LBL-7233
c.2

NUMERICAL ANALYSIS OF FLOW FIELDS GENERATED BY ACCELERATING FLAMES

John Kurylo
(Ph. D. thesis)

December 1977

Prepared for the U. S. Department of Energy
under Contract W-7405-ENG-48

NUMERICAL ANALYSIS OF FLOW FIELDS GENERATED BY ACCELERATING FLAMES

by

John Kurylo*

Department of Mechanical Engineering
University of California
Berkeley, California 94720

---

* Work performed as a Ph. D. Thesis partially supported by the U. S. Army Research Office under Grant No. DAAG-2977-G-0064.

NUMERICAL ANALYSIS OF FLOW FIELDS GENERATED BY ACCELERATING FLAMES

Contents

NUMERICAL ANALYSIS OF FLOW FIELDS GENERATED BY ACCELERATING FLAMES

John Kurylo

Lawrence Berkeley Laboratory
University of California
Berkeley, California 94720

ABSTRACT

Presented here is a numerical technique for the analysis of non-steady flow fields generated by accelerating flames in gaseous media. Of particular interest in the study is the evaluation of the non-steady effects on the flow field and the possible transition of the combustion process to detonation caused by an abrupt change in the burning speed of an initially steady flame propagating in an unconfined combustible gas mixture.

Optically recorded observations of accelerating flames established that the flow field can be considered to consist of non-steady flow fields associated with an assembly of interacting shock waves, contact discontinuities, deflagration and detonation fronts. In the analysis, these flow fields are treated as spatially one-dimensional, the influence of transport phenomena is considered to be negligible, and unburned and burned substances are assumed to behave as perfect gases with constant, but different, specific heats. The basis of the numerical technique is an explicit, two step, second order accurate, finite difference scheme employed to integrate the flow field equations expressed in divergence form. For this purpose the conservation equations are expressed in Eulerian form. The progress of and interaction between all the gasdynamic discontinuities specified above were treated explicitly

by a floating discontinuity fitting technique which was specially developed for this purpose.

Of particular importance in the computational scheme is the proper handling of deflagrations since the differentiating algorithm of the conservation equations cannot be applied across them. To accommodate this difficulty at each time interval, the Eulerian computational grid is shifted throughout the whole flow field so that it moves with the speed of the deflagration. The burning speed, governing the motion of the deflagration, is expressed in the form of a power law dependence on pressure and temperature immediately ahead of its front.

Each floating discontinuity fitting algorithm requires six to eight spatial grid points for the explicit handling of the discontinuity. This set of points defines a zone of influence. Discontinuities which propagate without interacting with each other nor with a plane, line, or point of symmetry, are considered to be well separated and their progress in the flow field established by the appropriate floating discontinuity fitting technique. When discontinuities are not well separated, that is when their zones of influence overlap, more elaborate difference algorithms are employed to ensure that differentiation, applicable only to the continuous portions of the flow field, is not carried out across a discontinuity. The actual interaction between non-reactive and reactive waves is considered to occur instantaneously. The resulting steady wave solution is obtained by the vector polar interaction technique, that is, by determining the point of intersection between the loci of end states in the plane of the two interaction invariants, pressure and particle velocity.

The technique is illustrated by a numerical example in which a steady flame experiences an abrupt change in its burning speed. Solutions correspond either to the eventual reestablishment of a steady state flow field commensurate with the burning speed or to the transition to detonation. A stability curve, a line of demarcation between the regimes of solution leading to steady and non-steady flow fields on the plane of the increment in the burning speed and its initial value, is presented. The results are in satisfactory agreement with experimental observations.

NOMENCLATURE

a	speed of sound
A	area
A	a/a_o (Figure)
B	constant Eq. (5.18)
A_t	transformation variable Eq. (4.9)
C_v	specific heat at constant volume
D	detonation wave speed
E_s	stagnation energy
G_p	function of pressure Eq. (5.11)
G_ρ	function of density Eq. (5.12)
H	Hugoniot
i	position index on the computational plane
j	geometry index (0, 1 and 2 for plane-, line- and point-symmetrical geometries)
J	Chapman-Jouguet detonation state
K	Chapman-Jouguet deflagration state
$\bar{M}$	mean molecular weight
M_n	shock Mach number
M_D	detonation Mach number
$\mathcal{M}$	$\bar{M}_u/\bar{M}_b$
n	time index on the computational plane
p	pressure
p	pressure ratio across the discontinuity
P	p/p_o (Figure)
R	gas constant

R	x'/x'_f (Figure)
RH	Rankine-Hugoniot
RL	Rayleigh Line
S	flame burning speed
t	transformed time variable Eq. (4.5a)
t'	time
T	temperature
T	$t'\sqrt{p_o v_o}/x'_f$ (Figure)
u	particle velocity
U	transformed velocity Eq. (4.11b)
U	u/a_o (Figure)
v	specific volume
W	transformation velocity Eq. (4.5b)
W	experimental data (Figure)
x	transformed space variable Eq. (4.5b)
x'	position
x'_f	flame position at the instant of initial acceleration
α	index Eq. (5.25)
β	asymptote of Hugoniot hyperbola
γ	ratio of specific heats
δ	temperature index Eq. (2.1)
ε	coefficient Eq. (5.1)
ζ,η,ξ	quadratic equation coefficients Eqs. (5.7), (5.16), (A.6)
θ	pressure index Eq. (2.1)
ν	specific volume ratio
ν	v/v_o (Figure)

π	explicit artificial viscosity
ρ	density
ρ	ρ/ρ_o (Figure)
$\tilde{\rho}$	transformed density Eq. (4.11a)
χ	constant Eq (5.12)
$[\,]$	conserved quantity Eqs. (5.1), (5.2)
$\|\ \|$	absolute value
Δ	step
+	node adjacent to a contact discontinuity
○	locus of end states (Figure)

Subscripts

b	state behind the combustion front
CJ	Chapman-Jouguet state
d	ratio of combusted to uncombusted parameters across the deflagration front
f	constant pressure combustion at local pressure
F	constant pressure combustion at reference pressure
h	head of the rarefaction
G	constant specific volume combustion at reference specific volume
i	position index on computational plane
J	Chapman-Jouguet state
N	von Neumann state
o	undisturbed reference state
s	shock

t	tail of the rarefaction
u	state ahead of the combustion front
x	position
1	state to the right of a discontinuity
2	state to the left of a discontinuity
3	state ahead of the discontinuity
4	state behind the discontinuity

Superscripts

n	time index on the computational plane
——	provisional value

1. INTRODUCTION

From the point of view of gasdynamics, all combustion processes can be classified as either detonations or deflagrations. Detonation, first described by Berthelot[1] in 1881, is the more dynamic process, characterized by a supersonic wave speed on the order of kilometers per second and a significant increase in presure, density, and fluid velocity. Deflagration is characterized by low subsonic burning speeds, negligible pressure changes, and significant increases in specific volume. However, deflagrations are endowed with the unique ability to generate pressure waves.[2-5]

