

THC FILE COPY

7

NPS72-86-004CR

AD-A180 280

NAVAL POSTGRADUATE SCHOOL

Monterey, California



DISC
ELECTRONIC
MAY 11 1987
A

CONTRACTOR REPORT

IMPULSIVE LOADING FROM A BARE
EXPLOSIVE CHARGE IN SPACE

by

Joseph Falcovitz

December 1986

Approved for public release; distribution unlimited.

Prepared for: Strategic Defense Initiative Office
The Pentagon
Washington, DC 20301-7100

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED			1b. RESTRICTIVE MARKINGS NONE		
2a. SECURITY CLASSIFICATION AUTHORITY			3. DISTRIBUTION / AVAILABILITY OF REPORT Approved for Public Release; Distribution Unlimited		
2b. DECLASSIFICATION / DOWNGRADING SCHEDULE					
4. PERFORMING ORGANIZATION REPORT NUMBER(S) NPS72-86-004CR			5. MONITORING ORGANIZATION REPORT NUMBER(S) NPS72-86-004CR		
6a. NAME OF PERFORMING ORGANIZATION JOSEPH FALCOVITZ		6b. OFFICE SYMBOL (If applicable) 72		7a. NAME OF MONITORING ORGANIZATION NAVAL POSTGRADUATE SCHOOL, CODE 72	
6c. ADDRESS (City, State, and ZIP Code) Research Contractor Naval Postgraduate School, Code 72 Monterey, CA 93943-5100			7b. ADDRESS (City, State, and ZIP Code) Space Systems Academic Group Monterey, CA 93943-5100		
8a. NAME OF FUNDING / SPONSORING ORGANIZATION Strategic Defense Initiative Office		8b. OFFICE SYMBOL (If applicable) SDIO/DEO		9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER MI PR-DGAA60045	
8c. ADDRESS (City, State, and ZIP Code) SDIO/DEO Washington, DC 20301-7100			10. SOURCE OF FUNDING NUMBERS		
			PROGRAM ELEMENT NO. PE63221	PROJECT NO.	TASK NO.
11. TITLE (Include Security Classification) Impulsive Loading From A Bare Explosive Charge in Space					
12. PERSONAL AUTHOR(S) JOSEPH FALCOVITZ					
13a. TYPE OF REPORT Contractor Report		13b. TIME COVERED FROM Jan 86 TO June 86		14. DATE OF REPORT (Year, Month, Day) December 1986	
15. PAGE COUNT 65					
16. SUPPLEMENTARY NOTATION					
17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number) Space Blast, Impulsive Loading, Explosive in Space		
FIELD	GROUP	SUB-GROUP			
19. ABSTRACT (Continue on reverse if necessary and identify by block number) Consider a planform target subjected to a normal impact of explosive products generated by detonating a bare charge in space. It is suggested that the loading impulse may be approximated by the total momentum of that portion of the fluid which impacts at the target. Assuming impulsive dynamic response, and assuming that the ensuing damage is proportional to the kinetic energy imparted to the structure by the blast, we get a particularly simple law: $\text{Damage} \sim W^2/R^4$ (W is charge mass, R is range). This model is an idealization of a solar panel (or antenna) extended in a paddle-like fashion from a relatively rigid and massive core structure. It is also shown that this law implies that no advantage can be realized by re-arranging the mass of a single bare charge in a cluster configuration of smaller sub-charges, which would be dispersed and detonated via an idealized "isotropic" scheme.					
20. DISTRIBUTION / AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS			21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED		
22a. NAME OF RESPONSIBLE INDIVIDUAL ALLEN E. FUHS, Distinguished Professor			22b. TELEPHONE (Include Area Code) (408) 646-2948		22c. OFFICE SYMBOL

ABSTRACT

This document

Similar to

to the 2d power

to the 4th power

Consider a platform target subjected to a normal impact of explosive products generated by detonating a bare charge in space. It is suggested that the loading impulse may be approximated by the total momentum of that portion of the fluid which impacts at the target. Assuming impulsive dynamic response, and assuming that the ensuing damage is proportional to the kinetic energy imparted to the structure by the blast, we get a particularly simple law: $\text{Damage} \sim W^2/R^4$ (W is charge mass, R is range). This model is an idealization of a solar panel (or antenna) extended in a paddle-like fashion from a relatively rigid and massive core structure. It is also shown that this law implies that no advantage can be realized by re-arranging the mass of a single bare charge in a cluster configuration of smaller sub-charges, which would be dispersed and detonated via an idealized isotropic scheme.

Keywords: gas dynamics, exhaust plumes, Sprengstoff.

ACKNOWLEDGEMENTS

This work is part of a study involving gas dynamics of exhaust plumes from spacecrafts. It was conducted under the cognizance of Distinguished Professor Allen E. Fuhs, who suggested extending our understanding of gasdynamics in space to the treatment of blast effects on spacecrafts. I wish to thank Professor Fuhs for his creative guidance and deeply appreciate his continuous support. The GRP code used for the blast computation is a product of mutual research conducted by Professor M. Ben-Artzi and myself. The fruitful collaboration of Professor Ben-Artzi is gratefully acknowledged.

Distribution Statement A is correct for this report. The material in this report is taken from open literature.
Per Professor Allen E. Fuhs, NPS/Code 67Fu

Distribution Statement	
CLASSIFICATION	☒
GROUP TAG	☐
UNCLASSIFIED	☐
Distribution Statement	
By _____	
Distribution/	
Availability Codes	
Distribution/	
Date _____	



TABLE OF CONTENTS

1.	INTRODUCTION.....	1
2.	IMPACT BLAST LOADING	3
3.	TARGET DYNAMIC RESPONSE.....	10
4.	CLUSTER CONFIGURATION.....	13
5.	DISCUSSION AND CONCLUSIONS	15
6.	REFERENCES	16
	APPENDIX A. The GRP Code.....	17
	A.1 Array Variables.....	18
	A.2 Major Parameters.....	20
	A.3 Labeled COMMON variables.....	21
	A.4 Description of Subroutines.....	24
	A.5 Listing of GRP Code.....	29
	APPENDIX B. Code for Re-Normalizing the Air Impulse.....	56
7.	DISTRIBUTION LIST	57

LIST OF FIGURES

Figure 2-1	Impact Blast Loading	7
Figure 2-2	Shock Reflection at Impact Phase.....	8
Figure 2-3	Limiting Cases of Shock Reflection (a) Initially Reflected Shock (Impact) (b) Stationary Shock	8
Figure 2-4	Impulse of Normally Reflected Blast Wave at Sea-Level and in Space	9
Figure 3-1	Cantilever Beam with Plastic Hinge.....	12
Figure 3-2	ChargeMass - Range - Damage Curves for Cantilever Beam.....	12
Figure 4-1	Target Intercept at Closest Approach.....	14
Figure 4-2	Spherical Cap Surrounding the Target.....	14
Figure A-1	Piecewise Linear Distribution of Flow Variables in Cells.....	53
Figure A-2	Intersection of Right and Left Adiabats for Solving Riemann Problem	54
Figure A-3	Wave Diagram Representing Solution to Riemann Problem.....	55

NOMENCLATURE (consistent units in m, kg, ms system)

C	Coefficient in ChargeMass-Range-Damage relationship ($\text{m kg}^{-1/2}$)
D_{CJ}	Speed of propagation of detonation wave at CJ point (m ms^{-1})
I	Impulse per unit area of target ($\text{kg m}^{-1} \text{ms}^{-1}$)
I	Dimensionless impulse $I = I(R) [4\pi R_0^2/W(2Q_0)^{1/2}]$
h	Beam thickness (m)
L	Length of cantilever beam (m)
m	Lagrange mass coordinate (kg)
M_p	Moment per unit length of plastic hinge (MPa m^2)
N	Number of sub-charges in a cluster configuration
P	Pressure (MPa)
P_s	Surface pressure (MPa)
Q₀	Explosive energy per unit mass (MJ kg^{-1})
R₀	Radius of spherical charge (m)
R	Range from center of charge (m)
S	Speed of propagation of shock wave (m ms^{-1})
t	Time (ms)
U	Flow velocity (m ms^{-1})
V	Velocity imparted to target by loading impulse (m ms^{-1})
W	Charge mass (kg)
Y	Plastic yield stress (MPa)
Z	Total momentum of an explosive charge (kg m ms^{-1})
α	Coefficient for dynamic pressure recovery
γ	Specific-heat ratio
γ_{CJ}	Specific-heat ratio of explosive products at CJ point
θ	Plastic rotation angle of cantilever beam
κ	Impact approximation impulse coefficient (presently $\kappa = 1$)
μ	Beam mass per unit area (kg m^{-2})
ρ	Fluid density (kg m^{-3})
ρ_p	Beam density (kg m^{-3})
φ	Mid-area angle of sub-charge spherical cap

EMPTY PAGE

1. INTRODUCTION

The advent of space-based weapon systems in our times has raised the prospects of future "Star Wars" conflicts, rendering the potential use of explosive devices against space targets a present day engineering reality. The warhead of choice in space seems to be of the fragmentation type, for obvious reasons. The effectiveness of fragments is unhampered by the space environment (lack of air may even be helpful). By contrast, bare charges in space are considerably less efficient than in air. One may wonder why this is so since in air, as in space, the same amount of chemical energy is released through the detonation process. The explanation is that the difference is in the much larger mass involved in the air blast, relative to the bare charge mass.

For a more comprehensive explanation, we take a close look at the process by which an explosive-driven air blast wave is generated. The explosive products effectively constitute a rapidly expanding spherical piston (typical initial speed around 6 km/sec), which drives an intense shock wave into the surrounding air. At a typical range of $100R_0$ (and with air density equal to about 1/1000 of charge density), the mass of air entrained by the shock is about 1000 times the charge mass. Thus, the highly concentrated initial explosive energy, has spread over a much larger mass than that of the charge, via the mechanism of wave propagation in compressible media, resulting in an increased momentum. For a comprehensive treatment of blast waves in air the reader is referred to Baker[1].

It is also worthwhile noting that explosive products in space typically attain hypersonic speed prior to impacting at the target. The flow velocity in an air blast is typically subsonic or somewhat supersonic. It is thus expected that the actual gasdynamic interaction between the blast flow and a stationary target, will be fundamentally different in these two cases.

We contend that blast effects in space may still be of practical interest for reasons such as the following :

- (i) Notwithstanding the poor efficiency of a bare charge, its use should not be ruled out altogether. Fragments would contribute to existing - and potentially hazardous - population of space debris, underlining the obvious fact that there is no absolutely safe standoff distance from an isotropic fragmentation warhead. A clean bare charge may thus be a reasonable alternative.
- (ii) Even a fragmentation warhead has some residual blast capacity, which has to be considered either as a factor in enhancing target damage, or as a threat to be reckoned with in determining a safe standoff distance.

The key idea of the present model is a combination of the assumption that target dynamic response is related primarily to total blast impulse, and the physically plausible notion that this impulse is equal to the total momentum of that portion of the expanding explosive products which impacts at the target. The sense in which this simple notion constitutes an approximation to a proper gasdynamic analysis of the interaction between the fluid and the target, is clarified in Ch. 2. In that chapter we also present an illuminating comparison between impulsive blast loading in air and in space.

In order to demonstrate the ChargeMass-Range-Damage relationship implied by our impact blast approximation, we chose a simple target model: A cantilever beam with a rigid-perfectly plastic stress-strain relationship. It represents an extended structural element such as a solar panel or an antenna. We make use of studies conducted by Mentel [2] and by Bodner and Symonds [3], which showed that by and large, the effect of accelerating the beam impulsively was to cause a rotation about a plastic hinge at the point of support. The final angle of rotation is generally proportional to the initial kinetic energy, so that equating damage with that angle, results in damage being proportional to the square of the impulse imparted to the target by the blast loading. A presentation of this dynamic response model, including a sample case, is given in Ch. 3.

Our ChargeMass-Range-Damage relationship may imply some far-reaching conclusions when applied to the analysis of a more general configuration than the single-charge/single-target case. In Ch. 4 we present a simple analysis of a sub-munition configuration of N bare charges, concluding that it seems to have no advantage in efficiency, relative to a single charge of equal mass. Sections 5 and 6 contain conclusions and references, correspondingly.

We conclude the introduction by listing the main assumptions made in the present study :

- (a) Blast loading and target response are uncoupled. This is true since typically the target mass is much larger than the mass of that portion of the explosive products which impacts on it.
- (b) Dynamic target response is independent of specific loading time history. It depends solely on total (time-integrated) impulse.
- (c) The target is a panel extended as a relatively supple cantilever. It is supported by a relatively rigid and massive core structure.
- (d) The charge is a sphere detonated at its center. The expansion is spherically symmetric.
- (e) Target surface is normal to local flow vector.
- (f) Target orbital velocity relative to the center of the charge is negligible, compared with the velocity of the expanding products.

2. IMPACT BLAST LOADING

Consider the expanding explosive products impacting at a target as shown in Fig. 2-1. By regarding the fluid as an ensemble of non-interacting particles moving at velocity $U(R,t)$, and by assuming a no-rebound normal impact at the surface, the pressure time history is given by :

$$P_s(t) = \rho(R,t)[U(R,t)]^2 \quad (2-1)$$

How is this simple impact mechanism related to the actual gasdynamic interaction between the expanding explosive products and the target? When a target is located at a range of at least several charge radii, two features in the free stream of the oncoming fluid are significant : The flow is highly hypersonic (Mach number 20 or higher), and the static pressure is very small, which means that $P + \rho U^2 \approx \rho U^2$. These facts were born out by a numerical computation which we performed for a typical high explosive characterized by the following parameters :

$$\rho_0 = 1800 \text{ (kg m}^{-3}\text{)}$$

$$\gamma_{CJ} = 3$$

$$D_{CJ} = 8 \text{ (m ms}^{-1}\text{)}$$

$$Q_0 = D_{CJ}^2/[2(\gamma_{CJ}^2 - 1)] = 4 \text{ (MJ kg}^{-1}\text{)}$$

(2-2)

Where Q_0 was determined by assuming that the detonation corresponded to the CJ point on the explosive Hugoniot curve, and that the detonation products were an ideal gas with a specific-heat ratio γ_{CJ} . The spherically expanding flow was computed by integrating the Euler equations for isentropic flow via a high-resolution conservative finite-difference scheme [4-6]. The initial conditions were the self-similar flow field of a just-detonated spherical charge given by Taylor [7]. The code GRP with which the computation was performed is described and listed in Appendix A.

Consider the flow at a stationary target, which begins at the moment of arrival of the expanding explosive products (Fig. 2-2). A qualitative description of the ensuing flow pattern is made by observing its evolution in time. Immediately following the initial (normal) impact, the fluid is stopped at the target by a backward-propagating shock wave reflected from the surface. Since the target is of

finite extent, the fluid between the shock and the surface is accelerated laterally, and streamlines that tend to curve around the target are being formed. If the oncoming flow were stationary, the flow field would evolve toward the familiar configuration of a detached bow-shock positioned at a relatively narrow standoff distance from the surface.

Let us find the post-shock pressure in these two limiting phases. In the initial phase, the fluid is stopped at the target by a reflected shock (Fig. 2-3a), and in the pseudo-stationary phase (Fig. 2-3b), the shock is stationary. In either case we find the post-shock pressure to be given by a pressure-recovery expression of the form :

$$P_2 = \alpha \rho U^2 \quad (2-3)$$

Where α is a constant related to the appropriate γ (assuming the expanded explosive products are an ideal gas). The governing equations in the reflected shock case are :

$$\rho(U+S) = \rho_2 S$$

$$\rho(U+S)^2 = P_2 \quad (2-4)$$

$$\rho(\gamma+1)/(\gamma-1) = \rho_2 \quad (\text{strong shock})$$

Where the unknowns are ρ_2 , P_2 , S .

The equations for the stationary shock case are :

$$\rho U = \rho_2 U_2$$

$$\rho U^2 = P_2 + \rho_2 U_2^2 \quad (2-5)$$

$$\rho(\gamma+1)/(\gamma-1) = \rho_2 \quad (\text{strong shock})$$

Where the unknowns are ρ_2 , U_2 , P_2 . Thus, solving for α in the two cases represented by equations (2-4) and (2-5), we get :

$$\text{Reflected shock} \quad \alpha = [(\gamma + 1)/2]^2 \quad (2-6)$$

$$\text{Stationary shock} \quad \alpha = 2/(\gamma + 1)$$

In either case, since the gas is not dense, the effective range of γ is somewhere between 1.0 and 1.4, so that setting $\alpha = 1$ is an approximation commensurate with the overall crudeness of the present impact blast model. Since the flow in the layer between the shock and the target is low subsonic (at least it is so away from target edges), the post-shock pressure is a reasonable substitute for the surface pressure. Also, $\alpha = 1$ is an appropriate approximation where the flow is so rarefied that it is collisionless. In this limit, $\alpha = 1$ corresponds to full thermal accommodation of re-emitted molecules from a presumably cold surface.

The foregoing analysis constitutes a justification of the impact approximation to the surface pressure (2-1). Now we turn to the task of evaluating the impulse which is defined as the time-integrated surface pressure. Using the impact approximation (2-1), the impulse is given by :

$$I(R) = \int_0^{\infty} P_s(t) dt = \int_0^{\infty} p(R,t) [U(R,t)]^2 dt \quad (2-7)$$

Let us introduce a Lagrange mass coordinate m which enables a transformation from the Euler system (R,t) to the Lagrange system (m,t) . The differential relation associated with this transformation at constant R is :

$$dm = 4\pi R^2 \rho(R,t) U(R,t) dt \quad (2-8)$$

Since it is assumed that the fluid is not accelerated at any (R,t) in the range of interest for blast loading, the velocity $U(R,t)$ can be regarded as function *solely of the mass coordinate*, so that $U(R,t) = U(m)$. Using (2-8) we are then able to cast the impact blast expression (2-7) in the following simple and physically appealing form :

$$I(R) = Z/4\pi R^2$$

$$Z = \int_0^W U(m) dm \quad (2-9)$$

The upper limit W in (2-9), which is consistent with the upper limit ∞ in (2-7), implies that the total impulse is somewhat overestimated, since it contains contributions from the innermost layers of the explosive products that will arrive at the target as $t \rightarrow \infty$.

The total momentum Z is thus a constant which can be evaluated for any specific explosive charge by numerical integration. We performed this computation with the code GRP described in Appendix A. In doing so for the typical explosive (2-2), we found out that the impulse (2-9) was a reasonable approximation at ranges as low as $R = 3R_0$. Furthermore, it was found that Z could be approximated by the maximum attainable momentum for the given charge mass and energy $W(2Q_0)^{1/2}$, to within about 6%. Apparently, the total momentum is not overly sensitive to the exact velocity distribution function $U(m)$, so that assuming a value of Z appropriate to the uniform distribution $U(m) = (2Q_0)^{1/2}$ is a reasonable approximation. Thus we finally arrive at the following closed-form approximation for the blast impulse:

$$I(R) = \kappa W(2Q_0)^{1/2} / 4\pi R^2 \quad (2-10)$$

$\kappa = 1$

Where the coefficient κ is retained in order to suggest that its value be determined more accurately from detailed experimental or computational data, in the event that such data become available. At present our best estimate is $\kappa = 1$.

There is one comparison, however, which can readily be made with available data. We refer to impulsive blast loading in air, such as given by Baker (Ref. 1, Fig. 6.3 in the supplement). The comparison is conveniently made with a non-dimensional form of (2-10), which is rewritten as:

$$\bar{I} = I(R) [4\pi R_0^2 / W(2Q_0)^{1/2}] = (R/R_0)^{-2} \quad (2-11)$$

The air blast data has to be converted to the same normalization scheme as in Eq. (2-11), before the comparison can be made. Considering the definition of $\bar{I}$ in (2-11) above, and the definition of scaled range and air blast impulse (Table 6.2 of Ref. 1), this conversion is done by multiplying the scaled air impulse and range by the following coefficients (sea-level air is assumed):

Impulse Multiplier	$\beta = 3(2\gamma)^{-1/2}(4\pi/3)^{1/3} (P_a/\rho_a Q_0)^{1/6} (\rho_a/\rho_0)^{1/2} = .01204$	
Range Multiplier	$\delta = (4\pi/3)^{1/3} (\rho_0 Q_0/P_a)^{1/3} = 67.06$	(2-12)
$\rho_a = 1.3 \text{ (kg m}^{-3}\text{)}$	$P_a = 0.1 \text{ (MPa)}$	$\gamma = 1.4$

The air blast conversion was done by a small code which is given in Appendix B. The air and space blast impulses are shown in Fig. 2-4. We note that at ranges larger than about 10 charge radii, the air blast impulse is higher than the space impulse, and the gap widens as the range increases. This observation is consistent with the qualitative explanation given in the introduction, which attributed this effect to the increase in the entrained air mass at higher range. At ranges lower than 10 charge radii, the air mass is relatively insignificant, so that one may expect the blast impulses in air and in space to be comparable. Indeed, the inverse-square variation of impulse with range is apparent for the air blast at low range. In absolute values, however, the low-range space impulse is higher by a factor of about 1.7. This might be interpreted as indicating that choosing $\kappa = 1/1.7$ would be the appropriate "calibration". However, we do not propose to do so, since we are not able to trace the various factors affecting the low-range impulse as given by Baker [1]; they may somehow depend on the presence of air, as well as on other parameters such as target size and equation of state of the explosion products.

