF TIME STEPS FOR THE PRESENT        COMB0020
C                     INTFGRATION                                       COMB0021
C     NNPCMX          NUMBER OF TIME INTEGRATIONS BETWEEN PICTURES      COMB0022
C     FUDG            SAFETY FACTOR FOR AUGMENTATION OF LINEAR          COMB0023
C                     STABILITY ANALYSIS                                COMB0024
C                    INPUT PARAMETERS IN OTHER ROUTINES                 COMB0025
C                    ----------------------------------                 COMB0026
C     GAMMA           RATIO OF SPECIFIC HEATS                           COMB002
C     PO              UNPFRTURBED PRESSURE                              COMB0028
C     PMAX            AMPLITUDE OF PRESSURE PERTURBATION                COM20029
C     TAU             DURATION OF PRESSURE PERTURBATION                 COMB0030
C     RMAX            RADIUS OF CYLINDER                                COMB0031
C     XM              MOLECULAR WEIGHT OF GAS                           COMB0032
C     TO              GAS STAGNATION TEMPERATURE                        COMB0033
C***********************************************************************COMB0034
C                    ------------------------------------------------   COMB0035
C                         SCALING RULES...SEE SUB. INITIL               COMB0036
C                    ------------------------------------               COMB003
C***********************************************************************COMB0038
C                     IN PUT TAPES                                      COMB0039
C                     ------------                                      COMB0040
C     BINARY TAPE 1         INPUT TAPE CONTAINING INFORMATION FOR THE   COMB0041
C                           CONTINUATION OF THE CALCULATION             COMB0042
C                    OUT PUT TAPES                                      COMB0043
C                    -------------                                      COMB0044
C     BINARY TAPE 3         OUTPUT TAPE CONTAINS INFORMATION FOR THE    COMB004
```

```
C                               CONTINUATION OF COMPUTATION                     COMB0048
C***********************************************************************************COMB0049
C ********************************************************************************* COMB0050
C        *************************************************************         COM20053
C                 ****************************************         COMB0054
C                               *************                               COMB0055
C                                                                               COMB0056
C                                                                               COMB005
C                                                                               COMB0058
C      MAIN                                                                     COMB0059
C*********  EXECUTIVE PROGRAM WHICH CONTROLS INPUT,OUTPUT AND                   COMB0060
C*********  LOGICAL FLOW FOR THE COMPUTATION     ****************************COMB0061
      COMMON PROPTY,GAMMA3,GAMMA1,GAMMA,DELR,DELO,PO,PMAX,PR,ER,TAU,IMAXCOMB0062
     1,JMAX,T,DTDR,DTDO,DELT,P,AO,ROO,NCYCMX,R
      DIMENSION PROPTY(32,108,4)
      COMMON/LE/FJMAX,IMAXM1,RAD(32),OVRLAY(12,3,32),OVRPRM(16,2,32)
      COMMON/NCPLOT/NTCYCL,ISTART,RC,OC
      COMMON/MIDDLE/VMAG,VDIR
      COMMON/PRTOPT/INTPNT,XTO,DETR(5),DTR(5),IDETR(5),PDM
      COMMON/TLDR/TTLDR
      COMMON/EDOT/EKF
      COMMON/PHIP/DLN,EN,V,EKFCT,DMF,DMFT,ELFZ,CPR3,FPHIFS,DMPINF,DMRINF
     1,CRE,CAP
      COMMON/QQXM/XM,TO,QKR
      COMMON/JPPT/JP,PTCRD
      DIMENSION PTCRD(501,2)
      DATA ZERO,ONE/0.0,1.0/
      REWIND 2
    5 CONTINUE
      NCLCM=0
      NTCYCL=0
      NCYCLE=0
      NPICTR=0
      NSAVE=0
      NPNTC=0
      JP=1
      TTLDR=0.0
      T=0.0
      NPRPLT=1
      DLN=2.0E-4
      PHI=1.0
      V=120.0
      FS=.695652174
      READ(5,666)ISTART,ISAVE
      IF(ISTART.LT.0) GO TO 186
      READ(5,668)RAD(2),RAD(3),RAD(4)
```

```
  668 FORMAT(4A10)
      RAD(1)=10H1
      RAD(5)=0.0
      WRITE(6,668) RAD(1),RAD(2),RAD(3),RAD(4)
      RAD(1)=10H554901
      WRITE(2) RAD(1),RAD(2),RAD(3),RAD(4)
      READ(5,666)IMAX,JMAX
      FJMAX=JMAX
      IMAXM1=IMAX-1                                                     COMB0101
      READ(5,666)NCYCMX,NNPCMX,KALKOM,INTPNT,IBSPLT,IPRPLT,JPR,IPTPLT,L
  666 FORMAT(9I5)
      WRITE(6,6066)NCYCMX,NNPCMX                                        COMB0076
 6066 FORMAT(39H0 TOTAL NUMBER OF CYCLES FOR THIS CASE=I4,4X,34HNUMBER O
     2F CYCLES BETWEEN PICTURES=I4///)                                  COMB0079
      READ(5,667)FUDG,D,EKFCT,RC,OC,EN
      PTCRD(1,1)=RC
      PTCRD(1,2)=OC
  667 FORMAT(7E10.7)
C       INITIALIZE MESH AND FIND DELT FOR CONVERG. AND STABILITY        COMB0083
      CALL INITIL                                                       COMB0084
      IF(ISTART.NE.0) CALL CHARGE
      CALL CONVRG(FUDG)                                                 COMB0086
      EKFS=EKFCT*EKF/DELT
      IF(EKFCT.EQ.0.0) EKFS=1.0
      DMFT=EN*R/(ROO*AO)
      DMF=DMFT*778.0*32.17/(AO*AO)
      PHITFS=PHI*FS
      ELFZ=540.0+178.2/PHITFS
      CPR3=0.276*0.591
      FPHIFS=1.0+1.0/PHITFS
      DMPINF=PO*GAMMA*GAMMA1
      DMRINF=144.0*XM/1545.0
      CRE=DLN*AO*ROO*23.66432/2.5E-5
      CAP=(1.0/300.0)**.0108
      NPNTC=MOD(NTCYCL,IPTPLT)
  757 CONTINUE
C       PRINT INITIAL FIELD                                             COMB008
      CALL SECOND(RTM)
      CALL PRTOUT(RTM)
      IMX=12
      JMX=36
      JDF=JMAX/JMX
      IF(ISTART)70,80,70
   70 CONTINUE
      WRITE(2) ZERO,T,NTCYCL,AO,R,JPR
      WRITE(2) VMAG,VDIR,TTLDR,GAMMA,PO,ROO,IMX,JMX,NPRPLT,
```

```
     1(((PROPTY(I,J,K),K=1,4),J=1,JMAX,JDF),I=1,IMAX,JDF)
  80 NCYCLE=NCYCLE+1
     NTCYCL=NTCYCL+1
     NPNTC=NPNTC+1
     T=T+DELT                                                    COMB0105
     CALL MVPNT(RC,OC)
     IF(MOD(NPNTC,L).NE.0) GO TO 801
     JP=JP+1
     PTCRD(JP,1)=RC
     PTCRD(JP,2)=OC
 801 CONTINUE
     EKF=EKFS
C    COMPUTE W(T+DELT) FOR FIRST ROW
     CALL FRSTRO
     DO 475 I=2,IMAX
     DO 475 K=1,4
 475 PROPTY(I,107,K)=PROPTY(I,1,K)
     DO 480 I=2,IMAX
     CALL VECTFR(I,0,OVRLAY(1,2,I))
     CALL VECTFR(I,1,OVRLAY(1,3,I))
     CALL VECTRG(I,0,OVRLAY(5,2,I))
     CALL VECTRG(I,1,OVRLAY(5,3,I))
     CALL VECTRS(I,0,OVRLAY(9,2,I))
     CALL VECTRS(I,1,OVRLAY(9,3,I))
 480 CONTINUE
     J=0
     DO 485 I=2,IMAXM1
     I1=I+1
     CALL QUAD1(I,J,OVRPRM(1,2,I1),I)
     CALL TEMPFW(OVRPRM(1,2,I1),OVRPRM(5,2,I1),I,J,1)
     CALL TEMPGW(OVRPRM(1,2,I1),OVRPRM(9,2,I1))
     CALL TEMPSW(OVRPRM(1,2,I1),OVRPRM(13,2,I1))
 485 CONTINUE
     DO 510 J=1,JMAX
     JP1=J+1
     DO 500 I=2,IMAX
     DO 490 K=1,12
     OVRLAY(K,1,I)=OVRLAY(K,2,I)
 490 OVRLAY(K,2,I)=OVRLAY(K,3,I)
     CALL VECTFR(I,JP1,OVRLAY(1,3,I))
     CALL VECTRG(I,JP1,OVRLAY(5,3,I))
     CALL VECTRS(I,JP1,OVRLAY(9,3,I))
 500 CONTINUE
     DO 501 I=2,IMAXM1
     I1=I+1
     DO 495 K=1,16
```

```
  495 OVRPRM(K,1,I1)=OVRPRM(K,2,I1)
      CALL QUAD1(I,J,OVRPRM(1,2,I1),I)
      CALL TEMPFW(OVRPRM(1,2,I1),OVRPRM(5,2,I1),I,J,1)
      CALL TEMPGW(OVRPRM(1,2,I1),OVRPRM(9,2,I1))
      CALL TEMPSW(OVRPRM(1,2,I1),OVRPRM(13,2,I1))
  501 CONTINUE
C  COMPUTE W(T+DELT)AT INTERIOR POINTS
      DO 502 I=3,IMAXM1
      CALL GENPT(I,J)
  502 CONTINUE
      IF(J.NE.1) GO TO 510
      DO 505 I=2,IMAX
      DO 505 K=1,4
      PROPTY(I,108,K)=PROPTY(I,1,K)
  505 PROPTY(I,1,K)=PROPTY(I,107,K)
  510 CONTINUE
      DO 515 I=2,IMAX
      DO 515 K=1,4
  515 PROPTY(I,1,K)=PROPTY(I,108,K)
C       COMPLETE MESH CALC AT I=1,J=1                                   COMB0150
      CALL CENTER
      CALL CHARGE                                                       COMB0162
      NCLCM=NCLCM+1
      NPICTR=NPICTR+1
      NSAVE=NSAVE+1
C       NOISE SUPPRESSOR                                                COMB0163
      DO 50 I=1,IMAX                                                    COMB0164
      DO 50 J=1,JMAX                                                    COMB0165
      IF(I.EQ.1.AND.J.GT.1) GO TO 50                                    COMB0166
C       TEST FOR NEGATIVE DENSITY                                       COMB016
      IF(PROPTY(I,J,1).LE.0.) GO TO 181                                 COMB0168
      DO 51 K=2,3                                                       COMB0169
C       ELIMINATE SPURIOUS VELOCITIES                                   COMB0170
      IF(ABS (PROPTY(I,J,K)).LE.1.E-14) PROPTY(I,J,K)=0.
   51 CONTINUE                                                          COMB0172
   50 CONTINUE
C       FIND NEW STABILITY NUMBER                                       COMB0158
      CALL CONVRG(FUDG)                                                 COMB0159
   81 CONTINUE
      IF(NPNTC-IPTPLT)820,810,810
  810 CONTINUE
      WRITE(2) ONE,T,NTCYCL,AO,R,JP
      WRITE(2)(PTCRD(I,1),PTCRD(I,2),I=1,JP)
      NPNTC=0
      JP=1
      PTCRD(1,1)=RC
```

```
      PTCRD(1,2)=OC
  820 CONTINUE
      IF(NCLCM-KALKOM)89,88,88
   88 CONTINUE
      WRITE(6,606)T,NTCYCL
  606 FORMAT(3X11HTOTAL TIME=E14.7,3X17HNUMBER OF CYCLES=I4)            COMB0186
      NCLCM=0
      IF(IPRPLT.NE.0) NPRPLT=MOD(NCYCLE,IPRPLT)
      WRITE(2) ZERO,T,NTCYCL,AO,R,JPR
      WRITE(2) VMAG,VDIR,TTLDR,GAMMA,PO,ROO,IMX,JMX,NPRPLT,
     1(((PROPTY(I,J,K),K=1,4),J=1,JMAX,JDF),I=1,IMAX,JDF)
   89 CONTINUE
C       TEST TO SEE IF FINISHED                                          COMB0174
      IF(NCYCLE-NCYCMX)890,91,91
  890 CONTINUE
C       TEST TO SEE IF PRINT TIME AFFIRMATIVE                            COMB0176
      IF(NPICTR-NNPCMX)901,90,90
   90 CONTINUE
C       PRINT DISPLAY OF FLOW FIELD ALONG THE RAYS 0,PI/2,PI,3*PI/2      COMB0180
      CALL SECOND(RTM)
      CALL PRTOUT(RTM)
C       RESTART PRINT CYCLE                                              COMB018
      NPICTR=0
  901 IF(ISAVE.EQ.0.OR.NSAVE.LT.ISAVE) GO TO 80
  902 CONTINUE
      REWIND 3
      WRITE(3)(((PROPTY(I,J,K),K=1,4),J=1,JMAX),I=1,IMAX)
      WRITE(3)T,NTCYCL,VMAG,VDIR,TTLDR,RC,OC
      WRITE(3) JP,(PTCRD(I,1),PTCRD(I,2),I=1,JP)
      NSAVE=0
      GO TO 80                                                          COMB0189
   91 CONTINUE
      CALL SECOND(RTM)
      CALL PRTOUT(RTM)
      IF(ISAVE)182,5,182
  181 WRITE(6,6666)                                                     COMB019
 6666 FORMAT(1H1,20X,63H********** ERROR ABORT  -  NEG. DENSITY IN THE FCOMB0198
     1IELD **********/21X,63H*******************************************COMB0199
     2********************)                                             COMB0200
      WRITE(6,707)(I,J,(PROPTY(I,J,K),K=1,4),I=1,IMAX)
  707 FORMAT(1X,2I4,4E23.7)
      GO TO 184                                                         COMB0201
  182 CONTINUE
      REWIND 3
      WRITE(3)      (((PROPTY(I,J,K),K=1,4),J=1,JMAX),I=1,IMAX)
      WRITE(3)T,NTCYCL,VMAG,VDIR,TTLDR,RC,OC
```

```
      WRITE(3) JP,(PTCRD(I,1),PTCRD(I,2),I=1,JP)
  184 GO TO 5
  186 CONTINUE
      ENDFILE 2
      END FILE 3
      STOP
      END                                                          COMB0209
```

```
      SUBROUTINE INITIL                                                 COMB0465
C*********  SETS UP THF INITIAL DATA IN THE ARRAY PROPTY AFTER          COMB0466
C*********  COMPLETING THE NORMALIZATION FROM INPUT DATA  **************COMB046
C      ********,***************************************************COMB0468
C      ************************************************************COMB0469
C      ************************************************************COMB0470
C      * * * * * * * * * * * * SCALING RULES * * * * * * * * * * * * * *COMB0471
C       * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * COMB0472
C      EQ1*(TO/RHOO),,FQ2*(TO/(RHOO*AO)),,EQ3*(TO/(RHOO*AO)),,          COMB0473
C      EQ4*(TO/(RHOO*AO**2)),,,,TO=(RMAX/AO)                            COMB0474
C       * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * COMB0475
C      **********    TIME IS SCALED BY ...THE REF TIME SCALE   **********COMB0476
C      **********    DISTANCE IS SCALED BY...COMBUSTION CHAMBER RADIUS***COMB047
C      **********    VELOCITY IS SCALED BY...REF SOUND SPEED   **********COMB0478
C      **********    DENSITY IS SCALED BY...REFERENCE DENSITY***********COMB0479
C      **********    PRFSSURE IS SCALED BY...REF SOUND SPEED**2*REF     COMB0480
C      **********    DENSITY******************************************COMB0481
C      ************************************************************COMB0482
C      ************************************************************COMB0483
C      ************************************************************COMB0484
      COMMON PROPTY,GAMMA3,GAMMA1,GAMMA,DELR,DELO,PO,PMAX,PR,ER,TAU,IMAXCOMB0485
     1,JMAX,T,DTDR,DTDO,DT,P,AO,ROO,NCYCMX,RMAX
      DIMENSION PROPTY(32,108,4)
      COMMON/LE/FJMAX,IMAXM1,RAD(32),OVRLAY(12,3,32),OVRPRM(16,2,32)
      COMMON/CNTRE/COSTH(108),SINTH(108),THTA(108)
      COMMON/MIDDLE/VMAG,VDIR
      COMMON/NCPLOT/NTCYCL,ISTART,RC,OC
      COMMON/EDOT/EKF
      COMMON/TLDR/TTLDR
      COMMON/QQXM/XM,TO,QKR
      COMMON/PRTOPT/INTPNT,XTO,DETR(5),DTR(5),IDETR(5),PDM
      COMMON/JPPT/JP,PTCRD
      DIMENSION PTCRD(501,2)
      DATA PI,TWOPI/3.14159265,6.28318531/
      READ(5,5)            GAMMA,PO,PMAX,TAU,RMAX                       COMB0490
      WRITE(6,65)                                                       COMB0491
   65 FORMAT(4X,5HGAMMA,5X,16HCHAMBER PRESSURE,2X,14HPULSE PRESSURE,4X,1COMB0492
     1                                                               5H COMB0493
     2PULSE DURATION,4X,                                                COMB0494
     3               14HCHAMBER RADIUS/20X,3HPSI,15X,3HPSI,15X,3HSEC,15XCOMB0495
     2,2HFT//)                                                          COMB0496
C      * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *COMB049
C      * * * * * *  BOMB PULSE PRESSURE AND TIME DURATION ARE ONLY      COMB0498
C      * * * * * *  TO BE REFERENCED WHEN INITIAL CONDITIONS ARE        COMB0499
C      * * * * * *  FOR A PULSE TYPE CALCULATION  * * * * * * * * * * * COMB0500
C      * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *COMB0501
```

```
      WRITE(6,4)            GAMMA,PO,PMAX,TAU,RMAX                     COMB0502
      FORMAT(1X,2E14.7,E18.7,E19.7,F12.4///)                           COMB0503
      READ(5,5)XM,ROO,OKR
      PO=PO                                                            COMB0519
      TO=(PO*144.)/((1545./XM)*ROO)
      WRITE(6,68)                                                      COMB0505
   68 FORMAT(3X,10HMOL WEIGHT,3X,19HCHAMBER TEMPERATURE//)             COMB0506
      WRITE(6,5)XM,TO                                                  COMB050
      WRITE(6,76)                                                      COMB0508
   76 FORMAT(1H ///)                                                   COMB0509
C       GAMMA  SPECIFIC HEAT RATIO                                     COMB0510
C       PO  UNPERTURBED PRESSURE                                       COMB0511
C       PMAX  AMPLITUDE OF PRESSURE PERTURBATION                       COMB0512
C       TAU IS DURATION OF PERT.                                       COMB0513
C       IMAX,JMAX MESH DIMENSIONS
C       XM  MOLECULAR WEIGHT, TO  CHAMBER TEMPERATURE                  COMB0515
    7 FORMAT(2I4)                                                      COMB0516
    5 FORMAT(4E14.7,2F10.4)                                            COMB051
      AO=SQRT (GAMMA*32.17*PO*144./ROO)                                COMB0521
C       RHO=RO/ROO=1.                                                  COMB0522
      RHO=1.                                                           COMB0523
      TO=RMAX/AO                                                       COMB0524
      XTO=TO*1000.                                                     COMB0525
      EQ2=ROO*AO/TO                                                    COMB0526
      P=PO*144.*32.17/(RMAX*EQ2)                                       COMB052
      DELR=1.0/FLOAT(IMAX-2)
      DELO=2.*PI/FJMAX
      GAMMA3=GAMMA-3.                                                  COMB0530
      GAMMA1=GAMMA-1.                                                  COMB0531
      PR=PMAX/PO                                                       COMB0532
      ER=PR/GAMMA1                                                     COMB0533
      WRITE(6,67)                                                      COMB0534
   67 FORMAT(1X,16HREDUCED PRESSURE,2X,15HREF SOUND SPEED,2X,11HREF DENSCOMB0535
     1ITY,6X,7HDELTA R,7X,11HDELTA THETA,4X,43HPRESSURE AND INTERNAL ENECOMB0536
     2RGY RATIO OF PULSE//)                                            COMB053
       WRITE(6,6)         P,    AO,ROO,DELR,DELO,PR,ER,XTO             COMB0538
    6 FORMAT(7E16.7///16H REF TIME SCALE=F15.5/3X,8HMILLISEC///)       COMB0539
      WRITE(6,566)                                                     COMB0540
  566 FORMAT(1X,117H**************************************** END OF INITCOMB0541
     1ILIZATION PHASE ************************************************/COMB0542
     2/////)                                                           COMB0543
      ELN=TWOPI/FJMAX
      RAD(1)=0.0
      DO 501 I=2,IMAX
  501 RAD(I)=RAD(I-1)+DELR
      DO 510 J=1,JMAX
```

```
      TH=J-1
      TH=TH*DELO
      THTA(J)=TH
      COSTH(J)=COS(TH)
  510 SINTH(J)=SIN(TH)
      EKF=P/GAMMA1
      DETR(1)=0.681
      DETR(2)=1.181
      DETR(3)=1.681
      DETR(4)=3.681
      DETR(5)=12.0*RMAX
      PDM=GAMMA*PO
      DO 520 I=1,4
      Q=DETR(I)/DETR(5)
      J=Q/DELR+1.0
      DTR(I)=(Q-RAD(J))/DELR
  520 IDETR(I)=J
      DTR(5)=1.0
      IDETR(5)=IMAX-2
      IF(ISTART)506,505,506
  505 CONTINUE
      READ(1)       (((PROPTY(I,J,K),K=1,4),J=1,JMAX),I=1,IMAX)
      READ(1) T,NTCYCL,VMAG,/DIR,TTLDR,RC,OC
      READ(1)  JP,(PTCRD(I,1),PTCRD(I,2),I=1,JP)
      RETURN
  506 CONTINUE
      CALL BOMB
  504 RETURN
      END                                                       COMB0562
```

```
      SUBROUTINE BOMB
      COMMON PROPTY,GAMMA3,GAMMA1,GAMMA,DELR,DELQ,PO,PMAX,PR,ER,TAU,IMAX
     1,JMAX,T,DTDR,DTDQ,DELT,P,AO,ROO,NCYCMX,R
      DIMENSION PROPTY(32,108,4)
      COMMON/LE/FJMAX,IMAXM1,RAD(32),OVRLAY(12,3,32),OVRPRM(16,2,32)
      COMMON/CNTRE/COSTH(108),SINTH(108),THTA(108)
      COMMON/MIDDLE/VMAG,VDIR
      DIMENSION XR(201),ZR(201),ZP(201),ZV(201)
      EQUIVALENCE (ZR,OVRLAY),(ZV,OVRLAY(202)),(ZP,OVRLAY(403)),
     1(XR,OVRLAY(604))
      PZ=GAMMA*PO
      DELTA=0.5
      E=P/GAMMA1
      READ(5,100) N,BR,BO
      READ(5,200)(XR(I),ZP(I),ZR(I),ZV(I),I=1,N)
      NM=N-1
      XZ=BR*COS(BO)
      YZ=BR*SIN(BO)
      RMXS=0.04
      PROPTY(1,1,1)=1.0
      PROPTY(1,1,2)=0.0
      PROPTY(1,1,3)=0.0
      PROPTY(1,1,4)=E
      DO 50 I=2,IMAXM1
      DO 50 J=1,JMAX
      XP=RAD(I)*COSTH(J)-XZ
      YP=RAD(I)*SINTH(J)-YZ
      RS=XP*XP+YP*YP
      IF(RS.GT.RMXS) GO TO 40
      RP=SQRT(RS)
      DR=R*RP
      CALL SEEK(DR,XR,NM,IJ)
      IF(IJ)30,20,30
   20 PRINT 300,DR,XR(1)
      STOP
   30 CONTINUE
      QQ=(DR-XR(IJ))/(XR(IJ+1)-XR(IJ))
      QR=ZR(IJ)+(ZR(IJ+1)-ZR(IJ))*QQ
      QP=ZP(IJ)+(ZP(IJ+1)-ZP(IJ))*QQ
      QV=ZV(IJ)+(ZV(IJ+1)-ZV(IJ))*QQ
      QR=QR/ROO
      QP=QP/PZ
      QV=QV/AO
      Q=QV/RP
      XU=XP*Q
      YV=YP*Q
```

```
      U=XU*COSTH(J)+YV*SINTH(J)
      V=YV*COSTH(J)-XU*SINTH(J)
      PROPTY(I,J,1)=QR
      PROPTY(I,J,2)=PROPTY(I,J,1)*U
      PROPTY(I,J,3)=PROPTY(I,J,1)*V
      PROPTY(I,J,4)=QP/GAMMA1+0.5*PROPTY(I,J,1)*(U*U+V*V)
      GO TO 50
   40 CONTINUE
      PROPTY(I,J,1)=1.0
      PROPTY(I,J,2)=0.0
      PROPTY(I,J,3)=0.0
      PROPTY(I,J,4)=E
   50 CONTINUE
      VMAG=0.0
      VDIR=0.0
      RETURN
  100 FORMAT(I10,2E10.7)
  200 FORMAT(4E15.8)
  300 FORMAT(11H SEEK ERROR,2E20.10)
      END
```

```
      SUBROUTINE CONVRG(FACTOR)                                         COMB0355 999999
C*********  COMPUTES A TIME INCREMENT TO BE USED IN DIFFERENCE EQUATION COMB0356
C*********  BASED ON C.F.L. CONDITION            *********************COMB035
      COMMON PROPTY,GAMMA3,GAMMA1,GAMMA,DELR,DELO,PO,PMAX,PR,ER,TAU,IMAXCOMB0358
     1,JMAX,T,DTDR,DTDO,DT,P,AO,ROO,NCYCMX,R
      DIMENSION PROPTY(32,108,4)
      COMMON/LE/FJMAX,IMAXM1,RAD(32),OVRLAY(12,3,32),OVRPRM(16,2,32)
      DT=0.37/(16.*.75/3.9445)                                          COMB0364
      DT=0.6/(16.*.75/3.9445)                                           COMB0365
      DT=DT*FACTOR                                                      COMB0366
      SQ=1./SQRT (2.)                                                   COMB036
      DO 1 I=2,IMAXM1
      DO 1 J=1,JMAX                                                     COMB0369
      IF(I.EQ.1.AND.J.GE.2)GO TO 1                                      COMB0370
      VELSQR=(PROPTY(I,J,2)**2+PROPTY(I,J,3)**2)/PROPTY(I,J,1)**2       COMB0371
      IF(VELSQR) 8,8,2                                                  COMB0372
    2 BG=PROPTY(I,J,4)/PROPTY(I,J,1)-0.5*VELSQR                         COMB0373
      IF(BG) 8,7,7                                                      COMB0374
    7 DENOM=SQRT (GAMMA*GAMMA1*BG)                                      COMB0375
      TERM=SQ/(DENOM+SQRT (VELSQR))
      TERMO=TERM*FACTOR*DELO *RAD(I)
      TERMR=TERM*FACTOR*DELR
      DTE=DT
      DT=AMIN1(DT,TERMR,TERMO)                                          COMB0380
      IF(DT-DTE) 150,8,8
  150 IT=I                                                              COMB0383
      JT=J                                                              COMB0384
    8 CONTINUE                                                          COMB0385
    1 CONTINUE                                                          COMB0386
      DTDR=DT/DELR                                                      COMB038
      DTDO=DT/DELO                                                      COMB0388
      RETURN                                                            COMB0392
      END                                                               COMB0393
```