Oppenheim[6] has established that if a sufficient amount of energy is deposited at a sufficiently high rate in an explosive gas, a detonation wave can be directly formed. Below a critical value of the initiation energy and power, there ensues a flame front that gradually recedes from that of the leading blast wave. Within the critical regime between the two, the non-steady mechanisms by which the process can accelerate from deflagration to detonation have been the subject of an intense wide-ranging experimental program of research over the last three decades. The development of and advances in optical techniques for viewing the flow field on a time scale commensurate with the non-steady mechanisms led Schmidt, Steinicke and Neubert[7,8] to produce the first photographic records of the development of detonation. These photographs provided the first insight into the nature of the non-steady mechanisms associated with the transition to detonation. As confirmed by stroboscopic-schlieren photographs in experiments by Oppenheim and associates,[6,9-11] the main mechanism of the transition is the action of the transverse waves generated by the deflagration process and the formation of their wave intersections and interactions.

The capability of deflagration to accelerate to detonation is important in the assessment of the dangers associated with unconfined vapour cloud explosions.. The rapidly increasing volume and rate at which flammable liquids and vapours are transported today throughout the world enhances the possibility of a variety of accidents that cause leaks and spills. An overview of the hazards due to such occurrences over the last 42 years was reviewed by Strehlow.[12] One hundred and eight fuel-air cloud explosions, the largest cloud encompassing 20 million cubic feet of combustible mixture, were documented. More recently, dangers associated with large-scale transport of fuels, in particular liquified natural gas (125,000 cubic meters or 5.6×10^7 kgs), gained a considerable amount of attention. This is exemplified by the reports of Fay[13,14] about the dispersion and flammability of LNG vapour clouds, and by the paper of Haverdings et al.[15] about the extent of damage which could be inflicted by the explosion of a fuel-air cloud resulting from the collision of a tanker at the entrance to Rotterdam harbor. As noted in the paper by Oppenhein, Kurylo, Cohen and Kamel,[16] the scope of potential danger from such clouds extends well beyond their initial boundaries. They concluded that detonative combustion of the cloud produces a higher level of potential damage than the deflagrative combustion of the cloud. These differences in the intensity of the blast waves and character of the flow fields persist until the front of the leading blast wave reaches a distance of five initial cloud radii. At greater distances, the question of whether transition to detonation did or did not take place inside the cloud is irrelevant. All this leads to the question: Under what conditions does deflagration maintain a steady state propagation rate and when does it accelerate to detonation? Such knowledge could lead to design requirements minimizing the occurrence of detonation by prohibiting design configurations, energy sources, and turbulence sources which could directly or indirectly trigger the transition to detonation.

In order to elucidate the conditions necessary to promote transition, Wagner et al.[17] performed experiments using screens to increase the flame propagation rate in unconfined stoichiometric hydrocarbon-air mixtures. They reported that centrally ignited steady deflagrations, upon passing through a screen, experienced significant increases in burning speed but did not transist to detonation. Rather they developed a new thermodynamic and gasdynamic state commensurate with the higher burning speed. However, they noted that flames with higher initial propagation rates (of approximately 50 meters per second obtained in oxygen enriched hydrocarbon-air mixtures) did go to detonation. In fact, their results indicate that increasing the initial flame propagation rate decreases the level of disturbance necessary for transition.

Therefore, the object of this study is the evaluation of the non-steady effects on the flow field and the possible acceleration of the combustion process to detonation, due to a disturbance at the flame front. The disturbance is considered to have the effect of causing an instantaneous change in the relative flame burning speed. Such phenomena are well established in the combustion literature. It is due to the effects of turbulence and has been exploited in the form of the so called Shchelkin turbolizers, used for detonation research.[17] In particular, it is the aim of this work to quantify the level of the disturbance necessary to trigger the transition to detonation.

The key to achieving this result lies in the unique feature of flames to generate pressure waves. By focusing attention on the pressure waves generated by the initial disturbance and determining their subsequent interaction with the flame and the remainder of the flow field, the evolution of the non-steady processes can be traced.

Over the last 3 decades many methods have been employed to solve complex gasdynamic problems. Analytical analysis has been able to predict the structure of blast waves under a variety of conditions,[18,19] including blast waves sustained by steady flames.[20] That analysis has been restricted to the case of self-similar solutions, whereas the problem now under consideration corresponds to a non-self similar flow field. Early numerical techniques based on the method of characteristics,[21-23] although exact in principle, suffered from losses in accuracy due to interpolation and extrapolation. A currently popular numerical technique[16,24-26] for removing the explicit computation of shock fronts was developed by von Neumann and Richtmyer[27] and later modified by Wilkins.[28] This technique utilizes the concept of explicit artificial viscosity. The smoothing action of explicit artificial viscosity transforms the discontinuous state properties across a shock front into smooth but rapidly varying quantities over a small number of computational zones. However, the excessive influence of the explicit artificial viscosity during the interaction process with the combustion front renders the technique inadequate for this problem.

During the last 15 years, great strides have been made in the theory of numerical analysis. The problems of the global conservation of mass, momentum, and energy, and the adequate treatment of shock fronts were overcome by the numerical differencing of the governing equations expressed in conservative[29] form and by the development of implicit artificial viscosity[30] techniques. The influence of the implicit artificial viscosity during the interaction process with a combustion front can be reduced through the technique of floating discontinuity fitting developed by Moretti.[31-33]

The primary objective of the work reported here is the development of a numerical technique for the analysis of the non-steady flow fields generated by accelerating flames in a gaseous media. The numerical technique in its final form is not limited in scope to flow fields containing only blast waves, nor is the combustion process restricted to the case of detonation or Chapman-Jouguet deflagration.[25,26] Rather the technique is capable of responding to and tracing the flow field processes which occur during the development of detonation. From the application of the technique, the level of disturbance necessary to trigger the transition to detonation can be specified.

Presented first is a discussion of the wave processes involved in the transition from deflagration to detonation. This provides physical reasons why the elementary wave processes of shock waves, contact discontinuities, deflagrations and detonations can be treated as plane discontinuities. The burning speed law governing the motion of the deflagration is then reported, followed by a brief note on the initial conditions corresponding to flow fields with plane, cylindrical, and spherical symmetry. Chapter 3 indicates the method employed in solving the problem of multiple wave interactions involving non-reactive and reactive waves. Chapter 4 introduces the explicit difference scheme for the integration of the conservation equations that describe the dynamic behavior of the non-steady flow field. A description of the algorithm for the floating discontinuity fitting technique for shock waves, contact discontinuities, deflagrations and detonations follows in Chapter 5. The next chapter is devoted to the specification of the numerical procedures required during the short period of time prior to and immediately after a wave interaction. The method of application of the numerical procedure is given in Chapter 7. Detailed results for two cases corresponding to the same initial flame burning speed, but experiencing different increments in the burning speed, are presented in the form of time-space wave diagrams, pressure-space profiles and pressure signatures. The stability curve, a line of demarcation between the regimes of solutions leading to steady and non-steady flow fields on the plane of the increment in burning speed and its initial value, is presented. Finally, Chapter 9 includes a summary of the significant features of the computational technique presented here, and conclusions regarding it.