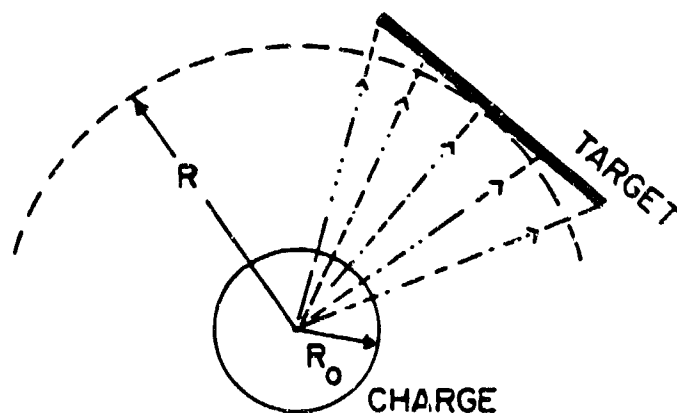


Figure 2-1. Impact Blast Loading

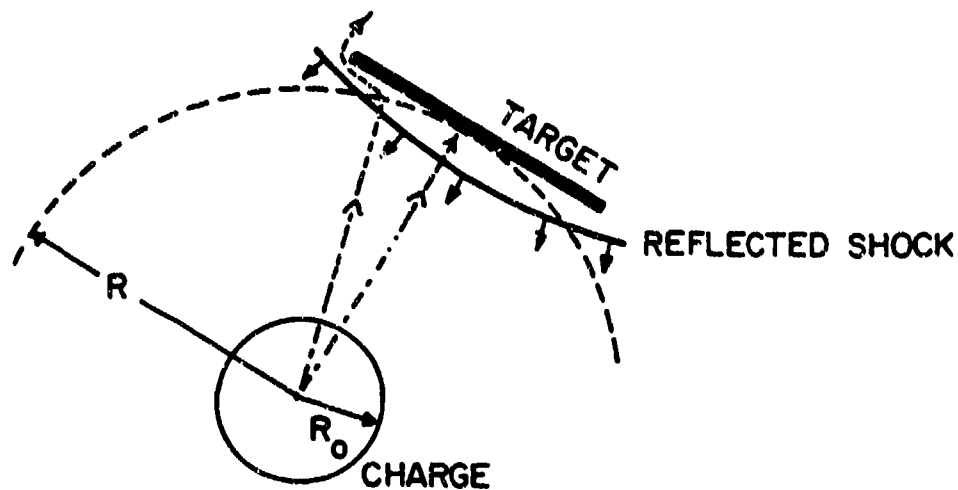


Figure 2-2. Shock Reflection at Impact Phase

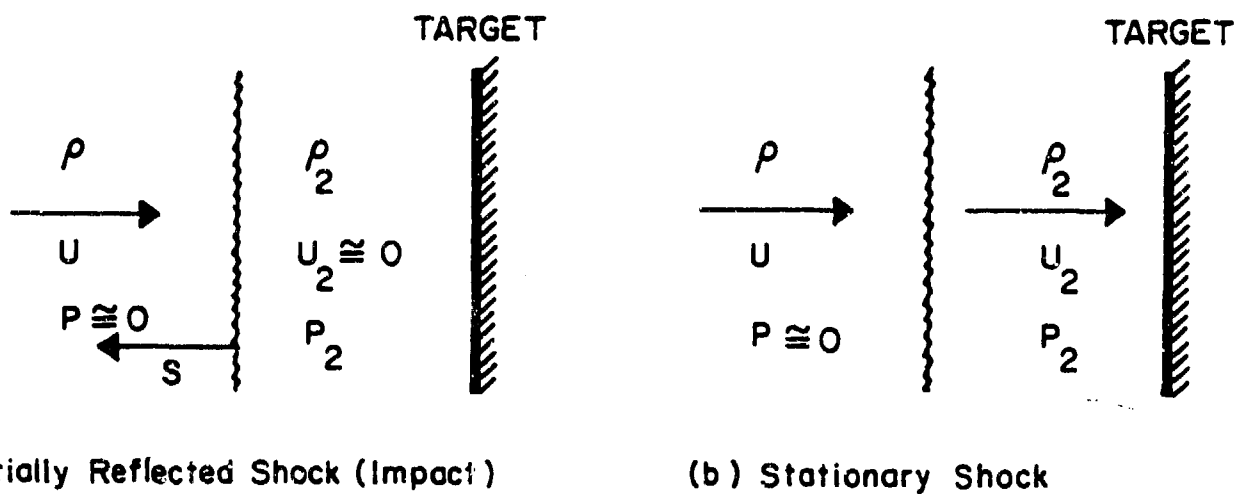


Figure 2-3. Limiting Cases of Shock Reflection

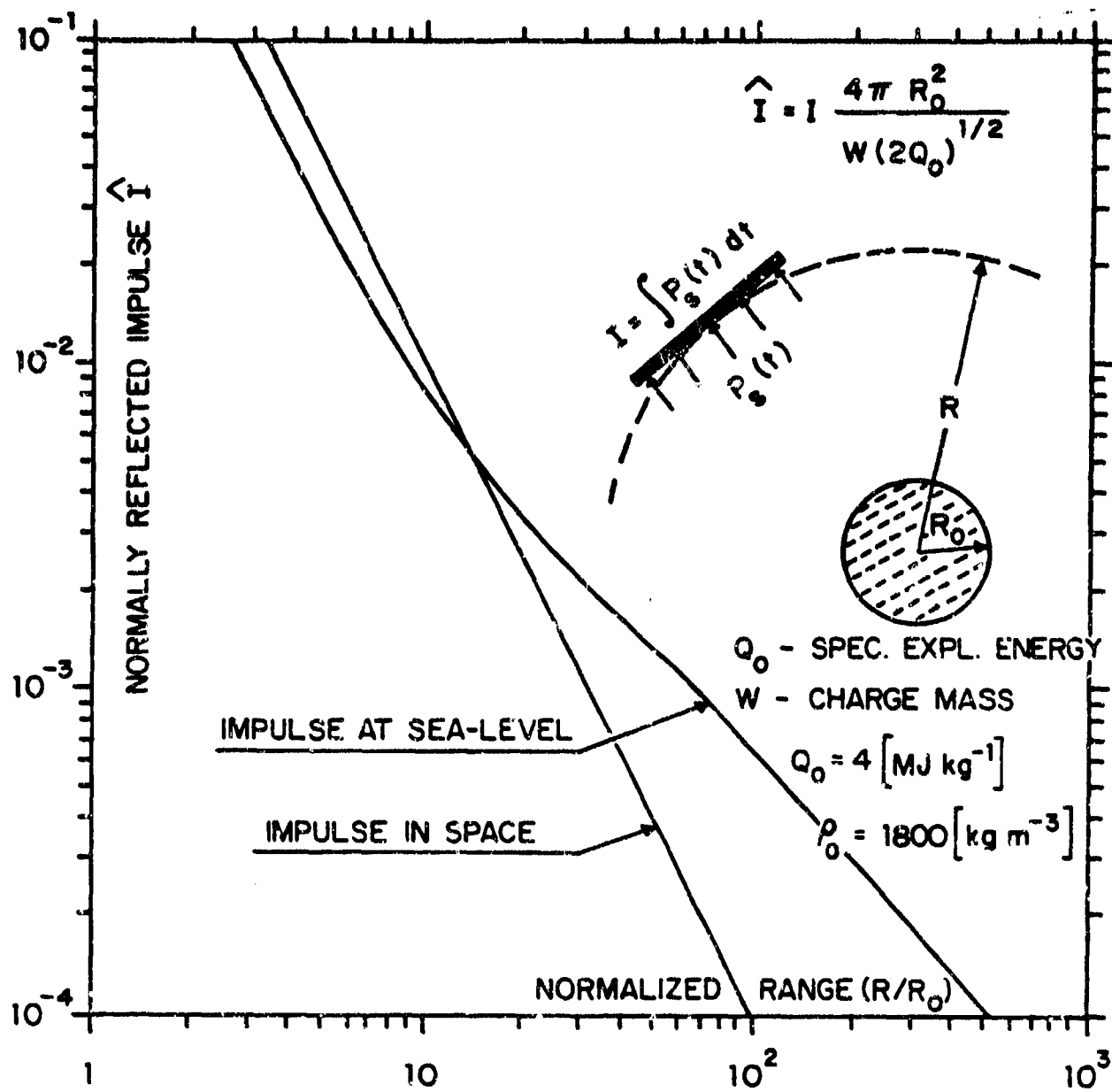


Figure 2-4. Impulse of Normally Reflected Blast Wave at Sea-Level and in Space

3. TARGET DYNAMIC RESPONSE

For the sake of constructing representative Charge-Mass-Range-Damage relations from our impact approximation to the blast impulse (2-10), we suggest a simple idealized structure as target model. It is a cantilever beam made of a metal characterized by a rigid-perfectly plastic stress-strain relation.

This model is supposed to represent an extended spacecraft component such as a solar panel or an antenna. The core structure is assumed to be much more massive and rigid than the extended structural element, so that the cantilever can be idealized as being rigidly supported. The sole dynamic and structural parameters are hence those of the cantilever.

For this purpose we make use of an experimental and theoretical investigation of uniform cantilever beams subjected to impulsive loading that was conducted by Mentel [2]. Aluminum alloy beams were held in a massive support that was gliding along a rail at speed V , until it was abruptly stopped by a very massive anvil. After the system came to rest, the beams were observed to have rotated through an angle θ about the point of support, with little deformation elsewhere (Fig. 3-1).

The theoretical model suggested by Mentel [2] for predicting $\theta(V)$, can be described as comprising two stages. Immediately following the impact, the beam commences rotating rigidly about the support point, with an angular momentum equal to the pre-collision moment of momentum about that point. This application of the principle of conservation of moment of momentum entails an abrupt re-distribution of velocity in the beam, with velocity being proportional to distance from support, and the tip moving at $1.5 V$. The angle θ is subsequently determined from the requirement that the rotational kinetic energy be dissipated as plastic hinge work $M_p \theta$. The resulting $\theta(V)$ expression is :

$$\theta = (3/8)\mu LV^2/M_p \quad (3-1)$$

We now make one more step in formulating the model, in that we postulate that *the angle θ is a measure of damage*. Using the following expressions for M_p , μ and V :

$$M_p = (1/4)Yh^2$$

$$\mu = \rho_p h \quad (3-2)$$

$$V = I(R)/\mu$$

We get from (2-10) and (3-1) the following ChargeMass-Range-Damage (W-R- θ) relationship :

$$\begin{aligned} R &= CW^{1/2} \\ C &= [(3/16\pi^2\theta) (LQ_0/\rho_p Yh^3)]^{1/4} \end{aligned} \tag{3-3}$$

We note that the effective range for a specified target and "damage level" θ , is proportional to the square root of the charge mass W .

Using the data for the typical explosive (2-2), and the following data for a specific aluminum beam, we get for this sample case :

$$\begin{aligned} h &= 0.002 \text{ (m)} \\ L &= 1.0 \text{ (m)} \\ \rho_p &= 2700 \text{ (kg m}^{-3}\text{)} \\ Y &= 300 \text{ (MPa)} \\ C &= 1.85 \theta^{-1/4} \text{ (m kg}^{-1/2}\text{)} \end{aligned} \tag{3-4}$$

The ChargeMass-Range-Damage relationship corresponding to this sample case is depicted in Fig. 3-2.

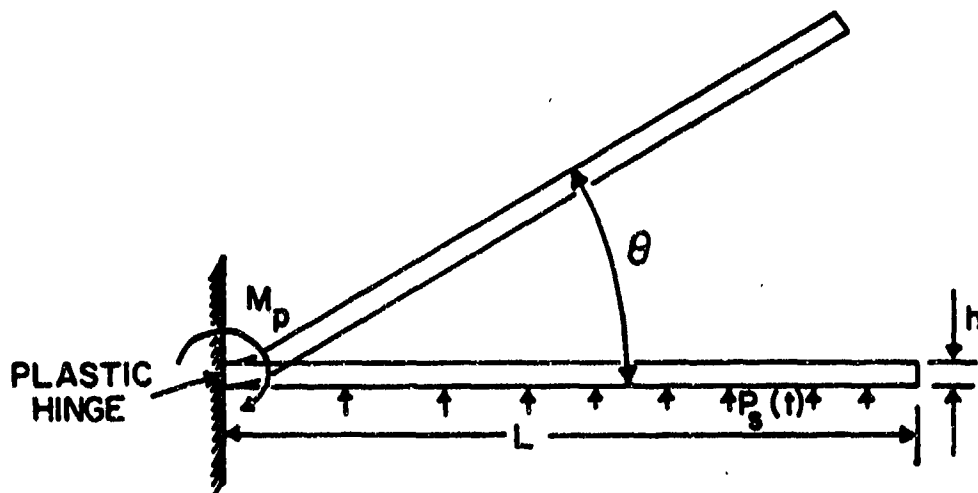


Figure 3-1. Cantilever Beam with Plastic Hinge

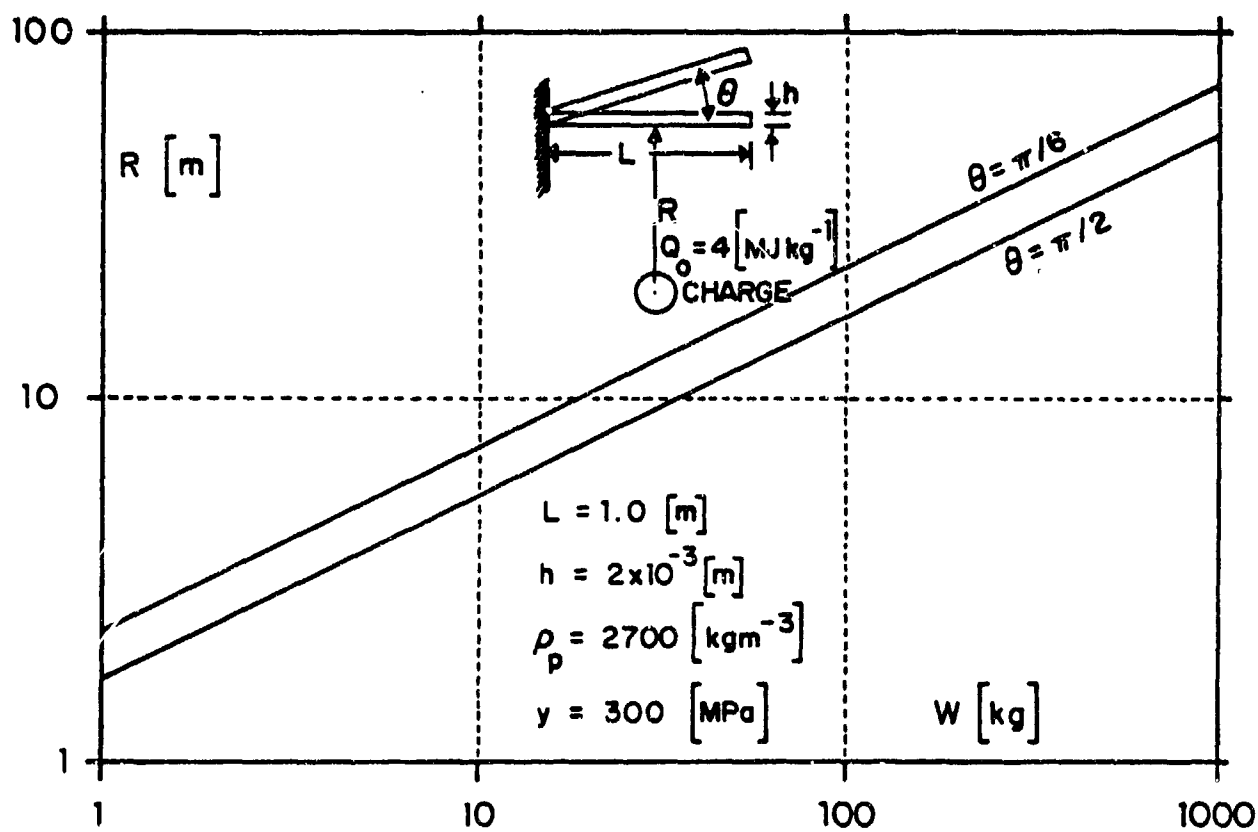


Figure 3-2. Charge Mass - Range - Damage Curves for Cantilever Beam

4. CLUSTER CONFIGURATION

In a cluster configuration, the gain in damage is presumably a result of a favorable design tradeoff between reduced charge mass and reduced range. Can such a gain be achieved for a space system, assuming the ChargeMass-Range-Damage law (3-3) to hold? It can be shown that by adopting some simple strategy of sub-munition dispersion and initiation, equation (3-3) implies no gain in target damage.

Let us assume for the sake of a reasonably simple analysis, that dispersion and initiation of sub-charges would take place according to the following scheme :

- (a) The N sub-charges appear to fan out from a common virtual center, moving at equal speeds. At subsequent times, their centers are uniformly distributed over an expanding spherical envelop.
- (b) The target moves at a constant velocity relative to the virtual center. Its point of closest approach to that center is at range R .
- (c) The timing for dispersion is chosen so that the target intersects (tangentially) with the spherical envelop at the point of closest approach (Fig. 4-1). This is also the point at which the blast from a single-charge configuration detonated at the virtual center, would have impacted at the target.
- (d) All sub-munitions are detonated at this "moment of closest approach".
- (e) It is assumed that each spherical cap of area $4\pi R^2/N$ will contain one, and only one, sub-charge. The probability of the charge location on that cap is assumed to be uniformly distributed. The expected location on the cap is hence that latitude line ϕ which divides the cap into two parts of equal area (Fig. 4-2).
- (f) It is assumed that the target is subjected to the blast of a single sub-charge, which is located on the mid-area latitude ϕ of the spherical cap that surrounds the target (Fig. 4-2).

Since the area of the spherical cap subtended by ϕ is $4\pi R^2/(2N)$, the angle ϕ is given by :

$$\sin(\phi/2) = (2N)^{-1/2} \quad (4-1)$$

We seek a comparison between the deflection θ for a single charge (W, R) , and the deflection θ_N in the sub-munition case ($W_N = W/N$, $R_N = 2R\sin(\phi/2)$). From the ChargeMass-Range-Damage law (3-3), using also Eq. (4-1), we get :

$$(\theta_N/\theta) = (W_N/W)^2 (R/R_N)^4 = 1/4$$

(4-2)

Consequently, there is no potential gain in a tradeoff between charge mass and range, for a cluster configuration with the aforementioned dispersion scheme. The factor $1/4$, along with the mass overhead inherent in constructing a multi-charge configuration, indicate that in causing blast damage, a single charge is more effective than an equal-mass isotropically dispersed cluster.

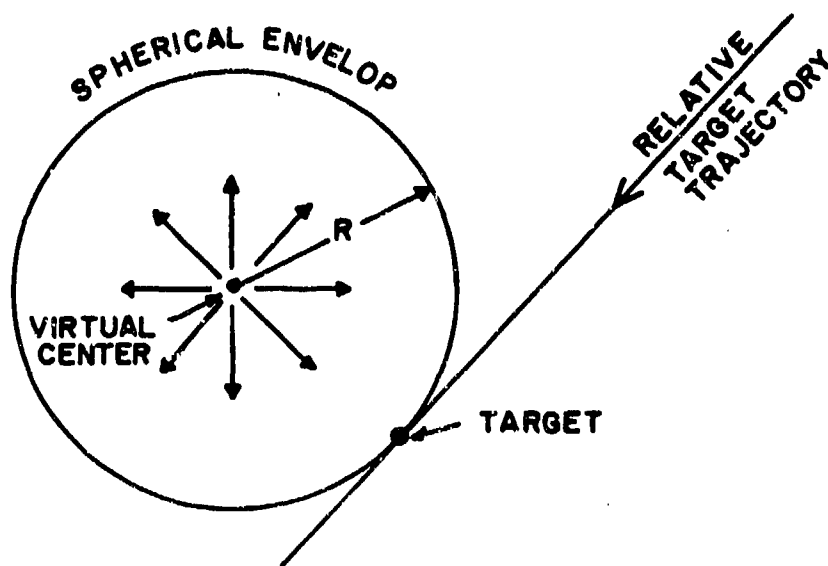


Figure 4-1. Target Intercept at Closest Approach

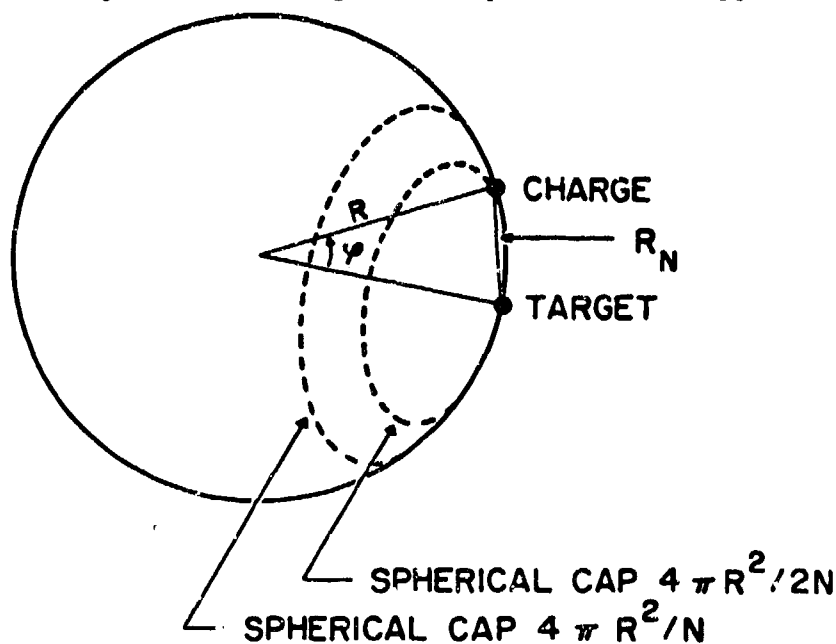


Figure 4-2. Spherical Cap Surrounding the Target

5. DISCUSSION AND CONCLUSIONS

Our analysis pertains to a bare explosive charge initiated at a point of closest approach to the target. We have shown that the loading impulse on a planform target is given by the impact approximation (2-7), which states that the impulse is proportional to the charge mass and inversely proportional to the range squared. The impulse in space has been compared with impulse in air at sea-level. It was found that the two are quite comparable at close range (10 charge radii or less), exhibiting identical variation with range. At far ranges, the impulse in air is the higher one. This is consistent with the notion that spreading the explosive energy over larger air mass results in larger momentum (and hence reflected impulse). We then proceeded to develop the Charge-Mass-Range-Damage law (3-3) for an impulse-responsive target, which states that blast damage is proportional to the square of the charge mass and inversely proportional to the fourth power of the range. These results were obtained by introducing extensive simplifications in the analysis of gasdynamic interaction, and in the analysis of dynamic target response. We have further shown that this damage law also implies that no gain can be achieved by an idealized cluster configuration of bare sub-charges, relative to a single charge of equal total mass.

It is worthwhile noting that all assumptions introduced in the course of formulating the impact blast approximation and the structural dynamic response to impulsive loading, imply that target damage is overestimated. The only exception is the approximation in setting $\alpha = 1$, which can be readily rectified by assigning to α the reflected shock value given in (2-6). Furthermore, we assumed that the pressure at the midpoint of the target, is the pressure everywhere on the target. Due to flow around the edges, the average pressure is lower than the midpoint pressure. Also, targets are not everywhere normal to the flow (and charge/target attitude is not a design parameter). Oblique impact obviously entails reduced target loading. In the area of structural dynamic response, a time-distributed loading function generally delivers less kinetic energy to the structure than an impulsive loading of equal total impulse, resulting in reduced deformation (damage). Thus, while the present model may be regarded as an over estimate when applied to a sure-fail analysis, it is particularly suitable in determining a sure-safe range.

6. REFERENCES

- [1] Baker, W. E., *Explosions in Air*, University of Texas Press, Austin and London, 1973.
- [2] Mentel, T. J., "The Plastic Deformation Due to Impact of a Cantilever Beam with an Attached Tip Mass", *Journal of Applied Mechanics*, pp. 515-524, 1958.
- [3] Bodner, S. R. and Symonds, P. S., "Experimental and Theoretical Investigation of the Plastic Deformation of Cantilever Beams Subjected to Impulsive Loading", *Journal of Applied Mechanics*, pp. 719-728, 1962.
- [4] Ben-Artzi, M., and Falcovitz, J., "A High-Resolution Upwind Scheme for Qausi 1-D Flows", *INRIA Workshop on Numerical Methods for Solving the Euler Equations of Fluid Dynamics*, F. Angrand and R. Glowinski (editors), Paris, Dec. 1983, SIAM Publication, Philadelphia, 1985.
- [5] Ben-Artzi, M., and Falcovitz, J., "A Second-Order Godunov-Type Scheme for Compressible Fluid Dynamics", *Journal of Computational Physics*, Vol 55, pp.1-32, 1984.
- [6] Ben-Artzi, M. and Falcovitz, J., "An Upwind Second-Order Scheme for Compressible Duct Flows", *SIAM Journal on Scientific and Statistical Computing*, Vol 7, p.744-768, 1986.
- [7] Taylor, G. I., "The Dynamics of the Combustion Products Behind Plane and Spherical Detonation Fronts in Explosives", *Proc. Roy. Soc. A*, Vol CC (1950), pp.235-247. Also: *The Scientific Papers of Sir Geoffrey Ingram Taylor*, Vol III, G. K. Batchelor, Editor, Cambridge Press, 1963.

APPENDIX A. The GRP Code

The purpose of this Appendix is to provide a concise description of the GRP code, and a listing of its CHARGE version. It is intended for users that have had prior experience in implementing schemes for solving the Euler equation of compressible flow. The theoretical background of GRP schemes constitutes the principles on which the code is founded. Some familiarity (at least) with this background, as given in References 4, 5 and 6 is indispensable to any implementation of GRP schemes. Reference 4 is recommended as an introduction. The planar GRP scheme is fully described in Reference 5, and the duct-flow GRP scheme on which the present CHARGE version is based is given in Reference 6. (In CHARGE version the flow is spherical and the "duct" area is set to $X(I)^2$, but the code can handle any area variation - see subroutines CROSS and RATIO below).