```
      SUBROUTINE MVPNT(RC,OC)
      COMMON PROPTY,GAMMA3,GAMMA1,GAMMA,DELR,DELO,PO,PMAX,PR,ER,TAU,IMAX
     1,JMAX,T,DTDR,DTDO,DT,P,AO,ROO   ,NCY
      DIMENSION PROPTY(32,108,4)
      COMMON/LE/FJMAX,IMAXM1,RAD(32),OVRLAY(12,3,32),OVRPRM(16,2,32)
      COMMON/CNTRE/COSTH(108),SINTH(108),THTA(108)
      DATA PI,TWOPI/3.14159265,6.28318531/
      I=INT(RC/DELR)+1
      IF(I.EQ.1) I=2
      J=INT(OC/DELO)+1
      IF(J.GT.JMAX) J=JMAX
      JP1=J+1
      IF(JP1.GT.JMAX) JP1=1
      U1=PROPTY(I,J,2)/PROPTY(I,J,1)
      U2=PROPTY(I+1,J,2)/PROPTY(I+1,J,1)
      U3=PROPTY(I,JP1,2)/PROPTY(I,JP1,1)
      U4=PROPTY(I+1,JP1,2)/PROPTY(I+1,JP1,1)
      V1=PROPTY(I,J,3)/PROPTY(I,J,1)
      V2=PROPTY(I+1,J,3)/PROPTY(I+1,J,1)
      V3=PROPTY(I,JP1,3)/PROPTY(I,JP1,1)
      V4=PROPTY(I+1,JP1,3)/PROPTY(I+1,JP1,1)
      A=(RC-RAD(I))/DELR
      B=(OC-THTA(J))/DELO
      C=A*B
      U=U1+A*(U2-U1)+B*(U3-U1)+C* (U4-U3-U2+U1)
      V=V1+A*(V2-V1)+B*(V3-V1)+C*(V4-V3-V2+V1)
      OC=OC+V*DT/RC
      RC=RC+U*DT
      IF(RC.GT. 1. ) RC=1.
      IF(RC.GE.0.) GO TO 10
      RC=-RC
      OC=OC+PI
   10 CONTINUE
      IF(OC.LT.0.0) OC=OC+TWOPI
      IF(OC.GE.TWOPI) OC=OC-TWOPI
      RETURN
      END
```

```
                SUBROUTINE FRSTRO
        C*********  COMPUTES VALUE OF W(T+DELT) ON THE CIRCLE R=DELR           COMB
        C*********  THE FIRST ROW OF NET POINTS CONTIGUOUS TO THE CENTER  ******COMB
000002          COMMON PROPTY,GAMMA3,GAMMA1,GAMMA,DELR,DELO,PO,PMAX,PR,ER,TAU,IMAXCOMB
               1,JMAX,T,DTDR,DTDO,DT,P,AO,ROO,NCYCMX,R
000002          DIMENSION PROPTY(32,108,4)
000002          COMMON/LE/FJMAX,IMAXM1,RAD(32),OVRLAY(12,3,32),OVRPRM(16,2,32)
000002          COMMON/CNTRE/COSTH(108),SINTH(108),THTA(108)
000002          COMMON/MIDDLE/VMAG,VDIR
000002          COMMON/FRS/FRSAV(4,108,3)
000002          RHO=PROPTY(1,1,1)
000004          E=PROPTY(1,1,4)
000005          RHOU=RHO*VMAG
000007          DO 20 J=1,JMAX
000011          DO 15 I=1,3
000012          DO 10 K=1,4
000013       10 FRSAV(K,J,I)=PROPTY(I,J,K)
000030       15 CONTINUE
000032          TH=VDIR-THTA(J)
000035          PROPTY(1,J,1)=RHO
000040          PROPTY(1,J,2)=RHOU*COS(TH)
000045          PROPTY(1,J,3)=RHOU*SIN(TH)
000052          PROPTY(1,J,4)=E
000054          DO 20 K=1,4
000055          PROPTY(1,J,K)=0.5*(PROPTY(1,J,K)+PROPTY(2,J,K))
000067       20 PROPTY(3,J,K)=0.5*(PROPTY(2,J,K)+PROPTY(3,J,K))
000102          DR=DELR
000104          DTR=DTDR
000105          DELR=0.5*DELR
000106          DTDR=2.0*DTDR
000107          RAD(1)=DELR
000110          RAD(2)=DR
000112          RAD(3)=DELR+DR
000113          DO 475 I=1,3
000114          DO 475 K=1,4
000115      475 PROPTY(I,107,K)=PROPTY(I,1,K)
000131          DO 480 I=1,3
000132          CALL VECTFR(I,0,OVRLAY(1,2,I))
000137          CALL VECTFR(I,1,OVRLAY(1,3,I))
000144          CALL VECTRG(I,0,OVRLAY(5,2,I))
000151          CALL VECTRG(I,1,OVRLAY(5,3,I))
000156          CALL VECTRS(I,0,OVRLAY(9,2,I))
000163          CALL VECTRS(I,1,OVRLAY(9,3,I))
000170      480 CONTINUE
000172          J=0
000173          DO 485 I=1,2
000175          I1=I+1
000177          CALL QUAD1(I,J,OVRPRM(1,2,I1),I)
000204          CALL TEMPFW(OVRPRM(1,2,I1),OVRPRM(5,2,I1),I1,J,1)
000214          CALL TEMPGW(OVRPRM(1,2,I1),OVRPRM(9,2,I1))
000222          CALL TEMPSW(OVRPRM(1,2,I1),OVRPRM(13,2,I1))
000230      485 CONTINUE
```

```
000232          DO 510 J=1,JMAX
000234          JP1=J+1
000236          DO 500 I=1,3
000237          DO 490 K=1,12
000240          OVRLAY(K,1,I)=OVRLAY(K,2,I)
000246      490 OVRLAY(K,2,I)=OVRLAY(K,3,I)
000253          CALL VECTFR(I,JP1,OVRLAY(1,3,I))
000257          CALL VECTRG(I,JP1,OVRLAY(5,3,I))
000264          CALL VECTRS(I,JP1,OVRLAY(9,3,I))
000271      500 CONTINUE
000273          DO 501 I=1,2
000275          I1=I+1
000277          DO 495 K=1,16
000300      495 OVRPRM(K,1,I1)=OVRPRM(K,2,I1)
000311          CALL QUAD1(I,J,OVRPRM(1,2,I1),I)
000315          CALL TEMPFW(OVRPRM(1,2,I1),OVRPRM(5,2,I1),I1,J,1)
000325          CALL TEMPGW(OVRPRM(1,2,I1),OVRPRM(9,2,I1))
000333          CALL TEMPSW(OVRPRM(1,2,I1),OVRPRM(13,2,I1))
000341      501 CONTINUE
000343          CALL GENPT(2,J)
000345          IF(J.NE.1) GO TO 510
000347          DO 505 I=1,3
000351          DO 505 K=1,4
000352          PROPTY(I,108,K)=PROPTY(I,1,K)
000360      505 PROPTY(I,1,K)=PROPTY(I,107,K)
000367      510 CONTINUE
000372          DO 515 I=1,3
000373          DO 515 K=1,4
000374      515 PROPTY(I,1,K)=PROPTY(I,108,K)
000410          DELR=DR
000411          DTDR=DTR
000413          RAD(1)=0.0
000414          RAD(2)=DELR
000415          RAD(3)=DELR+DELR
000416          DO 40 J=1,JMAX
000420          DO 40 K=1,4
000421          PROPTY(1,J,K)=FRSAV(K,J,1)
000430          FRSAV(K,J,1)=PROPTY(2,J,K)
000436          PROPTY(2,J,K)=FRSAV(K,J,2)
000443          PROPTY(3,J,K)=FRSAV(K,J,3)
000451       40 CONTINUE
000455          RETURN
000455          END
```

```
      SUBROUTINE VECTFR(II,JJ,R)                                        COMB0395
C*********  COMPUTES THE VALUE OF THE VECTOR FUNCTUON F(W) WHERE W      COMB0396
C*********  IS THE VECTOR OF CONSERVATION VARIABLES  ******************COMB039
      COMMON PROPTY,GAMMA3,GAMMA1,GAMMA,DELR,DELO,PO,PMAX,PR,ER,TAU,IMAXCOMB0398
     1,JMAX,T,DTDR,DTDO,DT,P,AO,ROO,NCYCMX,RR
      DIMENSION PROPTY(32,108,4)
      COMMON/LE/FJMAX,IMAXM1,RAD(32),OVRLAY(12,3,32),OVRPRM(16,2,32)
      DIMENSION R(4)                                                    COMB0404
      I=II                                                              COMB0405
      J=JJ                                                              COMB0406
      IF(J.LE.JMAX) GO TO 7                                             COMB040
      J=1                                                               COMB0408
    7 IF(J.EQ.0) J=JMAX                                                 COMB0409
      X=PROPTY(I,J,2)/PROPTY(I,J,1)                                     COMB0411
      Y=PROPTY(I,J,3)/PROPTY(I,J,1)                                     COMB0412
C       Z IS DISTANCE FROM ORIGIN                                       COMB0413
      Z=RAD(I)
      R(1)=-PROPTY(I,J,2)*Z                                             COMB0416
      R(2)=PROPTY(I,J,2)*GAMMA3*.5*X-GAMMA1*(PROPTY(I,J,4)-.5*Y*PROPTY  COMB041
     1(I,J,3))                                                          COMB0418
      R(2)=R(2)*Z                                                       COMB0419
      R(3)=-X*PROPTY(I,J,3)*Z                                           COMB0420
      R(4)=-(X*PROPTY(I,J,4)*GAMMA-GAMMA1*0.5*PROPTY(I,J,2)*(X**2+Y**2))COMB0421
     1*Z                                                                COMB0422
      RETURN                                                            COMB0423
      END                                                               COMB0424
      SUBROUTINE VECTRG(II,JJ,R)                                        COMB0426
C*********  COMPUTES THE VALUE OF THE VECTOR FUNCTUON G(W) WHERE W      COMB042
C*********  IS THE VECTOR OF CONSERVATION VARIABLES  ******************COMB0428
      COMMON PROPTY,GAMMA3,GAMMA1,GAMMA,DELR,DELO,PO,PMAX,PR,ER,TAU,IMAXCOMB0429
     1,JMAX,T,DTDR,DTDO,DT,P,AO,ROO,NCYCMX,RR
      DIMENSION PROPTY(32,108,4)
      COMMON/LE/FJMAX,IMAXM1,RAD(32),OVRLAY(12,3,32),OVRPRM(16,2,32)
      DIMENSION R(4)                                                    COMB0435
      I=II                                                              COMB0436
      J=JJ                                                              COMB043
      IF(J.LE.JMAX) GO TO 19                                            COMB0438
      J=1                                                               COMB0439
   19 IF(J.EQ.0) J=JMAX                                                 COMB0440
    9 X=PROPTY(I,J,3)/PROPTY(I,J,1)                                     COMB0443
    2 Y=PROPTY(I,J,2)/PROPTY(I,J,1)                                     COMB0446
    5 R(1)=-PROPTY(I,J,3)                                               COMB0458
      R(2)=R(1)*Y                                                       COMB0459
```

```
      R(3)=-0.5*GAMMA3*R(1)*X-GAMMA1*(PROPTY(I,J,4)-0.5*Y*PROPTY(I,J,2))COMB0460
      R(4)=-(GAMMA*X*PROPTY(I,J,4)-GAMMA1*0.5*PROPTY(I,J,3)*(X**2+Y**2))COMB0461
      RETURN                                                            COMB0462
      END                                                               COMB0463
      SUBROUTINE VECTRS(II,JJ,S)                                        COMB0583
C*********  COMPUTES THE VALUE OF THE VECTOR FUNCTUON S(W) WHERE W      COMB0584
C*********  IS THE VECTOR OF CONSERVATION VARIABLES  ******************COMB0585
      COMMON PROPTY,GAMMA3,GAMMA1,GAMMA,DELR,DELO,PO,PMAX,PR,ER,TAU,IMAXCOMB0586
     1,JMAX,T,DTDR,DTDO,DT,P,AO,ROO,NCYCMX,R
      DIMENSION PROPTY(32,108,4)
      COMMON/LE/FJMAX,IMAXM1,RAD(32),OVRLAY(12,3,32),OVRPRM(16,2,32)
      COMMON/EDOT/EKF
      COMMON/NCPLOT/NTCYCL,ISTART
      DIMENSION S(4),W(4)
      I=II                                                              COMB0593
      J=JJ                                                              COMB0594
      IF(J.LE.JMAX) GO TO 7                                             COMB0595
      J=1                                                               COMB0596
    7 IF(J.EQ.0) J=JMAX                                                 COMB059
      X=PROPTY(I,J,3)/PROPTY(I,J,1)                                     COMB0598
      Y=PROPTY(I,J,2)/PROPTY(I,J,1)                                     COMB0599
      PRS=GAMMA1*(PROPTY(I,J,4)-0.5*PROPTY(I,J,1)*(Y**2+X**2))
      IF(PRS.LT.0.0) GO TO 50
      S(1)=0.0
      S(2)=PROPTY(I,J,1)*X**2+PRS
      S(3)=-PROPTY(I,J,1)*X*Y                                           COMB0603
      S(4)=0.0
      RETURN                                                            COMB0605
   50 CONTINUE
      WRITE(6,707)((M,L,(PROPTY(L,M,K),K=1,4 ),M=1,JMAX),L=1,IMAX)
     2,J,I,(PROPTY(I,J,K),K=1,4),NTCYCL
  707 FORMAT(1H0//(2I4,4E23.7))
      STOP
      END                                                               COMB0606
      SUBROUTINE QUAD1(II,JJ,TEMPW1,LL)                                 COMB0650
C*********  COMPUTES W..SQUIGGLE(T+DELT), THE TEMPORARY VALUE          COMB0651
C*********  OF THE CONSERVATION VECTOR  ******************************COMB0652
      COMMON PROPTY,GAMMA3,GAMMA1,GAMMA,DELR,DELO,PO,PMAX,PR,ER,TAU,IMAXCOMB0653
     1,JMAX,T,DTDR,DTDO,DT,P,AO,ROO,NCYCMX,R
      DIMENSION PROPTY(32,108,4)
      COMMON/LE/FJMAX,IMAXM1,RAD(32),OVRLAY(12,3,32),OVRPRM(16,2,32)
      DIMENSION AVEW(4),TEMPW1(4),AVES(4)
      L=LL                                                              COMB0660
      I=II                                                              COMB0661
      J=JJ                                                              COMB0662
      IF(L.EQ.IMAXM1) GO TO 10
```

```
      HDELR=0.5*DELR
      IF(J.LE.JMAX) GO TO 7                                             COMB0663
      J=1                                                               COMB0664
    7 IF(J.EQ.0) J=JMAX                                                 COMB0665
      JP1=J+1                                                           COMB0666
      IF(JP1.LE.JMAX) GO TO 8                                           COMB066
      JP1=1                                                             COMB0668
    8 IF(JP1.EQ.0)JP1=JMAX                                              COMB0669
      DO 1 K=1,4                                                        COMB0670
                             AVES(K)=0.25*(OVRLAY(K+8,2,L)+OVRLAY(K+8,3,L)
     1+OVRLAY(K+8,3,L+1)+OVRLAY(K+8,2,L+1))*DT
    1 AVEW(K)=0.25*(PROPTY(I,J,K)+PROPTY(I+1,J,K)+PROPTY(I+1,JP1,K)+PROPCOMB0671
     XTY(I,JP1,K))                                                      COMB0672
      DO 2 K=1,4                                                        COMB0673
    2 TEMPW1(K)=AVEW(K)+((DTDR*(OVRLAY(K,2,L+1)-OVRLAY(K,2,L)+OVRLAY(K,3
     1,L+1)-OVRLAY(K,3,L))+DTDO*(OVRLAY(K+4,3,L)-OVRLAY(K+4,2,L)+OVRLAY(
     2K+4,3,L+1)-OVRLAY(K+4,2,L+1)))*.5+AVES(K))/(RAD(I)+HDELR)
      GO TO 13
   10 RADE=RAD(I)+DELR/2.
      RADI=RADE-DELR
      TEMPW1(1)=OVRPRM(1,2,I)*RADI/RADE
      TEMPW1(2)=-TEMPW1(1)*(OVRPRM(2,2,I)/OVRPRM(1,2,I))
      TEMPW1(3)=OVRPRM(3,2,I)
      EMSQ=OVRPRM(2,2,I)*OVRPRM(2,2,I)
      ENSQ=OVRPRM(3,2,I)*OVRPRM(3,2,I)
      EMNR=(EMSQ+ENSQ)/OVRPRM(1,2,I)
      PINTS=GAMMA1*(OVRPRM(4,2,I)-.5*EMNR)
      PEXTS=PINTS+DELR/RADI*ENSQ/OVRPRM(1,2,I)
      TEMPW1(4)=PEXTS/GAMMA1+.5*(TEMPW1(2)**2+TEMPW1(3)**2)/TEMPW1(1)
   13 RETURN
      END                                                               COMB0680
      SUBROUTINE TEMPFW(WTEMP,FOFW,II,JJ,IINDEX)
C*********  COMPUTES F(W..SQUIGGLE) WHERE W..SQUIGGLE IS THE TEMPORARY  COMB0609
C*********  VALUE OF THE CONSERVATION VECTOR      **********************COMB0610
      COMMON PROPTY,GAMMA3,GAMMA1,GAMMA,DELR,DELO,PO,PMAX,PR,ER,TAU,IMAXCOMB0611
     1,JMAX,T,DTDR,DTDO,DT,P,AO,ROO,NCYCMX,R
      DIMENSION PROPTY(32,108,4)
      COMMON/LE/FJMAX,IMAXM1,RAD(32),OVRLAY(12,3,32),OVRPRM(16,2,32)
      DIMENSION WTEMP(4),FOFW(4)                                        COMB061
      JJJ=II-1
      Z=FLOAT(JJJ)+FLOAT(IINDEX)/2.
      Z=Z*DELR
      X=WTEMP(2)/WTEMP(1)                                               COMB0621
      Y=WTEMP(3)/WTEMP(1)                                               COMB0622
      FOFW(1)=-WTEMP(2)*Z                                               COMB0623
      FOFW(2)=WTEMP(2)*GAMMA3*0.5*X-GAMMA1*(WTEMP(4)-0.5*Y*WTEMP(3))    COMB0624
```

```
      FOFW(2)=FOFW(2)*Z                                                 COMB0625
      FOFW(3)=-X*WTEMP(3)*Z                                             COMB0626
      FOFW(4)=-(X*WTEMP(4)*GAMMA-GAMMA1*0.5*WTEMP(2)*(X**2+Y**2))*Z     COMB062
      RETURN                                                            COMB0628
      END                                                               COMB0629
      SUBROUTINE TEMPGW(WTEMP,GOFW)                                     COMB0631
C*********  COMPUTES G(W..SQUIGGLE) WHERE W..SQUIGGLE IS THE TEMPORARY  COMB0632
C*********  VALUE OF THE CONSERVATION VECTOR     ***********************COMB0633
      COMMON PROPTY,GAMMA3,GAMMA1,GAMMA,DELR,DELO,PO,PMAX,PR,ER,TAU,IMAXCOMB0634
     1,JMAX,T,DTDR,DTDO,DT,P,AO,ROO,NCYCMX,R
      DIMENSION PROPTY(32,108,4)
      COMMON/LE/FJMAX,IMAXM1,RAD(32),OVRLAY(12,3,32),OVRPRM(16,2,32)
      DIMENSION WTEMP(4),GOFW(4)                                        COMB0640
      X=WTEMP(3)/WTEMP(1)                                               COMB0641
      Y=WTEMP(2)/WTEMP(1)                                               COMB0642
      GOFW(1)=-WTEMP(3)                                                 COMB0643
      GOFW(2)=GOFW(1)*Y                                                 COMB0644
      GOFW(3)=-0.5*GAMMA3*GOFW(1)*X-GAMMA1*(WTEMP(4)-0.5*Y*WTEMP(2))    COMB0645
      GOFW(4)=-(GAMMA*X*WTEMP(4)-GAMMA1*0.5*WTEMP(3)*(X**2+Y**2))       COMB0646
      RETURN                                                            COMB064
      END                                                               COMB0648
      SUBROUTINE TEMPSW(WTEMP,SOFW)                                     COMB0566
C*********  COMPUTES S(W..SQUIGGLE) WHERE W..SQUIGGLE IS THE TEMPORARY  COMB0564
C*********  VALUE OF THE CONSERVATION VECTOR     ***********************COMB0565
      COMMON PROPTY,GAMMA3,GAMMA1,GAMMA,DELR,DELO,PO,PMAX,PR,ER,TAU,IMAXCOMB056
     1,JMAX,T,DTDR,DTDO,DT,P,AO,ROO,NCYCMX,R
      DIMENSION PROPTY(32,108,4)
      COMMON/LE/FJMAX,IMAXM1,RAD(32),OVRLAY(12,3,32),OVRPRM(16,2,32)
      COMMON/EDOT/EKF
      COMMON/NCPLOT/NTCYCL,ISTART
      DIMENSION WTEMP(4),SOFW(4)                                        COMB0573
      X=WTEMP(3)/WTEMP(1)                                               COMB0574
      Y=WTEMP(2)/WTEMP(1)                                               COMB0575
      PRS=GAMMA1*(WTEMP(4)-0.5*WTEMP(1)*(Y**2+X**2))
      IF(PRS.LT.0.0) GO TO 50
      SOFW(1)=0.0
      SOFW(2)=WTEMP(1)*X**2+PRS
      SOFW(3)=-WTEMP(1)*X*Y                                             COMB0578
      SOFW(4)=0.0
      RETURN                                                            COMB0580
   50 CONTINUE
      WRITE(6,707) NTCYCL,WTEMP
  707 FORMAT(28HONEGATIVE PRESSURE IN TEMPSW/I8,4E23.7)
      STOP
      END                                                               COMB0581
      SUBROUTINE GENPT(IF,IL)
```

```
C*********  COMPUTES THE VALUE OF W(T+DELT) AT REGULAR MESH POINTS       COMB0724
C*********  AND AT BOUNDARY POINTS     *******************************COMB0725
      COMMON PROPTY,GAMMA3,GAMMA1,GAMMA,DELR,DELO,PO,PMAX,PR,ER,TAU,IMAXCOMB0726
     1,JMAX,T,DTDR,DTDO,DT,P,AO,ROO,NCYCMX,R
      DIMENSION PROPTY(32,108,4)
      COMMON/LE/FJMAX,IMAXM1,RAD(32),OVRLAY(12,3,32),OVRPRM(16,2,32)
      DIMENSION FDRVBR(4),GDRVBR(4),SAVEBR(4)
      I=IF
      M=IL
      JM1=1
      J=2
      JP1=3
      IM=I-1
      IP=I+1
      DO 4 K=1,4                                                        COMB0766
      K4=K+4
    1 FDRVBR(K)=(OVRPRM(K4,J,IP)-OVRPRM(K4,J,I)+OVRPRM(K4,JM1,IP)-
     1OVRPRM(K4,JM1,I))
      K8=K+8
    2 GDRVBR(K)=(OVRPRM(K8,J,IP)-OVRPRM(K8,JM1,IP)+OVRPRM(K8,J,I)-
     1OVRPRM(K8,JM1,I))
      K12=K+12
    3 SAVEBR(K)=.25*(OVRPRM(K12,J,IP)+OVRPRM(K12,J,I)+OVRPRM(K12,JM1,IP)
     1+OVRPRM(K12,JM1,I))
                              OVRKJI=0.5*(OVRLAY(K8,J,IP)+OVRLAY(K8,J,IM))
    4 PROPTY(I,M,K)=PROPTY(I,M,K)+((DTDR*((OVRLAY(K,J,IP)-OVRLAY(K,J,IM)
     1)+FDRVBR(K))*.25+DTDO*((OVRLAY(K4,JP1,I)-OVRLAY(K4,JM1,I))+GDRVBR
     2(K))*.25+DT*(OVRKJI+SAVEBR(K))*.5)/RAD(I))
      RETURN                                                            COMB077
      END                                                               COMB0778
      SUBROUTINE SEEK(E,EOUT,NOUT,I)
      DIMENSION EOUT(10)
      IF(EOUT(1)-E) 15,10,25
   10 I=1 $ RETURN
   15 PRINT 20,E,EOUT(1) $ I=0 $ RETURN
   20 FORMAT(11H SEEK ERROR,2E20.10)
   25 ITOP=1 $ IBOT=NOUT+1
   30 IF(IBOT-ITOP-1 ) 45,40,45
   40 I=ITOP $ RETURN
   45 I=ISHIFT(IBOT+ITOP,-1) $ IF(EOUT(I)-E) 50,55,60
   50 IBOT=I $ GO TO 30
   55 I=I-1 $ RETURN
   60 ITOP=I $ GO TO 30
      END
      SUBROUTINE CENTER
C*********  COMPUTES THE VALUE OF W(T+DELT) AT THE CENTER OF THE         COMB0683
```