2. FLOW FIELD PROCESSES

2.1. Wave Processes

This section discusses the wave processes which can occur in the flow field due to an increase in flame burning speed and in the course of transition to detonation. During the last 3 decades an intense wide-ranging experimental program aimed at elucidating the non-steady processes that govern the development of detonation has been carried out. In contrast to the basic experimental apparatus for detonation experiments, a long slender tube, which today remains practically unchanged from that first described by Berthelot,[1] it was the development of and advances in optical techniques for viewing the flow field that provided insight into the nature of the non-steady processes associated with the transition to detonation.

At the Fourth International Symposium on Combustion, Schmidt, Steinicke and Neubert[8] presented photographs of combustion waves in tubes obtained by the use of schlieren optics with a rotating drum-camera. Based on shock and flame traces in the time-distance domain as described by Schmidt et al.'s[7] interpretation of a photographic record of the development of detonation, Oppenheim and Stern[34] extensively analyzed their results. They proposed that the transition from deflagration to detonation was controlled by wave interaction phenomena involving shock waves, contact discontinuities, rarefaction waves and deflagrations. Experimental and theoretical investigations of shock-flame interactions begun by Chu[35] and Markstein[36,37] and enhanced by the ingeniously simple technique of Salamandra and Sevastyanova[38] for producing incident shock waves, concluded that the propagation of the flame was significantly influenced by the interaction. The ability of flames to generate pressure waves in confined and unconfined atmospheres was established in experiments by Laderman et al.[2-4] and Oppenheim, Kamel and Varvatsoulis.[5] It was shown that the generation of pressure waves by the flame could be theoretically modelled by accounting for the change in the rate of heat release brought about by the increase in the surface area of the combustion. This demonstrated the dynamic effects of combustion on the flow field. The dependence of transition on the interaction processes between the waves generated by accelerating flames in an explosive gas mixture was confirmed by strobscopic-schlieren photographs in experiments by Urtiew and Oppenheim.[6,9-11] Therefore, when considering the effects of an increase in flame burning speed on transition to detonation, it is essential to take proper account of the following elementary wave processes and of their interactions: shock waves, rarefaction waves, contact discontinuities and deflagrations.

2.2. Wave Treatment as a Discontinuity

Optically recorded experiments, notably in publications of Oppenheim and his associates,[4,5,39] show that the wave fronts of shock waves, contact discontinuities, detonations and flames have steep density gradients. Thus, all considerations associated with their structure and details of the progress of these waves, such as induction and relaxation phenomena and the effects of diffusion, viscosity and conductivity, can be neglected in favor of their dynamic effects. This yields a step-wise interpretation of the wave process which in reality has a more continuous character. The wave fronts are treated as plain surfaces in the flow field, across which finite instantaneous changes in state occur. In particular, flame fronts are treated as deflagrations, that is, discontinuities associated with a finite pressure change.

By accounting only for the dynamic effects of the wave processes, Laderman, Urtiew and Oppenheim[40] were able to successfully reconstruct a time-space wave diagram obtained from a streak-schlieren photograph and two associated pressure transducer records of a flame and shock undergoing numerous interactions and intersections. The experiment was carried out in an equimolar H_2-O_2 mixture maintained initially at room temperature and 100 mm Hg. The flame had an initial burning speed of 35.1 meters per second and the left running shocks had strengths of 1.14 and 1.024. Further details of the experiment can be found in Laderman, Urtiew and Oppenheim.[40] Figure 1 shows the streak-schlieren photograph with the insert displaying the pressure records at stations 1 and 2. Figures 2 and 3 show the time-space diagram obtained by a finite wave analysis and the close comparison of the experimental and analytical pressure profiles. Therefore, the conclusion that proper account of the progress of shock waves, contact discontinuities, detonations and flames in the flow field can be achieved by treating these elementary waves processes as plane finite waves is justified.

2.3 Flame Burning Speed Law

In analyzing the effect of an increase in flame burning speed on transition to detonation, the burning speed governing the motion of the flame was considered to be proportional to the thermodynamic state immediately ahead of its front. The specific form adopted was

$$S = S_o \left(\frac{T}{T_o}\right)^{\delta} \left(\frac{p}{p_o}\right)^{\theta} \tag{2.1}$$

where S is the normal burning speed, T and p are respectively the temperature and pressure immediately ahead of the flame front, while subscript o denotes the undisturbed, reference, conditions. Numerical values of the indices were selected so as to best fit the available experimental data. Such data were based on the experiments of Gilbert,[41] Goldenberg and Pelevin,[42] and the more recent experiments by Bradley and Hundy,[43] and Andrews and Bradley[44,45] Their correlations, valid over a wide ranges of pressures, temperatures and hydrocarbon-air mixtures, suggest that $\delta = 2.3$ and $\theta = 0.5$. In the case of a Chapman-Jouguet flame, the burning speed dependence on temperature, as shown in Appendix A, becomes

$$S_{CJ} = \frac{\sqrt{(1+\beta)\ R_o T}}{(1-\beta)} \left[\sqrt{\nu_f - \beta} - \sqrt{\nu_f - 1}\right] \tag{2.2}$$

where

$$\beta = \frac{\gamma_b - 1}{\gamma_b - 1}$$

and

$$\nu_f = \left(\frac{T_o}{T}\right) \nu_F + \left[1 - \frac{T_o}{T}\right] \left[\frac{\gamma_o}{\gamma_b}\ \frac{\gamma_b - 1}{\gamma_o - 1} - \frac{\bar{M}_o}{\bar{M}_b} + 1\right]$$

where subscript b refers to the combusted state behind the deflagration, R represents the gas constant, $\bar{M}$ denotes the mean molecular weight,

ν_F and ν_f refer to the specific volume ratio corresponding to a change of state at the reference and local pressures respectively. This relation arises due to the thermodynamic consideration associated with the Chapman-Jouguet condition of tangency of the Rayleigh line to the Hugoniot curve.