In GRP schemes, second-order accuracy is achieved by considering a piecewise linear interpolation of the flow in each cell (Fig. A-1), from which second-order accurate fluxes at each cell interface are evaluated through an analysis of a local Generalized Riemann Problem (GRP). Briefly stated, the GRP goes one step further than the Riemann Problem (RP), in that it seeks (analytically) the first time-derivative of the flow that evolves as the "diaphragm" is removed from the cell interface, at the origin of the centered (X,T) wave paths of the RP solution. The major computational subroutines are CYCEUL where the integration of conservation laws is performed, RIEMAN where the local Riemann Problems are solved by Newton-Raphson iterations, MAGA where the closed-form expressions derived from the GRP analysis [6] are used to compute flow time-derivatives along the contact surface, FLUXE where all the previously computed information is used to extrapolate the fluxes to mid-time-step ($T + DT/2$) which constitutes a second-order accurate flux.

The plan of this Appendix is as follows. Array variables, including those which carry conserved variables (mass, momentum and energy), are described in section A.1. This is followed by descriptions of general parameters (A.2), labeled COMMON variables (A.3) and all subroutines (A.4). We conclude by giving the CHARGE version listing (A.5), which should be consulted whenever a reading of this code description is attempted.

NOTE : The present CHARGE version was implemented in a GRP code version that had been converted to treat detonation waves as chemically reactive compressible flow. However, the detonation scheme is effectively neutralized by setting QDET=0 (in NETUNM). All variables pertaining to detonation, such as arrays Z(I), DZ(I), FIMZ(I), ZMDOT(I) and labeled COMMON variables containing Z in their names, should be ignored.

A.1 Array Variables

The code GRP is organized so that all major subroutines are called with standard list of array variables which represent the integration scheme (i.e. the conservation laws), local Riemann Problem solutions and second-order accurate fluxes. Virtually all array variables are initially defined in BEGIN (initial conditions), and are subsequently updated at each time step in CYCEUL. The following list explains the meaning of these variables. Some terms used in the list are defined below.

X(I)	grid point coordinate.
U(I)	velocity in cell I.
P(I)	pressure in cell I (computed from equation of state).
RO(I)	density in cell I. This variable is time-integrated according to the law of conservation of mass. (Computed in CYCEUL).
E(I)	total energy per unit volume (including kinetic energy) in cell I. This variable is time-integrated according to the law of conservation of (total) energy. (Computed in CYCEUL).
DU(I)	velocity difference in cell I.
DP(I)	pressure difference in cell I.
DRO(I)	density difference in cell I.
DG(I)	Lagrange sound velocity difference in cell I.
DXSI(I)	the Lagrange coordinate increment defined as $RO(I)*(X(I+1)-X(I))$, for cell I.
MIN(I)	inactive in present version.
US(I)	velocity at the contact surface obtained after the resolution of the local discontinuity at X(I) (Riemann Problem solution). It is denoted as U^* in References 4-6.
PS(I)	pressure at the contact surface obtained after the resolution of the local discontinuity at X(I) (Riemann Problem solution). It is denoted as P^* in References 4-6.
UIDOT(I)	time derivative of US(I) along the contact surface. (This derivative is the result of the GRP analysis. It is computed in MAGA. See Ref. 5 and 6).
PIDOT(I)	time derivative of PS(I) along the contact surface. (This derivative is the result of the GRP analysis. It is computed in MAGA. See Ref. 5 and 6).
FIMZ(I)	inactive in present version.
ZMDOT(I)	inactive in present version.

TENA(I)	momentum per unit volume $\rho(I)*U(I)$ in cell I. This variable is time-integrated according to the law of conservation of momentum. (Computed in CYCEUL).
FIRO(I)	mass flux at point X(I) (second-order accurate).
FIM(I)	momentum flux at point X(I) (second-order accurate).
FIE(I)	energy flux at point X(I) (second-order accurate).
GIP(I)	the pressure term in the momentum flux. It corresponds to $G(U)$ in References 4 and 6.
VOL(I)	volume of cell I.
Z(I)	inactive in present version.
DZ(I)	inactive in present version.

Glossary of terms used in the array variables list :

Cell I - the cell between grid points X(I) and X(I+1). All cell variables are averages per that interval.

Difference in cell I - the difference between values of variable at cell boundaries X(I+1) and X(I). Those values are obtained from "monotonized" piecewise linear distribution of each variable in each cell. (Fig. A-1).

Second-order accurate flux - the flux time-derivative at point X(I) is computed from the time-derivatives of pressure and velocity along contact surface PIDOT(I) and UIDOT(I) (in FLUXE). Then the the flux is extrapolated to the centered time point $(T+DT/2)$, using those derivatives. This centered value is the second-order flux for integrating the conservation laws between T and T+DT.

A.2 Major Parameters

A list of major parameters indicating their meaning and the routine in which they are defined, is given below. Those parameters defined in NETUNM are the run input. There is no reading of an input file in this version of GRP code (and the only output is the printed output).

L	number of grid points + 1 (main program)
LL	L - 1 (MAIN PROGRAM)
T	time (MAIN0)
DT	time step (MAIN0)
TMAX	maximum time (when T.GE.TMAX the run is terminated) (NETUNM)
TMUD	time for which next printing will take place (NETUNM)
DTMUD	printing time step (NETUNM)
NCYC	serial number of time step (integration cycles) (MAIN0)
COLELA	switch to evaluate cell differences by Colella's method when COLELA.NE.0 (NETUNM)
KEYMON	key for monotonization scheme (just one is presently provided when COLELA.EQ.0) (NETUNM)
NCYCPR	frequency of line printing at each cycle (time step) (NETUNM)
STAB	CFL stability coefficient. Must be smaller than 1. (NETUNM)
DTBA	next time step computed from stability criterion (CYCEUL)
DTKOD	former time step (MAIN0)
KDT	index of cell where DTBA was determined (CYCEUL)

A.3 Labeled COMMON variables

Labeled COMMONs are used primarily to transmit data to and from routines that perform the major computational steps of the GRP scheme, i.e, RIEMAN, MAGA and FLUXE; these routines are called from CYCEUL. When the value of any of those variables is needed for later use, whether for updating conservation variables (RO, TENA, E), or for printing, it is stored in the appropriate array. All labeled COMMON variables are grouped under labels that indicate their role, and their names are also mnemonic. Generally, suffix L means Left and suffix R means Right. It may indicate sides either with respect to a cell interface X(I), or with respect to the contact surface which separates the Right- and Left- propagating waves in a solution to the local Riemann Problem. We indicate by INPUT variables that are computed prior to calling the subroutine, and by OUTPUT variables whose value was computed within the subroutine and constitutes the result of calling that subroutine.

COMMON /STEP0/ Parameters related to the local Riemann Problem. This is the first step in the GRP scheme.

UL, PL, ROL, CL, GL, SL - velocity, pressure, density, sound speed, Lagrange sound speed and entropy, attributed to Left side of cell interface at point X(I). (INPUT)

USTAR, PSTAR - velocity and pressure at the contact surface obtained when the local discontinuity is resolved (i.e., the solution to the local Riemann Problem). The omission of L or R suffix indicates that P and U are continuous across the contact surface. (OUTPUT)

RSTARL, CSTARL, GSTARL - density, sound speed and Lagrange sound speed on the Left side of the contact surface. (OUTPUT)

WL - Lagrange velocity of propagation of the Left-moving shock, relative to the fluid. (OUTPUT)

UW(6) - velocity of propagation of each wave front (Fig. A-3), relative to the inertial system (X). (OUTPUT)

HELEML - logical variable. If HELEMLEQ.TRUE. the Left-propagating wave is a shock. Otherwise it is a (centered) rarefaction wave. (OUTPUT)

NFLUX - integer variable. It denotes the region in the Riemann solution wave structure, which contains the point X(I) for all time. Refer to Fig. A-3 for illustration. (OUTPUT)

LAMDAL, RATEL, TEMPL, TEMPSL, ZL, ZSTARL - inactive.

COMMON /STEP1/ Parameters related to the time-derivative evaluation of the GRP scheme, performed in MAGA. The time-derivatives of P and U along the contact surface are the main result of MAGA.

DUIDT, DPIDT - time-derivatives of velocity and pressure along contact surface. (OUTPUT)

ASTARL - The directional derivative of U along the fan characteristic at the trailing characteristic of the Left rarefaction wave. It is not evaluated when the Left wave is a shock. (See References 4-6) (OUTPUT)

DGIDTL, DRIDTL - time-derivatives of Lagrange sound speed and density along the left side of the contact surface. (OUTPUT)

DSDAL - Lagrange spatial derivative of entropy on the left side of contact surface, prior to removal of the partition at $X(I)$.

SH, RAT - the cross-section area and the x-derivative of $\ln(SH)$. They are user-defined in CROSS and RATIO respectively.

DSDASL - entropy derivative used in the special "sonic" case (i.e, when $NFLUX=2$ or $NFLUX=5$). See References 5,6 for details. (OUTPUT)

LAMDSL, DZDAL, BETACL, DZDASL - inactive.

COMMON /GRADS/ Used to transmit flow gradients (that exist in fluid prior to removal of the partition at $X(I)$) to MAGA.

DUDXIL, DPDXIL, DGDXIL, DRDXIL, DSDXIL - gradients of U, P, G, RO, S (with respect to Lagrange coordinate). They are computed in CYCEUL for transmission to MAGA. (INPUT)

DZDXIL - inactive.

COMMON /FI/ Used to return values of updated flux and cell-interface variables from FLUXE.

FIH1, FIH2, FIH3 - second-order flux of mass, momentum flow (just $RO*U^{**2}$) and energy. They are extrapolated to Half the time step $T + DT/2$. (OUTPUT)

GIH - the value of P at $T + DT/2$

UXN, PXN, GXN, ROXN - values of U, P, G, RO extrapolated to New time $T+DT$, at cell-interface. They are used in CYCEUL to get tentative (pre-monotonized) new cell differences. (OUTPUT)

ZXN, FIH4, ZMDOTL, ZMDOTR - inactive.

A.4 Description of Subroutines

MAIN PROGRAM

The task of this program is to allocate array space for the NMAT arrays required by the present version of GRP code. The length of each array is L. The allocation is done by calling MAIN0. This standard calling sequence is maintained hereafter, thus facilitating modifications.

MAIN0

This subroutine functions as an overall organization routine. It can be read as a kind of flow-chart of the entire computation. First, run set-up is done by calling once to NETUNM (data) and BEGIN (initial conditions). Then a loop over time steps is begun. In each cycle the integration by one time step is performed by calling CYCEUL, and subsequently boundary conditions are implemented by calling SAFAE. Whenever T.EQ.TMUD, results are printed by calling PRINT and TMUD is updated by adding DTMUD.

NETUNM

Here data are set for a particular run. User is invited to modify this routine. There is no input file. This routine is called just once from MAIN0. Note that the detonation data section is skipped when QDET.EQ.0.

BEGIN

Initial conditions are set-up in this routine. The configuration of some nominal case is given in present version. (In CHARGE version it is the detonated spherical charge, using the Taylor self similar solution as initial conditions). User is called to modify this routine so as to generate any other desired initial configuration.

TAYLOR

The purpose of this routine, along with ancillary routines INIDAT, RUNGE and DERIV, is to compute the self-similar Taylor solution [7] of a detonated spherical charge, and implement it as initial conditions for the GRP computation of the ensuing expansion. TAYLOR is called once by BEGIN.

The core of the solution is the numerical (Runge-Kutta) integration of two coupled ordinary differential equations. The integration variable is Ψ . (The flow velocity normalized by DCJ is given

by $U = \exp(-\psi)$). The two dependent variables are X - the normalized radial coordinate ($X = 1$ at the sphere boundary), and C - the normalized speed of sound. The integration is carried out by calling RUNGE, which in turn calls DERIV for the evaluation of derivatives. Data for the TAYLOR computation is set up by calling (just once) INIDAT.

The initial conditions needed in BEGIN are values of mass, momentum and (total) energy per cell. These are most accurately computed by spatially integrating the Taylor solution, resulting in lumped mass, momentum and energy per cell, which are then divided by the cell volume. This refinement is significant since gradients are high near the charge boundary ($X = 1$). A total mass and energy check for the entire sphere is performed and printed.

INIDAT, RUNGE, DERIV

Subroutines used only in conjunction with the Taylor initial conditions setup. See TAYLOR above.

RATIO, CROSS

User-defined routines. If $A(X)$ is the duct cross-section area, then $CROSS(X) = A(X)$ and $RATIO(X) = D[\ln(A(X))]/DX$.

CYCEUL

This is the central computation routine. All major stages of the GRP scheme are performed by calling specific subroutines from CYCEUL. Then $RO(I)$, $TENA(I)$ and $E(I)$ are updated to new time $T + DT$ by solving the appropriate conservation laws in CYCEUL.

The first loop (DO 1) performs a set of preparatory steps as follows :

- (a) CALL RIEMAN - Solving the local Riemann Problem at each $X(I)$.
- (b) CALL MAGA - Solving the local Generalized Riemann Problem at each $X(I)$.
- (c) CALL FLUXE - Computing second-order fluxes at $X(I)$.
- (d) Evaluation of cell-interface finite differences $DU(I)$, $DP(I)$, $DRO(I)$ in each cell. These will be used at the future time step (after monotonicization) for piecewise-linear interpolation of the flow in each cell. (See definition of $DUDXIL$, $DPDXIL$, ..., just preceding the call to MAGA in this loop).

Note that in present CHARGE version additional computation of PRESS, PULSE1, ..., PULSE4 has been added. It is just informative and does not interfere in any way with the execution of the

GRP scheme. The purpose of this computation is to monitor the numerical solution and to observe the accuracy within which the asymptotic value of the momentum integral Z (Eq. 2-9 above) is approached.

In the second loop (DO 2), the integration of the three conservation laws is performed, using second-order fluxes that had been computed in loop 1. Flow variables such as $P(I)$ and $U(I)$ are computed in this loop from the conserved variables. The cycle computation is concluded by calling BDOK1 for monotonicization of $DU(I)$, $DP(I)$ and $DRO(I)$.

SAFAE

In this routine user-defined boundary conditions are implemented. Present version (CHARGE) contains rigid wall at the center of the sphere $X(2)=0$, and an "open boundary" at the outer computational zone limit $X(L)$. The rigid wall condition is achieved by setting up a virtual antisymmetric cell next to the boundary cell, so that the solution to the local Riemann Problem will result in a non-moving contact surface ($USTAR=0$). The open boundary is an approximation to an ideally non-reflecting boundary. Here the virtual cell is $I=L$, and the flow in it is defined as a "continuation" of the flow in the adjacent last cell $I=LL$.

BDOK1

Here the tentative cell-interface differences $DV(I)$ are monotonicized according to neighboring average cell values $V(I-1)$, $V(I)$ and $V(I+1)$. The basic idea is that the cell-interface slope $DV(I)$ should have the same sign as the average slope $V(I+1)-V(I-1)$. When $V(I)$ is a local extremum $DV(I)$ is set to zero. Also, the absolute value of $DV(I)$ is constrained so that the jump from a cell-interface value to the adjacent average value $V(I)$, will never be of opposite sign to $DV(I)$.

DCOLE

When COLELA option is used (not in present CHARGE version), the pre-monotonicized slopes are simply the centered difference $(V(I+1)-V(I-1))/2$. Note that even under this option, the monotonicization routine BDOK1 is subsequently called.

PRINT

Printing of results. Reading this routine is self-explanatory. Note some features added for present CHARGE version. User is called to modify this routine to his specific needs.

SOF

Run termination when an error has been detected. ISTOP is an informative index. All printing of relevant information should be done at the calling routine prior to calling SOF. Note that the run is ended in SOF by deliberately causing a system error of computing $\text{SQRT}(-1)$. This is done in order to trigger printing of the sequence of calling routines by the operating system.

RIEMAN

Here a single Riemann Problem (RP) is solved by calling RIEMAN from CYCEUL. Referring to Fig. A-2, the RP is solved by finding the point of intersection (USTAR,PSTAR) of Left-propagating and Right-propagating shock/rarefaction adiabats in the (U,P) plane. Prior to the actual computation, the qualitative wave structure is determined. It is characterized by the index NCASE as follows :

NCASE = 1 - Left wave is rarefaction, Right wave is shock.

NCASE = 2 - Both waves are shock.

NCASE = 3 - Left wave is shock, Right wave is rarefaction.

NCASE = 4 - Both waves are rarefaction.

The computation of (USTAR,PSTAR) is coded separately for each case. Newton-Raphson iteration is employed, the first guess being the intersection of the Left and Right rarefaction branches (or their extrapolations), which is done in closed-form. Since in a smooth flow this guess is close to the exact (USTAR,PSTAR), little extra CPU effort is spent on subsequent Newton-Raphson iterations. These are truly needed only in regions of shock wave computation.

The computation in RIEMAN is concluded by computing $UW(1), \dots, UW(5)$ ($UW(6) = \text{infinity}$). From these wave speeds, the flux index NFLUX that denotes the location of the X-axis on the (X,T) wave diagram of the RP solution (Fig. A-3), is evaluated. It is later needed in subroutine FLUXE.

MAGA

The major purpose of this routine is to compute DUIDT and DPIDT along the contact surface of the RP solution. Since U and P are continuous across the contact, so are their time-derivatives along the contact. Thus, DUIDT and DPIDT are solved from a set of two linear equations. The coefficients of each equation are determined by GRP analysis of the wave on one side. See References 4-6 (particularly Ref. 6) for details.

FLUXE

The major task of this routine is to compute second-order fluxes. This is done in two phases. The first phase is up to statement 9 CONTINUE, where using NFLUX the X-suffixed values of flow variables and their time-derivatives are defined. An X-suffix means that the variable or its time-derivative are related to the line $X = X(I)$ on the (X,T) wave diagram (Fig. A-3). In the second phase, these variables and their time-derivatives are used to extend fluxes at $X(I)$ to Half-time-step (hence the suffix H), i.e. $T + DT/2$. It is these fluxes which are the second-order accurate fluxes for the integration of the conservation laws from T to $T + DT$. Also, cell-interface flow variables (suffix N) are extended to New time level $T + DT$. These are later used in defining cell differences DU(I), DP(I) and DRO(I) in CYCEUL.

A.5 Listing of GKP Code

		CHARGE VERSION	
C#OPTIONS LIST			CHAC001
IMPLICIT REAL*8(A-H,O-Z,*)			CHA0002
C PROGRAM GRP - GENERALIZED RIEMANN PROBLEM.			CHA0003
C EXPANSION OF A DETONATED SPHERICAL CHARGE IN VACUUM.			CHA0004
C INITIAL CONDITIONS FROM TAYLOR'S SELF SIMILAR SOLUTION.			CHA0005
COMMON B(102,26)			CHA0006
1, ENDB			CHA0007
COMMON /AB/A(50)			CHA0008
EQUIVALENCE (L,A(1)),(LL,A(2)),(T,A(3)),(DT,A(4)),(TMAX,A(5)),			CHA0009
1 (TMUD,A(6)),(DTMUD,A(7)),(JOB,A(8)),(NERI,A(9)),			CHA0010
2 (JJJ,A(10)),(KEYMON,A(11)),(NCYC,A(12))			CHA0011
EQUIVALENCE (COLELA,A(13))			CHA0012
EQUIVALENCE (LAGEUL,A(14))			CHA0013
EQUIVALENCE (UGAL,A(15))			CHA0014
EQUIVALENCE (KEYEK,A(16))			CHA0015
EQUIVALENCE (NCYCPR,A(17))			CHA0016
EQUIVALENCE (STAB,A(18)),(DTBA,A(19)),(DTKOD,A(20)),(KDT,A(21))			CHA0017
COMMON /MONIT/CASEAV(4),NC14(4),NF16(6),			CHA0018
1 NMONU(4),NMONP(4),NMONRO(4),NMONZ(4)			CHA0019
DIMENSION NZERO(26)			CHA0020
EQUIVALENCE (NZERO(1),NC14(1))			CHA0021
COMMON/PULS/PRESS(10),PULSE1(10),PULSE2(10),PULSE3(10),PULSE4(10)			CHA0022
C*****			CHA0023
DO 20 N=1,26			CHA0024
20 NZERO(N)=0			CHA0025
DO 21 N=1,4			CHA0026
21 CASEAV(N)=0.			CHA0027
DO 31 N=1,10			CHA0028
PRESS(N)=0.			CHA0029
PULSE1(N)=0.			CHA0030
PULSE2(N)=0.			CHA0031
PULSE3(N)=0.			CHA0032
PULSE4(N)=0.			CHA0033
31 CONTINUE			CHA0034
NMAT=26			CHA0035
C L=(LOC(ENDB)-LOC(B(1,1)))/NMAT			CHA0036
L=102			CHA0037
LL=L-1			CHA0038
NN=NMAT*LL			CHA0039
DO 1 J=1,L			CHA0040
DO 1 II=1,NMAT			CHA0041
1 B(I,II)=0.			CHA0042
CALL MAINO(L,B(1,1),B(1,2),B(1,3),B(1,4),B(1,5),			CHA0043
1 B(1,6),B(1,7),B(1,8),B(1,9),B(1,10),			CHA0044
2 B(1,11),B(1,12),B(1,13),B(1,14),B(1,15),			CHA0045
3 B(1,16),B(1,17),B(1,18),B(1,19),B(1,20),			CHA0046
4 B(1,21),B(1,22),B(1,23),B(1,24),B(1,25),			CHA0047
5 B(1,26))			CHA0048
STOP			CHA0049
END			CHA0050
SUBROUTINE MAINO		MAINO	CHA0051
1 (L,X,U,P,RO,G,E,DU,DP,DRO,DG,DXSI,MIN,			CHA0052
2 US,PS,UIDOT,PIDOT,			CHA0053
* FIMZ,ZMDOT,			CHA0054
3 TENA,FIRO,FIM,FIE,GIP,VOL,Z,DZ)			CHA0055
IMPLICIT REAL*8(A-H,O-Z,*)			CHA0056
DIMENSION X(L),U(L),P(L),RO(L),G(L),E(L),DU(L),DP(L),DRO(L),			CHA0057
1 DG(L),DXSI(L),MIN(L),			CHA0058
2 US(L),PS(L),UIDOT(L),PIDOT(L)			CHA0059
3 ,TENA(L),FIRO(L),FIM(L),FIE(L)			CHA0060
4 ,GIP(L),VOL(L),Z(L),DZ(L)			CHA0061
5 ,FIMZ(L),ZMDOT(L)			CHA0062
COMMON /AB/A(50)			CHA0063
EQUIVALENCE (LL,A(2)),(T,A(3)),(DT,A(4)),(TMAX,A(5)),			CHA0064
1 (TMUD,A(6)),(DTMUD,A(7)),(JOB,A(8)),(NERI,A(9)),			CHA0065
2 (JJJ,A(10)),(KEYMON,A(11)),(NCYC,A(12))			CHA0066
EQUIVALENCE (LAGEUL,A(14))			CHA0067
EQUIVALENCE (NCYCPR,A(17))			CHA0068
EQUIVALENCE (STAB,A(18)),(DTBA,A(19)),(DTKOD,A(20)),(KDT,A(21))			CHA0069
COMMON /TOT/AMTOT,ETOT,EKTOT,FPTOT,TENTOT			CHA0070
C*****			CHA0071
T=0.			CHA0072