```
C*********  CYLINDER BY SCALER AND VECTOR AVERAGING    *****************COMB0684
      COMMON PROPTY,GAMMA3,GAMMA1,GAMMA,DELR,DELO,PO,PMAX,PR,ER,TAU,IMAXCOMB0685
     1,JMAX,T,DTDR,DTDO,DT,P,AO,ROO,NCYCMX,R
      DIMENSION PROPTY(32,108,4)
      COMMON/LE/FJMAX,IMAXM1,RAD(32),OVRLAY(12,3,32),OVRPRM(16,2,32)
      COMMON/CNTRE/COSTH(108),SINTH(108),THTA(108)
      COMMON/MIDDLE/VMAG,VDIR
      COMMON/TLDR/TTLDR
      COMMON/EDOT/EKF
      COMMON/FRS/FRSAV(4,108,3)
      DIMENSION FNUM(4),GNUM(4),WXY(4,108,2),FXY(4,108,2),GXY(4,108,2),
     1X(108),Y(108)
      EQUIVALENCE (WXY,FRSAV(433)),(FXY,OVRPRM),(GXY,OVRLAY),
     1(X,OVRLAY(865)),(Y,OVRLAY(973))
      DATA PI,TWOPI/3.1415926,6.2831852/
      DR2=.5*DELR
      DO 10 J=1,JMAX
      X(J)=DELR*COSTH(J)
      Y(J)=DELR*SINTH(J)
      WXY(1,J,1)=PROPTY(2,J,1)
      WXY(4,J,1)=PROPTY(2,J,4)
      WXY(2,J,1)=PROPTY(2,J,2)*COSTH(J)-PROPTY(2,J,3)*SINTH(J)
      WXY(3,J,1)=PROPTY(2,J,2)*SINTH(J)+PROPTY(2,J,3)*COSTH(J)
      WXY(1,J,2)=FRSAV(1,J,1)
      WXY(4,J,2)=FRSAV(4,J,1)
      WXY(2,J,2)=FRSAV(2,J,1)*COSTH(J)-FRSAV(3,J,1)*SINTH(J)
      WXY(3,J,2)=FRSAV(2,J,1)*SINTH(J)+FRSAV(3,J,1)*COSTH(J)
      DO 20 K=1,4
      PROPTY(2,J,K)=FRSAV(K,J,1)
   20 CONTINUE
      DO 25 I=1,2
      CALL TEMPFW(WXY(1,J,I),FXY(1,J,I),1,J,1)
      CALL TEMPGW(WXY(1,J,I),GXY(1,J,I) )
      DO 25 K=1,4
   25 FXY(K,J,I)=FXY(K,J,I)/DR2
   10 CONTINUE
      DO 35 K=1,4
      DO 30 J=1,JMAX
      FXY(K,J,1)=0.5*(FXY(K,J,1)+FXY(K,J,2))
      GXY(K,J,1)=0.5*(GXY(K,J,1)+GXY(K,J,2))
   30 CONTINUE
      FNUM(K)=0.
      GNUM(K)=0.
   35 CONTINUE
      DENOM=0.
      DO 40 J=1,JMAX
```

```
      JP1=J+1
      IF(J.EQ.JMAX)JP1=1
      YDIF=Y(JP1)-Y(J)
      DENOM=(X(JP1)+X(J))*YDIF+DENOM
      XDIF=X(JP1)-X(J)
      DO 50 K=1,4
      FNUM(K)=FNUM(K)+(FXY(K,J,1)+FXY(K,JP1,1))*YDIF
   50 GNUM(K)=GNUM(K)+(GXY(K,J,1)+GXY(K,JP1,1))*XDIF
   40 CONTINUE
      DO 55 K=1,4
      FNUM(K)=FNUM(K)/DENOM
      GNUM(K)=GNUM(K)/DENOM
      PROPTY(1,1,K)=PROPTY(1,1,K)+DT*(FNUM(K)-GNUM(K))
   55 CONTINUE
      IF(ABS(PROPTY(1,1,2)).LE.1.0E-14) PROPTY(1,1,2)=0.0
      IF(ABS(PROPTY(1,1,3)).LE.1.0E-14) PROPTY(1,1,3)=0.0
      QA=VDIR
      VMAG=SQRT(PROPTY(1,1,2)**2+PROPTY(1,1,3)**2)/PROPTY(1,1,1)
      IF(VMAG.NE.0.0) VDIR=ATAN2(PROPTY(1,1,3),PROPTY(1,1,2))
      QA=ABS(VDIR-QA)
      IF(QA.GT.PI) QA=TWOPI-QA
      TTLDR=TTLDR+QA
      RETURN                                                            COMB0720
      END                                                               COMB0721
      SUBROUTINE CHARGE
C*********  COMPUTES BOUNDARY CONDITIONS AT THE PERIPHERY
C*********  OF THE CYLINDRICAL SURFACE    ******************************COMB0918
      COMMON PROPTY,GAMMA3,GAMMA1,GAMMA,DELR,DELO,PO,PMAX,PR,ER,TAU,IMAX
     1,JMAX,T,DTDR,DTDO,DT,P,AO,ROO,NCYCMX,R
      DIMENSION PROPTY(32,108,4)
      COMMON/LE/FJMAX,IMAXM1,RAD(32),OVRLAY(12,3,32),OVRPRM(16,2,32)
      IMAXM2=IMAX-2
      RINT=RAD(IMAX-2)
      REXT=RAD(IMAX)
      RQUOT=RINT/REXT
      DO 10 J=1,JMAX
      PROPTY(IMAX,J,1)=PROPTY(IMAXM2,J,1)*RQUOT
      PROPTY(IMAX,J,2)=-PROPTY(IMAX,J,1)*(PROPTY(IMAXM2,J,2)/PROPTY(IMAX
     X M2,J,1))
      PROPTY(IMAX,J,3)=PROPTY(IMAXM2,J,3)
      RUSQ=(PROPTY(IMAXM2,J,2)**2)/PROPTY(IMAXM2,J,1)
      RVSQ=(PROPTY(IMAXM2,J,3)**2)/PROPTY(IMAXM2,J,1)
      PINT=GAMMA1*(PROPTY(IMAXM2,J,4)-.5*(RUSQ+RVSQ))
      PEXT=PINT+2.*DELP/RINT*RVSQ
      PROPTY(IMAX,J,4)=PEXT/GAMMA1+.5*(PROPTY(IMAX,J,2)**2+PROPTY(IMAX,
     *J,3)**2)/PROPTY(IMAX,J,1)
```

```
   10 CONTINUE
      RETURN
      END
      SUBROUTINE PRTOUT(RTM)
C*********  PRINTS A TERSE VIEW OF THE FLOW FIELD ALONG THE            COMB1421
C***Q*****  RAYS 0,PI/2,PI,3*PI/2       *********************************COMB1422
      COMMON PROPTY,GAMMA3,GAMMA1,GAMMA,DELR,DELO,PO,PMAX,PR,ER,TAU,IMAXCOMB1423
     1,JMAX,T,DTDR,DTDO,DELT,P,AO,ROO,NCYCMX,R
      DIMENSION PROPTY(32,108,4)
      COMMON/NCPLOT/NTCYCL,ISTART,RC,OC
      COMMON/MIDDLE/VMAG,VDIR
      COMMON/PRTOPT/INTPNT,XTO,DETR(5),DTR(5),IDETR(5),PDM
      COMMON/TLDR/TTLDR
      DIMENSION DLP(5),DLPR(5)
      EQUIVALENCE(THTBAR,VDIR)
   75 LF=0
      L=1                                                              COMB1429
      ENREV=TTLDR/6.2831852
      WRITE(6,606) T,NTCYCL,ENREV,DELT,RTM
      TDM=XTO*T+TAU
      DO 100 J=1,5
      I=IDETR(J)
      P1=GAMMA1*(PROPTY(I,1,4)-0.5*(PROPTY(I,1,2)*PROPTY(I,1,2)
     1+PROPTY(I,1,3)*PROPTY(I,1,3))/PROPTY(I,1,1))
      P2=GAMMA1*(PROPTY(I+1,1,4)-0.5*(PROPTY(I+1,1,2)*PROPTY(I+1,1,2)
     1+PROPTY(I+1,1,3)*PROPTY(I+1,1,3))/PROPTY(I+1,1,1))
      DLP(J)=PDM*(P1+DTR(J)*(P2-P1))
      DLPR(J)=DLP(J)/PO
  100 DLP(J)=DLP(J)-PO
      WRITE(6,607) TDM,DETR,DLP,DLPR
  607 FORMAT(1H010X5HTIME=F8.5,9H MILLISEC/19H DISTANCE IN INCHES5E20.7/
     119H PRESS DIFF IN PSI 5E20.7/19H PRESSURE RATIO    5E20.7)
      IF(INTPNT.EQ.0) GO TO 1818
      J=1
      I=1
      WRITE(6,707) J,I,VMAG,VDIR
      WRITE(6,707)((J,I,(PROPTY(I,J,K),K=1,4),J=1,JMAX),I=1,IMAX)
  707 FORMAT(1X,2I4,4E23.7)
 1818 CONTINUE
      WRITE(6,6062)                                                    COMB1431
  606 FORMAT(12H1TOTAL TIME=E12.5,3X,17HNUMBER OF CYCLES=I4,3X,15HNUMBER
     1 OF REVS=F5.2,3X,10HTIME STEP=E12.5,3X,10HREAL TIME=F9.3///)
 6165 FORMAT(8X,5H RHO ,14X,3H U ,14X,3H V ,19X,10HINT ENERGY,8X,10H PRECOMB1434
     1SSURE ,10X,9H MACH NO.///)                                       COMB1435
 6061 FORMAT(1X,6E19.5)                                                COMB1479
 6062 FORMAT(1H0,20X,32HSOLUTION ALONG THE RAY THETA = 0///)            COMB1436
```

```
6063 FORMAT(1H0,20X,33HSOLUTION ALONG THE RAY THETA = PI///)          COMB143
6064 FORMAT(1H0,20X,35HSOLUTION ALONG THE RAY THETA = PI/2///)        COMB1438
6065 FORMAT(1H0,20X,37HSOLUTION ALONG THE RAY THETA = 3*PI/2///)      COMB1439
 402 CONTINUE
     WRITE(6,6165)                                                    COMB1432
     DO 1012 I=1,IMAX
     IF(L.NE.I.AND.I.EQ.1) GO TO 1013                                 COMB1441
     RHOX=PROPTY(I,L,1)                                               COMB1442
     UX=PROPTY(I,L,2)/RHOX                                            COMB1443
     VX=PROPTY(I,L,3)/RHOX                                            COMB1444
     EX=(PROPTY(I,L,4)/RHOX)-0.5*(UX**2+VX**2)                        COMB1445
     PX=GAMMA1*RHOX*EX                                                COMB1446
     XMX=SQRT((UX**2+VX**2)/(GAMMA*PX/RHOX))                          COMB144
     IF(L.EQ.1.AND.I.EQ.1) GO TO 1014                                 COMB1448
     GO TO 1011                                                       COMB1449
1014 TMPRO=RHOX                                                       COMB1450
     TMPEX=EX                                                         COMB1458
     TMPPX=PX                                                         COMB1459
     TMPXMX=XMX                                                       COMB1460
1013 CONTINUE
     XT=L-1
     THETSR=XT*DELO
     DIFFT=THTBAR-THETSR                                              COMB1455
     TMPUX=VMAG*COS(DIFFT)
     TMPVX=VMAG*SIN(DIFFT)
     WRITE(6,6061)TMPRO,TMPUX,TMPVX,TMPEX,TMPPX,TMPXMX
     GO TO 1012                                                       COMB1464
1011 WRITE(6,6061)           RHOX,UX,VX,EX,PX,XMX                     COMB1465
1012 CONTINUE                                                         COMB1466
     LF=LF+1
     GO TO(404,504,403,1001),LF
 404 CONTINUE
     L=JMAX/2+1                                                       COMB1469
     WRITE(6,6063)                                                    COMB1470
     GO TO 402                                                        COMB1472
 504 CONTINUE
     L=JMAX/4+1                                                       COMB1475
     WRITE(6,6064)                                                    COMB1476
     GO TO 402                                                        COMB1478
 403 CONTINUE
     L=3*(JMAX/4)+1                                                   COMB1482
     WRITE(6,6065)
     GO TO 402
1001 CONTINUE                                                         COMB1499
     RETURN                                                           COMB1500
     END                                                              COMB1501
```

```
      PROGRAM PRV(INPUT,OUTPUT,PUNCH)
      DIMENSION Y(3),F(3),Q(3)
      COMMON/COMPRV/GAMMA,DELTA
      EXTERNAL FPRV
      GAMMA=1.2
      RD=0.751666667
      PR1=113.5*144.0*32.17
      RHO1=0.53
      GAMMA1=GAMMA-1.0
      GAMMAP=GAMMA+1.0
      RZ=0.2*RD
      K=3
      P2P1=15.0
    5 CONTINUE
      PR2=P2P1*PR1
      SQ=(GAMMAP*PR2+GAMMA1*PR1)/(2.0*RHO1)
      S=SQRT(SQ)
      TZ=RZ/S*1000.0
      DELTA=S
      DMP=1.0/(32.17*144.0)
      AS=GAMMA*PR1/RHO1
      Q1=AS/SQ
      V=2.0*S/GAMMAP*(1.0-Q1)
      P=PR2
      R=GAMMAP*RHO1/(GAMMA1*(1.0+2.0*Q1/GAMMA1))
      L=5
      DX=0.001
   10 CONTINUE
      N=0
      Y(1)=P
      Y(2)=R
      Y(3)=V
      X=RZ
      H=-DX*X
      XMIN=-5.0*H
      PRINT 100,RD,RZ,DX,PR1,RHO1,SQ,S,AS,TZ
      PL=0.0
      PM=0.0
      PRINT 300
   20 CONTINUE
      CALL DEQ(X,H,3,Y,F,Q,FPRV)
      N=N+1
      IF(MOD(N,L).NE.1) GO TO 30
   25 CONTINUE
      SMV=ABS(DELTA-Y(3))
      A=SQRT(GAMMA*Y(1)/Y(2))
```

```
      EM=SMV/A
      VA=Y(3)/A
      YP=DMP*Y(1)
      PRINT 200,X,Y(1),Y(2),Y(3),A,EM,VA
      PRINT 400,YP,F(1),F(2),F(3)
      IF(PM.EQ.0.0)
     1PUNCH 500,X,YP,Y(2),Y(3)
      IF(PL.EQ.1.0) GO TO 40
   30 CONTINUE
      IF(X.GT.XMIN) GO TO 20
      IF(PM.EQ.3.0) GO TO 35
      IF(PM.EQ.2.0) GO TO 33
      IF(PM.EQ.1.0) GO TO 32
      PM=1.0
      XMIN=0.02*XMIN
      H=0.05*H
      L=10
      GO TO 20
   32 CONTINUE
      PM=2.0
      XMIN=0.01*XMIN
      H=0.01*H
      L=50
      GO TO 20
   33 CONTINUE
      PM=3.0
      XMIN=0.05*XMIN
      H=0.05*H
      L=2
      GO TO 20
   35 CONTINUE
      PL=1.0
      GO TO 25
   40 CONTINUE
      K=K-1
      IF(K.EQ.0) STOP
      IF(K.EQ.1) P2P1=3.85
      IF(K.EQ.2) P2P1=11.3
      CALL DEQSET
      GO TO 5
      RD=0.46
      PR1=313.5*144.0*32.17
      RHO1=1.464
  100 FORMAT(*1RD,RZ,DX,PR1,RHO1,SQ,S,AS,TZ*/(6E20.9))
  200 FORMAT(1H07E17.8)
  300 FORMAT(1H05X6HRADIUS11X1HP16X1HR16X1HV16X1HA16X1HM16X3HV/A)
```

```
  400 FORMAT(1H 7E17.8)
  500 FORMAT(4E15.8)
      END
      SUBROUTINE FPRV(X,N,Y,F)
      DIMENSION Y(3),F(3)
      COMMON/COMPRV/GAMMA,DELTA
      QLAM=X
      P=Y(1)
      R=Y(2)
      V=Y(3)
      Z=GAMMA*P/R
      AV=V/QLAM
      BV=V+DELTA
      CV=BV*BV-Z
      PP=-GAMMA*AV*BV/CV
      RR=PP/GAMMA
      F(1)=P*PP
      F(2)=R*RR
      F(3)=-AV-BV*RR
      RETURN
      END
```

APPENDIX D

```
      PROGRAM COMPLT(INPUT,OUTPUT,TAPE4,TAPE98,TAPE5=INPUT,TAPE6=OUTPUT)
      COMMON/CMPLT/PROPTY(12,36,4),RAD(12),THTA(36),SINTH(36),COSTH(36),
     1IMAX,IMAXM1,JMAX,NTCYCL,GAMMA,GAMMA1,T,TTLDR,VMAG,VDIR,LABLX(5),GD
      DIMENSION PTCRD(501,2)
      REWIND 4
      READ(4) RAD(1),RAD(2),RAD(3),RAD(4)
      RAD(1)=10H109904
      RAD(5)=0.0
      CALL PLOTS(999,RAD)
      CALL PLOT(8.5,0.0,-3)
   10 CONTINUE
      READ(4) Z,T,NTCYCL,AO,R,JP
      IF(EOF,4)80,20
   20 CONTINUE
      WRITE(6,100) Z,T,AO,R,JP,NTCYCL
      IF(Z)30,40,30
   30 CONTINUE
      READ(4) (PTCRD(I,1),PTCRD(I,2),I=1,JP)
      CALL PTPLT(PTCRD(1,1),PTCRD(1,2),JP)
      GO TO 10
   40 CONTINUE
      READ(4) VMAG,VDIR,TTLDR,GAMMA,PO,ROO,IMAX,JMAX,NPRPLT,
     1(((PROPTY(I,J,K),K=1,4),J=1,JMAX),I=1,IMAX)
      WRITE(6,200) GAMMA,PO,ROO,NPRPLT,IMAX,JMAX
      IMAXM1=IMAX-1
      FJMAX=JMAX
      GAMMA1=GAMMA-1.0
      GD=GAMMA1*GAMMA*PO
      DELR=1.0/FLOAT(IMAX-2)
      DELO=6.28318531/FJMAX
      RAD(1)=0.0
      DO 50 I=2,IMAX
   50 RAD(I)=RAD(I-1)+DELR
      DO 60 J=1,JMAX
      TH=J-1
      TH=TH*DELO
      THTA(J)=TH
      COSTH(J)=COS(TH)
   60 SINTH(J)=SIN(TH)
      CALL CALPLT
      IF(NPRPLT.EQ.0) CALL PRPLT(JP)
      GO TO 10
   80 CONTINUE
      CALL PLOT(0.0,0.0,999)
      STOP
  100 FORMAT(*0Z,T,AO,R,JP,NTCYCL*4E20.8,2I10)
  200 FORMAT(*0GAMMA,PO,ROO,NPRPLT,IMAX,JMAX*3E20.8,3I10)
      END
      SUBROUTINE CALPLT
      COMMON/CMPLT/PROPTY(12,36,4),RAD(12),THTA(36),SINTH(36),COSTH(36),
     1IMAX,IMAXM1,JMAX,NTCYCL,GAMMA,GAMMA1,T,TTLDR,VMAG,VDIR,LABLX(5),GD
      DIMENSION AA(12,36),CLEVELS(20),LABLY(3),HT(36),XX(361),YY(361),
     1UV(361),THT(361)
      EQUIVALENCE(HT,THTA)
      EXTERNAL FX,FY
      DATA PI,TWOPI/3.14159265,6.28318531/
```

```
      DATA RMAX,XYOR,DELXP,XYS/1.0,-1.125,.25,9./
      ENREV=TTLDR/TWOPI
      ENCODE(39,100,LABLX)T,NTCYCL,ENREV
  100 FORMAT(2HT=F7.4,I9,7H CYCLESF9.2,5H REVS)
      LABLY(1)=9HNUMERICAL
      LABLY(2)=8HSOLUTION
      LABLY(3)=0
      LABLX(5)=0
      Q=1.
      IMAXM2=IMAXM1-1
   85 CONTINUE
      DO 135 J=1,JMAX
      AA(1,J)=PROPTY(1,1,1)
  135 CONTINUE
      DO 140 I=2,IMAXM1
      DO 140 J=1,JMAX
      AA(I,J)=PROPTY(I,J,1)
  140 CONTINUE
      CALL CONTOUR(AA,12,IMAXM1,JMAX,IMAXM1,JMAX,0.,TWOPI,0.00,RMAX ,6.,
     *6.,-20,CLEVELS,11HPLOT OF RHO,LABLX,LABLY   ,FX,FY,Q)
      DO 145 J=1,JMAX
      AA(1   ,J)=GAMMA1*(PROPTY(1,1,4)-.5*(PROPTY(1,1,2)**2+PROPTY(1,1,3)
     ***2)/PROPTY(1,1,1))*GAMMA
  145 CONTINUE
      DO 150 I=2,IMAXM1
      DO 150 J=1,JMAX
      AA(I   ,J)=GAMMA1*(PROPTY(I,J,4)-.5*(PROPTY(I,J,2)**2+PROPTY(I,J,3)
     ***2)/PROPTY(I,J,1))*GAMMA
  150 CONTINUE
      CALL CONTOUR(AA,12,IMAXM1,JMAX,IMAXM1,JMAX,0.,TWOPI,0.00,RMAX ,6.,
     *6.,-20,CLEVELS,16HPLOT OF PRESSURE,LABLX,LABLY   ,FX,FY,Q)
      KK=1
      YY(KK)=0.
      XX(KK)=0.
      UV(KK)=VMAG
      THT(KK)=VDIR
      DO  91 I=2,IMAXM1
      DO 91 J=1,JMAX
      KK=KK+1
      UV(KK)=SQRT(PROPTY(I,J,2)**2+PROPTY(I,J,3)**2)/PROPTY(I,J,1)
      YY(KK)=SINTH(J)*RAD(I)
      XX(KK)=COSTH(J)*RAD(I)
      ATAKANG=0.0
      IF(UV(KK).EQ.0.0) GO TO 601
  602 ATAKANG=ATAN2(PROPTY(I,J,3),PROPTY(I,J,2))
  601 THT(KK)=HT(J)+ATAKANG
   91 CONTINUE
      CALL PLOT(2,,.58,-3)
      EMXY=0.
      KKK=IMAXM2*JMAX
      DO 160 I=1,KKK
      IF(UV(I)-EMXY)160,160,155
  155 EMXY=UV(I)
  160 CONTINUE
      IF(EMXY.EQ.0.0) RETURN
      CALL VECTORF(XX,YY,UV,THT,KK,XYOR,DELXP,XYOR,DELXP,EMXY,.5,XYS,XYS
```

```
     1,19HVECTOR PLOT OF UMAG,LABLX,LABLY)
      CALL PLOT(12.,-11.,-3)
  603 CONTINUE
      RETURN
      END
      FUNCTION FX(X)
      FX=X
      RETURN
      END
      FUNCTION FY(Y)
      FY=Y
      RETURN
      END
      SUBROUTINE PRPLT(II)
      COMMON/CMPLT/PROPTY(12,36,4),RAD(12),THTA(36),SINTH(36),COSTH(36),
     1IMAX,IMAXM1,JMAX,NTCYCL,GAMMA,GAMMA1,T,TTLDR,VMAG,VDIR,LABLX(5),GD
      DIMENSION X(36)
      I=II
      DO 18 J=1,JMAX
      X(J)=(PROPTY(I,J,4)-0.5*(PROPTY(I,J,2)**2+PROPTY(I,J,3)**2)/
     1PROPTY(I,J,1))*GD
   18 CONTINUE
      CALL PLOT(0,50,0.0,-3)
      M=INKPLOT(THTA,X,JMAX,LABLX   ,8HPRESSURE)
      CALL PLOT(1,50,0.0,-3)
      RETURN
      END
      SUBROUTINE PTPLT(X,Y,N)
      DIMENSION X(1),Y(1)
      DO 10 I=1,N
      YY=Y(I)
      XX=X(I)*COS(YY)
      YY=X(I)*SIN(YY)
      X(I)=XX
      Y(I)=YY
   10 CONTINUE
      M=INKPLOT(X,Y,N,1HX,1HY)
      CALL PLOT(2,0,0.0,-3)
      RETURN
      END
      FUNCTION INKPLOT (X,Y,N,XLABEL,YLABEL)                            PLT
C                                                                       PLT
C     IABS(INKPLOT) IS THE NUMBER OF GRAPHS (I.E. SETS OF AXES) WHICH   PLT
C     MAY STILL BE PLOTTED.   WHEN A NON-POSITIVE VALUE IS RETURNED, NO  PLT
C     PLOTTING TAKES PLACE, AND A NEGATIVE VALUE INDICATES SOME ERROR IN PLT
C     CALLING THIS ROUTINE (THE NATURE OF THE ERROR IS INDICATED ON THE  PLT
C     PRINTED OUTPUT).                                                  PLT
C                                                                       PLT
C     X AND Y ARE THE ARRAYS TO BE PLOTTED.                             PLT
C                                                                       PLT
C     XLABEL AND YLABEL ARE THE TEXT TO BE PRINTED UNDER THE AXES, THEY PLT
C     MUST BE IN HOLERITH FORM.                                         PLT
C                                                                       PLT
C     IF N IS NON-ZERO, ITS ABSOLUTE VALUE IS THE NUMBER OF POINTS IN   PLT
C     EACH ARRAY.    IF NEGATIVE, EACH DATA POINT WILL BE MARKED WITH AN PLT
C     -X- IN ADDITION TO THE LINE PLOT THAT IS ALWAYS DRAWN (IF TOO MANY PLT
```

```
C     DATA POINTS EXIST, SO THAT SUCH -X-S WOULD OVERLAP, THEY WILL NOT    PLT
C     BE DRAWN).                                                           PLT
C          N = 0 IS FOR MULTIPLE PLOTS ON THE SAME SET OF AXES ( MAXIMUM   PLT
C     OF TWENTY).    ALL PARAMETERS WILL HAVE THE VALUES OF THE LAST       PLT
C     CALL WITH NON-ZERO N AND THE MAX. AND MIN. VALUES OF BOTH ARRAYS     PLT
C     IN A CALL WITH N=0 MUST BE WITHIN PLOTTING RANGE OF THESE PREVIOUS   PLT
C     SCALE FACTORS.                                                       PLT
C                                                                          PLT
C                                                                          PLT
      DIMENSION X(1),Y(1),XLABEL(1),YLABEL(1)                              PLT
      COMMON /CALCOMP/ NPX,NPY,STEPS,NXLIM
      DATA WIDTH,PAPER,NPLOTS,MAXGRPH/11.5,0.0,0,50/
      DATA NTIMES,NGRAPHS,MAXPLOT / 0 , 0 , 20 /                           PLT
      DATA X0,Y0/1.0,0./
      DATA LX,LY/10.,10./
      DATA SIZE / 0.14 /
      DATA TIMZERO/1/
      REAL LX,LY
      INTEGER USERLIM,TIMZERO                                              PLT
C
      DX(T) = (T - XMIN) / SFX                                             PLT
      DY(T) = (T - YMIN) / SFY                                             PLT
C                                                                          PLT
C     USERLIM IS IS THE PAPER LIMIT SET BY THE USER., SEE CODING OF        PLT
C     PLOTS AND PLOT FOR THE DEFINITION OF NXLIM IN TERMS OF USERLIM.
C     WIDTH IS THE WIDTH OF THE PAPER ON WHICH THE PLOT WILL BE PLACED.
C     X0 AND Y0 ARE THE COORDINATES OF THE ORIGIN OF THE PLOT RELATIVE
C     TO THE LOWER LEFT CORNER OF THE PLOT PAGE.
C     PAPER IS INCHES OF PAPER ALREADY USED
C     NTIMES IS THE NUMBER OF TIMES THIS ROUTINE HAS BEEN CALLED.          PLT
C
C
      NTIMES = NTIMES + 1                                                  PLT
      INKPLOT = 0
C
      IF ( N .EQ. 0 )      GO TO 1900                                      PLT
      NPOINTS = IABS(N)                                                    PLT
C     CHECK NUMBER OF POINTS IN ARRAY.
      IF ( NPOINTS .LE. 1 )      GO TO 6000
C                                                                          PLT
      IF ( NTIMES .GT. TIMZERO )      GO TO 100                            PLT
      USERLIM = NXLIM/100 - 2
      LIMIT = MIN0( USERLIM , IFIX(0.5+WIDTH*FLOAT(MAXGRPH)))
      IF ( LIMIT .EQ. USERLIM )      MAXGRPH = IFIX(FLOAT(USERLIM)/WIDTH)  PLT
C
C     DRAW LEFT CUT LINE
      CALL PLOT(0.,-11.0,-3)
      Z = 0.
      IC = 3
      DO 55 J=1,55
      Z = Z + 0.2
      IC = 5 - IC
   55 CALL PLOT(0.,Z,IC)
C
  100 KIND = 0                                                             PLT
      IF ( N.LT.0 .AND. NPOINTS.LT.101)      KIND = 1                      PLT
```