2.4. Initial Conditions

The initial flow field conditions used in determining the effect of an increase in flame burning speed on transition to detonation are evaluated based on Kuhl, et al.'s[20] analysis of the pressure waves that can be generated by clouds of explosive gas mixtures, in an atmosphere which is initially at rest. Their self-similar analysis assumes that the combustible mixture gives rise to a flame front, justifiably characterized as a deflagration, which propagates into the medium ahead of it at a given burning speed. As noted in Section 2.1, a steady blast wave preceeds the flame. The combustible medium is treated as a thermally and calorically perfect gas. In addition, the boundary condition of a zero particle velocity in the burned regime immediately behind the deflagration is used. The space profiles of the gasdynamic parameters, i.e., the pressure p, density ρ, and particle velocity u, corresponding to a steady deflagration with a burning speed of 25 meters per second are presented in Figs. 4 to 6. These results are for a case of a typical hydrocarbon-air mixture with thermodynamic properties specified by

$\nu_F = 7,$

$a_o = 331$ m/sec,

$M = 1,$

$\gamma_o = \gamma_u = 1.3,$

$\gamma_b = 1.2$

where ν_F represents the ratio of specific volumes at the undisturbed pressure, a refers to speed of sound, M denotes the ratio of molecular weights and subscripts o, u, and b refer to the reference, uncombusted, and combusted states respectively. The undisturbed pressure, density, and Newton's speed of sound have been used to nondimensionalize the ordinates in Figs. 4 to 6. The space coordinate has been nondimensionalized with respect to the flame location. The values of j equal to 0, 1, and 2 correspond to flow fields with plane, line and point symmetry. Maximum pressure and shock strength occur in the case of plane-symmetrical flow. In the cylindrical and spherical cases, the profiles increase from the leading shock to the deflagration front. These gasdynamic-space profiles represent the state of the flow field at the instant the combustion front experiences a sudden increase in its burning speed. This increase is specified as part of the initial conditions. Appendix B contains a listing of the computer codes used in generating the initial conditions.

3. DISCONTINUITY INTERACTIONS

3.1 Vector Polar Method

This section discusses the method of analysis used to evaluate the wave interaction phenomena which occur during transition from deflagration to detonation. In the 1940's the method of characteristics was effectively employed in the analysis of non-steady gasdynamic problems. However, its application to wave interaction processes was restricted to the continuous domains of the phenomena thus capable of only tracing the details of the process. Then, in 1958, Oppenheim and Stern[34] introduced the vector polar method, the simplest and best known of all the polar methods developed in the 1950's, for analysis of wave phenomena. In contrast to using the method of characteristics, Oppenheim and Stern[34] took the approach that it is more important to evaluate the gasdynamic states and wave system attained by a given wave interaction after all the transients have died down, than to trace the details of its progress. This is the method used in the present calculations. Leaving the zone of interaction completely outside the scope of the method of analysis is justified by our experience that the duration of the wave interaction event in the course of development of detonation is short in comparison to the rest of the wave propagation. The vector polar method is based on the use of wave polars, that is diagrams representing the loci of states attainable by a given wave front without any consideration given to the details of the flow process across the front. By using a logarithmic scale of pressure and local speed of sound ratio for the ordinate and a linear scale of particle velocity for the abscissa, the plane of the polar diagram is rendered a vector character. Consequently, as pressure ratios are multiplied in crossing a discontinuity while the particle velocities are added, the solution for a given wave interaction is obtained by the addition of vectors, representing changes brought about by the action of the fronts that participate in the wave interaction process. The analysis assumes that the wave thickness is negligible in comparison to the flow field as a whole, and that the wave fronts are locally plane in the immediate vicinity of the interaction point. This thinness assumption has been justified in Section 2.2. For this purpose flame processes are treated as a deflagration, that is, a discontinuity associated with a finite pressure change. The most important aspect of the vector polar method is the facility it provides in solving interactions between shock waves, rarefaction waves, deflagrations, and contact discontinuities, thereby making tractable the problem of multiple wave interactions which occur as detonation develops. Phenomena associated with non-reactive and reactive interactions are discussed in Sections 3.2.1 and 3.2.2 respectively.

3.2. Interactions

3.2.1. Non-Reactive Interactions

Wave interactions in which none of the participating wave processes are driven by combustion constitute non-reactive interactions. Shock waves, rarefaction waves and contact discontinuities are examples of such wave processes. The reason for the distinction between non-reactive and reactive interactions will become evident in the succeeding section. During the transition from deflagration to detonation, the following non-reactive wave interaction systems were encountered:

(1) shock-shock collision
(2) shock-shock merging
(3) shock-contact discontinuity $a_2 > a_1$
(4) shock-contact discontinuity $a_2 < a_1$
(5) shock-plane of symmetry

where a_1 and a_2 refer to the local speed of sound in the states immediately to the right and left of the contact discontinuity. Interactions involving rarefactions waves can be handled by the numerical scheme discussed in the succeeding chapter. The gasdynamic states and resulting wave system for each interaction are evaluated by the vector polar method. Typical vector polar diagrams for interactions (1) to (5) are presented in Figs. 7 to 11. Included are the time-space wave diagrams. In the time-space diagrams the thin dashed lines indicate particle paths, the solid lines denote shocks, and the double lines denote contact discontinuities. In each case, wave processes are created and/or annihilated as a result of the interaction. For interaction (5) closed form analytical expressions for the reflected gasdynamic state parameters and shock Mach number in terms of the incident gasdynamic state parameters and shock Mach number exist.[46]

In the numerical computations a criteria for determining when the resulting wave pattern has been found is required. With reference to Fig. 7, after known states 0, 1, and 2 are located in the P-U and A-U planes, and the loci of attainable states 3 and 4 drawn from states 1 and 2, the criterion for determining the exact location of states 3 and 4 is, as explained by Oppenheim and Stern:[34]

> "The utility of the P-U plane is a direct consequence of the fact that each wave interaction is governed by the dynamic compatability condition, that is, the condition that any new domain bounded by waves resulting from the interaction must contain particles which are all at the same pressure and move with the same velocity irrespective of their previous history (i.e., irrespectively through which boundary they get into the domain in question). Since such a condition does not apply to another thermodynamic parameter, each interaction generates in principle a contact surface, that is a surface dividing two different states within the same domain, which is, of course, parallel to the particle velocity. In other words, the salient feature of the P-U hodograph plane is the property that the domain resulting from any wave interaction is represented there, and only there, by a single point, in spite of the fact that it may represent two states separated by a contact surface."
>
> "Unlike the P-U plane, the states on the two sides of the contact surface are here (A-U plane) represented by two points, having the same value of abscissa since they correspond to the same particle velocity."