	NCYC=0	CHA0073
	JJJ=0	CHA0074
	CALL NETUNM	CHA0075
	DELT=DT	CHA0076
	CALL BEGIN	CHA0077
1	(L,X,U,P,RO,G,E,DU,DP,DRO,DG,DXSI,MIN,	CHA0078
2	US,PS,UIDOT,PIDOT,	CHA0079
*	FIMZ,ZMDOT,	CHA0080
3	TENA,FIRO,FIM,FIE,GIP,VOL,Z,DZ)	CHA0081
	CALL SAFAE	CHA0082
1	(L,X,U,F,RO,G,E,DU,DP,DRO,DG,DXSI,MIN,	CHA0083
2	US,PS,UIDOT,PIDOT,	CHA0084
*	FIMZ,ZMDOT,	CHA0085
3	TENA,FIRO,FIM,FIE,GIP,VOL,Z,DZ)	CHA0086
1	NCYC=NCYC+1	CHA0087
C	TIME STEP CONTROL.	CHA0088
	DT=DTBA	CHA0089
	IF(DT.GT.1.1D0*DTKOD.AND.DTKOD.NE.0.) DT=1.1D0*DTKOD	CHA0090
	IF(NCYC.EQ.2) DT=DT/10.D0	CHA0091
	IF(NCYC.EQ.1) DT=0.	CHA0092
	IF(DT.EQ.0.) GO TO 11	CHA0093
	NHAD=((TMUD-T)/DT-1.D-10)	CHA0094
	IF(NHAD.GE.10) GO TO 11	CHA0095
	DT=(TMUD-T)/DFLOAT(NHAD+1)	CHA0096
11	CONTINUE	CHA0097
	T=T+DT	CHA0098
	IF((NCYC/NCYCPR)*NCYCPR.NE.NCYC.AND.NCYC.GT.NCYCPR) GO TO 33	CHA0099
	PRINT 10, NCYC,T,DT,KDT	CHA0100
10	FORMAT(1X,'NCYC=',I4,3X,'T=',D11.4,3X,'DT=',D11.4,3X,'KDT=',I4)	CHA0101
33	CONTINUE	CHA0102
	DTBA=DTMUD	CHA0103
	KDT=0	CHA0104
	NERI=1	CHA0105
	IF(DABS(T-TMUD).LT.1.D-8) NERI=0	CHA0106
	CALL CYCEUL	CHA0107
1	(L,X,U,P,RO,G,E,DU,DP,DRO,DG,DXSI,MIN,	CHA0108
2	US,PS,UIDOT,PIDOT,	CHA0109
*	FIMZ,ZMDOT,	CHA0110
3	TENA,FIRO,FIM,FIE,GIP,VOL,Z,DZ)	CHA0111
	CALL SAFAE	CHA0112
1	(L,X,U,P,RO,G,E,DU,DP,DRO,DG,DXSI,MIN,	CHA0113
2	US,PS,UIDOT,PIDOT,	CHA0114
*	FIMZ,ZMDOT,	CHA0115
3	TENA,FIRO,FIM,FIE,GIP,VOL,Z,DZ)	CHA0116
	IF(NERI.NE.0) GO TO 2	CHA0117
	CALL PRINT	CHA0118
1	(L,X,U,P,RO,G,E,DU,DP,DRO,DG,DXSI,MIN,	CHA0119
2	US,PS,UIDOT,PIDOT,	CHA0120
*	FIMZ,ZMDOT,	CHA0121
3	TENA,FIRO,FIM,FIE,GIP,VOL,Z,DZ)	CHA0122
	IF(DABS(T-TMUD).LT.1.D-8) TMUD=TMUD+DTMUD	CHA0130
2	CONTINUE	CHA0131
	DTKOD=DT	CHA0132
	IF(T.LT.TMAX-1.D-8) GO TO 1	CHA0133
	RETURN	CHA0134
	END	CHA0135
	SUBROUTINE NETUNM	NETUNM
	IMPLICIT REAL*8(A-H,O-Z,*)	CHA0136
	COMMON /AB/A(50)	CHA0137
	EQUIVALENCE (L,A(1))	CHA0138
	EQUIVALENCE (LL,A(2)),(T,A(3)),(DT,A(4)),(TMAX,A(5)),	CHA0139
1	(TMUD,A(6)),(DTMUD,A(7)),(JOB,A(8)),(NERI,A(9)),	CHA0140
2	(JJJ,A(10)),(KEYMON,A(11)),(NCYC,A(12))	CHA0141
	EQUIVALENCE (COLELA,A(13))	CHA0142
	EQUIVALENCE (LAGEUL,A(14))	CHA0143
	EQUIVALENCE (KEYEK,A(16))	CHA0144
	EQUIVALENCE (NCYCPR,A(17))	CHA0145
	EQUIVALENCE (STAB,A(18)),(DTBA,A(19)),(DTKOD,A(20)),(KDT,A(21))	CHA0146
	COMMON/DETO/QDET,PCJDET,RCJDET,UCJDET,DCJDET,PODET,ROODET,	CHA0147
1	RATE,TEMPC	CHA0148
	COMMON/DIFFUS/U2,P2,R02,ARW	CHA0149
	COMMON /DRAW/GODELX,GODELY,UMIN,UMAX,PMIN,PMAX,ROMIN,ROMAX	CHA0150
		CHA0151

```

1      ,XMIN,XMAX,SMIN,SMAX,IVERSA
COMMON /GAM/GAMA,NG,MU2,G1,G2,G3,G4,G5,G6,G7,G8,G9,G10,G11
1      ,G12,G13,G14,G15,G16,G17,G18,G19,G20,G21,G22,G23
2      ,G24,G25,G26,G27,G28,G29,G30,G31,G32,G33,G34,G35
REAL*8 NG,MU2
NAMELIST /IN/LIN,GAMA,DT,DTMUD,DTMUD,TMAX,
1      GODELX,GODELY,UMIN,UMAX,FMIN,PMAX,ROMIN,ROMAX,
2      SMIN,SMAX,IVERSA,KEYMON,COLELA,STAB
3      ,LAGEUL,KEYEK
4      ,QDET
C*****
LIN=L
LAGEUL=2
NCYCPR=1
KEYEK=1
TMUD=0.
DTMUD=10.D0
TMAX=100.D0
STAB=0.5D0
DT=1.D-2
KEYMON=1
GAMA=3.D0+1.D-6
QDET=0.04D0
RATE=0.
TEMPC=1.D50
GODELX=16D0
GODELY=20.D0
IVERSA=100
UMIN=0.
UMAX= 1.D0
PMIN=0.
PMAX=0.5D0
ROMIN=0.
ROMAX=3.D0
SMIN=0.
SMAX=0.03D0
COLELA=0.
READ IN
C
C
C
PRINT IN
GG=2.D0*GAMA/(GAMA-1.D0)
NG=GG
10  CONTINUE
MU2=(GAMA-1.D0)/(GAMA+1.D0)
G1=GAMA-1.D0
G2=1.D0-MU2
G3=2.D0/(3.D0*GAMA-1.D0)
G4=(GAMA+1.D0)/2.D0
G5=0.5D0*(3.D0*GAMA-1.D0)/(GAMA+1.D0)
G6=(GAMA+1.D0)/(2.D0*GAMA)
G7=2.D0/(GAMA-1.D0)
G8=(GAMA-1.D0)/(2.D0*GAMA)
G9=(GAMA+1.D0)/(4.D0*GAMA)
G10=1.D0/GAMA
G11=(GAMA+1.D0)/4.D0
G12=GAMA/(GAMA-1.D0)
G13=0.5D0*(GAMA-3.D0)/(GAMA+1.D0)
G14=0.5D0*(3.D0*GAMA-5.D0)/(GAMA+1.D0)
G15=GAMA*(3.D0*GAMA-1.D0)
G16=(GAMA+1.D0)/(2.D0*(GAMA-1.D0))
G17=GAMA+1.D0
G18=GAMA*(GAMA+1.D0)/(3.D0*GAMA-1.D0)
G19=(3.D0*GAMA-1.D0)/(GAMA+1.D0)
G20=2.D0*(GAMA-1.D0)/(3.D0*GAMA-1.D0)**2
G21=GAMA*(3.D0*GAMA-5.D0)/(3.D0*GAMA-1.D0)**2
GODELX=GODELX/2.54D0
GODELY=GODELY/2.54D0
CALL NAMPLT(IVERSA)
C
C
CALL LIMIT(1000.D0)
CALL PLOT(0.,0.5D0,-3)
C
PODET=0.

```

CHA0152
 CHA0153
 CHA0154
 CHA0155
 CHA0156
 CHA0157
 CHA0158
 CHA0159
 CHA0160
 CHA0161
 CHA0162
 CHA0163
 CHA0164
 CHA0165
 CHA0166
 CHA0167
 CHA0168
 CHA0169
 CHA0170
 CHA0171
 CHA0172
 CHA0173
 CHA0174
 CHA0175
 CHA0176
 CHA0177
 CHA0178
 CHA0179
 CHA0180
 CHA0181
 CHA0182
 CHA0183
 CHA0184
 CHA0185
 CHA0186
 CHA0187
 CHA0188
 CHA0189
 CHA0190
 CHA0191
 CHA0192
 CHA0193
 CHA0194
 CHA0195
 CHA0196
 CHA0197
 CHA0198
 CHA0199
 CHA0200
 CHA0201
 CHA0202
 CHA0203
 CHA0204
 CHA0205
 CHA0206
 CHA0207
 CHA0208
 CHA0209
 CHA0210
 CHA0211
 CHA0212
 CHA0213
 CHA0214
 CHA0215
 CHA0216
 CHA0217
 CHA0218
 CHA0219
 CHA0220
 CHA0221
 CHA0222
 CHA0223

```

      ROODET=0.
      PCJDET=0.
      UCJDET=0.
      DCJDET=0.
      RCJDET=0.
      IF(QDET.LE.0.) GO TO 100
C  DETONATION DATA
      QDET=0.04D0
      PODET=0.
      ROODET=1.8D0
      PCJDET=PODET-(GAMA-1.D0)*(-QDET)*ROODET+
1     DSQRT(((GAMA-1.D0)*QDET*ROODET)**2-2.D0*MU2*GAMA*
2     (-QDET)*PODET*ROODET)
      RCJDET=ROODET*((GAMA+1.D0)*PCJDET-PODET)/(GAMA*PCJDET)
      CCJ=DSQRT(GAMA*PCJDET/RCJDET)
      DCJDET=CCJ*RCJDET/ROODET
      UCJDET=DCJDET-CCJ
      PRINT 101
101  FORMAT(1H1,/,1X,'DETONATION DATA'/)
      PRINT 102, QDET,GAMA,TEMPC,RATE
102  FORMAT(/1X,'QDET,GAMA,TEMP,RATE=',4D18.8)
      PRINT 103, ROODET,PODET
103  FORMAT(/1X,'UNBURNED STATE      ROODET,PODET=',2D18.8)
      PRINT 104, DCJDET,PCJDET,RCJDET,UCJDET
104  FORMAT(/1X,'CJ POINT      DCJDET,PCJDET,RCJDET,UCJDET=',4D18.8)
100  CONTINUE
      RETURN
      END
SUBROUTINE BEGIN
1     (L,X,U,P,RO,G,E,DU,DP,DRO,DG,DXSI,MIN,
2     US,PS,UIDOT,PIDOT,
3     FIMZ,ZMDOT,
4     TENA,FIRO,FIM,FIE,GIP,VOL,Z,DZ)
      IMPLICIT REAL*8(A-H,O-Z,*)
      DIMENSION X(L),U(L),P(L),RO(L),G(L),E(L),DU(L),DP(L),DRO(L),
1     DG(L),DXSI(L),MIN(L),
2     US(L),PS(L),UIDOT(L),PIDOT(L)
3     ,TENA(L),FIRO(L),FIM(L),FIE(L)
4     ,GIP(L),VOL(L),Z(L),DZ(L)
5     ,FIMZ(L),ZMDOT(L)
      COMMON /AB/A(50)
      COMMON /GAM/GAMA,NG,MU2,G1,G2,G3,G4,G5,G6,G7,G8,G9,G10,G11
1     ,G12,G13,G14,G15,G16,G17,G18,G19,G20,G21,G22,G23
2     ,G24,G25,G26,G27,G28,G29,G30,G31,G32,G33,G34,G35
      REAL*8 NG,MU2
      EQUIVALENCE (LL,A(2))
      EQUIVALENCE (LAGEUL,A(14))
      EQUIVALENCE (UGAL,A(15))
      EQUIVALENCE (STAB,A(18)),(DTBA,A(19)),(DTKOD,A(20)),(KDT,A(21))
      COMMON/DETO/QDET,PCJDET,RCJDET,UCJDET,DCJDET,PODET,ROODET,
1     RATE,TEMPC
      COMMON /DRAW/GODELX,GODELY,UMIN,UMAX,PMIN,PMAX,ROMIN,ROMAX
1     ,XMIN,XMAX,SMIN,SMAX,IVERSA
      COMMON/GIT/ROLIM,ELIM,XGIT(200),ROGIT(200),ROUGIT(200),EGIT(200)
      COMMON /GITN/NP0
      LOGICAL CSOF
C*****
      DTBA=0.
      DTKOD=0.
      KDT=0
      PO=1.D-9
      RHOO=1.D-7
      UO=UCJDET
      UGAL=0.
      XO=0.
      X1=50.D0
      XCHARG=10.D0
      XMIN=XO
      XMAX=X1
      DX=(X1-XO)/(L-2.D0)
      DO 1 I=2,L
      X(I)=XO+(I-2.D0)*DX

```

```

1  CONTINUE
   X(L)=X1
   U0=U0*CROSS(XCHARG)
   DO 2 I=2,LL
   IF(I.GT.2) U(I)=U0/CROSS(X(I))
   P(I)=P0
   RO(I)=RH00
   Z(I)=0.
   GO TO (31,32), LAGEUL
31  CONTINUE
   E(I)=P(I)/((GAMA-1.D0)*RO(I))+0.5D0*U(I)**2+Z(I)*QDET
   GO TO 30
32  CONTINUE
   E(I)=P(I)/(GAMA-1.D0)+0.5D0*RO(I)*U(I)**2+Z(I)*RO(I)*QDET
30  CONTINUE
   Q(I)=DSQRT(GAMA*P(I)*RO(I))
2   CONTINUE
   DO 3 I=2,LL
   TENA(I)=RO(I)*U(I)
   VOL(I)=(X(I+1)-X(I))*(X(I+1)**2+X(I+1)*X(I)+X(I)**2)/3.D0
3   CONTINUE
C
C  INSERT DETONATED CHARGE FLOW FIELD FROM TAYLOR'S SOLUTION.
C
   CALL TAYLOR(GAMA)
   RONORM=RCJDET
   RUNORM=RCJDET*UCJDET
   ENORM=RCJDET*DCJDET**2
   XLIM=XGIT(NP0)
   NGIT=NP0-1
   XG1=XLIM
   XG2=XGIT(NGIT)
   AROIP =ROGIT (NP0)+ROLIM*XLIM**3/3.D0
   AROUIP=ROUGIT(NP0)
   AEIP  =EGIT (NP0)+ ELIM*XLIM**3/3.D0
   XP=X(2)/XCHARG
   DO 100 I=2,LL
   IP=I+1
   XI=XP
   AROI =AROIP
   AROUI=AROUIP
   AEI  =AEIP
   XP=X(IP)/XCHARG
   IF(DABS(XP-1.D0).LT.1.D-10) XP=1.D0
   CSOF=(XP.GE.1.D0)
   IF(XP.GE.XLIM) GO TO 101
C  UNIFORM FLOW REGION
   DELVOL=(XLIM-XP)*(XLIM**2+XLIM*XP+XP**2)/3.D0
   AROIP =ROGIT (NP0)+ROLIM*DELVOL
   AROUIP=ROUGIT(NP0)
   AEIP  =EGIT (NP0)+ ELIM*DELVOL
   GO TO 102
101 CONTINUE
C  NON UNIFORM FLOW REGION.
   IF(.NOT.CSOF) GO TO 104
C  LAST POINT. (THIS IS THE DETONATION FRONT POINT X=1).
   AROIP= 0.
   AROUIP=0.
   AEIP= 0.
   GO TO 102
104 CONTINUE
   IF(XP.LE.XG2) GO TO 103
   NGIT=NGIT-1
   IF(NGIT.LE.0) CALL SOF('BEGIN 104. NGIT.LE.0.')
   XG1=XG2
   XG2=XGIT(NGIT)
   GO TO 104
103 CONTINUE
   FRAC=(XP-XG1)/(XG2-XG1)
   IF(FRAC.LT.0.) CALL SOF('BEGIN 103. FRAC.LT.0.')
   IF(FRAC.GT.1.D0) CALL SOF('BEGIN 103. FRAC.GT.1.')
   AROIP =(1.D0-FRAC)*ROGIT (NGIT+1)+FRAC*ROGIT (NGIT)

```

CHA0296
 CHA0297
 CHA0298
 CHA0299
 CHA0300
 CHA0301
 CHA0302
 CHA0303
 CHA0304
 CHA0305
 CHA0306
 CHA0307
 CHA0308
 CHA0309
 CHA0310
 CHA0311
 CHA0312
 CHA0313
 CHA0314
 CHA0315
 CHA0316
 CHA0317
 CHA0318
 CHA0319
 CHA0320
 CHA0321
 CHA0322
 CHA0323
 CHA0324
 CHA0325
 CHA0326
 CHA0327
 CHA0328
 CHA0329
 CHA0330
 CHA0331
 CHA0332
 CHA0333
 CHA0334
 CHA0335
 CHA0336
 CHA0337
 CHA0338
 CHA0339
 CHA0340
 CHA0341
 CHA0342
 CHA0343
 CHA0344
 CHA0345
 CHA0346
 CHA0347
 CHA0348
 CHA0349
 CHA0350
 CHA0351
 CHA0352
 CHA0353
 CHA0354
 CHA0355
 CHA0356
 CHA0357
 CHA0358
 CHA0359
 CHA0360
 CHA0361
 CHA0362
 CHA0363
 CHA0364
 CHA0365
 CHA0366
 CHA0367

```

      AROUIP=(1.DO-FRAC)*ROUGIT(NGIT+1)+FRAC*ROUGIT(NGIT)
      AEIP =(1.DO-FRAC)*EGIT (NGIT+1)+FRAC*EGIT (NGIT)
102  CONTINUE
C  COMPUTE MASS, MOMENTUM AND ENERGY DENSITIES.
      IF(XP.LE.XLIM) GO TO 105
C  CONSERVATION-FORM DEFINITION OF MASS, MOMENTUM AND ENERGY DENSITY.
      DVOL=(XP-XI)*(XP**2+XP**XI+XI**2)/3.DO
      RO (I)=RONORM*(AROI - AROIP)/DVOL
      TENA(I)=RUNORM*(AROI-AROUI)/DVOL
      E (I)=ENORM *(AEI - AEIP)/DVOL
      GO TO 106
105  CONTINUE
C  UNIFORM FLOW REGION
      RO (I)=RONORM*ROLIM
      TENA(I)=0.
      E (I)=ENORM * ELIM
106  CONTINUE
      U(I)=TENA(I)/RO(I)
      P(I)=(GAMA-1.DO)*(E(I)-0.5DO*RO(I)*U(I)**2)
      PRINT 111,I,CSOF,U(I),P(I),RO(I),E(I)
111  FORMAT(/1X,'I,CSOF,U,P,RO,E=',I4,L3,4D14.4)
      IF(CSOF) GO TO 109
100  CONTINUE
109  CONTINUE
      DO 4 I=2,LL
      DXSI(I)=(X(I+1)-X(I))*RO(I)
      4  CONTINUE
      RETURN
      END
SUBROUTINE TAYLOR(GAMA)
      IMPLICIT REAL*8(A-H,O-Z,*)
C
C  TAYLOR -- SELF SIMILAR SPHERICAL DETONATION (CJ) FLOW FIELD
C
      COMMON /GGGG/G,G1,G2,G3,G4,G5,G6,G7,G8,G9,G10
      COMMON /PAR/RH00,Q0,ROCJ,DCJ,UCJ,PCJ,DPSI,PSIMAX,C0,U0
      COMMON /GITN/NP0
      COMMON/GIT/ROLIM,ELIM,XGIT(200),ROGIT(200),ROUGIT(200),EGIT(200)
C*****
      G=GAMA
      PRINT 101
101  FORMAT('1')
      PRINT 110
110  FORMAT(1X,'G. I. TAYLOR SOLUTION.  N,PSI,U,C,X/AM,AT,AE='//)
      CALL INIDAT
      X=1.DO
      Y=0.
      U=U0
      C=C0
      AM=0.
      AT=0.
      AE=0.
      PSI=-DLOG(U)
      DO 1 N=1,NP0
      XGIT (N)=X
      ROGIT (N)=AM
      ROUGIT(N)=AT
      EGIT (N)=AE
      PRINT 11, N,PSI,U,C,X,AM,AT,AE
11  FORMAT(1X,I4,4D14.5/5X,3D14.5)
      CALL RUNGE(N,PSI,X,C,AM,AT,AE,PSIN,XN,CN,AMN,ATN,AEN)
      PSI=PSIN
      U=DEXP(-PSI)
      X=XN
      C=CN
      AM=AMN
      AT=ATN
      AE=AEN
1  CONTINUE
      ROLIM=(C/C0)*G3
      ELIM=G5*(C/C0)*G4
      AM0=AM+(C/C0)*G3*X**3/3.DO

```

CHA0368
 CHA0369
 CHA0370
 CHA0371
 CHA0372
 CHA0373
 CHA0374
 CHA0375
 CHA0376
 CHA0377
 CHA0378
 CHA0379
 CHA0380
 CHA0381
 CHA0382
 CHA0383
 CHA0384
 CHA0385
 CHA0386
 CHA0387
 CHA0388
 CHA0389
 CHA0390
 CHA0391
 CHA0392
 CHA0393
 CHA0394
 CHA0395
 CHA0396
 CHA0397
 CHA0398
 CHA0399
 CHA0400
 CHA0401
 CHA0402
 CHA0403
 CHA0404
 CHA0405
 CHA0406
 CHA0407
 CHA0408
 CHA0409
 CHA0410
 CHA0411
 CHA0412
 CHA0413
 CHA0414
 CHA0415
 CHA0416
 CHA0417
 CHA0418
 CHA0419
 CHA0420
 CHA0421
 CHA0422
 CHA0423
 CHA0424
 CHA0425
 CHA0426
 CHA0427
 CHA0428
 CHA0429
 CHA0430
 CHA0431
 CHA0432
 CHA0433
 CHA0434
 CHA0435
 CHA0436
 CHA0437
 CHA0438
 CHA0439