```
      PAPER = PAPER + WIDTH
      IF ( PAPER .GT. FLOAT(LIMIT) )       GO TO 5000                         PLT
      NGRAPHS = NGRAPHS + 1
C     GET SCALE FACTORS.                                                      PLT
      CALL SCALE(X,LX,NPOINTS,1,XMIN,SFX)
      CALL SCALE(Y,LY,NPOINTS,1,YMIN,SFY)
C
C                                                                             PLT
C      ORIGIN IS NOW AT THE LEFT EDGE OF THE PLOT PAGE AND AT THE VERY
C     BOTTOM EDGE  OF THE PLOT PAPER ROLL. THIS EDGE IS 0.58 INCHES FROM
C     THE GRID ON THE PLOTTING PAPER ROLL.                                    PLT
C     MAKE SURE PEN IS WHERE IT SHOULD BE AND THEN REDEFINE ORIGIN.           PLT
      CALL PLOT(0.,-11.0,-3)                                                  PLT
      CALL PLOT(X0,Y0+0.58,-3)
C                                                                             PLT
C     FIND OUT HOW LONG THE Y-LABEL IS.
      DO 300 JK=1,20
      IF ( YLABEL(JK+1) .EQ. 0 )      GO TO 305
  300 CONTINUE
  305 NCHAR = MIN0(10*JK , IFIX(7./6.*LY/SIZE) )
C     DRAW Y-AXIS
      CALL AXIS(0.,0.,YLABEL,NCHAR,LY,90.0,YMIN,SFY,10.0)
C
C
C     DRAW CURVE
C     IF X=0 APPEAR ON GRAPH, DRAW DOTTED LINE THROUGH IT
      IF(XMIN.GE.0.) GO TO 340
      XVALUE=-XMIN/SFX
      IF(XVALUE.GE.LX) GO TO 340
      CALL PLOT(XVALUE,0.,3)
      JSTOP=LY/.2
      IC=3
      Z=0.
      DO 320 J=1,JSTOP
      IC=5-IC
      Z=Z+.2
320   CALL PLOT(XVALUE,Z,IC)
340   CALL LINE(X,Y,NPOINTS,1,KIND,4,XMIN,SFX,YMIN,SFY)
C
C
C     FIND OUT HOW LONG THE X-LABEL IS.
      DO 350 JK=1,20
      IF ( XLABEL(JK+1) .EQ. 0 )      GO TO 355
  350 CONTINUE
  355 NCHAR = MIN0( 10*JK , IFIX(7./6.*LX/SIZE) )
C     DRAW X-AXIS AT THE BOTTOM OF THE GRID
      CALL AXIS(0.,0.0    ,XLABEL,-NCHAR,LX,0.0,XMIN,SFX,10.0)
C     IF Y=0 APPEARS ON GRAPH, DRAW DOTTED LINE THROUGH IT
      IF (YMIN.GE.0.) GO TO 400
      YVALUE=-YMIN/SFY
      IF(YVALUE.GE.LY) GO TO 400
      CALL PLOT(0.,YVALUE,3)
      JSTOP =LX/.2
      IC=3
      Z=0.
      DO 370 J=1,JSTOP
```

```
      IC=5-IC
      Z=Z+.2
370   CALL PLOT(Z,YVALUE,IC)
C
C     DRAW GRAPH NUMBER JUST ABOVE GRID ON PLOT PAGE
C
400   CALL PLOT(LX-1.5, 9.80,-3)
      CALL SYMBOL(0.,0.40,0.14,10HGRAPH NO. ,0.0,0)                     PLT
      CALL NUMBER(1.20,0.40,0.14,FLOAT(NGRAPHS),0.0,-1)                 PLT
C
C     DRAW RIGHT CUT LINE.                                              PLT
      CALL PLOT(WIDTH+1.5-LX-X0,-11.0,-3)
      Z = 0.
      IC = 3
      DO 550 JK=1,55
      IC = 5 - IC
      Z = Z + 0.20
  550 CALL PLOT(0.,Z,IC)
C
C     ORIGIN IS AT LOWER LEFT CORNER OF NEW PAGE, MOVE PEN THERE.
      CALL PLOT(0.,0.,3)
C                                                                       PLT
      INKPLOT = MAXGRPH - NGRAPHS                                       PLT
      NPLOTS = 1                                                        PLT
C     SAVE FINAL VALUE PLOTTED SO THAT A PLOT NO. MAY BE ADDED NEXT TO  PLT
C     IT IF A MULTIPLE PLOT OCCURS.                                     PLT
      XSAVE = DX(X(NPOINTS))                                            PLT
      YSAVE = DY(Y(NPOINTS))                                            PLT
C                                                                       PLT
      RETURN                                                            PLT
C                                                                       PLT
C                                                                       PLT
C                                                                       PLT
C     N = 0, CHECK FOR TOO MANY PLOTS OR NO PREVIOUS SCALE VALUES.      PLT
 1900 IF ( NTIMES .EQ. 1 )     GO TO 4000                               PLT
      IF ( NPLOTS .EQ. MAXPLOT )     GO TO 7000                         PLT
C     THERE IS TO BE MULTIPLE PLOTTING.                                 PLT
      XMN = X(1)                                                        PLT
      XMX = X(1)                                                        PLT
      YMN = Y(1)                                                        PLT
      YMX = Y(1)                                                        PLT
      DO 2000 J=1,NPOINTS                                               PLT
      IF (XMN .GT. X(J) )      XMN = X(J)                               PLT
      IF (XMX .LT. X(J) )      XMX = X(J)                               PLT
      IF ( YMN .GT. Y(J) )      YMN = Y(J)                              PLT
 2000 IF ( YMX .LT. Y(J) )      YMX = Y(J)                              PLT
C     CHECK IF THIS RANGE OF VALUES WILL PLOT ON THE PAGE.              PLT
      IF ( XMN.LT.XMIN .OR. YMN.LT.YMIN )      GO TO 3000
      IF ( DX(XMX).GT.LX .OR. DY(YMX).GT.LY )      GO TO 3000
C     SINCE THE ORIGIN IS AT THE LOWER LEFT CORNER OF THE NEXT PLOTTING PLT
C     PAGE, REDEFINE ORIGIN.                                            PLT
      CALL PLOT(0.,-11.0,-3)                                            PLT
      CALL PLOT(X0-WIDTH,Y0+0.58,-3)                                    PLT
C     NUMBER THE FIRST PLOT.                                            PLT
      IF (NPLOTS.EQ.1)  CALL NUMBER(XSAVE+0.1,YSAVE-.05,.1,0.0,0.0,-1)  PLT
C     NUMBER THIS PLOT.                                                 PLT
```

```
      CALL NUMBER(.1+DX(X(NPOINTS)),DY(Y(NPOINTS))-.05,.1,FLOAT(NPLOTS)   PLT
     *                  ,0,0,-1)                                          PLT
C     DRAW CURVE.                                                         PLT
      CALL LINE(X,Y,NPOINTS,1,KIND,4,XMIN,SFX,YMIN,SFY)                   PLT
C     MOVE ORIGIN TO LOWER CORNER OF NEXT PLOT PAGE.                      PLT
      CALL PLOT(WIDTH-X0,-11.0,-3)                                        PLT
      NPLOTS = NPLOTS + 1                                                 PLT
      INKPLOT = MAXGRPH - NGRAPHS                                         PLT
C                                                                         PLT
      RETURN                                                              PLT
C                                                                         PLT
C                                                                         PLT
C     ERROR MESSAGES.                                                     PLT
C                                                                         PLT
3000  WRITE(98,6002)
      WRITE(98,3001) NTIMES
 3001 FORMAT (20X,*A MULTIPLE PLOT WAS ATTEMPTED AT THE*I6* -TH CALL OF   PLT
     *INKPLOT, BUT AT LEAST ONE MAXIMUM OR MINIMUM OF THE GIVEN*/ 15X,    PLT
     *      *VECTORS WAS NOT PLOTABLE ON THE SET OF AXES ALREADY DRAWN*)
      GO TO 9000                                                          PLT
C                                                                         PLT
4000  WRITE(98,6002)
      WRITE (98,4002)
      TIMZERO = TIMZERO + 1                                               PLT
 4002 FORMAT (20X,*YOU CANNOT ASK FOR A MULTIPLE PLOT WITHOUT EVER*       PLT
     *          * HAVING CALLED INKPLOT ROUTINE BEFORE.* )
      GO TO 9000                                                          PLT
C                                                                         PLT
5000  WRITE(98,6002)
      WRITE (98,5001) NTIMES
 5001 FORMAT (20X,*AFTER* I6                                              PLT
     *          * SUCCESSFUL CALLS TO THE INKPLOT ROUTINE, YOU ATTEMPTED TO  PLT
     *EXCEED THE PAPER LIMIT.*  /15X,*FURTHER CALLS TO INKPLOT WILL ONLY  PLT
     *RETURN CONTROL TO THE CALLING PROGRAM* )
      GO TO 9000                                                          PLT
C                                                                         PLT
6000  WRITE (98,6002)
      WRITE (98,6003) NTIMES
      IF ( NTIMES .EQ. TIMZERO )     TIMZERO = TIMZERO + 1                PLT
 6002 FORMAT (1X,13(5HERROR,5X))
 6003 FORMAT (20X, *THE NUMBER OF POINTS TO BE PLOTTED WAS NOT GREATER T
     *HAN ONE AT THE* ,I6, * -TH CALL OF THE INKPLOT ROUTINE* )
      GO TO 9000                                                          PLT
C                                                                         PLT
7000  WRITE (98,6002)
      WRITE (98,7001)  NTIMES,NGRAPHS,MAXPLOT
 7001 FORMAT (20X,*THE* I6 * -TH CALL OF INKPLOT ATTEMPTED TO MAKE TOO M  PLT
     *ANY MULTIPLE PLOTS ON THE* I4 * -TH SET OF AXES*//                  PLT
     *  15X,*DO YOU REALLY WANT MORE THAN* I3 * PLOTS ON ONE SET OF AXES  PLT
     * (QUESTION MARK)* )
C                                                                         PLT
 9000 JPLT811 = NGRAPHS - MAXGRPH                                         PLT
      WRITE(98,9001) JPLT811
 9001 FORMAT (60X,9HINKPLOT =I6 )
      WRITE(98,6002)
      INKPLOT = JPLT811                                                   PLT
```

```
      RETURN                                                              PLT
      END                                                                 PLT
      SUBROUTINE VECTORF(X,Y,R,THETA,N,X0,DX,Y0,DY,RREF,VREF,XSIZE,
     1 YSIZE,ITITLE,LABELX,LABELY)
C
C  VECTORF PRODUCES A VECTOR FIELD PLOT COMPLETE WITH AXES AND LABELS.
C  THE PLOTTER ORIGIN, (OR REFERENCE POINT) CORRESPONDS TO (X0,Y0).
C  ALL MOVEMENTS ARE RELATIVE TO THIS POINT.  VECTORF ASSUMES THIS
C  ORIGIN TO HAVE BEEN PROPERLY SET BY THE PROGRAMMER BEFORE ENTRANCE
C  TO VECTORF.  THIS IS DONE BY CALLING -PLOT- WITH A NEGATIVE THIRD
C  ARGUMENT.  E.G. CALL PLOT(2.,.58,-3)
C
C  X -- AN ARRAY CONTAINING UNSCALED ABSCISSA VALUES.
C  Y -- AN ARRAY CONTAINING UNSCALED ORDINATE VALUES.
C  R -- AN ARRAY CONTAINING UNSCALED VALUES OF THE VECTOR MAGNITUDES.
C          I.E. R(I) IS THE MAGNITUDE OF THE VECTOR AT (X(I),Y(I)).
C  THETA --AN ARRAY CONTAINING VALUES OF THE VECTOR DIRECTIONS, IN
C          RADIANS, WITH RESPECT TO THE POSITIVE X DIRECTION.
C          I.E. THETA (I) IS THE DIRECTION OF THE VECTOR AT (X(I),Y(I)).
C  N -- THE NUMBER OF VECTORS TO BE PLOTTED.  X, Y, R, AND THETA MUST
C          BE DIMENSIONED AT LEAST N IN THE CALLING ROUTINE.
C  X0 --   THE VALUE OF X AT THE REFERENCE POINT.  THIS MAY BE ANY NUMBER
C          LESS THAN OR EQUAL TO THE SMALLEST VALUE IN THE ARRAY X.
C          X0 IS THE VALUE AT THE LEFT END OF THE X-AXIS.
C  DX -- THE CHANGE IN X PER INCH ALONG THE X-AXIS.  DX SHOULD BE
C          CHOSEN SUCH THAT (X0+DX*XSIZE) .GE. THE LARGEST VALUE IN
C          THE ARRAY X. X0, DX, AND XSIZE SHOULD BE CHOSEN TO PROVIDE
C          A CONVENIENT SCALE ALONG THE X-AXIS.
C  Y0, DY -- ARE DEFINED FOR Y AND THE Y-AXIS AS X0, DX ARE FOR THE
C          X-AXIS.  Y0 IS THE VALUE AT THE LOWER END OF THE Y-AXIS.
C  RREF -- THE VECTOR MAGNITUDE WHICH IS REPRESENTED ON THE PLOT BY A
C          VECTOR VREF INCHES LONG.
C  VREF -- THE LENGTH, IN INCHES, OF THE VECTOR WITH MAGNITUDE RREF.
C          RREF/VREF IS USED AS THE SCALE FACTOR FOR R.
C  XSIZE -- THE LENGTH IN INCHES OF THE X-AXIS.  MINIMUM IS 7 INCHES.
C  YSIZE -- THE LENGTH IN INCHES OF THE Y-AXIS.  MAXIMUM IS 10 INCHES.
C  ITITLE -- AN ARRAY CONTAINING PACKED DISPLAY CODE,TERMINATED BY A
C          ZERO WORD.  THIS TEXT IS WRITTEN ACROSS THE TOP OF THE PLOT.
C          A HOLLERITH LITERAL ARGUMENT SUCH AS 17HVECTOR FIELD PLOT,
C          WILL PRODUCE SUCH AN ARRAY.  THE TEXT WILL BE CENTERED ALONG
C          THE LENGTH OF THE PLOT.
C  LABELX -- SAME AS ITITLE, BUT WRITTEN ALONG THE X-AXIS.
C  LABELY -- SAME AS ITITLE, BUT WRITTEN ALONG THE Y-AXIS.
C
C  VMIN IS THE MINIMUM SIZE VECTOR THAT WILL BE PLOTTED.  ALL VECTORS
C          WHICH, AFTER SCALING, ARE .LT. VMIN INCHES LONG WILL BE DRAWN
C          VMIN INCHES LONG.  THIS MINIMUM MAGNITUDE IS COMPUTED AND
C          STORED IN RMIN.
C  VREF, RREF, RMIN, AND VMIN ARE WRITTEN WITH APPROPRIATE TEXT DIRECTLY
C          UNDER THE TITLE OF THE PLOT.
C  IF THE VALUES X=0, OR Y=0 OCCUR ALONG THE X AND/OR Y AXES, DASHED
C          LINES WILL BE DRAWN TO INDICATE THEIR POSITION.
C  THE GRAPH SIZE, XSIZE*YSIZE, DOES NOT INCLUDE LABELS AND ANNOTATION.
C          AN ADDITIONAL SPACE OF ONE-HALF INCH SHOULD BE ALLOWED FOR
C          THIS PURPOSE BELOW THE X-AXIS, TO THE LEFT OF THE Y-AXIS, AND
C          ABOVE THE GRAPH.  I.E. THE TOTAL SPACE REQUIRED IS XSIZE+.5
```

```
C           BY YSIZE+1.
C   X0, Y0, XSIZE, AND YSIZE SHOULD BE CHOSEN TO ALLOW FOR THE LENGTH
C           OF THE PLOTTED VECTORS.  I.E. IF THE MAXIMUM LENGTH OF THE
C           PLOTTED VECTORS IS Z INCHES, THEN THE DISTANCE BETWEEN X0
C           AND MIN(X(I)) ALONG THE X-AXIS SHOULD BE .GE. Z, AND THE
C           DISTANCE BETWEEN (X0+DX*XSIZE) AND MAX(X(I)) SHOULD BE
C           .GE. Z.  LIKEWISE FOR THE Y-AXIS.
C
      DIMENSION X(1),Y(1),R(1),THETA(1),ITITLE(1),LABELX(1),LABELY(1)
      DIMENSION MSG(9),XX(5),YY(5)
      DATA DIV/10.0/
      DATA MAXCRS,VMIN,XARR,YARR,LESIGN/100,.06,-.03,.02,60558/
C
C   COUNT THE CHARACTERS IN LABELX, DRAW X-AXIS WITH ANNOTATION BELOW.
      KOUNT= -NCHARS(LABELX,MAXCRS)
      CALL AXIS(0.,0.,LABELX,KOUNT,XSIZE,0.0,X0,DX,DIV)
C   YAXIS WITH ANNOTATION TO THE LEFT.
      KOUNT=NCHARS(LABELY,MAXCRS)
      CALL AXIS(0.,0.,LABELY,KOUNT,YSIZE,90.0,Y0,DY,DIV)
C   WRITE TITLE ABOVE GRAPH, CENTERED ALONG LENGTH OF GRAPH.
      CALL TITLE(0.,YSIZE+.2,XSIZE,.21,0.,ITITLE)
C   COMPUTE SCALE FACTOR FOR VECTOR MAGNITUDES.
      DVDR=VREF/RREF
C   COMPUTE MAGNITUDE CORRESPONDING TO MINIMUM VECTOR SIZE.
      RMIN=VMIN/DVDR
C   PREPARE MESSAGE ABOUT VECTOR SIZES.  LESIGN CONTAINS THE CODE
C   FOR A LESS-THAN-OR-EQUAL SIGN, FOLLOWED BY THE CODE FOR A SPACE.
      ENCODE(80,20,MSG(1))VREF,RREF,LESIGN,RMIN,VMIN
   20 FORMAT(*VECTOR *,F4.2,* INCHES LONG = *,E9.2,*.  VALUES *,R2,E9.2,
     1  * ARE*,F4.2,* INCHES LONG.*)
C   WRITE MESSAGE UNDER TITLE, CENTERED ALONG LENGTH OF GRAPH.
      MSG(9)=0
      CALL TITLE(0.,YSIZE+.05,XSIZE,.1,0.,MSG)
C   COMPUTE LOCATION OF X=0 ALONG X-AXIS.  IF X=0 IS ON GRAPH, DRAW A
C   DASHED LINE TO INDICATE ITS POSITION.
      XZERO= -X0/DX
      IF((XZERO.GE..01).AND.(XZERO.LE.(XSIZE-.01)))CALL DASHLIN(XZERO,
     1  0.,XZERO,YSIZE,.08)
C   DO THE SAME FOR Y=0 ALONG Y-AXIS.
      YZERO= -Y0/DY
      IF((YZERO.GE..01).AND.(YZERO.LE.(YSIZE-.01)))CALL DASHLIN(0.,
     1  YZERO,XSIZE,YZERO,.08)
C
C   THIS IS THE ALGORITHM FOR PLOTTING THE VECTOR FIELD.
      DO100I=1,N
      SINTH=SIN(THETA(I))
      COSTH=COS(THETA(I))
C   COMPUTE POSITION OF POINT ON PLOT.  THIS WILL BE TAIL OF VECTOR.
      XX(1)=(X(I)-X0)/DX
      YY(1)=(Y(I)-Y0)/DY
C   COMPUTE LENGTH OF VECTOR ON PLOT.  SET TO VMIN IF .LT. VMIN.
      RT=R(I)*DVDR
      IF(RT.LT.VMIN)RT=VMIN
C   COMPUTE POSITION OF HEAD OF VECTOR.  THIS IS BOTH SECOND AND LAST
C   POINT USED IN DRAWING ARROW.
      XX(2)=XX(1)+RT*COSTH
```

```
      XX(5)=XX(2)
      YY(2)=YY(1)+RT*SINTH
      YY(5)=YY(2)
C  COMPUTE POSITION OF ARROWHEAD POINTS.  THESE ARE 3RD AND 4TH IN
C  DRAWING SEQUENCE.
      YSTH=YARR*SINTH
      XSTH=XARR*SINTH
      XCTH=XARR*COSTH
      YCTH=YARR*COSTH
      XX(3)=XX(2)+XCTH-YSTH
      XX(4)=XX(2)+XCTH+YSTH
      YY(3)=YY(2)+YCTH+XSTH
      YY(4)=YY(2)-YCTH+XSTH
C  SET INDICES FOR DRAWING ARROW FROM TAIL TO HEAD.
      K=0
      L=1
C  GET CURRENT PEN POSITION.
      CALL WHERE(XNOW,YNOW,STEPS)
C  DETERMINE WHETHER MOVING TO HEAD OR TAIL WILL TAKE FEWER PLOTTER
C  COMMANDS.  SINCE EACH COMMAND CAN CAUSE MOVEMENT IN X AND Y
C  DIRECTIONS SIMULTANEOUSLY, WE NEED TO KNOW THE MAXIMUM OF THE X AND
C  Y MOVEMENTS REQUIRED FOR EACH POINT.
      D1=AMAX1(ABS(XNOW-XX(1)),ABS(YNOW-YY(1)))
      D2=AMAX1(ABS(XNOW-XX(2)),ABS(YNOW-YY(2)))
      IF(D2.GT.D1)GO TO 200
C  IF TAIL IS CLOSER DRAW ARROW FROM TAIL TO HEAD.  IF HEAD IS CLOSER,
C  CHANGE K AND L TO DRAW ARROW FROM HEAD TO TAIL.
      K=6
      L= -1
C  LIFT PEN BEFORE MOVING TO ARROW.  MOVE TO ARROW.  THEN LOWER PEN AND
C  DRAW ARROW IN FOUR LINE SEGMENTS.  IPEN=3 LIFTS THE PEN.
C  IPEN=2 LOWERS THE PEN.
  200 IPEN=3
      DO 300J=1,5
      K=K+L
      CALL PLOT(XX(K),YY(K),IPEN)
      IPEN=2
  300 CONTINUE
  100 CONTINUE
      RETURN
      END
      SUBROUTINE CONTOUR(Z,KDIM,
     1 M,N,MM,NN,XA,XB,YA,YB,XG,YG,NCL,CL,ITITLE,LABELX,LABELY,FX,FY,KP)
      DATA TWOPI/6.28318530717958/
      COMMON/POLARC/RS,R0,THS,TH0
      COMMON/INDICES/MROW,NCOL,MMROW,NNCOL,KPOL
      COMMON/XYBNDS  /XMIN,XMAX,YMIN,YMAX,XSIZE,YSIZE,
     1HX,HY,XS,XSS,YS,YSS,FXA,FYA
      COMMON/CLEVELS/NLVLS,NLV,CLEVEL(50)
      COMMON/CAVIN/IDIM,DUM(4035)
      DIMENSION Z(1)
      DIMENSION CL(1)
C     Z(I,J) IS THE ORDINATE AT POINT X(J), Y(I)
C     MXN IS THE SIZE OF THE CALCULATED X-Y GRID
C     MMXNN IS THE SIZE OF THE EXPANDED(BY INTERPOLATION) X-Y GRID
C     XA,XB,YA,YB ARE THE MINIMUM AND MAXIMUM VALUES
```

```
C          OF X AND Y,
C      XG IS THE WIDTH OF THE GRAPH  IN INCHES.
C      YG IS THE HEIGHT OF THE GRAPH IN INCHES.
C      NCL IS THE NO. OF CONTOUR LEVELS
C      CL(I) ARE THE CONTOUR LEVELS
C      ITITLE CONTAINS THE PLOT TITLES IN 80 BCD CHARACTERS,
C            PLOT NAME IS FIRST 4 WORDS,
C            X-AXIS LABEL IS NEXT 3 WORDS,
C            Y-AXIS LABEL IS NEXT 3 WORDS.
C      THE X(I) ARE ASSUMED TO BE EQUALLY SPACED, AND
C      LIKEWISE, THE Y(I).
C      FX IS THE FUNCTION TO BE PLOTTED ALONG THE X-AXIS.
C      FY IS THE FUNCTION TO BE PLOTTED ALONG THE Y-AXIS.
      IDIM=KDIM
      MROW=M
                NCOL=N
                           MMROW=MM
                                        NNCOL=NN
      KPOL=0
      IF(KP.NE.0)GO TO 50
      XMIN=XA
                XMAX=XB
                           YMIN=YA
                                        YMAX=YB
      GO TO 55
   50 XMIN=-YB
      XMAX=YB
      YMIN=XMIN
      YMAX=XMAX
      THO=XA
      RO=YA
      THS=(XB-XA)/FLOAT(NNCOL-1)
      RS=(YB-YA)/FLOAT(MMROW-1)
      KPOL=1
   55 CONTINUE
      XSIZE=XG
                YSIZE=YG
                           NLVLS=XABSF(NCL)
      CALLPLOT(0,-.5*(11.-YSIZE),3)
      IF(NCL)1,9,9
    1 CLEVEL=HX=Z
                     L=0
      DO15I=1,NCOL
                     DO7J=1,MROW
                                    L=L+1
      IF(Z(L).LT.CLEVEL)4,5
    4 CLEVEL=Z(L)
    5 IF(Z(L).GT.HX)6,7
    6 HX=Z(L)
    7 CONTINUE
   15 L=L-M+IDIM
                     HX=(HX-CLEVEL)/FLOATF(NLVLS-1)
      DO8I=2,NLVLS
    8 CLEVEL(I)=CLEVEL(I-1)+HX
                                   GOTO11
    9 DO10I=1,NLVLS
```