3.2.2. Reactive Interactions

The wave interactions involving a wave process driven by combustion are reactive interactions. Deflagrations and detonations are examples of such a wave process. Whereas in non-reactive interactions the energy supporting the resulting wave system satisfies the Rankine-Hugoniot relations, in reactive interactions the Hugoniot relations must be satisfied in order to account for the chemical energy released in the form of heat during the combustion process. The complexity lies in attempting to uniquely determine this quantity of heat. The flame burning speed law (Section 2.3) provides the necessary relation. In the development of detonation, the following reactive wave interaction systems were encountered:

(1) initial deflagration acceleration
(2) shock-deflagration merging
(3) deflagration-contact discontinuity
(4) detonation-contact discontinuity
(5) detonation-shock merging

where interaction (1) represents the finite increment in the burning speed associated with the initial disturbance. Interactions (2) and (3) constitute the primary mechanisms for deflagration acceleration, while interactions (4) and (5) involve the detonation process.
The gasdynamic states and resulting wave system for interactions (1) to (3) were evaluated by the vector polar method. Typical polar diagrams are presented in Figs. 12 to 14. The non-steady analysis of interactions (4) and (5) is presented in Figs. 15 and 16 respectively. Time-space wave diagrams are included. The thick dashed line represents a deflagration, "CJ" indicates a Chapman-Jouguet deflagration and a thick solid line denotes detonation.

The wave system for interaction (1) was obtained by specifying the increase in burning speed given by the initial conditions, rather than by the flame burning speed law. With interactions (2) and (3), two and three wave systems respectively could arise, each depending on the shock and contact discontinuity strengths. Figures 13A and 14A present the final wave systems in the polar and time-space planes for interactions (2) and (3) in which the deflagration burning speeds increased but remained below the Chapman-Jouguet value. However, Figs. 13B and 14B present wave systems corresponding to Chapman-Jouguet deflagration. Due to the increased shock and contact discontinuity strengths, the interactions were of sufficient intensity to produce Chapman-Jouguet deflagration. Figures 13C and 14C indicate how the loci of end states, states 5 and 6, fail to intersect when burning speeds less than the Chapman-Jouguet value are assumed for case B. Figure 14D presents the results for a Chapman-Jouguet deflagration-contact discontinuity interaction.

The non-steady wave systems generated by interactions (4) and (5) were analyzed in the thermodynamic plane of pressure and specific volume, shown in Figs. 15 and 16. The pressure and specific volume in these figures have been nondimensionalized with respect to an undisturbed reference pressure and specific volume. Indicated in Figs. 15 and 16 are the Rankine-Hugoniot and Hugoniot curves, denoted by symbols R-H and H respectively, associated with the thermodynamic states 1 and 2, corresponding to the right and left hand sides of the contact discontinuity. In the analysis a detonation front is modelled as a shock front followed by a deflagration. Overdriven detonations arise in interactions (4) and (5) because the change of state across the shock front, associated with the detonation, propagating into state 1 must be compatible with its speed as it emerges from state 2. The thermodynamic states immediately behind the detonation front, just prior to the interaction and after the decay of the interaction's non-steady effects, are indicated by points J_2 and J_1 respectively. These states are the Chapman-Jouguet states associated with initial states 2 and 1. They are determined by the condition of tangency of the Rayleigh line, denoted by symbol RL, to the Hugoniot curve. The non-steady phenomena in interactions (4) and (5) are associated with the transition from state J_2 to state J_1. The decay from overdriven to Chapman-Jouguet detonation occurs due to the interaction of rarefaction waves, propagating at speeds greater than the detonation speed, with the leading detonation front. This condition is expressed by the inequality

$$a + u \geq D$$

where a and u represent the sound and particle speed immediately behind the detonation front and D represents the detonation front velocity. The inequality sign applies for overdriven detonations while the equality sign pertains to Chapman-Jouguet detonations.

In modelling the non-steady phenomena arising from interaction (4), the transition from state J_2 to state J_1 is assumed to occur over one time step characteristic of the method of calculation. In interaction (5), the differences between states J_2 and J_1 are large. The non-steady decay of the overdriven detonation is modelled initially as an instantaneous decay of the overdriven detonation to a level such that the pressure behind the detonation front falls slightly below the pressure at state J_2. This state is indicated in Fig. 16 by the dashed line. Then the decay to Chapman-Jouguet detonation occurs in 20 equal increments, requiring one time step, characteristic of the method of calculation, per increment.

4. NUMERICAL METHOD

4.1. Governing Conservation Equations

The most convenient form to cast the laws of conservation of mass, momentum, and energy, which describe the one-dimensional time-dependent flow of a compressible fluid, is the conservative form. If the area of the flow is variable, it is also convenient to include the area in the basic equations. In the absence of external body forces, energy, and mass sources, and with negligible effect of the transport processes of diffusion of mass, momentum, and energy, the basic governing equations can be written as

$$\frac{\partial(\rho A)}{\partial t'} = - \frac{\partial}{\partial x'} (\rho u A) \tag{4.1}$$

$$\frac{\partial(\rho u A)}{\partial t'} = - \frac{\partial}{\partial x'} (\rho u^2 A + pA) + p\frac{dA}{dx'} \tag{4.2}$$

$$\frac{\partial(E_S)}{\partial t'} = - \frac{\partial}{\partial x'} (u\{E_S + pA\}) \tag{4.3}$$

where

$$E_S = \rho A\left(C_V T + \frac{u^2}{2}\right)$$

$$p = \rho R T \tag{4.4}$$

where p, ρ, T, E_S, and u are the gasdynamic properties pressure, density, temperature, stagnation energy and particle velocity respectively; t', x' - independent variables time and position; A - area; C_V - specific heat at constant volume; and R the universal gas constant divided by the mean molecular weight.

For the purposes of simplicity and generality, the flow variables are nondimensionalized by scaling the pressure and density with respect to their reference values (p_o, ρ_o), scaling velocities by $(p_o/\rho_o)^{1/2}$, scaling energy and temperature by (p_o/ρ_o), scaling all lengths with respect to the flame's location at the instant of initial acceleration, x_f', and scaling time with respect to $x_f'/(p_o/\rho_o)^{1/2}$. Section 5.3 discusses the necessity of the following transformation of the independent variables in applying the floating flame fitting technique:

$$t = t' \tag{4.5a}$$

$$x = x' - Wt' \tag{4.5b}$$

where W is a time dependent but spatially independent quantity. Time and space derivatives become

$$\frac{\partial}{\partial t'} = \frac{\partial}{\partial t} - \left(W + t\,\frac{dW}{dt}\right)\frac{\partial}{\partial x} \tag{4.6a}$$

$$\frac{\partial}{\partial x'} = \frac{\partial}{\partial x} \tag{4.6b}$$

The nondimensionalized transformed governing equations can be written as

$$\frac{\partial(\rho A)}{\partial t} = -\frac{\partial}{\partial x}(\rho u A) - A_t\,\frac{\partial(\rho A)}{\partial x} \tag{4.7}$$

$$\frac{\partial(\rho u A)}{\partial t} = -\frac{\partial}{\partial x}\left(\rho u^2 A + pA\right) + p\frac{dA}{dx} - A_t\frac{\partial(\rho u A)}{\partial x} \qquad (4.8)$$

$$\frac{\partial(E_s)}{\partial t} = -\frac{\partial}{\partial x}\left(u\{E_s + pA\}\right) - A_t\frac{\partial(E_s)}{\partial x} \qquad (4.9)$$

where

$$E_s = \rho A\left(C_v T + \frac{u^2}{2}\right)$$

and

$$A_t = -W - t\frac{dW}{dt}$$

$$p = \rho R T \qquad (4.10)$$

4.2. Numerical Integration

These equations form a system of non-linear hyperbolic equations. The solution of this system can breakdown due to nonlinearity and exhibit a pattern of shock waves, contact surfaces with large energy differences, vortex sheets, and slip lines, the latter two surfaces being evident only in higher dimensions. Analytical analysis of these flows have been limited to cases of steady or self-similar flows, incorporating only a few such surfaces. As a characteristic length, the distance traversed by the flame prior to the instant of initial acceleration, is inherent in the problem, a self-similar analysis is ruled out.