TAYLOR

```

      AM0=AM0*5.D0*(G+1.D0)/G
      AE0=AE+(G5*(C/C0)**G4+0.5D0*(C/C0)**G3*U**2)*X**3/3.D0
      AE0=AE0*6.D0*(G+1.D0)*(G**2-1.D0)/G
      PRINT 22,AM0,AE0
22    FORMAT(//1X,'MASS AND ENERGY CHECK (SHOULD BE 1.)'//
1      1X,'M0=',D17.8,5X,'E0=',D17.8//)
      RETURN
      END
      SUBROUTINE INIDAT
      IMPLICIT REAL*8(A-H,O-Z,*)
      COMMON /GGGG/G,G1,G2,G3,G4,G5,G6,G7,G8,G9,G10
      COMMON /PAR/RH00,Q0,ROCJ,DCJ,UCJ,PCJ,DPSI,PSIMAX,C0,U0
      COMMON /GITN/NP0
      C*****
      NP0=200
      PSIMAX=10.D0
      U0=1.D0/(G+1.D0)
      C0=1.D0-U0
      DPSI=PSIMAX/DFLOAT(NP0)
      G1=G-1.D0
      G2=G1/2.D0
      G3=2.D0/(G-1.D0)
      G4=2.D0*G/(G-1.D0)
      G5=G/((G+1.D0)**2*(G-1.D0))
      RETURN
      END
      SUBROUTINE RUNGE(N,PSI,X,C,AM,AT,AE,PSIN,XN,CN,AMN,ATN,AEN)
      IMPLICIT REAL*8(A-H,O-Z,*)
      COMMON /GGGG/G,G1,G2,G3,G4,G5,G6,G7,G8,G9,G10
      COMMON /PAR/RH00,Q0,ROCJ,DCJ,UCJ,PCJ,DPSI,PSIMAX,C0,U0
      COMMON /GITN/NP0
      C*****
      H=DPSI
      H2=H/2.D0
      H6=H/6.D0
      CALL DERIV(PSI,X,C,AM,AT,AE,
1      DXDP1,DCDP1,DMDP1,DTDP1,DEDP1)
      CALL DERIV(PSI+H2,X+H2*DXDP1,C+H2*DCDP1,AM,AT,AE,
1      DXDP2,DCDP2,DMDP2,DTDP2,DEDP2)
      CALL DERIV(PSI+H2,X+H2*DXDP2,C+H2*DCDP2,AM,AT,AE,
1      DXDP3,DCDP3,DMDP3,DTDP3,DEDP3)
      CALL DERIV(PSI+H,X+H*DXDP3,C+H*DCDP3,AM,AT,AE,
1      DXDP4,DCDP4,DMDP4,DTDP4,DEDP4)
      PSIN=PSI+H
      XN=X+H6*(DXDP1+2.D0*(DXDP2+DXDP3)+DXDP4)
      CN=C+H6*(DCDP1+2.D0*(DCDP2+DCDP3)+DCDP4)
      AMN=AM+H6*(DMDP1+2.D0*(DMDP2+DMDP3)+DMDP4)
      ATN=AT+H6*(DTDP1+2.D0*(DTDP2+DTDP3)+DTDP4)
      AEN=AE+H6*(DEDP1+2.D0*(DEDP2+DEDP3)+DEDP4)
      RETURN
      END
      SUBROUTINE DERIV(PSI,X,C,AM,AT,AE,DXDP,DCDP,DMDP,DTDP,DEDP)
      IMPLICIT REAL*8(A-H,O-Z,*)
      COMMON /GGGG/G,G1,G2,G3,G4,G5,G6,G7,G8,G9,G10
      COMMON /PAR/RH00,Q0,ROCJ,DCJ,UCJ,PCJ,DPSI,PSIMAX,C0,U0
      COMMON /GITN/NP0
      C*****
      U=DEXP(-PSI)
      DXDP=0.5D0*X*(C-U+X)*(C+U-X)/C**2
      DCDP=-G2*U*(X-U)/C
      DMDP=-(C/C0)**G3*X**2*DXDP
      DTDP=DMDP*U
      DEDP=-(G5*(C/C0)**G4+0.5D0*(C/C0)**G3*U**2)*X**2*DXDP
      RETURN
      END
      DOUBLE PRECISION FUNCTION RATIO(X)
      IMPLICIT REAL*8(A-H,O-Z,*)
      C*****
      RATIO=0.
      IF(X.LE.1.D-8)RETURN
      RATIO=2.D0/X
      RETURN

```

END	CROSS	CHA0512
DOUBLE PRECISION FUNCTION CROSS(X)		CHA0513
IMPLICIT REAL*8(A-H,O-Z,*)		CHA0514
C*****		CHA0515
C CROSS=1.D0		CHA0516
C CROSS=X**2		CHA0517
C RETURN		CHA0518
C END		CHA0519
SUBROUTINE CYCEUL	CYCEUL	CHA0520
1 (L,X,U,P,RO,O,E,DU,DP,DRO,DG,DXSI,MIN,		CHA0521
2 US,PS,UIDOT,PIDOT,		CHA0522
3 FIMZ,ZMDOT,		CHA0523
4 TENA,FIRO,FIM,FIE,GIP,VOL,Z,DZ)		CHA0524
5 IMPLICIT REAL*8(A-H,O-Z,*)		CHA0525
1 DIMENSION X(L),U(L),P(L),RO(L),G(L),E(L),DU(L),DP(L),DRO(L),		CHA0526
2 DG(L),DXSI(L),MIN(L),		CHA0527
3 US(L),PS(L),UIDOT(L),PIDOT(L)		CHA0528
4 ,TENA(L),FIRO(L),FIM(L),FIE(L)		CHA0529
5 ,GIP(L),VOL(L),Z(L),DZ(L)		CHA0530
COMMON /AB/A(50)		CHA0531
EQUIVALENCE (LL,A(2)),(T,A(3)),(DT,A(4)),(COLELA,A(13))		CHA0532
EQUIVALENCE (KEYEK,A(16))		CHA0533
EQUIVALENCE (STAB,A(18)),(DTBA,A(19)),(DTKOD,A(20)),(KDT,A(21))		CHA0534
COMMON /GAM/GAMA,NG,MU2,G1,G2,G3,G4,G5,G6,G7,G8,G9,G10,G11		CHA0535
1 ,G12,G13,G14,G15,G16,G17,G18,G19,G20,G21,G22,G23		CHA0536
2 ,G24,G25,G26,G27,G28,G29,G30,G31,G32,G33,G34,G35		CHA0537
REAL*8 NG,MU2		CHA0538
COMMON /TOT/AMTOT,ETOT,EKTOT,EPTOT,TENTOT		CHA0539
COMMON /AZOV/ISAF,NORIMN,USAF,PSAF,ROSAF,GSFAF,ESAF,DPSAF		CHA0540
1 ,DXSIL,DXSIR		CHA0541
LOGICAL NORIMN		CHA0542
COMMON /STEP0/UL,PL,ROL,GL,UR,PR,ROR,GR,USTAR,PSTAR,		CHA0543
1 RSTARL,RSTARR,GSTARL,GSTARR,		CHA0544
2 CL,CR,CSTARL,CSTARR,SL,SR,WL,WR,UW(6)		CHA0545
3 ,LAMDAL,LAMDAR,RATEL,RATER,TEMPL,TEMPLR,TEMPSL,TEMPSR		CHA0546
4 ,ZL,ZR,ZSTARL,ZSTARR,NFLUX,HELEML,HELEMR		CHA0547
REAL*8 LAMDAL,LAMDAR		CHA0548
LOGICAL HELEML,HELEMR		CHA0549
COMMON /STEP1/DUIDT,DPIDT,DGIDTL,DGIDTR,DRIDTL,DRIDTR		CHA0550
2 ,ASTARL,ASTARR,LAMDSL,LAMDSR,DSDAL,DSDAR,DZDAL,DZDAR		CHA0551
3 ,RAT,SH		CHA0552
4 ,BETACL,BETACR,DSDASL,DSDASR,DZDASL,DZDASR		CHA0553
REAL*8 LAMDSL,LAMDSR,DSDAL,DSDAR,DZDAL,DZDAR		CHA0554
COMMON /GRADS/DUDXIL,DPDXIL,DGDIXIL,DRDXIL,DZDXIL,DSDXIL,		CHA0555
1 DUDXIR,DPDXIR,DGDIXIR,DRDXIR,DZDXIR,DSDXIR		CHA0556
COMMON /FI/FIH1,FIH2,FIH3,UXN,PXN,GXN,ROXN,ZXN		CHA0557
1 ,GIH		CHA0558
2 ,FIH4,ZMDOTL,ZMDOTR		CHA0559
COMMON/DETO/QDET,PCJDET,RCJDET,UCJDET,DCJDET,PODET,ROODET,		CHA0560
1 RATE,TEMPC		CHA0561
COMMON/PULS/PRESS(10),PULSE1(10),PULSE2(10),PULSE3(10),PULSE4(10)		CHA0562
DATA ERRP/0.D0/		CHA0563
DATA NERRP/0/		CHA0564
C DATA KOTZ/7777777777B/		CHA0565
C*****		CHA0566
DT2=DT/2.D0		CHA0567
UXN=0.		CHA0568
PXN=0.		CHA0569
ROXN=0.		CHA0570
ZXN=0.		CHA0571
DO 1 I=2,L		CHA0572
IM=I-1		CHA0573
UXNM=UXN		CHA0574
PXNM=PXN		CHA0575
ROXNM=ROXN		CHA0576
ZXNM=ZXN		CHA0577
UL=U(IM)+0.5D0*DU(IM)		CHA0578
PL=P(IM)+0.5D0*DP(IM)		CHA0579
ROL=RO(IM)+0.5D0*DRO(IM)		CHA0580
GL=DSQRT(GAMA*PL*ROL)		CHA0581
CL=GL/ROL		CHA0582
		CHA0583
		CHA0584
		CHA0585
		CHA0586
		CHA0587

```

ZL=Z(IM)+0.5D0*DZ(IM)
IF(ZL.GT.1.D0) ZL=1.D0
IF(ZL.LT.0.) ZL=0.
SL=PL/(G1*ROL**GAMA)
TEMPL=PL/ROL
UR=U(I)-0.5D0*DU(I)
PR=P(I)-0.5D0*DP(I)
ROR=RO(I)-0.5D0*DRO(I)
GR=DSQRT(GAMA*PR*ROR)
CR=GR/ROR
ZR=Z(I)-0.5D0*DZ(I)
IF(ZR.GT.1.D0) ZR=1.D0
IF(ZR.LT.0.) ZR=0.
SR=PR/(G1*ROR**GAMA)
TEMPR=PR/ROR

```

```

C CALL KIEMAN(L,I,MIN)

```

```

C
DUDXIL=DU(IM)/DXSI(IM)
DPDXIL=DP(IM)/DXSI(IM)
DRDXIL=DRO(IM)/DXSI(IM)
DGDIL=0.5D0*GL*(DPDXIL/PL+DRDXIL/ROL)
DZDXIL=DZ(IM)/DXSI(IM)
DSDXIL=SL*(DPDXIL/PL-GAMA*DRDXIL/ROL)
DUDXIR=DJ(I)/DXSI(I)
DPDXIR=DP(I)/DXSI(I)
DRDXIR=DRO(I)/DXSI(I)
DGDIR=0.5D0*GR*(DPDXIR/PR+DRDXIR/ROR)
DZDXIR=DZ(I)/DXSI(I)
DSDXIR=SR*(DPDXIR/PR-GAMA*DRDXIR/ROR)
SH=CROSS(X(I))
RAT=RATIO(X(I))

```

```

C CALL MAGA(L,I,MIN)

```

```

C
US(I)=USTAR
PS(I)=PSTAR
UIDOT(I)=DUIDT
PIDOT(I)=DPIDT

```

```

C CALL FLUXE(L,I,MIN)

```

```

C
FIRO(I)=FIH1
FIM(I)=FIH2
FIE(I)=FIH3
FIMZ(I)=FIH4
GIP(I)=GIH
DU(IM)=UXN-UXNM
DP(IM)=PXN-PXNM
DRO(IM)=ROXN-ROXNM
DZ(IM)=ZXM-ZXNM

```

```

C STATIONS OUTPUT

```

```

IF((I-42)*(I-62)*(I-82)*(I-102).NE.0) GO TO 1
NPU=0
IF(I.EQ.42) NPU=1
IF(I.EQ.62) NPU=2
IF(I.EQ.82) NPU=3
IF(I.EQ.102) NPU=4
IF(NPU.EQ.0) CALL SCF('FLUXE 90. NPU.EQ.0')
PRESS(NPU)=GIH+FIH2
PULSE1(NPU)=PULSE1(NPU)+DT*GIH
PULSE2(NPU)=PULSE2(NPU)+DT*(GIH+FIH2)
PULSE3(NPU)=PULSE3(NPU)+DT*FIH1*CROSS(X(I))
PULSE4(NPU)=PULSE4(NPU)+DT*FIH2*CROSS(X(I))
CONTINUE

```

```

1
C
AMTOT=0.
ETOT=0.
EKTOT=0.
EPTOT=0.
TENTOT=0.
FIL=FIRO(2)

```

```

CHA0588
CHA0589
CHA0590
CHA0591
CHA0592
CHA0593
CHA0594
CHA0595
CHA0596
CHA0597
CHA0598
CHA0599
CHA0600
CHA0601
CHA0602
CHA0603
CHA0604
CHA0605
CHA0606
CHA0607
CHA0608
CHA0609
CHA0610
CHA0611
CHA0612
CHA0613
CHA0614
CHA0615
CHA0616
CHA0617
CHA0618
CHA0619
CHA0620
CHA0621
CHA0622
CHA0623
CHA0624
CHA0625
CHA0626
CHA0627
CHA0628
CHA0629
CHA0630
CHA0631
CHA0632
CHA0633
CHA0634
CHA0635
CHA0636
CHA0637
CHA0638
CHA0639
CHA0640
CHA0641
CHA0642
CHA0643
CHA0644
CHA0645
CHA0646
CHA0647
CHA0648
CHA0649
CHA0650
CHA0651
CHA0652
CHA0653
CHA0654
CHA0655
CHA0656
CHA0657
CHA0658
CHA0659

```

	FI2=FIM (2)	CHA0660
	FI3=FIE (2)	CHA0661
	FI4=FIMZ(2)	CHA0662
	GI2=QIP(2)	CHA0663
	SH=CROSS(X(2))	CHA0664
	DO 2 I=2,LL	CHA0665
	IP=I+1	CHA0666
	FIM1=FI1	CHA0667
	FIM2=FI2	CHA0668
	FIM3=FI3	CHA0669
	FIM4=FI4	CHA0670
	GIM2=GI2	CHA0671
	SHM=SH	CHA0672
	FI1=FIRO(IP)	CHA0673
	FI2=FIM (IP)	CHA0674
	FI3=FIE (IP)	CHA0675
	FI4=FIMZ(IP)	CHA0676
	GI2=QIP (IP)	CHA0677
	SH=CROSS(X(IP))	CHA0678
	DVOL=VOL(I)	CHA0679
	ROOLD=RO(I)	CHA0680
	POLD=P(I)	CHA0681
	EOLD=E(I)	CHA0682
	UOLD=U(I)	CHA0683
	ZOLD=Z(I)	CHA0684
	ZKODM=ZOLD*ROOLD	CHA0685
	TOLD=POLD/ROOLD	CHA0686
	DX=X(IP)-X(I)	CHA0687
	DTVOL=DT/DVOL	CHA0688
C	RO(I)=RO(I)-DTVOL*(SH*FI1-SHM*FIM1)	CHA0689
	TENA(I)=TENA(I)-DTVOL*(SH*FI2-SHM*FIM2)-(DT/DX)*(GI2-GIM2)	CHA0690
	E(I)=E(I)-DTVOL*(SH*FI3-SHM*FIM3)	CHA0691
	U(I)=TENA(I)/RO(I)	CHA0692
	Z(I)=(ZKODM-DTVOL*(SH*FI4-SHM*FIM4))/RO(I)	CHA0693
	IF(Z(I).GT.1.D0) Z(I)=1.D0	CHA0694
	IF(Z(I).LT.0.) Z(I)=0.	CHA0695
C	UAV=U(I)	CHA0696
	ROAV=RO(I)	CHA0697
	EP=E(I)-0.5D0*ROAV*UAV**2	CHA0698
	IF(EP.GT.0.) GO TO 291	CHA0699
	NERRP=NERRP+1	CHA0700
	ERRP=ERRP+(1.D-3-EP)*DVOL	CHA0701
	IF(ERRP.GT.0.24D0) GO TO 291	CHA0702
	EP=1.D-8	CHA0703
291	CONTINUE	CHA0704
	IF(EP.LE.0.) GO TO 7001	CHA0705
	P(I)=G1*EP	CHA0706
	G(I)=DSQRT(GAMA*P(I)*RO(I))	CHA0707
C	UPC=DABS(U(I))+G(I)/RO(I)	CHA0708
	DTI=STAB*DX/UPC	CHA0709
	IF(DTI.GT.DTBA) GO TO 29	CHA0710
	DTBA=DTI	CHA0711
	KDT=I	CHA0712
29	CONTINUE	CHA0713
	DXSI(I)=RO(I)*DX	CHA0714
	ETOT=ETOT+E(I)*DVOL	CHA0715
	EPTOT=EPTOT+EP*DVOL	CHA0716
	AMTOT=AMTOT+RO(I)*DVOL	CHA0717
	TENTOT=TENTOT+TENA(I)*DVOL	CHA0718
2	CONTINUE	CHA0719
	EKTOT=ETOT-EPTOT	CHA0720
C	IF(COLELA.EQ.0.) GO TO 200	CHA0721
	CALL DCOLE(L,X,U,DU,MIN,1)	CHA0722
	CALL DCOLE(L,X,P,DP,MIN,2)	CHA0723
	CALL DCOLE(L,X,RO,DRO,MIN,3)	CHA0724
	CALL DCOLE(L,X,Z,DZ,MIN,4)	CHA0725
200	CONTINUE	CHA0726
	CALL BDOX1(L,X,U,DU,MIN,1)	CHA0727
		CHA0728
		CHA0729
		CHA0730
		CHA0731

```

CALL BDOK1(L,X,P,DP,MIN,2)
CALL BDOK1(L,X,RO,DRO,MIN,3)
CALL BDOK1(L,X,Z,DZ,MIN,4)
PRINT 901,(NN,PRESS(NN),PULSE1(NN),PULSE2(NN),
1 PULSE3(NN)/AMTOT,PULSE4(NN)/TENTOT,NN=1,4)
901 FORMAT(1X,2(' ',I3,5D11.3,' '))
IF(DABS(T-A(5)).LT.1.D-6) PRINT 911,NERRP,ERRP
911 FORMAT(/1X,'NERRP,ERRP=',I5,D15.5/)
RETURN
7001 CONTINUE
PRINT 7101, I,ROAV,UAV,DRO(I),DU(I),E(I),EP,ZNEW,ZNEW-1.DO,EPI
7101 FORMAT(/1X,'FROM CYCEUL. NEGATIVE EP. IN CELL I=',I6//
1 IX,'ROAV,UAV,DRO(I),DU(I)=' ,4D18.8//
2 IX,'E(I),EP,ZNEW,ZNEW-1,EPI=' ,5D14.6//)
CALL PRINT
1 (L,X,U,P,RO,G,E,DU,DP,DRO,DG,DXSI,MIN,
2 US,PS,UIDOT,PIDOT,
3 FIMZ,ZMDOT,
4 TENA,FIRO,FIM,FIE,GIP,VOL,Z,DZ)
CALL SOF('CYCEUL 7001, NEGATIVE EP')
RETURN
END
SUBROUTINE SAFAE SAFAE
1 (L,X,U,P,RO,G,E,DU,DP,DRO,DG,DXSI,MIN,
2 US,PS,UIDOT,PIDOT,
3 FIMZ,ZMDOT,
4 TENA,FIRO,FIM,FIE,GIP,VOL,Z,DZ)
IMPLICIT REAL*8(A-H,O-Z,*)
DIMENSION X(L),U(L),P(L),RO(L),G(L),E(L),DU(L),DP(L),DRO(L),
1 DG(L),DXSI(L),MIN(L),
2 US(L),PS(L),UIDOT(L),PIDOT(L)
3 ,TENA(L),FIRO(L),FIM(L),FIE(L)
4 ,GIP(L),VOL(L),Z(L),DZ(L)
5 ,FIMZ(L),ZMDOT(L)
COMMON /AB/A(50)
EQUIVALENCE (LL,A(2)),(T,A(3)),(DT,A(4)),(NCYC,A(12))
EQUIVALENCE (UGAL,A(15))
COMMON /GAM/GAMA,NG,MU2,G1,G2,G3,G4,G5,G6,G7,G8,G9,G10,G11
1 ,G12,G13,G14,G15,G16,G17,G18,G19,G20,G21,G22,G23
2 ,G24,G25,G26,G27,G28,G29,G30,G31,G32,G33,G34,G35
REAL*8 NG,MU2
COMMON/DETO/QDET,PCJDET,RCJDET,UCJDET,DCJDET,PODET,ROODET,
1 RATE,TEMPC
COMMON/DIFFUS/U2,P2,RO2,ARW
C*****
C RIGID B.C. AT I=2
C
U(1)=-U(2)
P(1)=P(2)
G(1)=G(2)
RO(1)=RO(2)
Z(1)=Z(2)
DU(1)=DU(2)
DP(1)=-DP(2)
DG(1)=-DG(2)
DRO(1)=-DRO(2)
DXSI(1)=DXSI(2)
C
C OUTFLOW B.C. AT I=L
C
U(L)=U(LL)+DU(LL)/2.DO
P(L)=P(LL)+DP(LL)/2.DO
RO(L)=RO(LL)+DRO(LL)/2.DO
G(L)=G(LL)+DG(LL)/2.DO
Z(L)=Z(LL)+DZ(LL)/2.DO
DU(L)=0.
DP(L)=0.
DG(L)=0.
DRO(L)=0.
DZ(L)=0.
DXSI(L)=DXSI(LL)

```

C

	RETURN	CHA0804
	END	CHA0805
	SUBROUTINE BDOK1(L,X,V,DV,MIN,NV)	CHA0806
	IMPLICIT REAL*8(A-H,O-Z,*)	CHA0807
	DIMENSION X(L),V(L),DV(L),MIN(L)	CHA0808
	COMMON /AB/A(50)	CHA0809
	EQUIVALENCE (LL,A(2)),(KEYMON,A(11))	CHA0810
	COMMON /DRAW/GODELX,GODELY,UMIN,UMAX,PMIN,PMAX,ROMIN,ROMAX	CHA0811
1	,XMIN,XMAX,SMIN,SMAX,IVERSA	CHA0812
1	COMMON /MONIT/CASEAV(4),NC14(4),NF16(6),	CHA0813
1	NMONU(4),NMONP(4),NMONRO(4),NMONZ(4)	CHA0814
	DIMENSION NMONV(4,4)	CHA0815
	EQUIVALENCE (NMONV(1,1),NMONU(1))	CHA0816
	DIMENSION NAMEV(4)	CHA0817
	DATA NAMEV/'U','P','RO','Z'/	CHA0818
	DATA EPS/1.D-9/	CHA0819
C*****		CHA0820
	GO TO (1,2,3,4), NV	CHA0821
1	AMIDA=(UMAX-UMIN)**2	CHA0822
	GO TO 9	CHA0823
2	AMIDA=(PMAX-PMIN)**2	CHA0824
	GO TO 9	CHA0825
3	AMIDA=(ROMAX-ROMIN)**2	CHA0826
	GO TO 9	CHA0827
4	AMIDA=1.D0	CHA0828
	GO TO 9	CHA0829
9	CONTINUE	CHA0830
	AMIDA=AMIDA*EPS**2	CHA0831
	EPSA=DSQRT(AMIDA)	CHA0832
	DO 29 I=2,LL	CHA0833
	ICAT=0	CHA0834
	IF(DABS(DV(I)).LE.EPSA) DV(I)=0.	CHA0835
	IF(DV(I).EQ.0.) GO TO 29	CHA0836
	VLEFT=V(I)-0.5D0*DV(I)	CHA0837
	VRIGHT=V(I)+0.5D0*DV(I)	CHA0838
	VM=V(I-1)	CHA0839
	VP=V(I+1)	CHA0840
	SIGN=(VP-V(I))*(V(I)-VM)	CHA0841
	IF(SIGN.GT.-AMIDA) GO TO 22	CHA0842
21	DV(I)=0.	CHA0843
	ICAT=1	CHA0844
	GO TO 20	CHA0845
22	CONTINUE	CHA0846
	SIGN=(VP-VM)*DV(I)	CHA0847
	IF(SIGN.GT.-AMIDA) GO TO 24	CHA0848
23	DV(I)=0.5D0*(VP-VM)	CHA0849
	VLEFT=V(I)-0.5D0*DV(I)	CHA0850
	VRIGHT=V(I)+0.5D0*DV(I)	CHA0851
	ICAT=2	CHA0852
24	SIGN=(VLEFT-VM)*DV(I)	CHA0853
	IF(SIGN.GT.-AMIDA) GO TO 26	CHA0854
25	VLEFT=VM	CHA0855
	VRIGHT=2.D0*V(I)-VLEFT	CHA0856
	DV(I)=VRIGHT-VLEFT	CHA0857
	ICAT=3	CHA0858
26	SIGN=(VP-VRIGHT)*DV(I)	CHA0859
	IF(SIGN.GT.-AMIDA) GO TO 28	CHA0860
27	VRIGHT=VP	CHA0861
	VLEFT=2.D0*V(I)-VRIGHT	CHA0862
	DV(I)=VRIGHT-VLEFT	CHA0863
	ICAT=3	CHA0864
28	IF(DABS(DV(I)).LE.0.5D0*DABS(VP-VM)) GO TO 31	CHA0865
30	DV(I)=0.5D0*(VP-VM)	CHA0866
	ICAT=4	CHA0867
31	CONTINUE	CHA0868
20	CONTINUE	CHA0869
	IF(DABS(DV(I)).GT.EPSA) GO TO 40	CHA0870
	DV(I)=0.	CHA0871
40	CONTINUE	CHA0872
	IF(ICAT.GT.0) NMONV(ICAT,NV)=NMONV(ICAT,NV)+1	CHA0873
29	CONTINUE	CHA0874
		CHA0877