```
 10 CLEVEL (I)=CL(I)
 11 HX=(XMAX-XMIN)/FLOATF(NCOL-1)
    HY=(YMAX-YMIN)/FLOATF(MROW-1)
    XS=(XMAX-XMIN)/FLOATF(NNCOL-1)
    YS=(YMAX-YMIN)/FLOATF(MMROW-1)
    FXA=FX(XMIN)
                       FYA=FY(YMIN)
    XSS=XG/(FX(XMAX)-FXA)
    YSS=YG/(FY(YMAX)-FYA)
  2 CALLINTERP(Z,IDIM)
    DO3NLV=1,NLVLS
  3 CALL SCAN(Z,FX,FY)
                           CALLLABEL(ITITLE,LABELX,LABELY,FX,FY)
    IF(KPOL.EQ.0)GOTO 200
    XCENT=XSIZE*.5
    YCENT=YSIZE*.5
    RZERO=XCENT*R0/YB
    RMAX=XCENT
    THETA1=XB
    IF(THETA1.GT.TWOPI)THETA1=TWOPI+AMOD(THETA1,TWOPI)
    IF(RZERO.GT..1)CALL ARC(XCENT,YCENT,RZERO,TH0,THETA1,1)
    CALL ARC(XCENT,YCENT,RMAX,TH0,THETA1,1)
    THH=TH0
    K=2
    IF(ABS(THETA1-(TWOPI+TH0)).LT..001)K=1
    DO100I=1,K
    SINTH=SIN(THH)
    COSTH=COS(THH)
    X0=XCENT+RZERO*COSTH
    Y0=YCENT+RZERO*SINTH
    X1=XCENT+RMAX*COSTH
    Y1=YCENT+RMAX*SINTH
    CALL DASHLIN(X0,Y0,X1,Y1,.1)
    THH=THETA1
100 CONTINUE
    CALL DASHLIN(XCENT,YCENT,0.,YCENT,.25)
    CALL DASHLIN(XCENT,YCENT,XCENT,YSIZE,.25)
    CALL DASHLIN(XCENT,YCENT,XCENT,0.,.25)
    CALL DASHLIN(XCENT,YCENT,XSIZE,YCENT,.25)
200 CALLPLOT(  INTF(XSIZE+6.),-11.,-3)
    RETURN
    END
    SUBROUTINE LABEL(ITITLE,LABELX,LABELY,FX,FY)
    COMMON/TEMP/Z(101)
    COMMON/XYBNDS/XA,XB,YA,YB,XSIZE,YSIZE,HX,HY,
   1XS,XSS,YS,YSS,FXA,FYA
    COMMON/INDICES/M,N,MM,NN,KPOL
    COMMON/CLEVELS/NCL,NLV,CL(50)
    DIMENSION ITITLE(1),LABELX(1),LABELY(1)
    DATA IS/18/
    J=0
          X=XA
                 P=0
                        G=XSIZE-.9
    DO10I=1,NN
                 XG=XSS*(FX(X)-FXA)
```

```
      IF(XG.GE.G)3,4
    3 X=XB
              XG=XSIZE
                          GOTO5
    4 IF(XG.GE.P)5,10
    5 J=J+1
               Z(J)=XG
      CALL SYMBOL(XG,-.07,.12,IS,0.,-1)
    7 CALLNUMBER(XG-.14,-.22,.07,X,0,5HE10.3)
    8 IF(X.EQ.XB)11,9
    9 P=XG+.9
   10 X=X+XS
   11 CALL TITLE(0.,-,4,XSIZE,.14,0.,LABELX)
      CALL TITLE(0.,YSIZE+.15,XSIZE,.21,0.,ITITLE)
      DO12I=1,J
   12 CALL SYMBOL(Z(I),YSIZE+.07,.12,IS,180.,-1)
      J=0
             Y=YA
                    P=0
                           G=YSIZE-.2
      DO20I=1,MM
                    YG=YSS*(FY(Y)-FYA)
      IF(YG.GE.G)13,14
   13 Y=YB
              YG=YSIZE
                          GOTO15
   14 IF(YG.GE.P)15,20
   15 J=J+1
               Z(J)=YG
      CALL SYMBOL(-,07,YG,.12,IS,270.,-1)
      CALLNUMBER(-.62,YG+.04,.07,Y,0,5HE10.3)
      IF(Y.EQ.YB)21,19
   19 P=YG+.9
   20 Y=Y+YS
   21 CALL TITLE(-.7,0.,YSIZE,.14,90.,LABELY)
      DO22I=1,J
   22 CALL SYMBOL(XSIZE+.07,Z(I),.12,IS,90.,-1)
      YI=.5*YSIZE+.1*FLOATF(NCL)
                                   DY=.2
      CALLNUMBER(XSIZE+.5,YI,.14,NCL,0,2HI2)
      CALLSYMBOL(XSIZE+.80,YI,.14,14HCONTOUR LEVELS,0,14)
      YI=YI-DY
   23 DO24I=1,NCL
      CALLSYMBOL(XSIZE+.8,YI+.05,0.10,I-1         ,0,-1)
       CALL NUMBER(XSIZE+1.0,YI,0.105,CL   (I),0,5HE15.5)
   24 YI=YI-DY
      RETURN
      END
      SUBROUTINE INTERP(AM,IDIM)
      DIMENSION AM(IDIM,1)
      COMMON/TEMP/Z(101)
      COMMON/XYBNDS/XA,XB,YA,YB,XG,YG,HX,HY,
     1XS,XSS,YS,YSS,FXA,FYA
      COMMON/INDICES/M,N,MM,NN  ,KPOL
      ZFUN(V)=A0+A1*V+A2*V**2
      N1=N-1
```

```
            M1=M-1
      IF(N-NN)16,15,14
   16 DO6I=1,M
      DO1J=1,N
    1 Z(J)=AM(I,J)
                     XY=XA
                              K=1
                                     T=HX+XA
      DO3J=2,N1
                CALLFIT(J,T,HX,A0,A1,A2)
    2 AM(I,K)=ZFUN(XY)
                          XY=XY+XS
                                     K=K+1
                                              IF(XY-T)2,2,3
    3 T=T+HX
    4 IF(K-NN)5,5,6
    5 AM(I,K)=ZFUN(XY)
                          K=K+1
                                   XY=XY+XS
                                              GOTO4
    6 CONTINUE
   15 IF(M-MM)17,13,14
   17 DO12I=1,NN
      DO7J=1,M
    7 Z(J)=AM(J,I)
                     K=1
                            XY=YA
                                     T=HY+YA
      DO9J=2,M1
                CALLFIT(J,T,HY,A0,A1,A2)
    8 AM(K,I)=ZFUN(XY)
                          XY=XY+YS
                                     K=K+1
                                              IF(XY-T)8,8,9
    9 T=T+HY
   10 IF(K-MM)11,11,12
   11 AM(K,I)=ZFUN(XY)
                          K=K+1
                                   XY=XY+YS
                                              GOTO10
   12 CONTINUE
   13 RETURN
   14 STOP12
      END
      SUBROUTINE SCAN(AM,FX,FY)
C AM IS THE MATRIX TO BE CONTOURED. MT AND NT ARE ITS X AND Y DIMENSIONS.
C CL(NLV) IS THE CONTOUR LEVEL.
C THE N  (X,Y)  VALUES OF ONE CONTOUR LINE ARE PLOTTED WHEN
C  THEY ARE AVAILABLE,
      DIMENSION AM(1)
      COMMON           /CLEVELS/NCL,NLV,CL(50)
      COMMON/INDICES/DUM(2),MT,NT,KPOL
      COMMON/CAVIN/DIM, IX,IY,IDX,IDY,ISS,
     1 NP,N,CV,IS,IS0,IX0,IY0,DCP,
     2    INX(8),INY(8),REC(800),X(1603),Y(1603)
      TYPE INTEGER REC ,DIM
```

```
      DATA(INY=                    0,1,1,1,0,-1,-1,-1)
      DATA(INX=-1,-1,0,1,1,1,0,-1)
      NP=ISS=0
      CV=CL(NLV)
      MT1=MT-1
                NT1=NT-1
      DO 110 I=1,MT1
      IF(AM(I)  -CV)55,110,110
   55 IF(AM(I+1)  -CV)110,57,57
   57 IX0=IX=I+1
                 IY0=IY=IS0=IS=1
                                        IDX=-1
                                                 IDY=0
      CALL TRACE(AM,FX,FY)
  110 CONTINUE
      J=MT-DIM
                DO20I=1,NT1
                               J=J+DIM
      IF(AM(J)-CV)15,20,20
   15 IF(AM(J+DIM)-CV)20,17,17
   17 IX0=IX=MT
                IY0=IY=I+1
                               IDX=0
                                      IDY=-1
                                               IS0=IS=7
      CALL TRACE(AM,FX,FY)
   20 CONTINUE
      J=MT+NT1*DIM+1
                   DO30I=1,MT1
                                    J=J-1
      IF(AM(J)-CV)25,30,30
   25 IF(AM(J-1)-CV)30,27,27
   27 IX0=IX=MT-I
                IY0=IY=NT
                               IDX=1
                                      IDY=0
                                               IS0=IS=5
      CALL TRACE(AM,FX,FY)
   30 CONTINUE
      J=NT+DIM+1
                DO40I=1,NT1
                               J=J-DIM
      IF(AM(J)-CV)35,40,40
   35 IF(AM(J-DIM)-CV)40,37,37
   37 IX0=IX=1
               IY0=IY=NT-I
                              IDX=0
                                     IDY=1
                                              IS0=IS=3
      CALL TRACE(AM,FX,FY)
   40 CONTINUE
      ISS=1
           L=0
      DO13J=2,NT1
                  L=L+DIM
      DO10I=1,MT1
```

```
                L=L+1
   IF(AM(L)-CV)5,10,10
 5 IF(AM(L+1)-CV)10,7,7
 7 K=L+1
   DO 9 ID = 1,NP
   IF(REC(ID)-K)9,10,9
 9 CONTINUE
   IX0=IX=I+1
               IY0=IY=J
                              IDX=-1
                                      IDY=0
                                             IS0=IS=1
   CALL TRACE(AM,FX,FY)
10 CONTINUE
13 L=L-MT1
   RETURN
   END
   SUBROUTINE TRACE(AM,FX,FY)
   DIMENSION AM(1)
   COMMON/POLARC/RS,R0,THS,TH0
   COMMON             /INDICES/DUM(2),MT,NT,KPOL
   COMMON/XYBNDS/XA,XB,YA,YB,XSIZE,YSIZE,HX,HY,
  1XS,XSS,YS,YSS,FXA,FYA
   COMMON/CAVIN/DIM, IX,IY,IDX,IDY,ISS,
  1 NP,N,CV,IS,IS0,IX0,IY0,DCP,
  2    INX(8),INY(8),REC(800),X(1603),Y(1603)
   COMMON/CLEVELS/NCL,NLV,CL(50)
   TYPE INTEGER REC,DIM
   N=0
        JY=DIM*(IY-1)+IX
                              MY=DIM*IDY+IDX+JY
 2 N=N+1
         IF(N-1600)3,3,32
 3 IF(IDX)5,4,6
 4 X(N)=FLOATF(IY-1)+FLOATF(IDY)*(AM(JY)-CV)/(AM(JY)-AM(DIM*IDY+JY))
   Y(N)=FLOATF(IX-1)
                         GOTO7
 5 NP=NP+1
          REC(NP)=JY
 6 Y(N)=FLOATF(IX-1)+FLOATF(IDX)*(AM(JY)-CV)/(AM(JY)-AM(JY+IDX))
   X(N)=FLOATF(IY-1)
 7 IS=IS+1
 8 IF(IS-8)10,10,9
 9 IS=IS-8
10 IDX=INX(IS)
               IDY=INY(IS)
   IX2=IX+IDX
              IY2=IY+IDY
                               IR=IDX*IDY
11 IF(ISS)13,15
13 IF(IS.NE.IS0.OR.IY.NE.IY0.OR.IX.NE.IX0)16,14
14 N=N+1
         X(N)=X
                Y(N)=Y
                       GOTO73
15 IF(IX2)151,73,151
```

```
151 IF(IY2)152,73,152
152 IF((IX2.LE.MT).AND.(IY2.LE.NT))16,73
 16 MY=DIM*IDY+IDX+JY
                          IF(IR)19,17,20
 17 IF(CV-AM(MY))18,18,2
 18 IX=IX2
          IY=IY2
 81 IS=IS+5
           JY=MY
                   GOTO8
 19 KY=JY+IDX
             LY=MY-IDX
                         GOTO21
 20 KY=MY-IDY
             LY=JY+IDY
 21 DCP=(AM(JY)+AM(KY)+AM(LY)+AM(MY))*.25
                                        IF(CV-DCP)23,23,22
 22 CALL GETPT(AM(JY))
    GO TO 7
 23 IF(IR)24,25,25
 24 IX=IX2
          IDX=-IDX
    CALL GETPT(AM(KY))
    IY=IY2
          IDY=-IDY
                      GOTO26
 25 IY=IY2
          IDY=-IDY
    CALL GETPT(AM(KY))
    IX=IX2
          IDX=-IDX
 26 IF(CV-AM(MY))81,81,28
 28 CALL GETPT(AM(MY))
                    IF(IR)29,30,30
 29 IX=IX+IDX
             IDX=-IDX
                         GOTO31
 30 IY=IY+IDY
             IDY=-IDY
 31 IF(CV-AM(LY))33,33,34
 33 IS=IS-1
           JY=LY
                   GOTO10
 34 CALL GETPT(AM(LY))
                    IF(IR)35,36,36
 35 IY=IY+IDY
             GOTO7
 36 IX=IX+IDX
             GOTO7
 32 PRINT103,CV
 73 IF(KPOL.EQ.0)GO TO 173
    DO174I=1,N
    THETA=X(I)*THS+TH0
    R=Y(I)*RS+R0
    X(I)=XSS*(R*COS(THETA)-FXA)
    Y(I)=YSS*(R*SIN(THETA)-FYA)
```

```
174 CONTINUE
    GO TO 175
173 DO74I=1,N
    X(I)=XSS*(FX(X(I)*XS+XA)-FXA)
 74 Y(I)=YSS*(FY(Y(I)*YS+YA)-FYA)
175 CONTINUE
    CALLSYMBOL(X,Y,.07,NLV-1              ,0,-1)
    DO75I=1,N
 75 CALLPLOT(X(I),Y(I),2)
    RETURN
103 FORMAT(1H0,23HA CONTOUR LINE AT LEVEL,E13.5,
   1 41H WAS TERMINATED BECAUSE IT CONTAINED MORE,
   2 23H THAN 1600 PLOT POINTS.)
    END
    SUBROUTINE GETPT(AM)
    COMMON/CAVIN/DIM, IX,IY,IDX,IDY,ISS,
   1 NP,N,CV,IS,IS0,IX0,IY0,DCP,
   2    INX(8),INY(8),REC(800),X(1603),Y(1603)
    N=N+1
            B=AM    -DCP
                              IF(B)2,1
  1 V=.5
           GOTO3
  2 V=.5*(AM    -CV)/B
  3 Y(N)=FLOATF(IX-1)+FLOATF(IDX)*V
                                         X(N)=FLOATF(IY-1)+FLOATF(IDY)*V
    RETURN
    END
    SUBROUTINE FIT(I,X,H,C,B,A)
    COMMON/TEMP/Z(101)
    W=0.5*(Z(I+1)-Z(I-1))/H
    A=0.5*(Z(I+1)+Z(I-1)-Z(I)-Z(I))/H**2
    C=Z(I)+X*(X*A-W)
                           B=W-2.*X*A
    RETURN
    END
    SUBROUTINE TITLE(X,Y,SIZE,HEIGHT,ANGLE,ITEXT)
    DIMENSION ITEXT(1)
    DATA SIX7,TWO7/.857142857,.285714286/
    CHSIZE=HEIGHT*SIX7
    MAXCHS=SIZE/CHSIZE
    NUMCHS=NCHARS(ITEXT,MAXCHS)
    START=.5*(SIZE-CHSIZE*FLOAT(NUMCHS)+TWO7*HEIGHT)
    TH=ANGLE*.0174533
    CALL SYMBOL(X+START*COS(TH),Y+START*SIN(TH),HEIGHT,ITEXT,ANGLE,
   1   NUMCHS)
    RETURN
    END
    FUNCTION NCHARS(ITEXT,MAXCHS)
    DIMENSION ITEXT(1)
    MAXWDS=MAXCHS/10+1
    DO1I=1,MAXWDS
    K=ITEXT(I).AND.7777B
    IF(K.EQ.0)GO TO 2
  1 CONTINUE
    I=MAXWDS
```

```
    2 NUMCHS=10*I
      DO3J=1,I
      L=I-J+1
      KTEST=ITEXT(L)
      DO3M=1,10
      K=KTEST.AND.77B
      IF((K.NE.0).AND.(K.NE.55B))GO TO 4
      KTEST=ISHIFT(KTEST,54)
      NUMCHS=NUMCHS-1
    3 CONTINUE
    4 NUMCHS=MIN0(NUMCHS,MAXCHS)
      NCHARS=NUMCHS
      RETURN
      END
      SUBROUTINE DASHLIN(X0,Y0,X1,Y1,DASH)
C
C  DRAWS A DASHED LINE FROM (X0,Y0) TO(X1,Y1).  THE DASHES AND SPACES
C  BETWEEN THEM ARE APPROXIMATELY OF LENGTH -DASH-.  THIS LENGTH IS
C  ADJUSTED SUCH THAT THE LINE IS COMPOSED OF EQUAL LENGTH DASHES,
C  AND BEGINS AND ENDS WITH A DASH.
C  ALL PARAMETERS ARE IN FLOATING POINT INCHES.
C
      XNOW=X0
      YNOW=Y0
      DX=X1-X0
      DY=Y1-Y0
      D=SQRT(DY*DY+DX*DX)
      NDASH=2*(IFIX(D/DASH)/2)+1
      DX=DX/FLOAT(NDASH)
      DY=DY/FLOAT(NDASH)
      CALL WHERE(XN,YN,STEPS)
      D1=AMAX1(ABS(XN-X0),ABS(YN-Y0))
      D2=AMAX1(ABS(XN-X1),ABS(YN-Y1))
      IF(D2.GT.D1)GO TO 2
      XNOW=X1
      YNOW=Y1
      DX=-DX
      DY=-DY
    2 IPEN=3
      CALL PLOT(XNOW,YNOW,IPEN)
      DO1I=1,NDASH
      XNOW=XNOW+DX
      YNOW=YNOW+DY
      IPEN=5-IPEN
      CALL PLOT(XNOW,YNOW,IPEN)
    1 CONTINUE
      RETURN
      END
      SUBROUTINE ARC(X0,Y0,R,TH0,TH1,IDASH)
C
C  DRAWS AN ARC OF RADIUS R ABOUT (X0,Y0) FROM THETA=TH0 TO
C  THETA=TH1, TH0.LT.TH1.  IF IDASH.EQ.0, THE ARC WILL BE SOLID.
C  IF IDASH.NE.0, THE ARC WILL BE DASHED.
C  X0, Y0, AND R ARE IN FLOATING POINT INCHES.
C  TH0 AND TH1 ARE IN RADIANS.
C
```

```
      DELTH=2.*ASIN(.05/R)
      THETA =TH0
      X=X0+R*COS(THETA)
      Y=Y0+R*SIN(THETA)
      X1=X0+R*COS(TH1)
      Y1=Y0+R*SIN(TH1)
      IPEN=3
    1 CALL PLOT(X,Y,+PEN)
      THETA=THETA+DELTH
      IPEN=5-IPEN
      IF(IDASH.NE,0)GO TO 2
      IPEN=2
    2 X=X0+R*COS(THETA)
      Y=Y0+R*SIN(THETA)
      IF(THETA.LT,TH1)GO TO 1
      CALL PLOT(X1,Y1,IPEN)
      RETURN
      END
```

APPENDIX E

```
      PROGRAM TRDL(INPUT,OUTPUT,TAPE1,TAPE2,TAPE3,TAPE5=INPUT,
     1TAPE6=OUTPUT)
C                                                                       COM 0001
C***********************************************************************COMB0002
C       TIME DEPENDENT EQUATIONS OF HYDRCDYNAMICS                       COM+0003
C***********************************************************************COM20004
C***********************************************************************COMB0005
C     FINITE DIFFERENCE APPROXIMATION ... LAX WENDROFF TWO STEP         COMB0008
C     BURSTEIN VARIATION ...                                            COMB0009
C                ************************************                   COMB0012
C                INPUT PARAMETERS IN MAIN ROUTINE                       COMB0013
C                -------------------------------                        COMB0014
C     ISTART      IF .EQ. 11111   NEW CASE                              COMB0015
C                 IF .EQ. 00000   CONTINUATION OF PREVIOUS CALCULATION  COM 0016
C     IMAX        MAX NUMBER OF MESH POINTS ON A RADIUS INCLUDING ONE   COMB0017
C                 POINT FOR THE BOUNDARY CCNDITION                      COMB0018
C     JMAX        MAX NUMBER OF RAYS                                    COMB0019
C     NCYCMX      TOTAL NUMBER OF TIME STEPS FOR THE PRESENT            COMB0020
C                 INTEGRATION                                           COMB0021
C     NNPCMX      NUMBER OF TIME INTEGRATICNS BETWEEN PICTURES          COMB0022
C     FUDG        SAFETY FACTOR FOR AUGMENTATION OF LINEAR              COMB0023
C                 STABILITY ANALYSIS                                    COMB0024
C                INPUT PARAMETERS IN OTHER ROUTINES                     COMB0025
C                ---------------------------------                      COMB0026
C     GAMMA       RATIO OF SPECIFIC HEATS                               COMB0027
C     PO          UNPERTURBED PRESSURE                                  COMB0028
C     PMAX        AMPLITUDE OF PRESSURE PERTURBATION                    COM20029
C     TAU         DURATION OF PRESSURE PERTURBATION                     COMB0030
C     RMAX        RADIUS OF CYLINDER                                    COMB0031
C     XM          MOLECULAR WEIGHT OF GAS                               COMB0032
C     TO          GAS STAGNATION TEMPERATURE                            COMB0033
C***********************************************************************COMB0034
C                ---------------------------------------                COMB0035
C                     SCALING RULES...SEE SUB. INITIL                   COMB0036
C                ---------------------------------------                COMB0037
C***********************************************************************COMB0038
C                 IN PUT TAPES                                          COMB0039
C                 ------------                                          COMB0040
C     BINARY TAPE 1     INPUT TAPE CONTAINING INFORMATION FOR THE       COMB0041
C                       CONTINUATION OF THE CALCULATION                 COMB0042
C                OUT PUT TAPES                                          COMB0043
C                 ------------                                          COMB0044
C     BINARY TAPE 3     OUTPUT TAPE CONTAINS INFORMATION FOR THE        COMB0047
C                       CONTINUATION OF COMPUTATION                     COM20048
C***********************************************************************COMB0049
C ********************************************************************* COMB0050
C        *****************************************************          COM2005
C                *************************************                  COMB0054
C                                ************                           COMB0055
C                                                                       COMB0056
C                                                                       COMB0057
C                                                                       COMB0058
C     MAIN                                                              COMB0059
C*********  EXECUTIVE PROGRAM WHICH CONTRCLS INPUT,OUTPUT AND           COMB0060
C*********  LOGICAL FLOW FOR THE COMPUTATION     ***********************COMB0061
      COMMON PROPTY,GAMMA3,GAMMA1,GAMMA,DELR,DELO,PO,PMAX,PR,ER,TAU,IMAX
```

```
     1,JMAX,T,DTDR,DTDO,DELT ,P,AO,ROO ,NCYCMX
      DIMENSION PROPTY(21,20,4)
      COMMON/LE/ FJMAX,IMAXM1,X,Y,RAD(41),OVRLAY(12,20,21),
     1OVRPRM(16,20,21),HT(40)
      COMMON/PHIP/DROP(5,20,21),PHI,EN,FS,DMF,DMFT,IM,XM
      COMMON/NCPLOT/NTCYCL,ISTART,HGAM,LABLX(5),LABLY(4)
      COMMON/PRTOPT/INTPNT
      COMMON/FZAB/A,B,BB
      CALL PLOTS(2,1)
      CALL PLOT(0.,-11.,-3)
    5 CONTINUE
      RAD(41)=-1.0
      READ(5,666)ISTART,ISAVE
      IF(ISTART.LT.0) GO TO 186
      CALL PLOT(2.,0.,-3)
      NCLCM=0
      NTCYCL=0
      NCYCLE=0
      NPICTR=0
      NSAVE=0
      T=0.0
      READ(5,668)LABLY(1),LABLY(2),LABLY(3)
      LABLY(4)=1H1
      WRITE(6,668)LABLY(4),LABLY(1),LABLY(2),LABLY(3)
      LABLY(4)=0
  668 FORMAT(4A10)
      READ(5,666)IMAX,JMAX
      FJMAX=JMAX
      IMAXM1=IMAX-1
      READ(5,666)NCYCMX,NNPCMX,KALKOM,INTPNT,IBSPLT,ISMTH
  666 FORMAT(9I5)
      WRITE(6,6066)NCYCMX,NNPCMX
 6066 FORMAT(1H0,38H TOTAL NUMBER OF CYCLES FOR THIS CASE=I4,4X,
     1                                                    34HNUMBER O
     2F CYCLES BETWEEN PICTURES=I4///)
      READ(5,667) FUDG,A
  667 FORMAT(7E10.7)
C       INITIALIZE MESH AND FIND DELT FOR CONVERG. AND STABILITY
      CALL INITIL
      CALL CONVRG(FUDG)
      DELZ2=0.5*DELR
      BB=3.333333333*A
      B=0.5*BB
  757 CONTINUE
C       PRINT INITIAL FIELD
      CALL PRTOUT(NTCYCL)
      CALL CALPLT
   80 NCYCLE=NCYCLE+1
      NTCYCL=NTCYCL+1
      T=T+DELT
      IF(EN.NE.0.0) CALL PHIDOTV
      CALL VECTR(1,IMAX)
      CALL QUAD(1,IMAXM1)
      CALL TMPVCT(2,IMAX)
C   COMPUTE W(T+DELT)AT INTERIOR POINTS
      CALL GENPT(2,IMAXM1)
```