4.3. Numerical Scheme[47]

Over the last 3 decades, many methods have been employed to solve one-dimensional gasdynamic problems under a variety of circumstances. The most successful early techniques were based on the method of characteristics.[21-23] Although they were exact in principle, they suffered losses in accuracy due to interpolation and extrapolation. Also, the method of characteristics did not lend itself as readily to the digital computer as the more efficient finite difference technique. The method of shock capturing, a technique to remove the explicit computation of the discontinuity, developed by von Neumann and Richtmyer[27] and as modified by Wilkins,[28] utilized the concept of explicit artificial viscosity. The rapid changes of state produced by shocks waves were treated by replacing pressure, p, in the Lagrangian conversation equations with $p + \pi$ where π is a psuedo-viscous pressure depending on the form of the explicit artificial viscosity. The smoothing action of explicit viscosity transformed the discontinuous state properties across the discontinuity into smooth but rapidly varying quantities over a small number of cells, on either side of which the discontinuity relations were satisfied. As the whole flow field could be treated as a single continuous flow field, the method found many applications.[16,24-26] However, the excessive influence of the explicit artificial viscosity during the interaction of discontinuities with the combustion front rendered the technique inadequate in this problem.

The finite difference methods used to solve one-dimensional gasdynamic problems over the last 15 years and the difficulties encountered by these methods are discussed in Richtmyer and Morton[30] and Ames.[48] The two most prevelent problems were concerned with the global conservation of mass, momentum, and energy and the adequate treatment of discontinuities. The solution to these problems were accomplished by casting the governing equations into conservative form and by the development of implicit artificial viscosity techniques. Differencing equations in conservative form ensures global conservation of mass, momentum, and energy.[29] The method of shock capturing utilizing implicit artificial viscosity is based on generalizing the concept of a solution of Euler's equations to include weak solutions, i.e., discontinuities.[30]

Hence the shock capturing technique developed by MacCormack[49] is employed in the solution of this complicated one-dimensional gasdynamic flow. MacCormack's method consists of an explicit, noncentered difference scheme applied to the equations of motion in conservative form. The artificial viscosity is implicit and a direct result of the numerical scheme as clearly shown by Tyler.[50] The method conserves global mass, momentum, and energy and will automatically capture any imbedded shocks which appear in the flow field. For the computational advantage of maintaining mass, momentum, and energy fluxes relatively constant through rapid area changes, a change in the dependent variables is made as follows

$$\tilde{\rho} = \rho A \tag{4.11a}$$

$$U = \tilde{\rho} u = \tilde{\rho} u A \tag{4.11b}$$

$$E_s = \frac{pA}{\gamma - 1} + \frac{U^2}{2\tilde{\rho}} \tag{4.11c}$$

where Eq. (4.10) has been used to eliminate the explicit temperature dependence of E_S.

In terms of these variables, the basic Eqs. (4.7) to (4.10) become

$$\frac{\partial \tilde{\rho}}{\partial t} = - \frac{\partial U}{\partial x} - A_t \frac{\partial \tilde{\rho}}{\partial x} \tag{4.12}$$

$$\frac{\partial U}{\partial t} = - \frac{\partial}{\partial x} \left(\frac{U^2}{\tilde{\rho}} + pA \right) + p \frac{dA}{dx} - A_t \frac{\partial U}{\partial x} \tag{4.13}$$

$$\frac{\partial E_S}{\partial t} = - \frac{\partial}{\partial x} \left(\frac{U}{\tilde{\rho}} \{E_S + pA\} \right) - A_t \frac{\partial E_S}{\partial x} \tag{4.14}$$

$$p = \left(E_S - \frac{U^2}{2\tilde{\rho}} \right) \frac{\gamma - 1}{A} \tag{4.15}$$

The basic algorithm in the MacCormack technique consists of a two step process, a predictor step and a corrector step. Both the predictor and corrector are made up of simple forward and backward differences of first order accuracy, but the combined two step process has a second order accuracy.[49] The numerical artificial viscosity required for accurate simulation of shock waves in the flow is applied implicitly where needed.[50] The predictor portion of the bilevel difference method is defined by

$$\tilde{\rho}_i^{\overline{n+1}} = \tilde{\rho}_i^n - \frac{\Delta t}{\Delta x} \left[U_{i+1}^n - U_i^n \right]$$

$$- A_t \frac{\Delta t}{\Delta x} \left[\tilde{\rho}_{i+1}^n - \tilde{\rho}_i^n \right] \tag{4.16}$$

$$\overline{U_i^{n+1}} = U_i^n - \frac{\Delta t}{\Delta x}\left[\left(\frac{U^2}{\tilde{\rho}} + pA\right)_{i+1}^{n} - \left(\frac{U^2}{\tilde{\rho}} + pA\right)_{i}^{n}\right]$$

$$+ \Delta t\left(p\,\frac{dA}{dx}\right)_{i+1/2}^{n} - A_t\,\frac{\Delta t}{\Delta x}\left(U_{i+1}^n - U_i^n\right) \tag{4.17}$$

$$\overline{E_{s_i}^{n+1}} = E_{s_i}^n - \frac{\Delta t}{\Delta x}\left[\left\{\frac{U}{\tilde{\rho}}\,(E_s + pA)\right\}_{i+1}^{n}\right.$$

$$\left. - \left\{\frac{U}{\tilde{\rho}}\,(E_s + pA)\right\}_{i}^{n}\right] - A_t\,\frac{\Delta t}{\Delta x}\left[E_{s_{i+1}}^n - E_{s_i}^n\right] \tag{4.18}$$

$$\overline{p_i^{n+1}} = \left[\overline{E_{s_i}^{n+1}} - \frac{\overline{U_i^{n+1}}^{\,2}}{2\tilde{\rho}_i}\right]\frac{\gamma - 1}{\overline{A_i^{n+1}}} \tag{4.19}$$

where subscript i refers to the ith node at spatial location $x = x_o + i\Delta x$ in the spatial mesh of constant spacing Δx, while superscript n refers to time $t = t_o + n\Delta t$ where Δt is the time increment that the solution is advanced during each cycle of the predictor and corrector. The bar indicates the provisional nature of the values given by the predictor. the predictor uses only forward time and space differences.