BDOK1

```

RETURN
END
SUBROUTINE DCOLE(L,X,V,DV,MIN,NV)
IMPLICIT REAL*8(A-H,O-Z,*)
DIMENSION X(L),V(L),DV(L),MIN(L)
COMMON /AB/A(50)
EQUIVALENCE (LL,A(2))
C*****
DO 1 I=2,LL
IM=I-1
IP=I+1
DV(I)=0.5D0*(V(IP)-V(IM))
1 CONTINUE
RETURN
END
SUBROUTINE PRINT
1 (L,X,U,P,RO,G,E,DU,DP,DRO,DG,DXSI,MIN,
2 US,PS,UIDOT,PIDOT,
3 FIMZ,ZMDOT,
4 TENA,FIRO,FIM,FIE,GIP,VOL,Z,DZ)
IMPLICIT REAL*8(A-H,O-Z,*)
DIMENSION X(L),U(L),P(L),RO(L),G(L),E(L),DU(L),DP(L),DRO(L),
1 DG(L),DXSI(L),MIN(L),
2 US(L),PS(L),UIDOT(L),PIDOT(L)
3 ,TENA(L),FIRO(L),FIM(L),FIE(L)
4 ,GIP(L),VOL(L),Z(L),DZ(L)
5 ,FIMZ(L),ZMDOT(L)
COMMON /TOT/AMTOT,ETOT,EKTOT,EPTOT,TENTOT
COMMON /STEP0/UL,PL,ROL,GL,UR,PR,ROR,GR,USTAR,PSTAR,
1 RSTARL,RSTARR,GSTARL,GSTARR,
2 CL,CR,CSTARL,CSTARR,SL,SR,WL,WR,UW(6)
3 ,LAMDAL,LAMDAR,RATEL,RATER,TEMPL,TEMPLR,TEMPSL,TEMPSR
4 ,ZL,ZR,ZSTARL,ZSTARR,NFLUX,HELEML,HELEMR
REAL*8 LAMDAL,LAMDAR
LOGICAL HELEML,HELEMR
COMMON /AB/A(50)
EQUIVALENCE (LL,A(2)),(T,A(3)),(NCYC,A(12)),(DT,A(4))
EQUIVALENCE (UGAL,A(15))
C COMMON/DIFFUS/U2,P2,RO2,ARM
COMMON/DETO/QDET,PCJDET,RCJDET,UCJDET,DCJDET,P0DET,RO0DET,
1 RATE,TEMPC
COMMON /CAM/GAMA,NG,MU2,G1,G2,G3,G4,G5,G6,G7,G8,G9,G10,G11
1 ,G12,G13,G14,G15,G16,G17,G18,G19,G20,G21,G22,G23
2 ,G24,G25,G26,G27,G28,G29,G30,G31,G32,G33,G34,G35
REAL*8 NG,MU2
COMMON /MONIT/CASEAV(4),NC14(4),NF16(6),
1 NMONU(4),NMONP(4),NMONRO(4),NMONZ(4)
DIMENSION CASAV1(4)
LOGICAL FULLPR
C*****
FULLPR=.TRUE.
PRINT 1
1 FORMAT(1H1)
PRINT 2, T,DT,NCYC
2 FORMAT(1X,10X,'RESULTS AT T=',D11.5,5X,'DT=',D11.5,5X,'NCYC=',
1 15//)
PRINT 3, AMTOT,ETOT,EKTOT,EPTOT,TENTOT
3 FORMAT(1X,'AMTOT=',D20.14,2X,'ETOT,EKTOT,EPTOT=',3D22.14/
1 1X,'TENTOT=',D21.14//)
4 FORMAT(1X,' I', ' X', ' U', ' P', '
1 ' RO', ' G', ' Z', '
2 ' DU', ' DP', ' DRO', '
3 ' DG', ' DZ', '
44 FORMAT(1X,' ' , ' US', ' PS', '
1 ' ZMDOT', ' FIMZ', ' AMDOT', '
2 ' AMDOTN', ' TEMP', ' ENTALP', '
3 ' AMACH', ' ENTRO', '
5 FORMAT(1X)
IF (UGAL.NE.0.) PRINT 6, UGAL
6 FORMAT(/11X,'INITIAL VELOCITY CORRESPONDS TO UGAL=',D15.6/)
DO 10 I=1,L
IF (MOD(I,10).NE.1) GO TO 11

```

```

      PRINT 5
      PRINT 4
      PRINT 44
      PRINT 5
11  CONTINUE
      PRINT 12,I,X(I),U(I),P(I),RO(I),G(I),Z(I),DU(I),DP(I),DRO(I),
1  DG(I),DZ(I)
12  FORMAT(1X,I3,6D12.5,5D11.4)
      ENTRO=P(I)/RO(I)*GAMA
      IF(.NOT.FULLPR) GO TO 131
      IF(I.EQ.1) GO TO 131
      IM=I-1
      UL=U(IM)+0.5*DU(IM)
      PL=P(IM)+0.5*DP(IM)
      ROL=RO(IM)+0.5*DRO(IM)
      OL=G(IM)+0.5*DG(IM)
      CL=OL/ROL
      ZL=Z(IM)+0.5*DZ(IM)
      IF(ZL.LT.0.) ZL=0.
      UR=U(I)-0.5*DU(I)
      PR=P(I)-0.5*DP(I)
      OR=G(I)-0.5*DG(I)
      ROR=RO(I)-0.5*DRO(I)
      CR=OR/ROR
      ZR=Z(I)-0.5*DZ(I)
      IF(ZR.LT.0.) ZR=0.
      IF(PL.LE.0.) PL=1.D-8
      IF(PR.LE.0.) PR=1.D-8
      CALL RIEMAN(L,I,MIN)
      XI=X(I)
      RSTAR=RSTARL
      IF(USTAR.LT.0.) RSTAR=RSTARR
      ZSTAR=ZL
      IF(USTAR.LY.0.) ZSTAR=ZR
      AMACH=USTAR/DSQRT(GAMA*PSTAR/RSTAR)
      AMDOT=RSTAR*USTAR*CROSS(XI)
      IF(I.NE.2) GO TO 132
      AMDOT0=AMDOT
      IF(DABS(AMDOT0).LT.1.D-12) AMDOT0=1.D0
132 CONTINUE
      AMDOTN=AMDOT/AMDOT0
      ENTALP=(GAMA/(GAMA-1.D0))*PSTAR/RSTAR+0.5D0*USTAR**2+QDET*ZSTAR
      ARW=1.D0
      TEMP=PSTAR/(RSTAR*ARW)
      PRINT 13,US(I),PS(I),
1  ZMDOT(I),FIMZ(I),AMDOT,AMDOTN,TEMP,ENTALP,AMACH,ENTRO
13  FORMAT(4X,12X,5D12.5,6D11.4)
131 CONTINUE
10  CONTINUE
C  JOB STATISTICS
      DO 40 I=1,4
      CASAV1(I)=0.
      IF (NC14(I).NE.0) CASAV1(I)=CASEAV(I)/DFLOAT(NC14(I))
40  CONTINUE
      PRINT 30
      FORMAT(//1X,10('*'),3X,'JOB STATISTICS',3X,10('*'))//
      PRINT 31,(NC14(I),I=1,4)
      PRINT 31, 'NO. OF VARIOUS CASES IN RIEMAN SOLVER  NC14(NCASE)=' ,
1  4I10)
      PRINT 301, (CASAV1(I),I=1,4)
301  FORMAT(/1X,'AVERAGE NUMBER OF ITERATIONS IN RIEMAN SOLVER',
1  1X,' CASAV1(NCASE)=' ,4(F6.2,4X))
      PRINT 32,(NF16(I),I=1,6)
32  FORMAT(/1X,'NO. OF VARIOUS FLUX CASES  NF16(NFLUX)=' ,6I10)
      ICAT0=4
      PRINT 33,(NMONU(I),I=1,ICAT0),(NMONP(I),I=1,ICAT0),
1  (NMONRO(I),I=1,ICAT0),(NMONZ(I),I=1,ICAT0)
33  FORMAT(/1X,'NO. OF MONOTONICITY INTERVENTIONS FOR EACH VAR.',
1  1X,'IN EACH CATEGORY.'/
1  1X,'NMONU (ICAT)=' ,4I10/
1  1X,'NMONP (ICAT)=' ,4I10/
1  1X,'NMONRO (ICAT)=' ,4I10/

```

CHA0950
 CHA0951
 CHA0952
 CHA0953
 CHA0954
 CHA0955
 CHA0956
 CHA0957
 CHA0958
 CHA0959
 CHA0960
 CHA0961
 CHA0962
 CHA0963
 CHA0964
 CHA0965
 CHA0966
 CHA0967
 CHA0968
 CHA0969
 CHA0970
 CHA0971
 CHA0972
 CHA0973
 CHA0974
 CHA0975
 CHA0976
 CHA0977
 CHA0978
 CHA0987
 CHA0988
 CHA0989
 CHA0990
 CHA0991
 CHA0992
 CHA0993
 CHA0994
 CHA0995
 CHA0996
 CHA0997
 CHA0998
 CHA0999
 CHA1000
 CHA1001
 CHA1002
 CHA1003
 CHA1004
 CHA1005
 CHA1006
 CHA1007
 CHA1008
 CHA1009
 CHA1010
 CHA1011
 CHA1012
 CHA1013
 CHA1014
 CHA1015
 CHA1016
 CHA1017
 CHA1018
 CHA1019
 CHA1020
 CHA1021
 CHA1022
 CHA1023
 CHA1024
 CHA1025
 CHA1026
 CHA1027
 CHA1028
 CHA1029

1	IX, 'NMONZ (ICAT)=' ,4I10/)	CHA1030
	RETURN	CHA1031
	END	CHA1032
	SUBROUTINE SOF(ISTOP)	CHA1034
	IMPLICIT REAL*8(A-H,O-Z,*)	CHA1035
	DIMENSION ISTOP(1)	CHA1036
	PRINT 1, ISTOP	CHA1037
1	FORMAT(/1X,3H***,2X,20A4,3X,3H***//)	CHA1038
	PRINT 1	CHA1039
	XX=-1.D0	CHA1040
	YY=DSQRT(XX)	CHA1041
	STOP	CHA1042
	END	CHA1043
	SUBROUTINE RIEMAN(L,I,MIN)	CHA1310
	IMPLICIT REAL*8(A-H,O-Z,*)	CHA1311
	DIMENSION MIN(L)	CHA1312
	COMMON /STEP0/UL,PL,ROL,GL,UR,PR,ROR,GR,USTAR,PSTAR,	CHA1313
1	RSTARL,RSTARR,GSTARL,GSTARR,	CHA1314
2	CL,CR,CSTARL,CSTARR,SL,SR,WL,WR,UW(6)	CHA1315
3	,LAMDAL,LAMDAR,RATEL,RATER,TEMPL,TEMPSR	CHA1316
4	,ZL,ZR,ZSTARL,ZSTARR,NFLUX,HELEML,HELEMR	CHA1317
	REAL*8 LAMDAL,LAMDAR	CHA1318
	LOGICAL HELEML,HELEMR	CHA1319
	COMMON /STEP1/DUIDT,DPIDT,DGIDTL,DGIDTR,DRIDTL,DRIDTR	CHA1320
2	,ASTARL,ASTARR,LAMDSL,LAMDSR,DSDAL,DSDAR,DZDAL,DZDAR	CHA1321
3	,RAT,SH	CHA1322
4	,BETACL,BETACR,DSDASL,DSDASR,DZDASL,DZDASR	CHA1323
	REAL*8 LAMDSL,LAMDSR,DSDAL,DSDAR,DZDAL,DZDAR	CHA1324
	COMMON /DRAW/GODELX,GODELY,UMIN,UMAX,PMIN,PMAX,ROMIN,ROMAX	CHA1325
1	,XMIN,XMAX,SMIN,SMAX,IVERSA	CHA1326
	COMMON /GAM/GAMA,NG,MU2,G1,G2,G3,G4,G5,G6,G7,G8,G9,G10,G11	CHA1327
1	,G12,G13,G14,G15,G16,G17,G18,G19,G20,G21,G22,G23	CHA1328
2	,G24,G25,G26,G27,G28,G29,G30,G31,G32,G33,G34,G35	CHA1329
	REAL*8 NG,MU2	CHA1330
	COMMON /AB/A(50)	CHA1331
	COMMON /MONIT/CASEAV(4),NC14(4),NF16(6),	CHA1332
1	NMONU(4),NMONP(4),NMONRO(4),NMONZ(4)	CHA1333
	C*****	CHA1334
	DATA NMAX/63/	CHA1335
	DATA EPS/1.D-8/	CHA1336
	DATA NTRY/0/	CHA1337
	C*****	CHA1338
	UW(6)=1.D20	CHA1339
	WL=0.	CHA1340
	WR=0.	CHA1341
	ZETAL=PL*G8	CHA1342
	ZETAR=PR*G8	CHA1343
	CLG=CL/GAMA	CHA1344
	CRG=CR/GAMA	CHA1345
	ZSTARL=ZL	CHA1346
	ZSTARR=ZR	CHA1347
	IF (ZETAL.LT.ZETAR) GO TO 102	CHA1348
C	LEFT PRESSURE IS HIGHER	CHA1349
101	CONTINUE	CHA1350
	EVERR=(PL-PR)/PR	CHA1351
	USR=UR+CRG*EVERR/DSQRT(1.D0+G6*EVERR)	CHA1352
	SRR=USR	CHA1353
	UEL=UL-G7*CL*(ZETAR-ZETAL)/ZETAL	CHA1354
	SLL=UEL	CHA1355
	NL=2	CHA1356
	NR=2	CHA1357
	IF (USR.GE.UL) NL=1	CHA1358
	IF (UEL.LE.UR) NR=1	CHA1359
	IF (DABS(EVERR).LT.EPS) GO TO 100	CHA1360
	IF (NL.EQ.2.AND.NR.EQ.1) GO TO 7001	CHA1361
	GO TO 100	CHA1362
C	RIGHT PRESSURE IS HIGHER	CHA1363
102	CONTINUE	CHA1364
	EVERL=(PR-PL)/PL	CHA1365
	USL=UL-CLG*EVERL/DSQRT(1.D0+G6*EVERL)	CHA1366
	SLL=USL	CHA1367
	UER=UR+G7*CR*(ZETAL-ZETAR)/ZETAR	CHA1368

```

      SRR=UER
      NL=2
      NR=2
      IF (UER.GE.UL) NL=1
      IF (USL.LE.UR) NR=1
      IF (DABS(EVERL).LT.EPS) GO TO 100
      IF (NL.EQ.1.AND.NR.EQ.2) GO TO 7001
      GO TO 100
100  CONTINUE
      IF (NL.EQ.1.AND.NR.EQ.2) NCASE=1
      IF (NL.EQ.2.AND.NR.EQ.2) NCASE=2
      IF (NL.EQ.2.AND.NR.EQ.1) NCASE=3
      IF (NL.EQ.1.AND.NR.EQ.1) NCASE=4
      IF (DABS(PI-PR)+DABS(UL-UR).LT.EPS*(PMAX-UMIN)) NCASE=4
      UMIDA=EPS*DMAX1(CL,CR)
      DUDZL=-G7*CL/ZETAL
      DUDZR= G7*CR/ZETAR
      ZETA=(-(UR-UL)+ZETAR*DUDZR-ZETAL*DUDZL)/(DUDZR-DUDZL)
      IF (ZETA.LE.0.) GO TO 7002
      N=0
      GO TO (1,2,3,4), NCASE
C   THE CASE ES
1   ITYPE=NCASE
   HELEML=.FALSE.
   HELEMR=.TRUE.
11  N=N+1
   IF (N.GT.NMAX) GO TO 7003
   ZETAF=ZETA
   UEL=UL-G7*CL*(ZETAF-ZETAL)/ZETAL
   PPR=(ZETAF/ZETAR)**NG
   EVERR=PPR-1.DO
   SQRR=DSQRT(1.DO+G6*EVERR)
   USR=UR+CRG*EVERR/SQRR
   DU=UEL-USR
   IF (DABS(DU).LE.UMIDA) GO TO 10
   DUDZR=NG*CRG*(PPR/ZETAF)*(1.DO+G9*EVERR)/SQRR**3
   ZETA=ZETAF+DU/(DUDZR-DUDZL)
   GO TO 11
10  CONTINUE
   USTAR=(UEL+USR)/2.DO
   IF (DABS(USTAR).LT.EPS*UMAX) USTAR=0.
   PSTAR=PPR*PR
   CSTARL=CL+(UL-USTAR)/G7
   RSTARL=GAMA*PSTAR/CSTARL**2
   GSTARL=CSTARL*RSTARL
C   EQU. NO. 69.01 OF THE BOOK BY COURANT-FRIEDRICHS.
   WWR=G11*(USTAR-UR)*ROR
   WR=WWR+DSQRT(GR**2+WWR**2)
   RSTARR=ROR*WR/(WR-ROR*(USTAR-UR))
   GSTARR=DSQRT(GAMA*PSTAR*RSTARR)
   CSTARR=GSTARR/RSTARR
   WRE=WR/ROR+UR
   UW(1)=UL-CL
   UW(2)=USTAR-CSTARL
   UW(3)=USTAR
   UW(4)=WRE
   UW(5)=WRE
   GO TO 5
C   THE CASE SS
2   ITYPE=NCASE
   HELEML=.TRUE.
   HELEMR=.TRUE.
21  N=N+1
   IF (N.GT.NMAX) GO TO 7003
   ZETAF=ZETA
   PF=ZETAF**NG
   PPL=PF/PL
   PPR=PF/PR
   EVERL=PPL-1.DO
   EVERR=PPR-1.DO
   SQRL=DSQRT(1.DO+G6*EVERL)
   SQRR=DSQRT(1.DO+G6*EVERR)

```

CHA1369
 CHA1370
 CHA1371
 CHA1372
 CHA1373
 CHA1374
 CHA1375
 CHA1376
 CHA1377
 CHA1378
 CHA1379
 CHA1380
 CHA1381
 CHA1382
 CHA1383
 CHA1384
 CHA1385
 CHA1386
 CHA1387
 CHA1388
 CHA1389
 CHA1390
 CHA1391
 CHA1392
 CHA1393
 CHA1394
 CHA1395
 CHA1396
 CHA1397
 CHA1398
 CHA1399
 CHA1400
 CHA1401
 CHA1402
 CHA1403
 CHA1404
 CHA1405
 CHA1406
 CHA1407
 CHA1408
 CHA1409
 CHA1410
 CHA1411
 CHA1412
 CHA1413
 CHA1414
 CHA1415
 CHA1416
 CHA1417
 CHA1418
 CHA1419
 CHA1420
 CHA1421
 CHA1422
 CHA1423
 CHA1424
 CHA1425
 CHA1426
 CHA1427
 CHA1428
 CHA1429
 CHA1430
 CHA1431
 CHA1432
 CHA1433
 CHA1434
 CHA1435
 CHA1436
 CHA1437
 CHA1438
 CHA1439
 CHA1440

```

USL=UL-CLG*EVERL/SQRL
USR=UR+CRG*EVERR/SQRR
DU=USL-USR
IF (DABS(DU).LE.UMIDA) GO TO 20
DUDZL=-NG*CLG*(PPL/ZETAF)*(1.D0+G9*EVERL)/SQRL**3
DUDZR= NG*CRG*(PPR/ZETAF)*(1.D0+G9*EVERR)/SQRR**3
ZETA=ZETAF+DU/(DUDZR-DUDZL)
GO TO 21
20 CONTINUE
USTAR=(USL+USR)/2.D0
IF(DABS(USTAR).LT.EPS*UMAX) USTAR=0.
PSTAR=(PPL*PL+PPR*PR)/2.D0
WNR=G11*(USTAR-UR)*ROR
WR=WNR+DSQRT(GR**2+WNR**2)
WWL=-G11*(USTAR-UL)*ROL
WL=WWL+DSQRT(GL**2+WWL**2)
RSTARL=ROL*WL/(WL+ROL*(USTAR-UL))
RSTARR=ROR*WR/(WR-ROR*(USTAR-UR))
GSTARL=DSQRT(GAMA*PSTAR*RSTARL)
GSTARR=DSQRT(GAMA*PSTAR*RSTARR)
CSTARL=GSTARL/RSTARL
CSTARR=GSTARR/RSTARR
WLE=-WL/ROL+UL
WRE=WR/ROR+UR
UW(1)=WLE
UW(2)=WLE
UW(3)=USTAR
UW(4)=WRE
UW(5)=WRE
GO TO 5
C THE CASE SE
3 ITYPE=NCASE
HELEML=.TRUE.
HELEMR=.FALSE.
31 N=N+1
IF (N.GT.NMAX) GO TO 7003
ZETAF=ZETA
UER=UR+G7*CR*(ZETAF-ZETAR)/ZETAR
PPL=(ZETAF/ZETAL)**NG
EVERL=PPL-1.D0
SQRL=DSQRT(1.D0+G6*EVERL)
USL=UL-CLG*EVERL/SQRL
DU=USL-UER
IF (DABS(DU).LE.UMIDA) GO TO 30
DUDZL=-NG*CLG*(PPL/ZETAF)*(1.D0+G9*EVERL)/SQRL**3
ZETA=ZETAF+DU/(DUDZR-DUDZL)
GO TO 31
30 CONTINUE
USTAR=(USL+UER)/2.D0
IF(DABS(USTAR).LT.EPS*UMAX) USTAR=0.
PSTAR=PPL*PL
CSTARR=CR-(UR-USTAR)/G7
RSTARR=GAMA*PSTAR/CSTARR**2
GSTARR=CSTARR*RSTARR
WWL=-G11*(USTAR-UL)*ROL
WL=WWL+DSQRT(GL**2+WWL**2)
WLE=-WL/ROL+UL
RSTARL=ROL*WL/(WL+ROL*(USTAR-UL))
GSTARL=DSQRT(GAMA*PSTAR*RSTARL)
CSTARL=GSTARL/RSTARL
UW(1)=WLE
UW(2)=WLE
UW(3)=USTAR
UW(4)=USTAR+CSTARR
UW(5)=UR+CR
GO TO 5
C THE CASE EE
4 ITYPE=NCASE
HELEML=.FALSE.
HELEMR=.FALSE.
PSTAR=ZETA**NG
USTAR=UL-G7*CL*(ZETA-ZETAL)/ZETAL