```
C       COMPLETE MESH CALC AT I=1,J=1
      IF(ISMTH.NE.0) CALL SMOOTH
      CALL CHARGE
      NCLCM=NCLCM+1
      NPICTR=NPICTR+1
      NSAVE=NSAVE+1
C       NOISE SUPPRESSOR
      DO 50 I=1,IMAX
      DO 50 J=1,JMAX
C       TEST FOR NEGATIVE DENSITY
      IF(PROPTY(I,J,1).LE.0.) GO TO 181
      DO 51 K=2,3
C       ELIMINATE SPURIOUS VELOCITIES
      IF(ABS (PROPTY(I,J,K)).LE.1.E-14) PROPTY(I,J,K)=0.
   51 CONTINUE
   50 CONTINUE
C       FIND NEW STABILITY NUMBER
      CALL CONVRG(FUDG)
C       TEST TO SEE IF FINISHED
      IF(NCYCLE-NCYCMX)81,91,91
   81 CONTINUE
      IF(NCLCM-KALKOM)89,88,88
   88 CONTINUE
      WRITE(6,606)T,NTCYCL
  606 FORMAT(3X11HTOTAL TIME=E14.7,3X17HNUMBER OF CYCLES=I4)
      CALL CALPLT
      NCLCM=0
   89 CONTINUE
C       TEST TO SEE IF PRINT TIME AFFIRMATIVE
      IF(NPICTR-NNPCMX)901,90,90
   90 CONTINUE
C       PRINT DISPLAY OF FLOW FIELD ALONG THE RAYS 0,PI/2,PI,3*PI/2
      CALL PRTOUT(NTCYCL)
C       RESTART PRINT CYCLE
      NPICTR=0
  901 IF(ISAVE.EQ.0.OR.NSAVE.LT.ISAVE) GO TO 80
  902 CONTINUE
      REWIND 3
      WRITE(3)(((PROPTY(I,J,K),K=1,4),J=1,JMAX),I=1,IMAX),T,NTCYCL
      WRITE(3)IM,(((DROP(K,J,I),K=1,3),J=1,JMAX),I=1,IM)
      NSAVE=0
      GO TO 80
   91 CONTINUE
      CALL CALPLT
      CALL PRTOUT(NTCYCL)
      IF(ISAVE)182,5,182
  181 WRITE(6,6666)
 6666 FORMAT(1H1,20X,63H********** ERROR ABORT  -  NEG. DENSITY IN THE F
     1IELD **********/21X,63H*************************************M**********
     2*********************)
      WRITE(6,606)T,NTCYCL
      JS=MAX0(J-2,1)
      JF=MIN0(J+2,JMAX)
      IS=MAX0(I-2,1)
      IF=MIN0(I+2,IMAX)
      WRITE(6,707)((J,I,(PROPTY(I,J,K),K=1,4),J=JS,JF),I=IS,IF)
```

```
  707 FORMAT(1H0//(2I4,4E23.7))
      GO TO 184
  182 CONTINUE
      REWIND 3
      WRITE(3)(((PROPTY(I,J,K),K=1,4),J=1,JMAX),I=1,IMAX),T,NTCYCL
      WRITE(3)IM,(((DROP(K,J,I),K=1,3),J=1,JMAX),I=1,IM)
  184 GO TO 5
  186 CONTINUE
      ENDFILE 3
      CALL PLOT(0.,0.,999)
      STOP
      END
      SUBROUTINE INITIL
C*********  SETS UP THE INITIAL DATA IN THE ARRAY PROPTY AFTER
C*********  COMPLETING THE NORMALIZATION FROM INPUT DATA  **************
C       ***************************************************************
C       ***************************************************************
C       ***************************************************************
C       * * * * * * * * * * * * * SCALING RULES * * * * * * * * * * * * *
C        * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C       EQ1*(TO/RHO0),,EQ2*(TO/(RHO0*AO)),,EQ3*(TO/(RHO0*AO)),,
C       EQ4*(TO/(RHO0*AO**2)),,,,TO=(RMAX/AO)
C        * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C       **********   TIME IS SCALED BY ...THE REF TIME SCALE   **********
C       **********   DISTANCE IS SCALED BY...COMBUSTION CHAMBER RADIUS***
C       **********   VELOCITY IS SCALED BY...REF SOUND SPEED   **********
C       **********   DENSITY IS SCALED BY...REFERENCE DENSITY************
C       **********   PRESSURE IS SCALED BY...REF SOUND SPEED**2*REF
C       **********   DENSITY****************************************
C       ***************************************************************
      COMMON PROPTY,GAMMA3,GAMMA1,GAMMA,DELR,DELO,PO,PMAX,PR,ER,TAU,IMAX
     1,JMAX,T,DTDR,DTDO,DT,P,AO,ROO   ,NCY
      DIMENSION PROPTY(21,20,4)
      COMMON/LE/ FJMAX,IMAXM1,X,Y,RAD(41),OVRLAY(12,20,21),
     1OVRPRM(16,20,21),HT(40)
      COMMON/PHIP/DROP(5,20,21),PHI,EN,FS,DMF,DMFT,IM,XM
      COMMON/NCPLOT/NTCYCL,ISTART,HGAM,LABLX(5),LABLY(4)
      DATA PI,TWOPI/3.14159265,6.28318531/
      READ(5,5)             GAMMA,PO,PMAX,TAU,RMAX
      WRITE(6,65)
   65 FORMAT(4X,5HGAMMA,5X,16HCHAMBER PRESSURE,2X,14HPULSE PRESSURE,4X,1
     1                                                                5H
     2PULSE DURATION,4X,
     3                14HCHAMBER RADIUS/20X,3HPSI,15X,3HPSI,15X,3HSEC,15X
     2,2HFT//)
C       * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C       * * * * * *  BOMB PULSE PRESSURE AND TIME DURATION ARE ONLY
C       * * * * * *  TO BE REFERENCED WHEN INITIAL CONDITIONS ARE
C       * * * * * *  FOR A PULSE TYPE CALCULATION  * * * * * * * * * * *
C       * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      WRITE(6,4)               GAMMA,PO,PMAX,TAU,RMAX
    4 FORMAT(1X,2E14.7,E18.7,E19.7,F12.4///)
      READ(5,5) XM,TO,VFACT,EN
      WRITE(6,68) EN
   68 FORMAT(3X10HMOL WEIGHT3X19HCHAMBER TEMPERATURE10X2HN=E10.3//)
      WRITE(6,5)XM,TO
```

```
      WRITE(6,76)
   76 FORMAT(1H ///)
C       GAMMA  SPECIFIC HEAT RATIO
C       PO  UNPERTURBED PRESSURE
C       PMAX  AMPLITUDE OF PRESSURE PERTURBATION
C       TAU IS DURATION OF PERT.
C       IMAX,JMAX MESH DIMENSIONS
C       XM  MOLECULAR WEIGHT, TO  CHAMBER TEMPERATURE
    7 FORMAT(2I4)
    5 FORMAT(4E14.7,2F10.4)
      P0=PO
      R00=(P0*144.)/((1545./XM)*T0)
      A0=SQRT (GAMMA*32.17*P0*144./R00)
C      RHO=R0/R00=1.
      RHO=1.
      TO=RMAX/AO
      Y=TO
      XTO=TO*1000.
      EQ2=R00*AO/TO
      P=P0*144.*32.17/(RMAX*EQ2)
      DELR=3.0/FLOAT(IMAXM1)
      DELO=2.*PI/FJMAX
      GAMMA1=GAMMA-1.
      GAMMA3=GAMMA-3.
      PHI=1.0
      FS=0.695652174
      DMFT=TO/R00
      DMF=DMFT*778.0*32.17/(AO*AO)
      HGAM=0.0
      PR=PMAX/PO
      ER=PR/GAMMA1
      WRITE(6,67)
   67 FORMAT(1X,16HREDUCED PRESSURE,2X,15HREF SOUND SPEED,2X,11HREF DENS
     1ITY,6X,7HDELTA R,7X,11HDELTA THETA,4X,43HPRESSURE AND INTERNAL ENE
     2RGY RATIO OF PULSE//)
       WRITE(6,6)          P,     AO,R00,DELR,DELO,PR,ER,XTO
    6 FORMAT(7E16.7///16H REF TIME SCALE=F15.5/3X,8HMILLISEC///)
      WRITE(6,566)
  566 FORMAT(1X,117H******************************************* END OF INIT
     1ILIZATION PHASE ***********************************************/
     2/////)
      RAD(1)=0.0
      DO 501 I=2,IMAX
 501  RAD(I)=RAD(I-1)+DELR
      HT(1)=0.0
      DO 502 J=2,JMAX
  502 HT(J)=HT(J-1)+DELO
      IF(ISTART)506,505,506
  505 CONTINUE
      READ(1) (((PROPTY(I,J,K),K=1,4),J=1,JMAX),I=1,IMAX),T,NTCYCL
      READ(1) IM,(((DROP(K,J,I),K=1,3),J=1,JMAX),I=1,IM)
      GO TO 504
  506 CONTINUE
      IQ=IMAX/10
      IM3=IMAX-3*IQ
      IM2=IM3+1
```

```
      IF(VFACT.NE.0.0) GO TO 512
      IS=IM2
      EPS=(PMAX-PO)/(PO*GAMMA)
      B=TWOPI/3.0
      UI=0.4
      QU=UI
      QK=-UI*B-SQRT(B*B-1.0)
      QKE=EPS/QK
      DO 510 I=1,IM3
      READ(5,600) SR,Q1,Q2,Q3
      SU=Q1/SR
      SP=GAMMA1*(Q3-0.5*Q1*SU)
      BZ=B*RAD(I)
      Q1=COS(BZ)
      Q2=QKE*B*SIN(BZ)
      Q3=QKE*Q1
      Q1=EPS*Q1
      DO 510 J=1,JMAX
      TH=HT(J)
      CTH=COS(TH)
      STH=SIN(TH)
      PRS=1.0-Q1*CTH
      RHOQ=SR*PRS
      PRS=SP*PRS
      U=SU*(1.0-Q2*STH)
      V=Q3*CTH
      PROPTY(I,J,1)=RHOQ
      PROPTY(I,J,2)=RHOQ*U
      PROPTY(I,J,3)=RHOQ*V
      PROPTY(I,J,4)=PRS/GAMMA1+0.5*RHOQ*(U*U+V*V)
  510 CONTINUE
      READ(5,600)((PROPTY(I,1,K),K=1,4),I=IM2,IMAX)
      GO TO 520
  512 CONTINUE
      IPOP=3
      IS=1
      JS=(3*JMAX)/4-1
      JF=JS+4
      QN=PI/(6.0*DELO)
      GAMEXP=GAMMA1/GAMMA
      GAMINV=1.0/GAMMA
      READ(5,600)((PROPTY(I,1,K),K=1,4),I=1,IMAX)
      EPOP=0.0
      DO 515 I=1,IM3
  515 EPOP=EPOP+PROPTY(I,1,4)
      EPOP=VFACT*EPOP*FJMAX
      SMA=0.0
      DO 518 J=JS,JF
  518 SMA=SMA+SIN(QN*HT(J))**GAMEXP
      A=(GAMMA1*EPOP/SMA)**(GAMMA/GAMMA1)
  520 CONTINUE
      DLN=3.2808E-4
      TL=8700./(15.0622979-ALOG(PO))
      TRM=1.0-TL/1175.69
      RHOL=14.4+TRM*(86.82-60.654*TRM)+6.3431*SQRT(TRM)+17.716
     1*TRM**0.333333333
```

```
      WL=PI*RHOL*DLN**3/6.0
      UD=120.0
      VD=0.0
      IM=IMAX-4*IQ
      Q1=(QU*AO-UD)/FLOAT(IM-1)
      Q2=9.0/FLOAT(IM-1)
      DO 522 I=1,IM
      QI=I-1
      Q=1.0+QI*Q2
      DROP(1,1,I)=WL/Q**3
      DROP(2,1,I)=UD+QI*Q1
  522 DROP(3,1,I)=0.0
      DO 530 J=2,JMAX
      DO 525 I=1,IM
      DO 525 K=1,3
  525 DROP(K,J,I)=DROP(K,1,I)
      DO 530 I=IS,IMAX
      DO 530 K=1,4
      PROPTY(I,J,K)=PROPTY(I,1,I
  530 CONTINUE
  504 IF(EN.NE.0.0) GO TO 545
      DO 540 I=1,IMAX
      DO 540 J=1,JMAX
      DO 540 K=1,5
      DROP(K,J,I)=0.0
  540 CONTINUE
  545 CONTINUE
      IF(VFACT.EQ.0.0) RETURN
      SU=PROPTY(IPOP,1,2)/PROPTY(IPOP,1,1,
      SP=GAMMA1*(PROPTY(IPOP,1,4)-0.5*PROPTY(IPOP,1,2)*SU)
      DO 550 J=JS,JF
      PRP=0.5*A*SIN(QN*HT(J))
      RHOQ=PROPTY(IPOP,1,1)+PRP**GAMINV
      PRS=SP+PRP
      PROPTY(IPOP,J,1)=RHOQ
      PROPTY(IPOP,J,2)=RHOQ*SU
      PROPTY(IPOP,J,4)=PRS/GAMMA1+0.5*PROPTY(IPOP,J,2)*SU
      PRP=0.5*PRP
      RHOQ=PROPTY(IPOP,1,1)+PRP**GAMINV
      PRS=SP+PRP
      PROPTY(IPOP-1,J,1)=RHOQ
      PROPTY(IPOP-1,J,2)=RHOQ*SU
      PROPTY(IPOP-1,J,4)=PRS/GAMMA1+0.5*PROPTY(IPOP-1,J,2)*SU
      PROPTY(IPOP+1,J,1)=PROPTY(IPOP-1,J,1)
      PROPTY(IPOP+1,J,2)=PROPTY(IPOP-1,J,2)
      PROPTY(IPOP+1,J,4)=PROPTY(IPOP-1,J,4)
  550 CONTINUE
      RETURN
  600 FORMAT(4E15.8)
      END
      SUBROUTINE CHARGE
C*********  COMPUTES BOUNDARY CONDITIONS AT THE PERIPHERY
C*********  OF THE CYLINDRICAL SURFACE    ********************************
      COMMON PROPTY,GAMMA3,GAMMA1,GAMMA,DELR,DELO,PO,PMAX,PR,ER,TAU,IMAX
     1,JMAX,T,DTDR,DTDO,DT,P,AO,ROO      ,NCY
      DIMENSION PROPTY(21,20,4)
```

```
      COMMON/LE/ FJMAX,IMAXM1,X,Y,RAD(41),OVRLAY(12,20,21),
     1OVRPRM(16,20,21),HT(40)
      COMMON/PHIP/DROP(5,20,21),PHI,EN,FS,DMF,DMFT,IM,XM
      COMMON/NCPLOT/NTCYCL,ISTART,HGAM,LABLX(5),LABLY(4)
      DO 60 J=1,JMAX
      JP1=J+1
      IF(JP1.GT.JMAX) JP1=1
      JM1=J-1
      IF(JM1.EQ.0) JM1=JMAX
      DO 50 K=1,4
      FD=OVRPRM(K+4,J,2)+OVRPRM(K+4,JM1,2)
      GD=OVRPRM(K+8,J,2)-OVRPRM(K+8,JM1,2)
      PROPTY(1,J,K)=PROPTY(1,J,K)+(DTDR*(CVRLAY(K,U,2)+FD)*0.5+DTDO*
     1((OVRLAY(K+4,JP1,1)-OVRLAY(K+4,JM1,1))*0.5+GD)*0.5)
   50 CONTINUE
      PROPTY(1,J,1)=PROPTY(1,J,1)+DT*DROP(4,J,1)
      PROPTY(1,J,2)=0.0
      PROPTY(1,J,4)=PROPTY(1,J,4)+DT*DROP(5,J,1)
      DO 60 K=1,4
      PROPTY(IMAX,J,K)=PROPTY(IMAX-1,J,K)+PROPTY(IMAX-1,J,K)
     1-PROPTY(IMAX-2,J,K)
   60 CONTINUE
      RETURN
      END
      SUBROUTINE CONVRG(FACTOR)
C**********  COMPUTES A TIME INCREMENT TO BE USED IN DIFFERENCE EQUATION
C**********  BASED ON C.F.L. CONDITION                *********************
      COMMON PROPTY,GAMMA3,GAMMA1,GAMMA,DELR,DELO,PO,PMAX,PR,ER,TAU,IMAX
     1,JMAX,T,DTDR,DTDO,DT
     1,P,AO,ROO,NCY
      DIMENSION PROPTY(21,20,4)
      COMMON/LE/ FJMAX,IMAXM1,X,Y,RAD(41),OVRLAY(18,20,21),
     1OVRPRM(16,20,21),HT(40)
      DT=0.3/(16.*.75/3.9445)
      DT=DT*FACTOR
      SQ=1./SQRT (2.)
      DO 1 I=2,IMAXM1
      DO 1 J=1,JMAX
      VELSQR=(PROPTY(I,J,2)**2+PROPTY(I,J,3)**2)/PROPTY(I,J,1)**2
      IF(VELSQR) 8,8,2
    2 BG=PROPTY(I,J,4)/PROPTY(I,J,1)-0.5*VELSQR
      IF(BG) 8,7,7
    7 DENOM=SQRT (GAMMA*GAMMA1*BG)
      TERM=SQ/(DENOM+SQRT (VELSQR))
      TERMO=TERM*FACTOR*DELO
      TERMR=TERM*FACTOR*DELR
      DTE=DT
      DT=AMIN1(DT,TERMR,TERMO)
      IF(DT-DTE) 150,8,8
  150 IT=I
      JT=J
    8 CONTINUE
    1 CONTINUE
      DTDR=DT/DELR
      DTDO=DT/DELO
      RETURN
```

```
      END
      SUBROUTINE VECTR(IF,IL)
C*********  COMPUTES THE VALUES OF THE VECTOR FUNCTIONS OF W WHERE W
C*********  IS THE VECTOR OF CONSERVATION VARIABLES  ******************
      COMMON PROPTY,GAMMA3,GAMMA1,GAMMA,DELR,DELO,PO,PMAX,PR,ER,TAU,IMAX
     1,JMAX,T,DTDR,DTDO,DT,P,AO,ROO,NCY
      DIMENSION PROPTY(21,20,4)
      COMMON/LE/ FJMAX,IMAXM1,X,Y,RAD(41),OVRLAY(12,20,21),
     1OVRPRM(16,20,21),HT(40)
      DO 50 I=IF,IL
      ZZ=RAD(I)
      ZF=FZ(ZZ)
      DO 50 J=1,JMAX
      U=PROPTY(I,J,2)/PROPTY(I,J,1)
      V=PROPTY(I,J,3)/PROPTY(I,J,1)
      PRS=GAMMA1*(PROPTY(I,J,4)-0.5*PROPTY(I,J,1)*(U*U+V*V))
      OVRLAY(1,J,I)=-PROPTY(I,J,2)
      OVRLAY(2,J,I)=OVRLAY(1,J,I)*U
      OVRLAY(3,J,I)=OVRLAY(1,J,I)*V
      OVRLAY(4,J,I)=-PROPTY(I,J,4)-PRS
      OVRLAY(5,J,I)=-PROPTY(I,J,3)
      OVRLAY(6,J,I)=OVRLAY(3,J,I)
      OVRLAY(7,J,I)=OVRLAY(5,J,I)*V-PRS
      OVRLAY(8,J,I)=OVRLAY(4,J,I)*V
      OVRLAY(9,J,I)=OVRLAY(1,J,I)*ZF
      OVRLAY(10,J,I)=OVRLAY(2,J,I)*ZF
      OVRLAY(11,J,I)=OVRLAY(3,J,I)*ZF
      OVRLAY(12,J,I)=-PROPTY(I,J,4)*U*ZF
      OVRLAY(2,J,I)=OVRLAY(2,J,I)-PRS
      OVRLAY(4,J,I)=OVRLAY(4,J,I)*U
   50 CONTINUE
      RETURN
      END
      SUBROUTINE QUAD(LF,LL)
C*********  COMPUTES W..SQUIGGLE(T+DELT), THE TEMPORARY VALUE
C*********  OF THE CONSERVATION VECTOR   ****************************
      COMMON PROPTY,GAMMA3,GAMMA1,GAMMA,DELR,DELO,PO,PMAX,PR,ER,TAU,IMAX
     1,JMAX,T,DTDR,DTDO,DT,P,AO,ROO,NCY
      DIMENSION PROPTY(21,20,4)
      COMMON/LE/ FJMAX,IMAXM1,X,Y,RAD(41),OVRLAY(12,20,21),
     1OVRPRM(16,20,21),HT(40)
      COMMON/PHIP/DROP(5,20,21),PHI,EN,FS,DMF,DMFT,IM,XM
      DIMENSION AVEW(4)
      IM1=IM-1
      DO 60 J=1,JMAX
      JP1=J+1
      IF(JP1.GT.JMAX) JP1=1
      DO 40 L=LF,LL
      DO 2 K=1,4
    1 AVEW(K)=0.25*(PROPTY(L,J,K)+PROPTY(L+1,J,K)+PROPTY(L+1,JP1,K)
     1+PROPTY(L,JP1,K))
      TEMPW1=AVEW(K)+(DTDR*(OVRLAY(K,J,L+1)-OVRLAY(K,J,L)+OVRLAY(K,
     1JP1,L+1)-OVRLAY(K,JP1,L))    +DTDO*(OVRLAY(K+4,JP1,L)-OVRLAY(K+4,J
     2,L)+OVRLAY(K+4,JP1,L+1)-OVRLAY(K+4,J,L+1)))*0.5+0.25*(OVRLAY(K+8
     3,J,L)+OVRLAY(K+8,JP1,L)+OVRLAY(K+8,JP1,L+1)+OVRLAY(K+8,J,L+1))*DT
    2 OVRPRM(K,J,L+1)=TEMPW1
```

```
   40 CONTINUE
      IF(EN.EQ.0.0) GO TO 60
      DO 50 L=LF,IM1
      OVRPRM(1,J,L+1)=OVRPRM(1,J,L+1)+0.25*(DROP(4,J,L)+DROP(4,JP1,L)+
     1DROP(4,JP1,L+1)+DROP(4,J,L+1))*DT
      OVRPRM(4,J,L+1)=OVRPRM(4,J,L+1)+0.25*(DROP(5,J,L)+DROP(5,JP1,L)+
     1DROP(5,JP1,L+1)+DROP(5,J,L+1))*DT
   50 CONTINUE
   60 CONTINUE
      RETURN
      END
      SUBROUTINE TMPVCT(IF,IL)
      COMMON PROPTY,GAMMA3,GAMMA1,GAMMA,DELR,DELO,PO,PMAX,PR,ER,TAU,IMAX
     1,JMAX,T,DTDR,DTDO,DT,P,AO,ROO,NCY
      DIMENSION PROPTY(21,20,4)
      COMMON/LE/ FJMAX,IMAXM1,X,Y,RAD(41),OVRLAY(12,20,21),
     1OVRPRM(16,20,21),HT(40)
      DELZ2=0.5*DELR
      DO 50 I=IF,IL
      ZZ=RAD(I)-DELZ2
      ZF=FZ(ZZ)
      DO 50 J=1,JMAX
      U=OVRPRM(2,J,I)/OVRPRM(1,J,I)
      V=OVRPRM(3,J,I)/OVRPRM(1,J,I)
      PRS=GAMMA1*(OVRPRM(4,J,I)-0.5*OVRPRM(1,J,I)*(U*U+V*V))
      OVRPRM(5,J,I)=-OVRPRM(2,J,I)
      OVRPRM(6,J,I)= OVRPRM(5,J,I)*U
      OVRPRM(7,J,I)= OVRPRM(5,J,I)*V
      OVRPRM(8,J,I)=-OVRPRM(4,J,I)-PRS
      OVRPRM(9,J,I)=-OVRPRM(3,J,I)
      OVRPRM(10,J,I)= OVRPRM(7,J,I)
      OVRPRM(11,J,I)= OVRPRM(9,J,I)*V-PRS
      OVRPRM(12,J,I)= OVRPRM(8,J,I)*V
      OVRPRM(13,J,I)= OVRPRM(5,J,I)*ZF
      OVRPRM(14,J,I)= OVRPRM(6,J,I)*ZF
      OVRPRM(15,J,I)= OVRPRM(7,J,I)*ZF
      OVRPRM(16,J,I)=-OVRPRM(4,J,I)*U*ZF
      OVRPRM(6,J,I)= OVRPRM(6,J,I)-PRS
      OVRPRM(8,J,I)= OVRPRM(8,J,I)*U
   50 CONTINUE
      RETURN
      END
      SUBROUTINE GENPT(IF,IL)
C*********  COMPUTES THE VALUE OF W(T+DELT) AT REGULAR MESH POINTS
C*********  AND AT BOUNDARY POINTS        ******************************
      COMMON PROPTY,GAMMA3,GAMMA1,GAMMA,DELR,DELO,PO,PMAX,PR,ER,TAU,IMAX
     1,JMAX,T,DTDR,DTDO,DT
     1,P,AO,ROO,NCY
      DIMENSION PROPTY(21,20,4)
      COMMON/LE/ FJMAX,IMAXM1,X,Y,RAD(41),OVRLAY(12,20,21)
     1OVRPRM(16,20,21),HT(40)
      COMMON/PHIP/DROP(5,20,21),PHI,EN,FS,DMF,DMFT,IM,XM
      DIMENSION FDRVBR(4),GDRVBR(4),SAVEBR(4)
      DO 10 J=1,JMAX
   50 JP1=J+1
      JM1=J-1
```


5 ⊕

```
      IF(JP1.LE.JMAX) GO TO 7
      JP1=1
    7 IF(JM1.EQ.0)JM1=JMAX
      DO 5 I=IF,IL
      DO 4 K=1,4
      K4=K+4
    1 FDRVBR(K)=(OVRPRM(K4,J,I+1)-OVRPRM(K4,J,I)+OVRPRM(K4,JM1,I+1)-
     1OVRPRM(K4,JM1,I))
      K8=K+8
    2 GDRVBR(K)=(OVRPRM(K8,J,I+1)-OVRPRM(K8,JM1,I+1)+OVRPRM(K8,J,I)-
     1OVRPRM(K8,JM1,I))
      K12=K+12
    3 SAVEBR(K)=.25*(OVRPRM(K12,J,I+1)+OVRPRM(K12,J,I)+OVRPRM(K12,JM1,I
     4+1)+OVRPRM(K12,JM1,I))
    4 PROPTY(I,J,K)=PROPTY(I,J,K)+(DTDR*((OVRLAY(K,J,I+1)-OVRLAY(K,J,I-1
     1))+FDRVBR(K))*.25+DTDO*((OVRLAY(K4,JP1,I)-OVRLAY(K4,JM1,I))+GDRVBR
     2(K))*.25+DT*((OVRLAY(K8,J,I+1)+OVRLAY(K8,J,I-1))*.5+SAVEBR(K))*.5)
    5 CONTINUE
      IF(EN.EQ.0.0) GO TO 10
      DO 8 I=IF,IM
      PROPTY(I,J,1)=PROPTY(I,J,1)+DT*DROP(4,J,I)
    8 PROPTY(I,J,4)=PROPTY(I,J,4)+DT*DROP(5,J,I)
   10 CONTINUE
      RETURN
      END
      SUBROUTINE PRTOUT(CYCLE)
C*********  PRINTS A TERSE VIEW OF THE FLOW FIELD ALONG THE
C*********  RAYS 0,PI/2,PI,3*PI/2       ***********************************
      COMMON PROPTY,GAMMA3,GAMMA1,GAMMA,DELR,DELO,PO,PMAX,PR,ER,TAU,IMAX
     1,JMAX,T,DTDR,DTDO,DELT ,P,AO,ROO ,NCYCMX
      DIMENSION PROPTY(21,20,4)
      COMMON/PRTOPT/INTPNT
      INTEGER CYCLE
   75 LF=0
      L=1
      WRITE(6,606) T,CYCLE
      IF(INTPNT.EQ.0) GO TO 1818
      WRITE(6,707)((J,I,(PROPTY(I,J,K),K=1,4),J=1,JMAX),I=1,IMAX)
  707 FORMAT(1X,2I4,4E23.7)
 1818 CONTINUE
      WRITE(6,6062)
  606 FORMAT(1H110X11HTOTAL TIME=E12.5,3X,17HNUMBER OF CYCLES=I4)
 6165 FORMAT(8X,5H RHO ,14X,3H U ,14X,3H V ,19X,10HINT ENERGY,8X,10H PRE
     1SSURE ,10X,9H MACH NO.///)
 6061 FORMAT(1X,6E19.5)
 6062 FORMAT(1H0,20X,32HSOLUTION ALONG THE RAY THETA = 0///)
 6063 FORMAT(1H0,20X,33HSOLUTION ALONG THE RAY THETA = PI///)
 6064 FORMAT(1H0,20X,35HSOLUTION ALONG THE RAY THETA = PI/2///)
 6065 FORMAT(1H0,20X,37HSOLUTION ALONG THE RAY THETA = 3*PI/2///)
  402 CONTINUE
      WRITE(6,6165)
      DO 1012 I=1,IMAX
      RHOX=PROPTY(I,L,1)
      UX=PROPTY(I,L,2)/RHOX
      VX=PROPTY(I,L,3)/RHOX
      EX=(PROPTY(I,L,4)/RHOX)-0.5*(UX**2+VX**2)
```