The finite difference form for the corrector is as follows

$$\tilde{\rho}_i^{n+1} = \frac{1}{2}\left[\tilde{\rho}_i^{n} + \tilde{\rho}_i^{\overline{n+1}} - \frac{\Delta t}{\Delta x}\left(U_i^{\overline{n+1}} - U_{i-1}^{\overline{n+1}}\right) - A_t\,\frac{\Delta t}{\Delta x}\left(\tilde{\rho}_i^{\overline{n+1}} - \tilde{\rho}_{i-1}^{\overline{n+1}}\right)\right] \qquad (4.20)$$

$$U_i^{n+1} = \frac{1}{2}\left[U_i^{n} + U_i^{\overline{n+1}} - \frac{\Delta t}{\Delta x}\left\{\left(\frac{U^2}{\tilde{\rho}} + pA\right)_i^{\overline{n+1}} - \left(\frac{U^2}{\tilde{\rho}} + pA\right)_{i-1}^{\overline{n+1}}\right\} + \Delta t\left(p\,\frac{dA}{dx}\right)_{i-1/2}^{\overline{n+1}} - A_t\,\frac{\Delta t}{\Delta x}\left(U_i^{\overline{n+1}} - U_{i-1}^{\overline{n+1}}\right)\right] \qquad (4.21)$$

$$E_{s_i}^{n+1} = \frac{1}{2}\left[E_{s_i}^{n} + E_{s_i}^{\overline{n+1}} - \frac{\Delta t}{\Delta x}\left(\left\{\frac{U}{\tilde{\rho}}\,(E_s + pA)\right\}_i^{\overline{n+1}} - \left\{\frac{U}{\tilde{\rho}}\,(E_s + pA)\right\}_{i-1}^{\overline{n+1}}\right) - A_t\,\frac{\Delta t}{\Delta x}\left(E_{s_i}^{\overline{n+1}} - E_{s_{i-1}}^{\overline{n+1}}\right)\right] \qquad (4.22)$$

$$p_i^{n+1} = \left[E_{s_i}^{n+1} - \frac{{U_i^{n+1}}^2}{2\tilde{\rho}_i^{n+1}}\right]\frac{\gamma - 1}{A_i^{n+1}} \qquad (4.23)$$

The corrector step utilizes a forward time step, but a backward spatial step based on the predicted solution. The alternating use of forward and backward spatial derivatives models the fact that in hyperbolic systems at each location information travels by waves in both the positive and negative x direction. Completing the corrector at each mesh point advances the solution one complete time step. In order to proceed to the next point in time, the entire process of predictor and corrector is repeated. Examination of the set of difference equations shows that the mass, momentum, and energy are conserved during the calculations; that is, the differenced quantities at interior points of the mesh appear exactly twice during a sweep through the mesh, each time with opposite signs.

4.4. Time Step Criteria

The numerical stability of methods of the MacCormack type, namely, those of Lax and Wendroff[30] can not presently be completely analyzed in their general nonlinear form. The most successful attempt in this respect to date is to first linearize the set of basic differential equations and then to obtain a bound for the maximum amplification of any Fourier component of the solution by the difference method applied to the linearized set. Difference methods found unstable on locally linearized differential equations can be expected to be unstable in the general nonlinear case. Two conditions inherent in such an analysis are that the boundary conditions have no effect on the stability and that the exact solution to the basic equations is smooth. Even though the results apply principally to regions of flow away from the boundaries and in which there are no discontinuities, such as shocks, they are taken as representative of the whole flow field. The stability of the MacCormack scheme has been treated very thoroughly in the literature.[51]

In order to avoid stability problems, the following criterion must hold between the time and space step sizes

$$\frac{\Delta t(|u| + a + |A_t|)}{\Delta x} \leq 1 \tag{4.24}$$

where a is the local speed of sound in the gas, and A_t is related to the transformation of the spatial coordinate. This condition is the well known Courant-Friedrichs-Lewy (CFL) condition that often appears in fluid dynamics. This condition represents the best bound that can be realized in numerical methods. For all the calculations which have been carried out in the present research, a more conservative condition was used

$$\frac{\Delta t(|u| + a + |A_t|)}{\Delta x} = 0.7 \tag{4.25}$$

5. FLOATING DISCONTINUITY FITTING

5.1. Floating Shock Fitting

Explicit and implicit artificial viscosity shock capturing techniques require no special treatment to deal with discontinuities. Therefore they have become an extremely popular way of computing. However, despite its present popularity, the results obtained with these techniques forces us to agree with Moretti's[32] conclusion that, at times, shock capturing is a poor interpretation of the physical phenomena. As shock interactions with the combustion front represent the primary means of initial flame front acceleration, proper physical modelling of the interacting process is required.

To determine whether shock-combustion front interactions could be modelled appropriately, a test for the pressure-space profile of a shock wave was made. The solid line in Fig. 17 represents the exact pressure-space profile for shock reflection from a solid plane wall corresponding to an incident shock, Mach number 1.505, travelling into a quiescent air medium. Corresponding to that same instant in time, the "plus" profile represents calculations performed by the contemporary explicit artificial viscosity scheme, the Cloud Code,[24] while the "dotted" profile corresponds to MacCormack's implicit artificial viscosity scheme. For both numerical schemes the velocity and position of the shock agree with the exact results. Previous calculations, using the explicit artificial viscosity scheme, indicated an excessive influence of artificial viscosity on the shock-combustion front interaction process. In addition, allowable time steps were reduced. This precluded the method's usefulness. With MacCormack, where artificial viscosity is limited, the captured shock is distributed over six to eight spatial grid stations over which oscillation exists. Oscillation and overshoots are emblematic of numerical differencing across discontinuous or non-smooth, rapidly varying quantities. To properly model the shock-combustion front interaction process, interpretation of the shock's physical phenomena is necessary.

A technique known as floating shock fitting,[52] when used in conjunction with MacCormack's scheme, produces a pressure-space profile indistinguishable from the exact solution. Floating shock fitting is a technique for explicitly computing the discontinuity. A description of this combined scheme follows.