```

CHA1441
CHA1442
CHA1443
CHA1444
CHA1445
CHA1446
CHA1447
CHA1448
CHA1449
CHA1450
CHA1451
CHA1452
CHA1453
CHA1454
CHA1455
CHA1456
CHA1457
CHA1458
CHA1459
CHA1460
CHA1461
CHA1462
CHA1463
CHA1464
CHA1465
CHA1466
CHA1467
CHA1468
CHA1469
CHA1470
CHA1471
CHA1472
CHA1473
CHA1474
CHA1475
CHA1476
CHA1477
CHA1478
CHA1479
CHA1480
CHA1481
CHA1482
CHA1483
CHA1484
CHA1485
CHA1486
CHA1487
CHA1488
CHA1489
CHA1490
CHA1491
CHA1492
CHA1493
CHA1494
CHA1495
CHA1496
CHA1497
CHA1498
CHA1499
CHA1500
CHA1501
CHA1502
CHA1503
CHA1504
CHA1505
CHA1506
CHA1507
CHA1508
CHA1509
CHA1510
CHA1511
CHA1512

```

      IF(DABS(USTAR).LT.EPS*UMAX) USTAR=0.
      CSTARL=CL+(UL-USTAR)/G7
      CSTARR=CR-(UR-USTAR)/G7
      RSTARL=GAMA*PSTAR/CSTARL**2
      RSTARR=GAMA*PSTAR/CSTARR**2
      GSTARL=RSTARL*CSTARL
      GSTARR=RSTARR*CSTARR
      UW(1)=UL-CL
      UW(2)=USTAR-CSTARL
      UW(3)=USTAR
      UW(4)=USTAR+CSTARR
      UW(5)=UR+CR
      N=1
      GO TO 5
5     CONTINUE
      DO 6 K=1,6
      NFLUX=K
      IF (UW(K).GE.0.) GO TO 61
6     CONTINUE
      NFLUX=6
61    CONTINUE
      NC14(NCASE)=NC14(NCASE)+1
      CASEAV(NCASE)=CASEAV(NCASE)+DFLOAT(N)
      NF16(NFLUX)=NF16(NFLUX)+1
      IF(NTRY.CE.2)GO TO 666
      IF(I.NE.2.AND.I.NE.L) GO TO 666
      PRINT 667,I,NFLUX,NCASE,PL,UL,ROL,PR,UR,ROR,USTAR,PSTAR,RSTARL,
1     RSTARR,(KK,UW(KK),KK=1,6)
667   FORMAT(/1X,'I,NFLUX,NCASE=',3I5/1X,'PL,UL,ROL,PR,UR,ROR=',6D12.4/
1     1X,'USTAR,PSTAR,RSTARL,RSTARR=',4D13.4/
2     1X,'KK,UW(KK)='',6(I4,2X,D13.4)/)
      NTRY=NTRY+1
666   CONTINUE
      RETURN
7001  CONTINUE
      PRINT 7101, PL,UL,PR,UR,ZETAL,ZETAR,SLL,SRR,NL,NR,I
7101  FORMAT(/1X,'FROM RIEMAN. AN IMPOSSIBLE CASE OF EXPANSION/SHOCK'
1     //1X,'PL,UL,PR,UR=',4D25.14//
2     1X,'ZETAL,ZETAR,SLL,SRR=',4D25.14//
3     1X,'NL,NR,I='',3I10//)
      CALL SOF('7001')
7002  CONTINUE
      PRINT 7102, ZETA,DUDZL,DUDZR,ZETAL,ZETAR,PL,UL,PR,UR,N,NCASE,I
7102  FORMAT(/1X,'FROM RIEMAN. NEGATIVE PRESSURE AT THE INTERSECTION',
1     1X,'OF L AND R EXPANSION BRANCHES'//
2     1X,'IT MEANS THAT A CAVITATION TENDS TO FORM. THIS',
3     1X,'POSSIBILITY IS EXCLUDED IN PRESENT VERSION'//
4     1X,'ZETA,DUDZL,DUDZR,ZETAL,ZETAR,PL,UL,PR,UR=',9D10.3//
5     1X,'N,NCASE,I='',3I10//)
      CALL SOF('7002')
7003  CONTINUE
      PRINT 7103, I,N,NCASE,DU,UMIDA,EPS,PL,UL,PR,UR,
1     ZETA,ZETAF,ZETAL,ZETAR,DUDZL,DUDZR
7103  FORMAT(/1X,'FROM RIEMAN. NUMBER OF ITERATIONS EXCEEDED.'//
1     1X,'I,N,NCASE,DU,UMIDA,EPS=',3I6,3D18.6//
2     1X,'PL,UL,PR,UR,ZETA,ZETAF=',6D18.10//
3     1X,'ZETAL,ZETAR,DUDZL,DUDZR=',4D18.10//)
      CALL SOF('7003')
      RETURN
      END
C$OPTIONS LIST
      SUBROUTINE MAGA(L,I,MIN)
      IMPLICIT REAL*8(A-H,O-Z,$)
      DIMENSION MIN(L)
      COMMON /GAM/GAMA,NG,MU2,G1,G2,G3,G4,G5,G6,G7,G8,G9,G10,G11
1     ,G12,G13,G14,G15,G16,G17,G18,G19,G20,G21,G22,G23
2     ,G24,G25,G26,G27,G28,G29,G30,G31,G32,G33,G34,G35
      REAL*8 NG,MU2
      COMMON/DETO/QDET,PCJDET,RCJDET,UCJDET,DCJDET,PODET,ROODET,
1     RATE,TEMPC
      COMMON /STEP0/UL,PL,ROL,GL,UR,PR,ROR,GR,USTAR,PSTAR,
1     RSTARL,RSTARR,GSTARL,GSTARR,

```

CHA1513
 CHA1514
 CHA1515
 CHA1516
 CHA1517
 CHA1518
 CHA1519
 CHA1520
 CHA1521
 CHA1522
 CHA1523
 CHA1524
 CHA1525
 CHA1526
 CHA1527
 CHA1528
 CHA1529
 CHA1530
 CHA1531
 CHA1532
 CHA1533
 CHA1537
 CHA1538
 CHA1539
 CHA1540
 CHA1541
 CHA1542
 CHA1543
 CHA1544
 CHA1545
 CHA1546
 CHA1547
 CHA1548
 CHA1549
 CHA1550
 CHA1551
 CHA1552
 CHA1553
 CHA1554
 CHA1555
 CHA1556
 CHA1557
 CHA1558
 CHA1559
 CHA1560
 CHA1561
 CHA1562
 CHA1563
 CHA1564
 CHA1565
 CHA1566
 CHA1567
 CHA1568
 CHA1569
 CHA1570
 CHA1571
 CHA1572
 CHA1573
 CHA1574
 CHA1575
 CHA1576
 CHA1577
 CHA1578
 CHA1579
 CHA1580
 CHA1581
 CHA1582
 CHA1583
 CHA1584
 CHA1585
 CHA1586
 CHA1587

MAGA

```

2      CL,CR,CSTARL,CSTARR,SL,SR,WL,WR,UW(6)
3      ,LAMDAL,LAMDAR,RATEL,RATER,TEMPL,TEMPLR,TEMPSL,TEMPSR
4      ,ZL,ZR,ZSTARL,ZSTARR,NFLUX,HELEML,HELEMR
      REAL*8 LAMDAL,LAMDAR
      LOGICAL HELEML,HELEMR
      COMMON /STEP1/DUIDT,DPIDT,DGIDTL,DGIDTR,DRIDTL,DRIDTR
2      ,ASTARL,ASTARR,LAMDSL,LAMDSR,DSDAL,DSDAR,DZDAL,DZDAR
3      ,RAT,SH
4      ,BETACL,BETACR,DSDASL,DSDASR,DZDASL,DZDASR
      REAL*8 LAMDSL,LAMDSR,DSDAL,DSDAR,DZDAL,DZDAR
      COMMON /GRADS/DUDXIL,DPDXIL,DGDYIL,DRDXIL,DZDXIL,DSDXIL,
1      DUDXIR,DPDXIR,DGDYIR,DRDXIR,DZDXIR,DSDXIR
      COMMON /AB/A(50)
      REAL*8 LU,LP,LRO,LLAMDA
      DATA EPS/1.D-6/
C*****
C WE HERE SOLVE FOR THE TIME-DERIVATIVES ALONG THE CONTACT SURFACE,
C NAMELY DUIDT,DPIDT. FROM THESE WE ALSO OBTAIN THE OTHER
C TIME-DERIVATIVES (SEE COMMON /STEP1/).
C WE COMPUTE THE COEFFICIENTS FOR TWO EQUATIONS FOR DUIDT,DPIDT. THESE
C ARE      AAL*DUIDT+BBL*DPIDT=DDL
C      AAR*DUIDT+BBR*DPIDT=DDR
C*****
      IF(SH.LE.EPS)RAT=0.
C
C LEFT SIDE OF CONTACT
C
      IF (.NOT.HELEML) GO TO 12
11  CONTINUE
C LEFT SHOCK
      DP=PSTAR-PL
      DU=USTAR-UL
      Z2=0.5D0/(PSTAR+MU2*PL)
      LU=DU*(0.5D0*ROL+MU2*Z2*GL**2)-GL**2/WL-WL
      LRO=-0.5D0*DP/ROL
      LP=-2.D0-MU2*Z2*DP
      AAL=2.D0-Z2*DP
      BBL=Z2*DU+WL/GSTARL**2+1.D0/WL
      DDL=LU*DUDXIL+LRO*DRDXIL+LP*DPDXIL
      DDL=DDL-WL*USTAR*RAT/RSTARL
1      +UL*RAT*(-GAMA*PL/WL+DU*(GAMA*PL*MU2*Z2+0.5D0))
      GO TO 10
12  CONTINUE
C LEFT REFRACTION
      A1=DUDXIL+DPDXIL/GL
      BETA=GSTARL/CL
      SQB=DSQRT(BETA)
      ASTARL=A1-(CL/(G15*SL))*DSDXIL*(BETA**G5-1.D0)
      AAL=1.D0
      BBL=1.D0/GSTARL
      DDL=-GSTARL*ASTARL/SQB
      DSDAL=DSDXIL
      DZDAL=DZDXIL
      DSDASL=DSDXIL*SQB
      DZDASL=DZDXIL*SQB
      GEOM=RAT*((GAMA-1.D0)*UL+2.D0*CL)*
1      (BETA**G13-1.D0)/(ROL*(GAMA-3.D0))
1      -4.D0*RAT*CL*(BETA**G14-1.D0)/(ROL*(3.D0*GAMA-5.D0))
      ASTARL=ASTARL-GEOM
      EVER1= GSTARL*GEOM/SQB
      EVER2=-RAT*USTAR*CSTARL
      DDL=DDL+EVER1+EVER2
      GO TO 10
10  CONTINUE
C
C RIGHT SIDE OF CONTACT
C
      IF (.NOT.HELEMR) GO TO 22
21  CONTINUE
C RIGHT SHOCK
      DP=PSTAR-PR
      DU=USTAR-UR

```

```

Z2=0.5D0/(PSTAR+MU2*PR)
LU=DU*(0.5D0*ROR+MU2*Z2*GR**2)+GR**2/WR+WR
LRO=-0.5D0*DP/ROR
LP=-2.D0-MU2*Z2*DP
AAK=2.D0-Z2*DP
BBR=Z2*DU-WR/GSTARR**2-1.D0/WR
DDR=LU*DUDXIR+LRO*DRDXIR+LP*DPDXIR
DDR=DDR+WR*USTAR*RAT/RSTARR
1  +UR*RAT*(GAMA*PR/WR+DU*(GAMA*PR*MU2*Z2+0.5D0))
22  GO TO 20
C  CONTINUE
C  RIGHT RAREFACTION
A1=DUDXIR-DPDXIR/GR
BETA=GSTARR/GR
SQB=DSQRT(BETA)
ASTARR=A1+(CR/(G1.5*SR))*DSDXIR*(BETA**G5-1.D0)
AAR=1.D0
BBR=-1.D0/GSTARR
DDR=GSTARR*ASTARR/SQB
DSDAR=DSDXIR
DZDAR=DZDXIR
DSDASR=DSDXIR*SQB
DZDASR=DZDXIR*SQB
GEOM=RAT*(-(GAMA-1.D0)*UR+2.D0*CR)*(BETA**G13-1.D0)
1  /(ROR*(GAMA-5.D0))
2  -4.D0*RAT*CR*(BETA**G14-1.D0)/(ROR*(3.D0*GAMA-5.D0))
ASTARR=ASTARR+GEOM
EVER1=GSTARR*GEOM/SQB
EVER2=RAT*USTAR*CSTARR
DDR=DDR+EVER1+EVER2
20  GO TO 20
C  CONTINUE
DET=AAL*BBR-AAR*BBL
DUIDT=(DDL*BBR-DDR*BBL)/DET
DPIDT=-((DDL*AAR-DDR*AAL)/DET)
DRIDTL=DPIDT/CSTARL**2
DRIDTR=DPIDT/CSTARR**2
RETURN
END
SUBROUTINE FLUXE(L,Y,MIN)
IMPLICIT REAL*8(A-H,O-Z,9)
DIMENSION MIN(L)
COMMON /AB/A(50)
EQUIVALENCE (DT,A(4)),(NCYC,A(12))
COMMON /GAM/GAMA,NG,MU2,G1,G2,G3,G4,G5,G6,G7,G8,G9,G10,G11
1  ,G12,G13,G14,G15,G16,G17,G18,G19,G20,G21,G22,G23
2  ,G24,G25,G26,G27,G28,G29,G30,G31,G32,G33,G34,G35
REAL*8 NG,MU2
COMMON /GRADS/DUDXIL,DPDXIL,DGDXIL,DRDXIL,DZDXIL,DSDXIL,
1  DUDXIR,DPDXIR,DGDXIR,DRDXIR,DZDXIR,DSDXIR
COMMON /STEP0/UL,PL,ROL,GL,UR,PR,ROR,GR,USTAR,PSTAR,
1  RSTARL,RSTARR,GSTARL,GSTARR,
2  CL,CR,CSTARL,CSTARR,SL,SR,WL,WR,UW(6)
3  ,LAMDAL,LAMDAR,RATEL,RATER,TEMPL,TEMPSL,TEMPSR
4  ,ZL,ZR,ZSTARL,ZSTARR,NFLUX,HELEML,HELEMR
REAL*8 LAMDAL,LAMDAR
LOGICAL HELEML,HELEMR
COMMON /STEP1/DUIDT,DPIDT,DGIDTL,DGIDTR,DRIDTL,DRIDTR
2  ,ASTARL,ASTARR,LAMDSL,LAMDSR,DSDAL,DSDAR,DZDAL,DZDAR
3  ,RAT,SH
4  ,BETACL,BETACR,DSDASL,DSDASR,DZDASL,DZDASR
REAL*8 LAMDSL,LAMDSR,DSDAL,DSDAR,DZDAL,DZDAR
COMMON /DETO/QDET,PCJDET,RCJDET,UCJDET,PODET,ROODET,
1  RATE,TEMPC
COMMON /FI/FIH1,FIH2,FIH3,UXN,PXN,GXN,ROXH,ZXN
1  ,GIH
2  ,FIH4,ZMDOTL,ZMDOTR
REAL*8 LAMDA0
C*****
C  RO,U,P,Z AND THEIR (XI,T) DERIVATIVES AT EULERIAN POINT X=X(I).
C*****
DT2=DT/2.D0

```

FLUXE

```

1      GO TO (1,2,3,4,5,6),NFLUX
CONTINUE
C
C      NFLUX=1.   LINE X=0 IS TO THE LEFT OF LEFT WAVE.
C
      UX=UL
      PX=PL
      ROX=ROL
      ZX=ZL
      GX=GL
      DUDXIX=DUDXIL
      DPDXIX=DPDXIL
      DRDXIX=DRDXIL
      DZDXIX=DZDXIL
      DUDTX=-DPDXIL
      DRODTX=-ROL**2*DUDXIL
      DPDTX=-GL**2*DUDXIL
      DRODTX=DRODTX-RAT*ROL*UL
      DPDTX=DRODTX*CL**2
      DZDTX=0.
      GO TO 9
      CONTINUE
6
C
C      NFLUX=6.   LINE X=0 IS TO THE RIGHT OF RIGHT WAVE.
C
      UX=UR
      PX=PR
      ROX=ROR
      ZX=ZR
      GX=GR
      DUDXIX=DUDXIR
      DPDXIX=DPDXIR
      DRDXIX=DRDXIR
      DZDXIX=DZDXIR
      DUDTX=-DPDXIR
      DPDTX=-GR**2*DUDXIR
      DRODTX=-ROR**2*DUDXIR
      DRODTX=DRODTX-RAT*ROR*UR
      DPDTX=DRODTX*CR**2
      DZDTX=0.
      GO TO 9
      CONTINUE
2
C
C      NFLUX=2.   SONIC CASE (LEFT).
C
      BETA0=(MU2*(UL/CL+G7))*((1.D0/MU2)
      SQB0=DSQRT(BETA0)
      A1=DUDXIL+DPDXIL/GL
      A0=A1-(CL/(G15*SL))*DSDXIL*(BETA0**G5-1.D0)
      EVER1=-((GAMA-1.D0)*UL+2.D0*CL)*(BETA0**G13-1.D0)/(GAMA-5.D0)
      EVER2=4.D0*CL*(BETA0**G14-1.D0)/(3.D0*GAMA-5.D0)
      EVER=(EVER1+EVER2)*RAT/ROL
      A0=(A0+EVER)
      DPDAIX=GL*BETA0*A0
      C0=MU2*(UL+G7*CL)
      IF(C0.LT.0.) CALL SOF('FLUXE 2.  C0 NEGATIVE.')
```

```

CHA1732
CHA1733
CHA1734
CHA1735
CHA1736
CHA1737
CHA1738
CHA1739
CHA1740
CHA1741
CHA1742
CHA1743
CHA1744
CHA1745
CHA1746
CHA1747
CHA1748
CHA1749
CHA1750
CHA1751
CHA1752
CHA1753
CHA1754
CHA1755
CHA1756
CHA1757
CHA1758
CHA1759
CHA1760
CHA1761
CHA1762
CHA1763
CHA1764
CHA1765
CHA1766
CHA1767
CHA1768
CHA1769
CHA1770
CHA1771
CHA1772
CHA1773
CHA1774
CHA1775
CHA1776
CHA1777
CHA1778
CHA1779
CHA1780
CHA1781
CHA1782
CHA1783
CHA1784
CHA1785
CHA1786
CHA1787
CHA1788
CHA1789
CHA1790
CHA1791
CHA1792
CHA1793
CHA1794
CHA1795
CHA1796
CHA1797
CHA1798
CHA1799
CHA1800
CHA1801
CHA1802
CHA1803
```

```

C      NFLUX=5.    SONIC CASE (RIGHT).
C
      BETA0=(MU2*(-UR/CR+G7))**(1.D0/MU2)
      SQB0=DSQRT(BETA0)
      A1=DUDXIR-DPDXIR/GR
      A0=A1+(CR/(G15*SR))*DSDXIR*(BETA0**G5-1.D0)
      EVER1=(-(GAMA-1.D0)*UR+2.D0*CR)*(BETA0**G13-1.D0)/(GAMA-3.D0)
      EVER2=-4.D0*CR*(BETA0**G14-1.D0)/(3.D0*GAMA-5.D0)
      EVER=(EVER1+EVER2)*RAT/ROR
      A0=(A0+EVER)
      DPDA=-GR*BETA0*A0
      C0=MU2*(-UR+G7*CR)
      IF(C0.LT.0.) CALL SOF('FLUXE 5.  CG NEGATIVE.')
      UX=-C0
      ROX=GR*BETA0/C0
      ZX=ZR
      PX=ROX*C0**2/GAMA
      GX=ROX*C0
      DPDA=DPDA-RAT*UX*C0*DSQRT(BETA0)
      DUDX=CR*BETA0**(-1.D0/G4)/G4
      DPDA=PR*BETA0**MU2/G6
      DRODBX=ROR*BETA0**(-MU2)/G4
      DSDAX=SQB0*DSDAR
      DZDAX=SQB0*DZDAR
      DRODAX=DPDA/C0**2-(ROX/(GAMA*SR))*DSDAX
      DUDAX=A0
      DGDAX=0.5D0*GAMA*(PX*DRODAX+ROX*DPDA)/GX
      GO TO 9
3     CONTINUE
C
C      NFLUX=3.    LINE X=0 IS BETWEEN THE LEFT WAVE AND THE CONTACT.
C
      UX=USTAR
      PX=PSTAR
      ROX=RSTARL
      ZX=ZL
      GX=GSTARL
      DUDXIX=-DPIDT/GSTARL**2
      DPDXIX=-DUIDT
      DUDXIX=DUDXIX-RAT*USTAR/RSTARL
      DZDXIX=DZDXIL
      DZDTX=0.
      IF (.NOT.HELEML) GO TO 32
31    CONTINUE
C LEFT SHOCK.
      DRDXIX=(RSTARL/WL)**2*(3.D0*DUIDT
1         +DPIDT*(1.D0+3.D0*(WL/GSTARL)**2)/WL
2         +DUDXIL*WL*((GL/WL)**2+3.D0)+3.D0*DPDXIL
3         +DRDXIL*(WL/ROL)**2)
      EVER1=UL*RSTARL**2*RAT*((GL/WL)**2+1.D0)/(ROL*WL)
      EVER2=2.D0*RSTARL*USTAR*RAT/WL
      DRDXIX=DRDXIX+EVER1+EVER2
      DRODTX=-DUDXIX*ROX**2
      GO TO 33
32    CONTINUE
      BETA=GSTARL/GL
      SQB=DSQRT(BETA)
      DPDA=ASTARL*GSTARL
      DPDA=GSTARL*(ASTARL+RAT*USTAR*CSTARL/(GL*  SQB))
      G41=1.D0/G4+0.5D0
      DRODA=(DRDXIL-DPDXIL/(CL*CL)) *BETA**G41+DPDA/(CSTARL**2)
      DRDXIX= DRODA/SQB+DPIDT/(GSTARL*CSTARL**2)
      DRODA=DPDA/CSTARL**2-(RSTARL/(GAMA*SL))*DSDASL
      DRODTX=-DUDXIX*ROX**2
      DRDXIX=DRODA/SQB+DRODTX/GSTARL
33    CONTINUE
      DUDTX=DUIDT
      DPDTX=DPIDT
      GO TO 9
4     CONTINUE
C

```

CHA1804
 CHA1805
 CHA1806
 CHA1807
 CHA1808
 CHA1809
 CHA1810
 CHA1811
 CHA1812
 CHA1813
 CHA1814
 CHA1815
 CHA1816
 CHA1817
 CHA1818
 CHA1819
 CHA1820
 CHA1821
 CHA1822
 CHA1823
 CHA1824
 CHA1825
 CHA1826
 CHA1827
 CHA1828
 CHA1829
 CHA1830
 CHA1831
 CHA1832
 CHA1833
 CHA1834
 CHA1835
 CHA1836
 CHA1837
 CHA1838
 CHA1839
 CHA1840
 CHA1841
 CHA1842
 CHA1843
 CHA1844
 CHA1845
 CHA1846
 CHA1847
 CHA1848
 CHA1849
 CHA1850
 CHA1851
 CHA1852
 CHA1853
 CHA1854
 CHA1855
 CHA1856
 CHA1857
 CHA1858
 CHA1859
 CHA1860
 CHA1861
 CHA1862
 CHA1863
 CHA1864
 CHA1865
 CHA1866
 CHA1867
 CHA1868
 CHA1869
 CHA1870
 CHA1871
 CHA1872
 CHA1873
 CHA1874
 CHA1875