```
      PX=GAMMA1*RHOX*EX
      XMX=SQRT((UX**2+VX**2)/(GAMMA*PX/RHOX))
 1011 WRITE(6,6061)              RHOX,UX,VX,EX,PX,XMX
 1012 CONTINUE
      LF=LF+1
      GO TO(404,504,403,1001),LF
  404 CONTINUE
      L=JMAX/2+1
      WRITE(6,6063)
      GO TO 402
  504 CONTINUE
      L=JMAX/4+1
      WRITE(6,6064)
      GO TO 402
  403 CONTINUE
      L=3*(JMAX/4)+1
      WRITE(6,6065)
      GO TO 402
 1001 CONTINUE
      RETURN
      END
      FUNCTION FZ(Z)
      COMMON/FZAB/A,B,BB
      DZ=Z-2.1
      IF(DZ)10,10,20
   10 FZ=0.0
      RETURN
   20 FZ=(BB*DZ-A)/(DZ*(B*DZ-A)+1.0)
      RETURN
      END
      SUBROUTINE PHIDOTV
      COMMON PROPTY,GAMMA3,GAMMA1,GAMMA,DELR,DELO,PO,PMAX,PR,ER,TAU,IMAX
     1,JMAX,T,DTDR,DTDO,DT,P,AO,ROO      ,NCY
      DIMENSION PROPTY(21,20,4)
      COMMON/LE/ FJMAX,IMAXM1,X,Y,RAD(41),OVRLAY(12,20,21),
     1OVRPRM(16,20,21),HT(40)
      COMMON/PHIP/DROP(5,20,21),PHI,EN,FS,DMF,DMFT,IM,XM
      DIMENSION PHIIN(4),TT(12),DHRT(12)
      DATA TWOPI,PI,RZ/6.28318531,3.14159265,1544./
      DATA SPI,ONE3/1.9098593  ,0.333333333/
      DATA G,CPSINF/32.17,.3539658/
      DATA TT,DHRT/10800.0,9900.0,9000.0,8100.0,7200.0,6300.0,5400.0,
     X4500.0,3600.0,2700.0,1800.0,536.4,225.0,1010.0,1860.0,2580.0,
     X3380.0,4160.0,4900.0,5620.0,6300.0,6970.0,7540.0,8150.0/
      DLT=Y*DT
      DTA=DT/AO
      PHITFS=PHI*FS
      ELFZ=540.0+178.2/PHITFS
      WLMIN=0.001*DROP(1,1,2)
      DO 100 J=1,JMAX
      DO 90 I=1,IM
      WL=DROP(1,J,I)
      V1=DROP(2,J,I)
      V2=DROP(3,J,I)
      IF(WL.GT.WLMIN) GO TO 5
      OVRPRM(1,J,I+1)=0.0
```

```
      OVRPRM(2,J,I+1)=V1
      OVRPRM(3,J,I+1)=V2
      OVRPRM(4,J,I+1)=HT(J)
      OVRPRM(5,J,I+1)=RAD(I)
      DROP(4,J,I)=0.0
      DROP(5,J,I)=0.0
      GO TO 90
    5 CONTINUE
      DO 10 K=1,4
   10 PHIIN(K)=PROPTY(I,J,K)
      ELITLE=(PHIIN(4)-0.5*(PHIIN(2)**2+PHIIN(3)**2)/PHIIN(1))/PHIIN(1)
      PINF=PHIIN(1)*GAMMA1*ELITLE*ROO*AO*AO/(144.*8)
      RHOINF=PHIIN(1)*ROO
      TINF=PINF*144.0*XM/(1545.*RHOINF)
      U1=PHIIN(2)/PHIIN(1)*AO
      U2=PHIIN(3)/PHIIN(1)*AO
      DUV1=U1-V1
      DUV2=U2-V2
      UVMG=SQRT(DUV1*DUV1+DUV2*DUV2)
      A300=3.73E-5*TINF+1.855
      AP=A300*(PINF/300.)**.0108
      IF(PINF-2131.)25,20,20
   20 TL=1175.69
      RHOL=14.4
      GO TO 30
   25 CONTINUE
      TL=8700./(15.0622979-ALOG(PINF))
      TRM=1.0-TL/1175.69
      RHOL=14.4+TRM*(86.82-60.654*TRM)+6.3431*SQRT(TRM)+17.716*TRM**ONE3
   30 CONTINUE
      DLN=(SPI*WL/RHOL)**ONE3
      SK=.691716667E-4-2.31388889E-8*TL
      CP=.137857+TL*(.52715E-3-TL*.119907E-6)
      EMFZ=TWOPI*DLN*SK/CP*AP
      PR3=0.591
      UINF=2.5E-5*SQRT(TINF/560.)
      RE=RHOINF*DLN*UVMG/UINF
      EMFD=EMFZ*(1.0+0.276*SQRT(RE)*PR3)
      PSI=EN*EMFD*(1.+1./PHITFS)
      DROP(4,J,I)=PSI*DMFT
      CALL SEEK(TINF,TT,11,IJ)
   60 FORMAT(11H SEEK ERROR,2E20.10)
      IF(IJ)80,70,80
   70 PRINT 60,TINF,TT(1)
      STOP
   80 CONTINUE
      DHR=DHRT(IJ)+(DHRT(IJ+1)-DHRT(IJ))*(TINF-TT(IJ))/(TT(IJ+1)-TT(IJ))
      XIS=EN*EMFD*(DHR-ELFZ)
      DROP(5,J,I)=XIS*DMF
      OVRPRM(1,J,I+1)=WL-EMFD*DLT
      IF(OVRPRM(1,J,I+1).LT.0.0) OVRPRM(1,J,I+1)=0.0
      CD=20.25*RE**(-0.84)
      CDR=CD*RHOINF/(RHOL*DLN)*DLT
      OVRPRM(2,J,I+1)=V1+CDR*DUV1*ABS(DUV1)
      OVRPRM(3,J,I+1)=V2+CDR*DUV2*ABS(DUV2)
      OVRPRM(4,J,I+1)=HT(J)+V2*DTA
```

```
      OVRPRM(5,J,I+1)=RAD(I)+V1*DTA
   90 CONTINUE
  100 CONTINUE
      CALL MSHVL
      RETURN
      END
      SUBROUTINE MSHVL
      COMMON PROPTY,GAMMA3,GAMMA1,GAMMA,DELR,DELO,PO,PMAX,PR,ER,TAU,IMAX
     1,JMAX,T,DTDR,DTDO,DT,P,AO,ROO    ,NCY
      DIMENSION PROPTY(21,20,4)
      COMMON/LE/ FJMAX,IMAXM1,X,Y,RAD(41),OVRLAY(12,20,21),
     1OVRPRM(16,20,21),HT(40)
      COMMON/PHIP/DROP(5,20,21),PHI,EN,FS,DMF,DMFT,IM,XM
      DATA TWOPI/6.28318531/
      DO 60 I=2,IM
      DO 20 J=1,JMAX
      Q=(RAD(I)-OVRPRM(5,J,I))/(OVRPRM(5,J,I+1)-OVRPRM(5,J,I))
      DO 10 K=1,4
   10 OVRPRM(K+5,J,I)=OVRPRM(K,J,I)+Q*(OVRPRM(K,J,I+1)-OVRPRM(K,J,I))
   20 CONTINUE
      DO 50 J=1,JMAX
      Q=HT(J)-OVRPRM(9,J,I)
      JJ=J+SIGN(1.0,Q)
      IF(JJ)30,25,30
   25 JJ=JMAX
      Q=Q/(OVRPRM(9,JJ,I)-TWOPI-OVRPRM(9,J,I))
      GO TO 45
   30 IF(JJ-JMAX)40,40,35
   35 JJ=1
      Q=Q/(OVRPRM(9,JJ,I)+TWOPI-OVRPRM(9,J,I))
      GO TO 45
   40 Q=Q/(OVRPRM(9,JJ,I)-OVRPRM(9,J,I))
   45 CONTINUE
      DO 50 K=1,3
   50 DROP(K,J,I)=OVRPRM(K+5,J,I)+Q*(OVRPRM(K+5,JJ,I)-OVRPRM(K+5,J,I))
   60 CONTINUE
      RETURN
      END
      SUBROUTINE SMOOTH
      COMMON PROPTY,GAMMA3,GAMMA1,GAMMA,DELR,DELO,PO,PMAX,PR,ER,TAU,IMAX
     1,JMAX,T,DTDR,DTDO,DELT ,P,AO,ROO ,NCYCMX
      DIMENSION PROPTY(21,20,4)
      COMMON/LE/ FJMAX,IMAXM1,X,Y,RAD(41),OVRLAY(12,20,21),
     1OVRPRM(16,20,21),HT(40)
      SMK=0.5
      DTH=SMK*DTDO
      DZ=SMK*DTDR
      DO 20 I=2,IMAXM1
      V2=PROPTY(I,JMAX,3)/PROPTY(I,JMAX,1)
      DO 20 J=1,JMAX
      V1=V2
      V2=PROPTY(I,J,3)/PROPTY(I,J,1)
      DV=ABS(V2-V1)
      JM1=J-1
      IF(JM1.EQ.0) JM1=JMAX
      DO 20 K=1,4
```

```
 20 OVRPRM(K,J,I)=DV*(PROPTY(I,J,K)-PROPTY(I,JM1,K))
    DO 30 I=2,IMAXM1
    DO 30 J=1,JMAX
    JP1=J+1
    IF(JP1.GT.JMAX) JP1=1
    DO 30 K=1,4
 30 PROPTY(I,J,K)=PROPTY(I,J,K)+DTH*(OVRPRM(K,JP1,I)-OVRPRM(K,J,I))
    DO 40 J=1,JMAX
    U2=PROPTY(1,J,2)/PROPTY(1,J,1)
    DO 40 I=2,IMAXM1
    U1=U2
    U2=PROPTY(I,J,2)/PROPTY(I,J,1)
    DU=ABS(U2-U1)
    DO 40 K=1,4
 40 OVRPRM(K,J,I)=DU*(PROPTY(I,J,K)-PROPTY(I-1,J,K))
    DO 50 J=1,JMAX
    DO 50 I=2,IMAXM1
    DO 50 K=1,4
 50 PROPTY(I,J,K)=PROPTY(I,J,K)+DZ*(OVRPRM(K,J,I+1)-OVRPRM(K,J,I))
    RETURN
    END
    SUBROUTINE CALPLT
    COMMON PROPTY,GAMMA3,GAMMA1,GAMMA,DELR,DELO,PO,PMAX,PR,ER,TAU,IMAX
   1,JMAX,T,DTDR,DTDO,DELT ,P,AO,ROO ,NCYCMX
    DIMENSION PROPTY(21,20,4)
    COMMON/LE/ FJMAX,IMAXM1,X,Y,RAD(41),OVRLAY(12,20,21),
   1OVRPRM(16,20,21),HT(40)
    DIMENSION XX( 420),YY( 420),UV( 420),THT( 420)
    DIMENSION AA(20,21),CLEVELS(20)
    EQUIVALENCE (OVRLAY,XX),(OVRLAY( 421),YY),(OVRLAY( 841),UV),
   1(OVRLAY(1261),THT),(OVRLAY(1681),AA)
    COMMON/NCPLOT/NTCYCL,ISTART,HGAM,LABLX(5),LABLY(4)
    EXTERNAL FX,FY
    DATA PI,TWOPI/3.14159265,6.28318531/
    TD=T*Y*1000.
100 FORMAT(2HT=F7.3,I9,7H CYCLES)
    ENCODE(25,100,LABLX)TD,NTCYCL
    LABLX(5)=0
    Q=0.0
    GO TO 145
    DO 140 I=1,IMAX
    DO 140 J=1,JMAX
    AA(J,I)=PROPTY(I,J,1)
140 CONTINUE
    CALL CONTOUR(AA,20,JMAX,IMAX,JMAX,IMAX,0.0,3.0,0.0,TWOPI,10.0,
   *10.0,-15,CLEVELS,11HPLOT OF RHO,LABLX,LABLY   ,FX,FY,Q)
145 CONTINUE
    DO 150 I=1,IMAX
    DO 150 J=1,JMAX
    AA(J,I)=GAMMA1*(PROPTY(I,J,4)-.5*(PROPTY(I,J,2)**2+PROPTY(I,J,3)
   ***2)/PROPTY(I,J,1))*GAMMA
150 CONTINUE
    CALL CONTOUR(AA,20,JMAX,IMAX,JMAX,IMAX,0.0,3.0,0.0,TWOPI,10.0,
   *10.0,-15,CLEVELS,16HPLOT OF PRESSURE,LABLX,LABLY   ,FX,FY,Q)
    RAD(41)=-RAD(41)
    IF(RAD(41).LT.0.0) GO TO 603
```

```
      KK=0
      DO 91 I=1,IMAX
      DO 91 J=1,JMAX
      KK=KK+1
      UV(KK)=SQRT(PROPTY(I,J,2)**2+PROPTY(I,J,3)**2)/PROPTY(I,J,1)
      IF(UV(KK).NE.0.0) GO TO 80
      KK=KK-1
      GO TO 91
   80 CONTINUE
      XX(KK)=RAD(I)
      YY(KK)=HT(J)
      THT(KK)=ATAN2(PROPTY(I,J,3),PROPTY(I,J,2))
   91 CONTINUE
      CALL PLOT(2.,.50,-3)
      EMXY=0.
      DO 160 I=1,KK
      IF(UV(I)-EMXY)160,160,155
  155 EMXY=UV(I)
  160 CONTINUE
      CALL VECTORF(XX,YY,UV,THT,KK,0.0,0.3,0.0,.628318531,EMXY,0.5,10.0,
     X10.0,19HVECTOR PLOT OF UMAG,LABLX,LABLY)
      CALL PLOT(12.,-11.,-3)
  603 CONTINUE
      RETURN
      END
      FUNCTION FX(X)
      FX=X
      RETURN
      END
      FUNCTION FY(Y)
      FY=Y
      RETURN
      END
      SUBROUTINE VECTORF(X,Y,R,THETA,N,X0,DX,Y0,DY,RREF,VREF,XSIZE,
     1 YSIZE,ITITLE,LABELX,LABELY)
C
C  VECTORF PRODUCES A VECTOR FIELD PLOT COMPLETE WITH AXES AND LABELS.
C  THE PLOTTER ORIGIN, (OR REFERENCE POINT) CORRESPONDS TO (X0,Y0).
C  ALL MOVEMENTS ARE RELATIVE TO THIS POINT.  VECTORF ASSUMES THIS
C  ORIGIN TO HAVE BEEN PROPERLY SET BY THE PROGRAMMER BEFORE ENTRANCE
C  TO VECTORF.  THIS IS DONE BY CALLING -PLOT- WITH A NEGATIVE THIRD
C  ARGUMENT.  E.G. CALL PLOT(2.,.58,-3)
C
C  X -- AN ARRAY CONTAINING UNSCALED ABSCISSA VALUES.
C  Y -- AN ARRAY CONTAINING UNSCALED ORDINATE VALUES.
C  R -- AN ARRAY CONTAINING UNSCALED VALUES OF THE VECTOR MAGNITUDES.
C         I.E. R(I) IS THE MAGNITUDE OF THE VECTOR AT (X(I),Y(I)).
C  THETA --AN ARRAY CONTAINING VALUES OF THE VECTOR DIRECTIONS, IN
C         RADIANS, WITH RESPECT TO THE POSITIVE X DIRECTION.
C         I.E. THETA (I) IS THE DIRECTION OF THE VECTOR AT (X(I),Y(I)).
C  N -- THE NUMBER OF VECTORS TO BE PLOTTED.  X, Y, R, AND THETA MUST
C         BE DIMENSIONED AT LEAST N IN THE CALLING ROUTINE.
C  X0 --  THE VALUE OF X AT THE REFERENCE POINT.  THIS MAY BE ANY NUMBER
C         LESS THAN OR EQUAL TO THE SMALLEST VALUE IN THE ARRAY X.
C         X0 IS THE VALUE AT THE LEFT END OF THE X-AXIS.
C  DX -- THE CHANGE IN X PER INCH ALONG THE X-AXIS.  DX SHOULD BE
```

```
C          CHOSEN SUCH THAT (X0+DX*XSIZE) .GE. THE LARGEST VALUE IN
C          THE ARRAY X. X0, DX, AND XSIZE SHOULD BE CHOSEN TO PROVIDE
C          A CONVENIENT SCALE ALONG THE X-AXIS.
C   Y0, DY -- ARE DEFINED FOR Y AND THE Y-AXIS AS X0, DX ARE FOR THE
C          X-AXIS.  Y0 IS THE VALUE AT THE LOWER END OF THE Y-AXIS.
C   RREF -- THE VECTOR MAGNITUDE WHICH IS REPRESENTED ON THE PLOT BY A
C          VECTOR VREF INCHES LONG.
C   VREF -- THE LENGTH, IN INCHES, OF THE VECTOR WITH MAGNITUDE RREF.
C          RREF/VREF IS USED AS THE SCALE FACTOR FOR R.
C   XSIZE -- THE LENGTH IN INCHES OF THE X-AXIS.  MINIMUM IS 7 INCHES.
C   YSIZE -- THE LENGTH IN INCHES OF THE Y-AXIS.  MAXIMUM IS 10 INCHES.
C   ITITLE -- AN ARRAY CONTAINING PACKED DISPLAY CODE,TERMINATED BY A
C          ZERO WORD.  THIS TEXT IS WRITTEN ACROSS THE TOP OF THE PLOT.
C          A HOLLERITH LITERAL ARGUMENT SUCH AS 17HVECTOR FIELD PLOT,
C          WILL PRODUCE SUCH AN ARRAY.  THE TEXT WILL BE CENTERED ALONG
C          THE LENGTH OF THE PLOT.
C   LABELX -- SAME AS ITITLE, BUT WRITTEN ALONG THE X-AXIS.
C   LABELY -- SAME AS ITITLE, BUT WRITTEN ALONG THE Y-AXIS.
C
C   VMIN IS THE MINIMUM SIZE VECTOR THAT WILL BE PLOTTED.  ALL VECTORS
C          WHICH, AFTER SCALING, ARE .LT. VMIN INCHES LONG WILL BE DRAWN
C          VMIN INCHES LONG.  THIS MINIMUM MAGNITUDE IS COMPUTED AND
C          STORED IN RMIN.
C   VREF, RREF, RMIN, AND VMIN ARE WRITTEN WITH APPROPRIATE TEXT DIRECTLY
C          UNDER THE TITLE OF THE PLOT.
C   IF THE VALUES X=0, OR Y=0 OCCUR ALONG THE X AND/OR Y AXES, DASHED
C          LINES WILL BE DRAWN TO INDICATE THEIR POSITION.
C   THE GRAPH SIZE, XSIZE*YSIZE, DOES NOT INCLUDE LABELS AND ANNOTATION.
C          AN ADDITIONAL SPACE OF ONE-HALF INCH SHOULD BE ALLOWED FOR
C          THIS PURPOSE BELOW THE X-AXIS, TO THE LEFT OF THE Y-AXIS, AND
C          ABOVE THE GRAPH.  I.E. THE TOTAL SPACE REQUIRED IS XSIZE+.5
C          BY YSIZE+1.
C   X0, Y0, XSIZE, AND YSIZE SHOULD BE CHOSEN TO ALLOW FOR THE LENGTH
C          OF THE PLOTTED VECTORS.  I.E. IF THE MAXIMUM LENGTH OF THE
C          PLOTTED VECTORS IS Z INCHES, THEN THE DISTANCE BETWEEN X0
C          AND MIN(X(I)) ALONG THE X-AXIS SHOULD BE .GE. Z, AND THE
C          DISTANCE BETWEEN (X0+DX*XSIZE) AND MAX(X(I)) SHOULD BE
C          .GE. Z.  LIKEWISE FOR THE Y-AXIS.
C
      DIMENSION X(1),Y(1),R(1),THETA(1),ITITLE(1),LABELX(1),LABELY(1)
      DIMENSION MSG(9),XX(5),YY(5)
      DATA DIV/10.0/
      DATA MAXCRS,VMIN,XARR,YARR,LESIGN/100,.06,-.03,.02,6055B/
C
C   COUNT THE CHARACTERS IN LABELX, DRAW X-AXIS WITH ANNOTATION BELOW.
      KOUNT= -NCHARS(LABELX,MAXCRS)
      CALL AXIS(0.,0.,LABELX,KOUNT,XSIZE,0.0,X0,DX,DIV)
C   YAXIS WITH ANNOTATION TO THE LEFT.
      KOUNT=NCHARS(LABELY,MAXCRS)
      CALL AXIS(0.,0.,LABELY,KOUNT,YSIZE,90.0,Y0,DY,DIV)
C   WRITE TITLE ABOVE GRAPH, CENTERED ALONG LENGTH OF GRAPH.
      CALL TITLE(0.,YSIZE+.2,XSIZE,.21,0.,ITITLE)
C   COMPUTE SCALE FACTOR FOR VECTOR MAGNITUDES.
      DVDR=VREF/RREF
C   COMPUTE MAGNITUDE CORRESPONDING TO MINIMUM VECTOR SIZE.
      RMIN=VMIN/DVDR
```

```
C  PREPARE MESSAGE ABOUT VECTOR SIZES.  LESIGN CONTAINS THE CODE
C  FOR A LESS-THAN-OR-EQUAL SIGN, FOLLOWED BY THE CODE FOR A SPACE.
      ENCODE(80,20,MSG(1))VREF,RREF,LESIGN,RMIN,VMIN
   20 FORMAT(*VECTOR *,F4.2,* INCHES LONG = *,E9.2,*.  VALUES *,R2,E9.2,
     1  * ARE*,F4.2,* INCHES LONG.*)
C  WRITE MESSAGE UNDER TITLE, CENTERED ALONG LENGTH OF GRAPH.
      MSG(9)=0
      CALL TITLE(0.,YSIZE+.05,XSIZE,.1,0.,MSG)
C  COMPUTE LOCATION OF X=0 ALONG X-AXIS.  IF X=0 IS ON GRAPH, DRAW A
C  DASHED LINE TO INDICATE ITS POSITION.
      XZERO= -X0/DX
      IF((XZERO.GE..01).AND.(XZERO.LE.(XSIZE-.01)))CALL DASHLIN(XZERO,
     1  0.,XZERO,YSIZE,.08)
C  DO THE SAME FOR Y=0 ALONG Y-AXIS,
      YZERO= -Y0/DY
      IF((YZERO.GE..01).AND.(YZERO.LE.(YSIZE-.01)))CALL DASHLIN(0.,
     1  YZERO,XSIZE,YZERO,.08)
C
C  THIS IS THE ALGORITHM FOR PLOTTING THE VECTOR FIELD.
      DO100I=1,N
      SINTH=SIN(THETA(I))
      COSTH=COS(THETA(I))
C  COMPUTE POSITION OF POINT ON PLOT.  THIS WILL BE TAIL OF VECTOR.
      XX(1)=(X(I)-X0)/DX
      YY(1)=(Y(I)-Y0)/DY
C  COMPUTE LENGTH OF VECTOR ON PLOT.  SET TO VMIN IF .LT. VMIN.
      RT=R(I)*DVDR
      IF(RT.LT.VMIN)RT=VMIN
C  COMPUTE POSITION OF HEAD OF VECTOR.  THIS IS BOTH SECOND AND LAST
C  POINT USED IN DRAWING ARROW.
      XX(2)=XX(1)+RT*COSTH
      XX(5)=XX(2)
      YY(2)=YY(1)+RT*SINTH
      YY(5)=YY(2)
C  COMPUTE POSITION OF ARROWHEAD POINTS.  THESE ARE 3RD AND 4TH IN
C  DRAWING SEQUENCE.
      YSTH=YARR*SINTH
      XSTH=XARR*SINTH
      XCTH=XARR*COSTH
      YCTH=YARR*COSTH
      XX(3)=XX(2)+XCTH-YSTH
      XX(4)=XX(2)+XCTH+YSTH
      YY(3)=YY(2)+YCTH+XSTH
      YY(4)=YY(2)-YCTH+XSTH
C  SET INDICES FOR DRAWING ARROW FROM TAIL TO HEAD.
      K=0
      L=1
C  GET CURRENT PEN POSITION.
      CALL WHERE(XNOW,YNOW,STEPS)
C  DETERMINE WHETHER MOVING TO HEAD OR TAIL WILL TAKE FEWER PLOTTER
C  COMMANDS.  SINCE EACH COMMAND CAN CAUSE MOVEMENT IN X AND Y
C  DIRECTIONS SIMULTANEOUSLY, WE NEED TO KNOW THE MAXIMUM OF THE X AND
C  Y MOVEMENTS REQUIRED FOR EACH POINT.
      D1=AMAX1(ABS(XNOW-XX(1)),ABS(YNOW-YY(1)))
      D2=AMAX1(ABS(XNOW-XX(2)),ABS(YNOW-YY(2)))
      IF(D2.GT.D1)GO TO 200
```