C NFLUX=4, LINE X=0 IS BETWEEN THE CONTACT AND THE RIGHT WAVE.

```

C
  DFDXIX=-DUIDT
  UX=USTAR
  PX=FSTAR
  ROX=RSTARR
  ZX=ZR
  OX=GSTARR
  DUDXIX=-DPIDT/GSTARR**2
  DUDXIX=DUDXIX-RAT*USTAR/RSTARR
  DFDXIX=-DUIDT
  DZDXIX=DZDXIL
  DZDTX=0.
  IF (.NOT. HELEMR) GO TO 42
41 CONTINUE
C RIGHT SHOCK
  DRDXIX=(RSTARR/WR)**2*(3.*DUIDT
1    -DPIDT*(1.D0+3.D0*(WR/GSTARR)**2)/WR
2    -DUDXIX*WR*((OR/WR)**2+3.D0)+3.D0*DPDXIR
3    +DRDXIR*(WR/ROX)**2)
  EVER1=OR*RSTARR**2*RAT*((OR/WR)**2+1.D0)/(ROX*WR)
  EVER2=2.D0*RSTARR*USTAR*RAT/WR
  DRDXIX=DUDXIX*EVER1-EVER2
  DRODTX=-DUDXIX*ROX**2
  GO TO 43
42 CONTINUE
C RIGHT RAREFACTION
  BETA=GSTARR/OR
  SQB=DSQRT(BETA)
  DPDA=-ASTARR*GSTARR
  DPDA=-GSTARR*(ASTARR+RAT*USTAR*GSTARR/(OR*SQB))
  G41=1.D0/G4+0.5D0
  DRODA=(DRDXIR-DPDXIR/(CR*CR)) *BETA**G41+DPDA/(GSTARR**2)
  DRDXIX=DRODA/SQB-DPIDT/(GSTARR*GSTARR**2)
  DRODA=DPDA/GSTARR**2-(RSTARR/(GAMA*SR))*DSASR
  DRODTX=-DUDXIX*ROX**2
  DRDXIX=DRODA/SQB-DRODTX/GSTARR
43 CONTINUE
  DUDTX=DUIDT
  DPDTX=DPIDT
  GO TO 9
9 CONTINUE
C*****
C FLUXES CENTERED AT TIME T(N+1/2) AT EULERIAN POINT X=X(I).
C*****
  FI1=ROX*UX
  FI2=ROX*UX**2+PX
  FI2=FI2-PX
  FI3=UX*(G12*PX+0.5D0*ROX*UX**2)
  FI4=ZX*ROX*UX
  FI3=FI3+QDET*FI4
  ROU00=ROX*UX
  GO TO(10,20,30,40,50,60), NFLUX
10 CONTINUE
60 CONTINUE
  DFDXI1=DRDXIX*UX+ROX*DUDXIX
  DFDXI2=DRDXIX*UX**2+2.D0*ROX*UX*DUDXIX+D*DXIX
  DFDXI2=DFDXI2-DPDXIX
  DFDXI5=DUDXIX*(G12*PX+0.5D0*ROX*UX**2)
1  +UX*(G12*DPDXIX+0.5D0*DRDXIX*UX**2+ROX*UX*DUDXIX)
  DFDXI4=ZX*DFDXI1+ROX*UX*DZDXIX
  DFDXI3=DFDXI3+QDET*DFDXI4
  DFIDT1=DRODTX*UX+ROX*DUDTX
  DFIDT2=DRODTX*UX**2+2.D0*ROX*UX*DUDTX+DPDTX
  DFIDT2=DFIDT2-DPDTX
  DFIDT3=DUDTX*(G12*PX+0.5D0*ROX*UX**2)
1  +UX*(G12*DPDTX+0.5D0*DRODTX*UX**2+ROX*UX*DUDTX)
  DFIDT4=ZX*DFIDT1+ROX*ZX*DZDTX
  DFIDT3=DFIDT3+QDET*DFIDT4
  FIDOT1=-ROU00*DFDXI1+DFIDT1
  FIDOT2=-ROU00*DFDXI2+DFIDT2
  FIDOT3=-ROU00*DFDXI3+DFIDT3

```

CHA1876
CHA1877
CHA1878
CHA1879
CHA1880
CHA1881
CHA1882
CHA1883
CHA1884
CHA1885
CHA1886
CHA1887
CHA1888
CHA1889
CHA1890
CHA1891
CHA1892
CHA1893
CHA1894
CHA1895
CHA1896
CHA1897
CHA1898
CHA1899
CHA1900
CHA1901
CHA1902
CHA1903
CHA1904
CHA1905
CHA1906
CHA1907
CHA1908
CHA1909
CHA1910
CHA1911
CHA1912
CHA1913
CHA1914
CHA1915
CHA1916
CHA1917
CHA1918
CHA1919
CHA1920
CHA1921
CHA1922
CHA1923
CHA1924
CHA1925
CHA1926
CHA1927
CHA1928
CHA1929
CHA1930
CHA1931
CHA1932
CHA1933
CHA1934
CHA1935
CHA1936
CHA1937
CHA1938
CHA1939
CHA1940
CHA1941
CHA1942
CHA1943
CHA1944
CHA1945
CHA1946
CHA1947

	FIDOT4=-ROU00*DFDXI4+DFIDT4	CHA1948
	UXDOT=-RGUG0*DUDXI4+DUDTX	CHA1949
	PXDOT=-ROU00*DPDXIX+DPDTX	CHA1950
	ROXDOT=-ROU00*DRDXIX+DRODTX	CHA1951
	ZXDOT=-ROU00*DZDXIX+DZDTX	CHA1952
	FIH1=FI1+DT2*FIDOT1	CHA1953
	FIH2=FI2+DT2*FIDOT2	CHA1954
	GIH=PX+DT2*PXDOT	CHA1955
	FIH3=FI3+DT2*FIDOT3	CHA1956
	FIH4=FI4+DT2*FIDOT4	CHA1957
	UXN=UX+DT*UXDOT	CHA1958
	PXN=PX+DT*PXDOT	CHA1959
	ROXN=ROX+DT*ROXDOT	CHA1960
	ZXN=ZX+DT*ZXDOT	CHA1961
	IF(ZXN.LT.0.) ZXN=0.	CHA1962
	GO TO 90	CHA1963
20	CONTINUE	CHA1964
	EVO=GL*DSQRT(BETA0)	CHA1965
201	CONTINUE	CHA1966
	DFIDA1=DRODAX*UX+ROX*DUDAX	CHA1967
	DFIDA2=DRODAX*UX**2+2.D0*ROX*UX*DUDAX+DPDAX	CHA1968
	DFIDA2=DFIDA2-DPDAX	CHA1969
	DFIDA3=DUDAX*(G12*PX+0.5D0*ROX*UX**2)	CHA1970
1	+UX*(G12*DPDAX+0.5D0*DRODAX*UX**2+ROX*UX*DUDAX)	CHA1971
	DFIDA4=ZX*DFIDA1+ROX*UX*DZDAX	CHA1972
	FIDOT1=-EVO*DFIDA1	CHA1973
	FIDOT2=-EVO*DFIDA2	CHA1974
	FIDOT3=-EVO*DFIDA3	CHA1975
	FIDOT4=-EVO*DFIDA4	CHA1976
	FIH1=FI1+DT2*FIDOT1	CHA1977
	FIH2=FI2+DT2*FIDOT2	CHA1978
	FIH3=FI3+DT2*FIDOT3	CHA1979
	FIH4=FI4+DT2*FIDOT4	CHA1980
	GA=DGDAX	CHA1981
	IF(NFIUX.EQ.5)GA=-GA	CHA1982
	DROUA=UX*DRODAX+ROX*DUDAX	CHA1983
	BETAPR=0.5D0*DSQRT(BETA0)*(GA-DROUA)	CHA1984
	FIH2=FIH2-DPDBX*BETAPR*DI2	CHA1985
	UXDOT=-EVO*DUDAX+BETAPR*DUDIX	CHA1986
	PXDOT=-EVO*DPDAX+BETAPR*DPDBX	CHA1987
	GIH=PX+DT2*PXDOT	CHA1988
	ROXDOT=-EVO*DRODAX+BETAPR*DRODBX	CHA1989
	ZXDOT=-EVO*DZDAX	CHA1990
	UXN=UX+DT*UXDOT	CHA1991
	PXN=PX+DT*PXDOT	CHA1992
	ROXN=ROX+DT*ROXDOT	CHA1993
	ZXN=ZX+DT*ZXDOT	CHA1994
	IF(ZXN.LT.0.) ZXN=0.	CHA1995
	GO TO 90	CHA1996
50	CONTINUE	CHA1997
	EVO=-GR*DSQRT(BETA0)	CHA1998
	GO TO 201	CHA1999
30	CONTINUE	CHA2000
40	CONTINUE	CHA2001
	GO TO 60	CHA2002
90	CONTINUE	CHA2003
	RETURN	CHA2004
	END	CHA2005

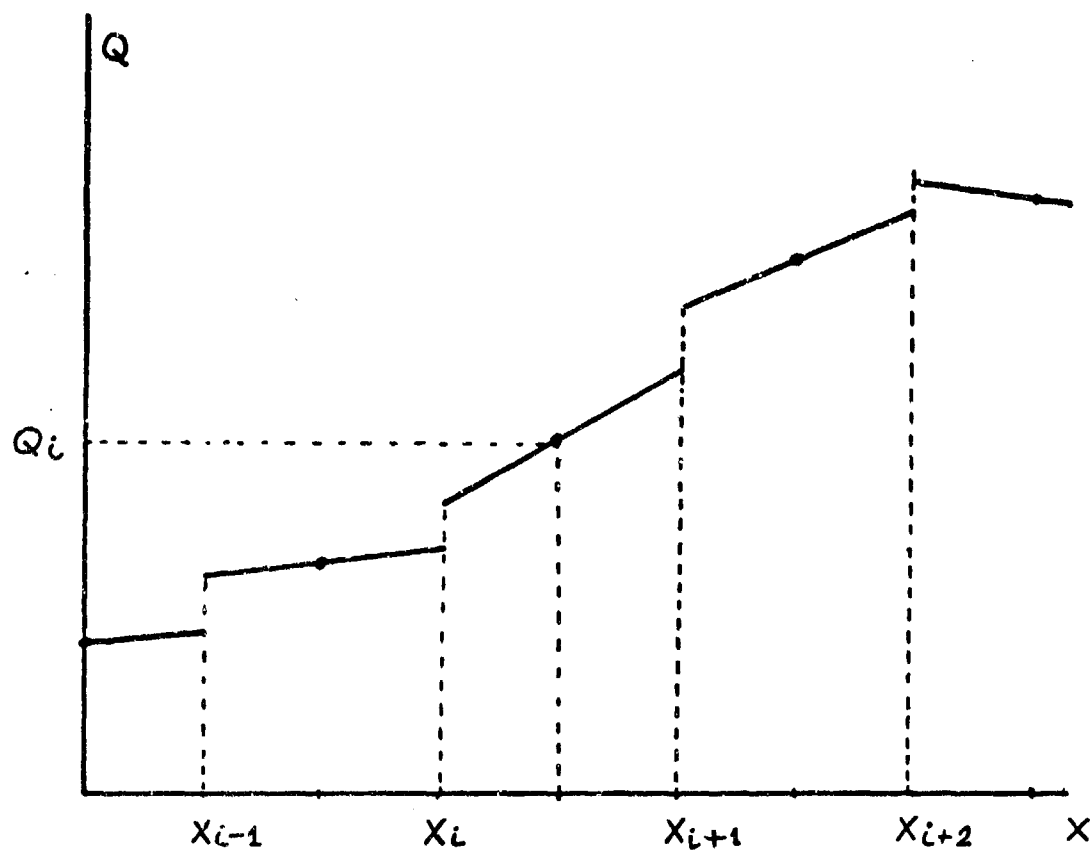


Figure A-1. Piecewise Linear Distribution of Flow Variables in Cells

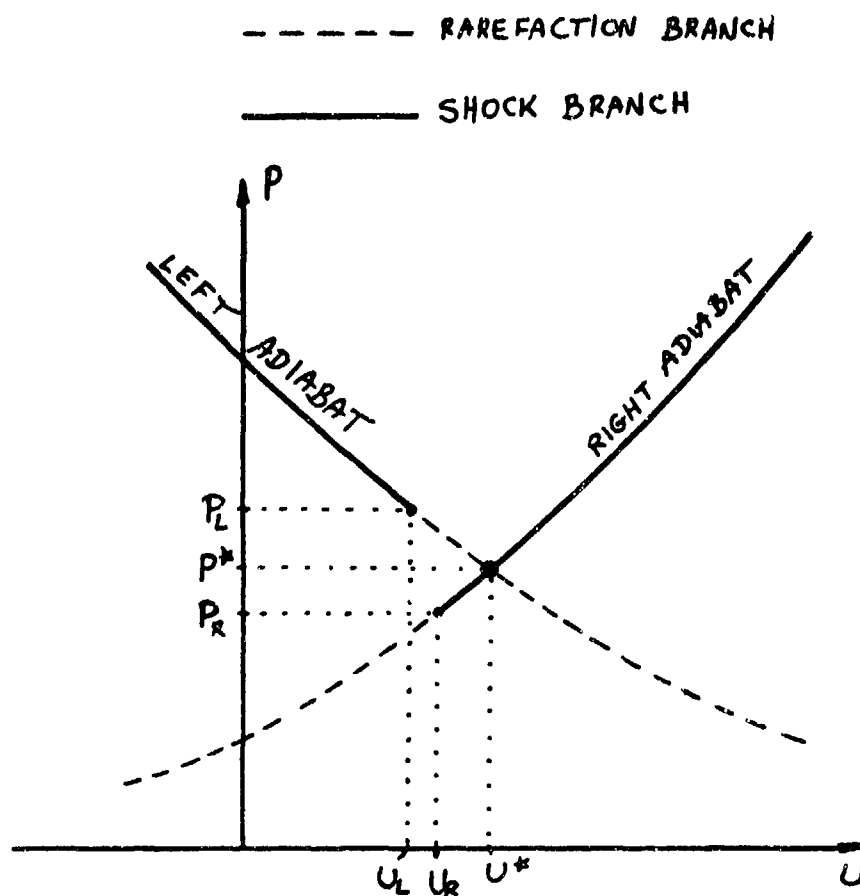


Figure A-2. Intersection of Right and Left Adiabats for Solving Riemann Problem

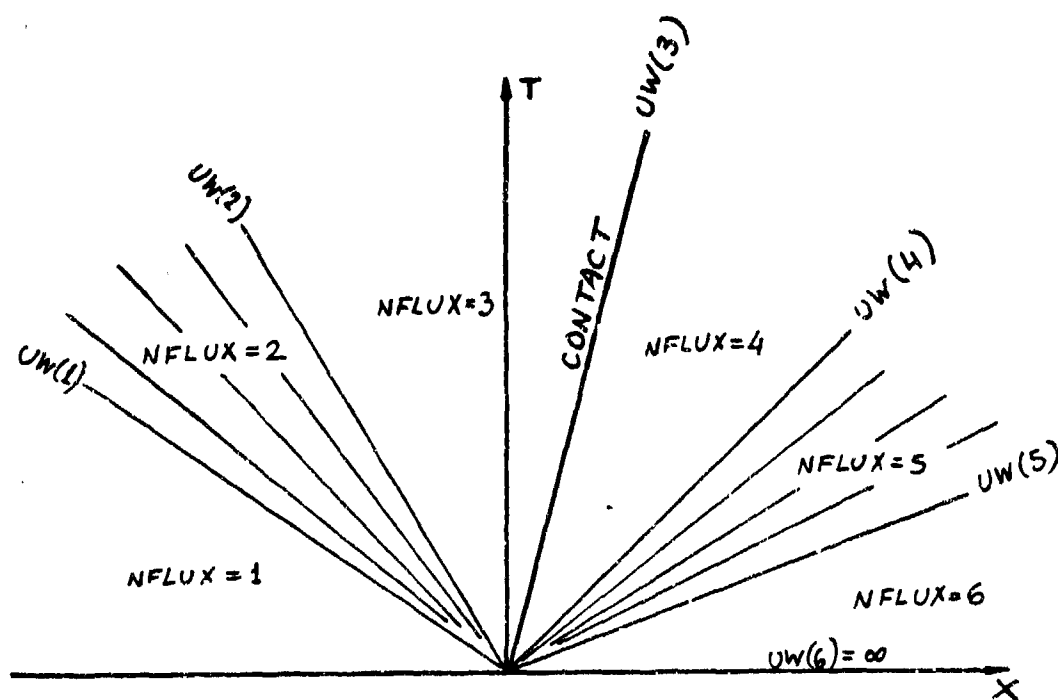


Figure A-3. Wave Diagram Representing Solution to Riemann Problem

APPENDIX B. Code for Re-Normalizing the Air Impulse

```

1      IMPLICIT REAL*8(A-H,O-Z)
C      CODE RENORM -- C TRANSFORMATION OF TOTAL REFLECTED IMPULSE FROM
C      BAKER'S CHART TO SPACE-NORMALIZED VALUES.
C      DATA FROM FIG. 6.3 (SUPPLEMENT) IN BAKER'S BOOK "EXPLOSIONS IN AIR"
2      REAL*4 RB,IB,RS,IS,ISBARE
3      DIMENSION RB(21),IB(21)
4      DIMENSION RS(21),IS(21),ISBARE(21)
5      DATA RB/.05,.06,.07,.08,.09,.1,.2,.3,.4,.5,.6,.7,.8,.9,1.,
1      2.,3.,4.,5.,6.,7./
6      DATA IB/4.4,3.06,2.30,1.83,1.50,1.27,.457,.293,.221,.178,.149,
1      .128,.113,.099,.0885,.0376,.0236,.0173,.0136,.0113,.0095/
7      PAI=4.D0*DATAN(1.D0)
8      G=1.4D0
9      PA=0.1D0
10     RHOA=1.3D0
11     RH00=1800.D0
12     Q0=4.D0
13     BETA=DSQRT(RHOA/RH00)*(PA/(RH00*Q0))*((1.D0/6.D0)
14     GOREM=(3.D0/DSQRT(2.D0*G))*((4.D0*PAI/3.D0))*((1.D0/3.D0)
15     BETA=BETA*GOREM
16     DELTA=((4.D0*PAI/3.D0)*(RH00*Q0/PA))*((1.D0/3.D0)
17     PRINT 11, BETA, DELTA
18 11   FORMAT(/1X,'RESULTS WITH BETA, DELTA=',2D16.7//
1      1X,' N', ' RB', ' IB', ' RS', ' IS', ' ISBARE' //)
2      2X, ' '
19     DO 1 N=1,21
20     RS(N)=RB(N)*DELTA
21     IS(N)=IB(N)*BETA
22     ISBARE(N)=1.D0/RS(N)**2
23     PRINT 2, N, RB(N), IB(N), RS(N), IS(N), ISBARE(N)
24 2     FORMAT(1X,I4,2E12.4,2X,2E12.4,2X,E12.4)
25 1     CONTINUE
26     END

```

REN00010
 REN00020
 REN00030
 REN00040
 REN00050
 REN00060
 REN00070
 REN00080
 REN00090
 REN00100
 REN00110
 REN00120
 REN00130
 REN00140
 REN00150
 REN00160
 REN00170
 REN00180
 REN00190
 REN00200
 REN00210
 REN00220
 REN00230
 REN00240
 REN00250
 REN00260
 REN00270
 REN00280
 REN00290
 REN00300
 REN00310
 REN00320
 REN00330
 REN00340

RESULTS WITH BETA, DELTA= 0.1204163D-01 0.6706157D+02

N	RB	IB	RS	IS	ISBARE
1	0.5000E-01	0.4400E+01	0.3353E+01	0.5298E-01	0.8894E-01
2	0.6000E-01	0.3060E+01	0.4024E+01	0.3685E-01	0.6177E-01
3	0.7000E-01	0.2300E+01	0.4694E+01	0.2770E-01	0.4538E-01
4	0.8000E-01	0.1830E+01	0.5365E+01	0.2204E-01	0.3474E-01
5	0.9000E-01	0.1500E+01	0.6036E+01	0.1806E-01	0.2745E-01
6	0.1000E+00	0.1270E+01	0.6706E+01	0.1529E-01	0.2224E-01
7	0.2000E+00	0.4570E+00	0.1341E+02	0.5503E-02	0.5559E-02
8	0.3000E+00	0.2930E+00	0.2012E+02	0.3528E-02	0.2471E-02
9	0.4000E+00	0.2210E+00	0.2682E+02	0.2661E-02	0.1390E-02
10	0.5000E+00	0.1780E+00	0.3353E+02	0.2143E-02	0.8894E-03
11	0.6000E+00	0.1490E+00	0.4024E+02	0.1794E-02	0.6177E-03
12	0.7000E+00	0.1280E+00	0.4694E+02	0.1541E-02	0.4538E-03
13	0.8000E+00	0.1130E+00	0.5365E+02	0.1361E-02	0.3474E-03
14	0.9000E+00	0.9900E-01	0.6036E+02	0.1192E-02	0.2745E-03
15	0.1000E+01	0.8850E-01	0.6706E+02	0.1066E-02	0.2224E-03
16	0.2000E+01	0.3760E-01	0.1341E+03	0.4528E-03	0.5559E-04
17	0.3000E+01	0.2360E-01	0.2012E+03	0.2842E-03	0.2471E-04
18	0.4000E+01	0.1730E-01	0.2682E+03	0.2083E-03	0.1390E-04
19	0.5000E+01	0.1360E-01	0.3353E+03	0.1638E-03	0.8894E-05
20	0.6000E+01	0.1130E-01	0.4024E+03	0.1361E-03	0.6177E-05
21	0.7000E+01	0.9510E-02	0.4694E+03	0.1144E-03	0.4538E-05

6. DISTRIBUTION LIST

No. of Copies

1. Defense Technical Information Center
Cameron Station
Alexandria, VA 22314 2
2. Library, Code 0142
Naval Postgraduate School
Monterey, CA 93943-5100. 2
3. Department Chairman, Code 67
Department of Aeronautics
Naval Postgraduate School
Monterey, CA 93943-5100. 1
4. Distinguished Professor Allen E. Fuhs
Space Systems Academic Group, Code 72
Naval Postgraduate School
Monterey, CA 93943-5100. 5
5. Dr. Neil Griff
SDIO/DEO
Washington, DC 20301-7100. 3
6. Mr. Bruce Pierce
SDIO/DEO
Washington, DC 20301-7100. 1
7. Dr. Joseph Falcovitz
Code 72
Naval Postgraduate School
Monterey, CA 93943-5100. 5

8. Professor R. E. Bali
 Department of Aeronautics, Code 67
 Naval Postgraduate School
 Monterey, CA 93943-5100. 1

9. Professor Y. S. Shin
 Department of Mechanical Engineering, Code 69
 Naval Postgraduate School
 Monterey, CA 93943-5100. 1

10. Research Administration Office
 Code 012
 Naval Postgraduate School
 Monterey, CA 93943-5100. 1

11. Dr. P. Avizonis
 Air Force Weapons Laboratory
 Kirtland Air Force Base, NM 87117 1

12. Dr. John Lawless
 Space Power Inc.
 1977 Concourse Drive
 San Jose, CA 95131 1

13. Dr. Mark Thornton
 Boeing Aerospace Company
 Post Office Box 3999
 Seattle, WA 98124-2499 1

14. LT. Mark Price
 AFRPL
 Edwards AFB, CA 93523 1

No. of Copies

15. Mr. Arthur W. Rogers
Space Systems Division
Hughes Aircraft Co.
P. O. Box 92919, Los Angeles, CA 90009 1

16. LCOL Rick Babcock, USAF
Air Force Geophysical Laboratory
Hanscomb Field
Bedford, MA 01730 1