```
C  IF TAIL IS CLOSER DRAW ARROW FROM TAIL TO HEAD.  IF HEAD IS CLOSER,
C  CHANGE K AND L TO DRAW ARROW FROM HEAD TO TAIL.
      K=6
      L= -1
C  LIFT PEN BEFORE MOVING TO ARROW.  MOVE TO ARROW.  THEN LOWER PEN AND
C  DRAW ARROW IN FOUR LINE SEGMENTS.  IPEN=3 LIFTS THE PEN.
C  IPEN=2 LOWERS THE PEN.
  200 IPEN=3
      DO 300J=1,5
      K=K+L
      CALL PLOT(XX(K),YY(K),IPEN)
      IPEN=2
  300 CONTINUE
  100 CONTINUE
      RETURN
      END
      SUBROUTINE CONTOUR(Z,KDIM,
     1 M,N,MM,NN,XA,XB,YA,YB,XG,YG,NCL,CL,ITITLE,LABELX,LABELY,FX,FY,KP)
      DATA TWOPI/6.28318530717958/
      COMMON/POLARC/RS,R0,THS,TH0
      COMMON/INDICES/MROW,NCOL,MMROW,NNCOL ,KPOL
      COMMON/XYBNDS  /XMIN,XMAX,YMIN,YMAX,XSIZE,YSIZE,
     1HX,HY,XS,XSS,YS,YSS,FXA,FYA
      COMMON/CLEVELS/NLVLS,NLV,CLEVEL(50)
      COMMON/CAVIN/IDIM,DUM(4035)
      DIMENSION Z(1)
      DIMENSION CL(1)
C     Z(I,J) IS THE ORDINATE AT POINT X(J), Y(I)
C     MXN IS THE SIZE OF THE CALCULATED X-Y GRID
C     MMXNN IS THE SIZE OF THE EXPANDED(BY INTERPOLATION) X-Y GRID
C     XA,XB,YA,YB ARE THE MINIMUM AND MAXIMUM VALUES
C          OF X AND Y.
C     XG IS THE WIDTH OF THE GRAPH  IN INCHES.
C     YG IS THE HEIGHT OF THE GRAPH IN INCHES.
C     NCL IS THE NO. OF CONTOUR LEVELS
C     CL(I) ARE THE CONTOUR LEVELS
C     ITITLE CONTAINS THE PLOT TITLES IN 80 BCD CHARACTERS,
C          PLOT NAME IS FIRST 4 WORDS,
C          X-AXIS LABEL IS NEXT 3 WORDS,
C          Y-AXIS LABEL IS NEXT 3 WORDS.
C     THE X(I) ARE ASSUMED TO BE EQUALLY SPACED, AND
C     LIKEWISE, THE Y(I).
C     FX IS THE FUNCTION TO BE PLOTTED ALONG THE X-AXIS.
C     FY IS THE FUNCTION TO BE PLOTTED ALONG THE Y-AXIS.
      IDIM=KDIM
      MROW=M
                NCOL=N
                         MMROW=MM
                                     NNCOL=NN
      KPOL=0
      IF(KP.NE.0)GO TO 50
      XMIN=XA
                 XMAX=XB
                           YMIN=YA
                                      YMAX=YB
      GO TO 55
```

```
50 XMIN=-YB
   XMAX=YB
   YMIN=XMIN
   YMAX=XMAX
   TH0=XA
   R0=YA
   THS=(XB-XA)/FLOAT(NNCOL-1)
   RS=(YB-YA)/FLOAT(MMROW-1)
   KPOL=1
55 CONTINUE
   XSIZE=XG
               YSIZE=YG
                              NLVLS=XABSF(NCL)
   CALLPLOT(0,-.5*(11.-YSIZE),3)
   IF(NCL)1,9,9
 1 CLEVEL=HX=Z
                   L=0
   DO15I=1,NCOL
                    DO7J=1,MROW
                                 L=L+1
   IF(Z(L).LT.CLEVEL)4,5
 4 CLEVEL=Z(L)
 5 IF(Z(L).GT.HX)6,7
 6 HX=Z(L)
 7 CONTINUE
15 L=L-M+IDIM
                    HX=(HX-CLEVEL)/FLOATF(NLVLS-1)
   DO8I=2,NLVLS
 8 CLEVEL(I)=CLEVEL(I-1)+HX
                              GOTO11
 9 DO10I=1,NLVLS
10 CLEVEL (I)=CL(I)
11 HX=(XMAX-XMIN)/FLOATF(NCOL-1)
   HY=(YMAX-YMIN)/FLOATF(MROW-1)
   XS=(XMAX-XMIN)/FLOATF(NNCOL-1)
   YS=(YMAX-YMIN)/FLOATF(MMROW-1)
   FXA=FX(XMIN)
                    FYA=FY(YMIN)
   XSS=XG/(FX(XMAX)-FXA)
   YSS=YG/(FY(YMAX)-FYA)
 2 CALLINTERP(Z,IDIM)
   DO3NLV=1,NLVLS
 3 CALL SCAN(Z,FX,FY)
                       CALLLABEL(ITITLE,LABELX,LABELY,FX,FY)
   IF(KPOL.EQ.0)GOTO 200
   XCENT=XSIZE*.5
   YCENT=YSIZE*.5
   RZERO=XCENT*R0/YB
   RMAX=XCENT
   THETA1=XB
   IF(THETA1.GT.TWOPI)THETA1=TWOPI+AMOD(THETA1,TWOPI)
   IF(RZERO.GT..1)CALL ARC(XCENT,YCENT,RZERO,TH0,THETA1,1)
   CALL ARC(XCENT,YCENT,RMAX,TH0,THETA1,1)
   THH=TH0
   K=2
   IF(ABS(THETA1-(TWOPI+TH0)).LT..001)K=1
```

```
      DO100I=1,K
      SINTH=SIN(THH)
      COSTH=COS(THH)
      X0=XCENT+RZERO*COSTH
      Y0=YCENT+RZERO*SINTH
      X1=XCENT+RMAX*COSTH
      Y1=YCENT+RMAX*SINTH
      CALL DASHLIN(X0,Y0,X1,Y1,.1)
      THH=THETA1
  100 CONTINUE
      CALL DASHLIN(XCENT,YCENT,0.,YCENT,.25)
      CALL DASHLIN(XCENT,YCENT,XCENT,YSIZE,.25)
      CALL DASHLIN(XCENT,YCENT,XCENT,0.,.25)
      CALL DASHLIN(XCENT,YCENT,XSIZE,YCENT,.25)
  200 CALLPLOT(   INTF(XSIZE+6.),-11,,-3)
      RETURN
      END
      SUBROUTINE LABEL(ITITLE,LABELX,LABELY,FX,FY)
      COMMON/TEMP/Z(101)
      COMMON/XYBNDS/XA,XB,YA,YB,XSIZE,YSIZE,HX,HY,
     1XS,XSS,YS,YSS,FXA,FYA
      COMMON/INDICES/M,N,MM,NN,KPOL
      COMMON/CLEVELS/NCL,NLV,CL(50)
      DIMENSION ITITLE(1),LABELX(1),LABELY(1)
      DATA IS/18/
      J=0
            X=XA
                   P=0
                         G=XSIZE-.9
      DO10I=1,NN
                   XG=XSS*(FX(X)-FXA)
      IF(XG.GE.G)3,4
    3 X=XB
             XG=XSIZE
                        GOTO5
    4 IF(XG.GE.P)5,10
    5 J=J+1
              Z(J)=XG
      CALL SYMBOL(XG,-.07,.12,IS,0.,-1)
    7 CALLNUMBER(XG-.14,-.22,.07,X,0,5HE10.3)
    8 IF(X.EQ.XB)11,9
    9 P=XG+.9
   10 X=X+XS
   11 CALL TITLE(0.,-.4,XSIZE,.14,0.,LABELX)
      CALL TITLE(0.,YSIZE+.15,XSIZE,.21,0.,ITITLE)
      DO12I=1,J
   12 CALL SYMBOL(Z(I),YSIZE+.07,.12,IS,180.,-1)
      J=0
            Y=YA
                   P=0
                         G=YSIZE-.2
      DO20I=1,MM
                   YG=YSS*(FY(Y)-FYA)
      IF(YG.GE.G)13,14
   13 Y=YB
             YG=YSIZE
```

```
                    GOTO15
14 IF(YG.GE.P)15,20
15 J=J+1
            Z(J)=YG
   CALL SYMBOL(-.07,YG,.12,IS,270.,-1)
   CALLNUMBER(-.62,YG+.04,.07,Y,0,5HE10.3)
   IF(Y.EQ.YB)21,19
19 P=YG+.9
20 Y=Y+YS
21 CALL TITLE(-.7,0.,YSIZE,.14,90.,LABELY)
   DO22I=1,J
22 CALL SYMBOL(XSIZE+.07,Z(I),.12,IS,90.,-1)
   YI=.5*YSIZE+.1*FLOATF(NCL)
                                  DY=.2
   CALLNUMBER(XSIZE+.5,YI,.14,NCL,0,2HI2)
   CALLSYMBOL(XSIZE+.80,YI,.14,14HCONTOUR LEVELS,0,14)
   YI=YI-DY
23 DO24I=1,NCL
   CALLSYMBOL(XSIZE+.8,YI+.05,0.10,I-1          ,0,-1)
    CALL NUMBER(XSIZE+1.0,YI,0.105,CL    (I),0,5HE15.5)
24 YI=YI-DY
   RETURN
   END
   SUBROUTINE INTERP(AM,IDIM)
   DIMENSION AM(IDIM,1)
   COMMON/TEMP/Z(101)
   COMMON/XYBNDS/XA,XB,YA,YB,XG,YG,HX,HY,
  1XS,XSS,YS,YSS,FXA,FYA
   COMMON/INDICES/M,N,MM,NN   ,KPOL
   ZFUN(V)=A0+A1*V+A2*V**2
   N1=N-1
            M1=M-1
   IF(N-NN)16,15,14
16 DO6I=1,M
   DO1J=1,N
 1 Z(J)=AM(I,J)
                  XY=XA
                       K=1
                           T=HX+XA
   DO3J=2,N1
                CALLFIT(J,T,HX,A0,A1,A2)
 2 AM(I,K)=ZFUN(XY)
                    XY=XY+XS
                          K=K+1
                              IF(XY-T)2,2,3
 3 T=T+HX
 4 IF(K-NN)5,5,6
 5 AM(I,K)=ZFUN(XY)
                    K=K+1
                        XY=XY+XS
                               GOTO4
 6 CONTINUE
15 IF(M-MM)17,13,14
17 DO12I=1,NN
   DO7J=1,M
 7 Z(J)=AM(J,I)
```

```
                   K=1
                          XY=YA
                                    T=HY+YA
      DO9J=2,M1
                   CALLFIT(J,T,HY,A0,A1,A2)
    8 AM(K,I)=ZFUN(XY)
                        XY=XY+YS
                                K=K+1
                                              IF(XY-T)8,8,9
    9 T=T+HY
   10 IF(K-MM)11,11,12
   11 AM(K,I)=ZFUN(XY)
                        K=K+1
                              XY=XY+YS
                                          GOTO10
   12 CONTINUE
   13 RETURN
   14 STOP12
      END
      SUBROUTINE SCAN(AM,FX,FY)
C AM IS THE MATRIX TO BE CONTOURED. MT AND NT ARE ITS X AND Y DIMENSIONS.
C CL(NLV) IS THE CONTOUR LEVEL.
C THE N  (X,Y)  VALUES OF ONE CONTOUR LINE ARE PLOTTED WHEN
C  THEY ARE AVAILABLE.
      DIMENSION AM(1)
      COMMON           /CLEVELS/NCL,NLV,CL(50)
      COMMON/INDICES/DUM(2),MT,NT,KPOL
      COMMON/CAVIN/DIM, IX,IY,IDX,IDY,ISS,
     1 NP,N,CV,IS,IS0,IX0,IY0,DCP,
     2    INX(8),INY(8),REC(800),X(1603),Y(1603)
      TYPE INTEGER REC ,DIM
      DATA(INY=                   0,1,1,1,0,-1,-1,-1)
      DATA(INX=-1,-1,0,1,1,1,0,-1)
      NP=ISS=0
      CV=CL(NLV)
      MT1=MT-1
                NT1=NT-1
      DO 110 I=1,MT1
      IF(AM(I)  -CV)55,110,110
   55 IF(AM(I+1)  -CV)110,57,57
   57 IX0=IX=I+1
                   IY0=IY=IS0=IS=1
                                   IDX=-1
                                          IDY=0
      CALL TRACE(AM,FX,FY)
  110 CONTINUE
      J=MT-DIM
                DO20I=1,NT1
                           J=J+DIM
      IF(AM(J)-CV)15,20,20
   15 IF(AM(J+DIM)-CV)20,17,17
   17 IX0=IX=MT
                 IY0=IY=I+1
                           IDX=0
                                 IDY=-1
                                        IS0=IS=7
```

```
      CALL TRACE(AM,FX,FY)
   20 CONTINUE
      J=MT+NT1*DIM+1
                        DO30I=1,MT1
                                       J=J-1
      IF(AM(J)-CV)25,30,30
   25 IF(AM(J-1)-CV)30,27,27
   27 IX0=IX=MT-I
                     IY0=IY=NT
                                  IDX=1
                                          IDY=0
                                                  IS0=IS=5
      CALL TRACE(AM,FX,FY)
   30 CONTINUE
      J=NT+DIM+1
                     DO40I=1,NT1
                                  J=J-DIM
      IF(AM(J)-CV)35,40,40
   35 IF(AM(J-DIM)-CV)40,37,37
   37 IX0=IX=1
                  IY0=IY=NT-I
                               IDX=0
                                       IDY=1
                                               IS0=IS=3
      CALL TRACE(AM,FX,FY)
   40 CONTINUE
      ISS=1
            L=0
      DO13J=2,NT1
                     L=L+DIM
      DO10I=1,MT1
                     L=L+1
      IF(AM(L)-CV)5,10,10
    5 IF(AM(L+1)-CV)10,7,7
    7 K=L+1
      DO 9 ID = 1,NP
      IF(REC(ID)-K)9,10,9
    9 CONTINUE
      IX0=IX=I+1
                     IY0=IY=J
                               IDX=-1
                                       IDY=0
                                               IS0=IS=1
      CALL TRACE(AM,FX,FY)
   10 CONTINUE
   13 L=L-MT1
      RETURN
      END
      SUBROUTINE TRACE(AM,FX,FY)
      DIMENSION AM(1)
      COMMON/POLARC/RS,R0,THS,TH0
      COMMON          /INDICES/DUM(2),MT,NT,KPOL
      COMMON/XYBNDS/XA,XB,YA,YB,XSIZE,YSIZE,HX,HY,
     1XS,XSS,YS,YSS,FXA,FYA
      COMMON/CAVIN/DIM, IX,IY,IDX,IDY,ISS,
     1 NP,N,CV,IS,IS0,IX0,IY0,DCP,
```

```
  2    INX(8),INY(8),REC(800),X(1603),Y(1603)
   COMMON/CLEVELS/NCL,NLV,CL(50)
   TYPE INTEGER REC,DIM
   N=0
        JY=DIM*(IY-1)+IX
                                MY=DIM*IDY+IDX+JY
 2 N=N+1
           IF(N-1600)3,3,32
 3 IF(IDX)5,4,6
 4 X(N)=FLOATF(IY-1)+FLOATF(IDY)*(AM(JY)-CV)/(AM(JY)-AM(DIM*IDY+JY))
   Y(N)=FLOATF(IX-1)
                           GOTO7
 5 NP=NP+1
            REC(NP)=JY
 6 Y(N)=FLOATF(IX-1)+FLOATF(IDX)*(AM(JY)-CV)/(AM(JY)-AM(JY+IDX))
   X(N)=FLOATF(IY-1)
 7 IS=IS+1
 8 IF(IS-8)10,10,9
 9 IS=IS-8
10 IDX=INX(IS)
               IDY=INY(IS)
   IX2=IX+IDX
              IY2=IY+IDY
                               IR=IDX*IDY
11 IF(ISS)13,15
13 IF(IS.NE.IS0.OR.IY.NE.IY0.OR.IX.NE.IX0)16,14
14 N=N+1
          X(N)=X
                 Y(N)=Y
                               GOTO73
15 IF(IX2)151,73,151
151 IF(IY2)152,73,152
152 IF((IX2.LE.MT).AND.(IY2.LE.NT))16,73
16 MY=DIM*IDY+IDX+JY
                        IF(IR)19,17,20
17 IF(CV-AM(MY))18,18,2
18 IX=IX2
           IY=IY2
81 IS=IS+5
            JY=MY
                 GOTO8
19 KY=JY+IDX
              LY=MY-IDX
                          GOTO21
20 KY=MY-IDY
              LY=JY+IDY
21 DCP=(AM(JY)+AM(KY)+AM(LY)+AM(MY))*.25
                                          IF(CV-DCP)23,23,22
22 CALL GETPT(AM(JY))
   GO TO 7
23 IF(IR)24,25,25
24 IX=IX2
           IDX=-IDX
   CALL GETPT(AM(KY))
   IY=IY2
           IDY=-IDY
```

```
                                  GOTO26
 25 IY=IY2
               IDY=-IDY
    CALL GETPT(AM(KY))
    IX=IX2
               IDX=-IDX
 26 IF(CV-AM(MY))81,81,28
 28 CALL GETPT(AM(MY))
                         IF(IR)29,30,30
 29 IX=IX+IDX
                  IDX=-IDX
                                 GOTO31
 30 IY=IY+IDY
                  IDY=-IDY
 31 IF(CV-AM(LY))33,33,34
 33 IS=IS-1
               JY=LY
                             GOTO10
 34 CALL GETPT(AM(LY))
                         IF(IR)35,36,36
 35 IY=IY+IDY
                  GOTO7
 36 IX=IX+IDX
                  GOTO7
 32 PRINT103,CV
 73 IF(KPOL.EQ.0)GO TO 173
    DO174I=1,N
    THETA=X(I)*THS+THO
    R=Y(I)*RS+RO
    X(I)=XSS*(R*COS(THETA)-FXA)
    Y(I)=YSS*(R*SIN(THETA)-FYA)
174 CONTINUE
    GO TO 175
173 DO74I=1,N
    X(I)=XSS*(FX(X(I)*XS+XA)-FXA)
 74 Y(I)=YSS*(FY(Y(I)*YS+YA)-FYA)
175 CONTINUE
    CALLSYMBOL(X,Y,.07,NLV-1              ,0,-1)
    DO75I=1,N
 75 CALLPLOT(X(I),Y(I),2)
    RETURN
103 FORMAT(1H0,23HA CONTOUR LINE AT LEVEL,E13.5,
   1 41H WAS TERMINATED BECAUSE IT CONTAINED MORE,
   2 23H THAN 1600 PLOT POINTS.)
    END
    SUBROUTINE GETPT(AM)
    COMMON/CAVIN/DIM, IX,IY,IDX,IDY,ISS,
   1 NP,N,CV,IS,IS0,IX0,IY0,DCP,
   2    INX(8),INY(8),REC(800),X(1603),Y(1603)
    N=N+1
             B=AM    -DCP
                                IF(B)2,1
  1 V=.5
            GOTO3
  2 V=.5*(AM    -CV)/B
  3 Y(N)=FLOATF(IX-1)+FLOATF(IDX)*V
```

```
                                        X(N)=FLOATF(IY-1)+FLOATF(IDY)*V
      RETURN
      END
      SUBROUTINE FIT(I,X,H,C,B,A)
      COMMON/TEMP/Z(101)
      W=0.5*(Z(I+1)-Z(I-1))/H
      A=0.5*(Z(I+1)+Z(I-1)-Z(I)-Z(I))/H**2
      C=Z(I)+X*(X*A-W)
                        B=W-2.*X*A
      RETURN
      END
      SUBROUTINE TITLE(X,Y,SIZE,HEIGHT,ANGLE,ITEXT)
      DIMENSION ITEXT(1)
      DATA SIX7,TWO7/.857142857,.285714286/
      CHSIZE=HEIGHT*SIX7
      MAXCHS=SIZE/CHSIZE
      NUMCHS=NCHARS(ITEXT,MAXCHS)
      START=.5*(SIZE-CHSIZE*FLOAT(NUMCHS)+TWO7*HEIGHT)
      TH=ANGLE*.0174533
      CALL SYMBOL(X+START*COS(TH),Y+START*SIN(TH),HEIGHT,ITEXT,ANGLE,
     1   NUMCHS)
      RETURN
      END
      FUNCTION NCHARS(ITEXT,MAXCHS)
      DIMENSION ITEXT(1)
      MAXWDS=MAXCHS/10+1
      DO1I=1,MAXWDS
      K=ITEXT(I).AND.7777B
      IF(K.EQ.0)GO TO 2
    1 CONTINUE
      I=MAXWDS
    2 NUMCHS=10*I
      DO3J=1,I
      L=I-J+1
      KTEST=ITEXT(L)
      DO3M=1,10
      K=KTEST.AND.77B
      IF((K.NE.0).AND.(K.NE.55B))GO TO 4
      KTEST=ISHIFT(KTEST,54)
      NUMCHS=NUMCHS-1
    3 CONTINUE
    4 NUMCHS=MIN0(NUMCHS,MAXCHS)
      NCHARS=NUMCHS
      RETURN
      END
      SUBROUTINE DASHLIN(X0,Y0,X1,Y1,DASH,
C
C  DRAWS A DASHED LINE FROM (X0,Y0) TO(X1,Y1).  THE DASHES AND SPACES
C  BETWEEN THEM ARE APPROXIMATELY OF LENGTH -DASH-.  THIS LENGTH IS
C  ADJUSTED SUCH THAT THE LINE IS COMPOSED OF EQUAL LENGTH DASHES,
C  AND BEGINS AND ENDS WITH A DASH.
C  ALL PARAMETERS ARE IN FLOATING POINT INCHES.
C
      XNOW=X0
      YNOW=Y0
      DX=X1-X0
```

```
      DY=Y1-Y0
      D=SQRT(DY*DY+DX*DX)
      NDASH=2*(IFIX(D/DASH)/2)+1
      DX=DX/FLOAT(NDASH)
      DY=DY/FLOAT(NDASH)
      CALL WHERE(XN,YN,STEPS)
      D1=AMAX1(ABS(XN-X0),ABS(YN-Y0))
      D2=AMAX1(ABS(XN-X1),ABS(YN-Y1))
      IF(D2.GT.D1)GO TO 2
      XNOW=X1
      YNOW=Y1
      DX=-DX
      DY=-DY
    2 IPEN=3
      CALL PLOT(XNOW,YNOW,IPEN)
      DO1I=1,NDASH
      XNOW=XNOW+DX
      YNOW=YNOW+DY
      IPEN=5-IPEN
      CALL PLOT(XNOW,YNOW,IPEN)
    1 CONTINUE
      RETURN
      END
      SUBROUTINE ARC(X0,Y0,R,TH0,TH1,IDASH)
C
C  DRAWS AN ARC OF RADIUS R ABOUT (X0,Y0) FROM THETA=TH0 TO
C  THETA=TH1, TH0.LT.TH1.  IF IDASH.EQ.0, THE ARC WILL BE SOLID.
C  IF IDASH.NE.0, THE ARC WILL BE DASHED.
C  X0, Y0, AND R ARE IN FLOATING POINT INCHES.
C  TH0 AND TH1 ARE IN RADIANS.
C
      DELTH=2.*ASIN(.05/R)
      THETA =TH0
      X=X0+R*COS(THETA)
      Y=Y0+R*SIN(THETA)
      X1=X0+R*COS(TH1)
      Y1=Y0+R*SIN(TH1)
      IPEN=3
    1 CALL PLOT(X,Y,+PEN)
      THETA=THETA+DELTH
      IPEN=5-IPEN
      IF(IDASH.NE.0)GO TO 2
      IPEN=2
    2 X=X0+R*COS(THETA)
      Y=Y0+R*SIN(THETA)
      IF(THETA.LT.TH1)GO TO 1
      CALL PLOT(X1,Y1,IPEN)
      RETURN
      END
          IDENT   SEEK
          ENTRY   SEEK
+         VFD     42/0LSEEK,18/4
SEEK      BSS    1               .ENTRY LINE
          SB7    1
          SB6    -B7
          SA1    B1              .X1=E
```

```
         SA2    B2           .X2*ET(1)
         SX0    B7           .SET X0*1
         SA3    B3           .X3*N
         FX4    X2-X1        .ET(1)-E
         SX7    X3           .X7*N
         ZR     X4,S200      .IF X4*0, GOTOS200
         SX6    X0           .X6*1=ITOP
         SA2    X3+B2        .X2*ET(IBOT)
         PL     X4,S300      .IF X4=+, GOTO S300
         FX5    X2-X1        .X5*ET(IBOT)-E
         ZR     X5,S500      .IF X5*0, GOTO S500
         FX3    X7+X0        .IBOT=N+1
         IX4    X3-X6        .X4*IBOT-ITOP
         PL     X5,S603      .IF X5=+, GOTOS603
S400     SB7    B0           .SET I=0 FOR ERROR
S200     SX7    B7           .SET I=1
S500     SA7    B4           .     OR I*N
         EQ     B0,B0,SEEK   .RETURN
S603     IX7    X4-X0
         FX5    X3+X6        .IBOT+ITOP
         AX2    B7,X5        .DIVIDE BY 2
         SB5    X2+B6        .I*I-1
         SA4    B5+B2        .X4*ET(I)
         ZR     X7,S800      .IF X7=0, GOTO S800
         FX5    X4-X1        .X5=ET(I)-E
         BX7    X3           .STORE IBOT
         LX3    X2           .X3=I=IBOT
         IX4    X2-X6        .X4=IBOT-ITOP
         PL     X5,S603
         BX3    X7           .RESTORE X3=IBOT
         SX6    X2           .X6*ITOP=I
         IX4    X3-X2        .X4*IBOT-ITOP
         EQ     B0,B0,S603
S800     SA6    B4
         EQ     B0,B0,SEEK
S300     FX4    X2-X1        .X4=ET(IBOT)-E
         ZR     X4,S500
         PL     X4,S400
         FX3    X7+X0        .IBOT=N+1
         IX4    X3-X6        .X4*IBOT-ITOP
S703     IX7    X4-X0
         FX5    X3+X6        .IBOT+ITOP
         AX2    B7,X5        .DIVIDE BY 2
         SB5    X2+B6        .I*I-1
         SA4    B5+B2        .X4*ET(I)
         ZR     X7,S800      .IF X7=0, GOTO S800
         FX5    X1-X4        .X5*E-ET(I)
         BX7    X3           .STORE IBOT
         LX3    X2           .X3*I=IBOT
         IX4    X2-X6        .X4*IBOT-ITOP
         PL     X5,S703
         BX3    X7           .RESTORE X3=IBOT
         SX6    X2           .X6=ITOP=I
         IX4    X3-X2        .X4*IBOT-ITOP
         EQ     B0,B0,S703
         END
```