Except in the neighborhood of the discontinuity, all mesh points are computed using MacCormack's predictor-corrector algorithm to integrate Eqs. (4.12) to (4.15). Because the discontinuities are permitted to float about grid positions, special provisions must be taken when evaluating the spatial derivatives of conserved quantities at mesh points in the neighborhood of a discontinuity. Consider the situation of a right running shock in the computational plane between times n and n+1 as shown in Fig. 18A. There is difficulty in computing the predictor level of MacCormack's scheme at node (n,i), the high pressure side of the shock. Performing the forward spatial derivative using nodal points s and i leads to instabilities resulting from large truncation errors, while Moretti's[31] recommendation of using values at the shock and the three adjacent mesh points introduces too large a numerical domain of dependence. Results replacing forward differences with backward differences in the predictor remained stable for axis speeds less than the shock induced particle speeds. The cause of the instability is thought to involve the improper transmission of information along the three characteristics adjacent to the shock.[53] Instead the following second order accurate approximation for the forward derivatives is used[54]

$$\frac{\partial []}{\partial x} = \frac{2(2-\varepsilon)}{1+\varepsilon} []_s^n + (2\varepsilon - 3) []_i^n + \frac{(1-\varepsilon)(2\varepsilon - 1)}{1+\varepsilon} []_{i-1}^n \quad (5.1)$$

where

$$\varepsilon = \frac{(x)_s^n - (x)_i^n}{\Delta x}$$

and [] represents a conserved quantity. In Fig. 18B a similar situation occurs, but involving the corrector level at node (n+1, i+1), the high pressure side of the shock. The backward derivative is approximated by

$$\frac{\partial []}{\partial x} = \frac{2(2-\varepsilon)}{1+\varepsilon} []_s^{\overline{n+1}} + (2\varepsilon - 3) []_{i+1}^{\overline{n+1}} + \frac{(1-\varepsilon)(2\varepsilon - 1)}{1+\varepsilon} []_{i+2}^{\overline{n+1}} \quad (5.2)$$

where

$$\varepsilon = \frac{(x)_{i+1}^{\overline{n+1}} - (x)_s^{\overline{n+1}}}{\Delta x}$$

For the low pressure side, stable computations are obtained by reversing the forward and/or backward differences in the predictor and/or corrector levels.

Figure 19 presents the five identifiable cases of shock motion relative to the uniformly translating difference grid. The lower and upper mesh lines represent nodal positions at the beginning and end of the time step under consideration, the thick solid line is the shock trajectory, and the thin upward and downward arrowed lines are symbols used to indicate application of MacCormack's predictor or corrector step respectively at the note indicated by the head. The leftward or rightward direction of the arrowed lines denotes whether a forward or backward space difference is to be used. The strength of the shock is simply determined from the Rankine-Hugoniot relations for pressure as a function of shock strength. The predicted shock strength uses predicted values of pressure, while the corrected shock strength relies on corrected pressure values. To maintain higher order accuracy, the shock's trajectory is evaluated on the basis of the average of the initial and predicted speeds. After completion of the corrector step, a corrected shock speed and strength are calculated. These are retained as the initial shock speed and strength during the next cycle. The vertical arrowed lines appear because the discontinuity floated over a node of the mesh. In such cases the predicted and corrected values of the gasdynamic variables are determined by applying the Rankine-Hugoniot relations to upstream conditions at the beginning of the time step. The predicted values of the gasdynamic variables are based on the initial shock strength while the corrected values use an average of the initial and predicted shock strengths. The inverted "T" and "T" symbols indicate the use of Eqs. (5.1) and (5.2) respectively.

A technique developed by Moretti et al.[33,55] utilizing the information carried along the characteristic intersecting the shock on the high pressure side, i.e., the compatability equation along with the Rankine-Hugoniot relations to determine the gasdynamic states about the shock, was tested. Moretti's technique showed no advantages over floating shock fitting. In addition, Salas[52] mentions the difficulty of determining the origin of the characteristic reaching the shock in the immediate viscinity of other discontinuities.

MacCormack's scheme combined with the floating shock fitting techniue properly models the shock. The appropriate difference schemes for all five cases are thus completely described.

5.2. Floating Contact Discontinuity Fitting

A Contact discontinuity is an interface separating two fluids having a common pressure and fluid velocity but different densities and specific energies. Contact discontinuities are created by the following flow field interaction systems:

1) shock-shock collision and merging (Figs. 7 and 8)
2) shock-deflagration merging (Fig. 13)
3) deflagration-contact discontinuity (Fig. 14)

Strictly speaking, numerical differentiation of density and specific internal energy across the surface is not permitted. Figure 20 shows the resulting density-space profile of a contact surface after 40 cycles of the MacCormack scheme. Though the oscillation is minimal and negligibly influences shock wave calculations, the sensitivity of the combustion front parameters to small changes in density is significant.

The floating contact discontinuity fitting technique devised by the author, as suggested by Chorin,[56] overcomes this difficulty by treating the contact discontinuity as a physical discontinuity. With reference to Fig. 21A, the basis for calculating the contact discontinuity's motion and associated discontinuous gasdynamic states is the generalization of the solution of Riemann's[57] problem to include any two arbitrary independent gasdynamic states, 1 and 2. States 3 and 4 are the required gasdynamic states on either side of the contact surface. A rapidly converging algorithm[58] is used for solving the Riemann problem. Figure 21B shows the trajectory of the contact discontinuity, denoted by the dashed line, in the computational plane between times n and n+1. Parameters at locations i+1 and i correspond to states 1 and 2 respectively while values for states 3 and 4 reside at the "+" nodes which travel with the contact surface. The "bursting star" symbol is used to identify states 1 through 4. In the course of the calculations the shock strengths associated with the Riemann solution were monitored and found to vary insignificantly from that of sound waves. Figure 22 presents the difference schemes in symbolic form used in the four identifiable cases of contact discontinuity motion relative to the difference grid.

5.3. Floating Flame Fitting

This section describes floating flame fitting, a novel technique for encorporating a flame, treated as a deflagration, into a bilevel difference scheme. The technique is based on the ability of sound waves to traverse a flame from positions upstream and downstream of the flame. Equations (2.1) and (2.2) relate the burning speed to the value of the thermodynamic pressure and temperature immediately ahead of the flame. In order to avoid the complications of evaluating the thermodynamic state immediately ahead of the deflagration as the deflagration progresses through a stationary mesh, eventually crossing a mesh point, this technique requires that the deflagration always coincide with one of the nodes. In addition the computational mesh moves with the deflagration velocity. This is accomplished by judiciously distributing the initial mesh about the deflagration position and transforming the independent time and space coordinates of the governing differential equations as shown in Eqs. (4.5a) and (4.5b).

Figure 23 shows the deflagration trajectory, denoted by a thick dashed line, and the nodal distribution about the deflagration during time instant n to n+1. Nodes i and i+1 represent the gasdynamic states of the medium immediately ahead of, denoted by 3, and behind, denoted by 4, the deflagration respectively. For the case of a perfect gas, the conservation relations governing the motion of the deflagration can be expressed as[59]

Continuity: $\nu_d = v_4/v_3 = (S + u_3 - u_4)/S$ (5.3)

Momentum: $P_d = p_4/p_3 = 1 + \gamma_3 S^2(1 - \nu_d)/a_3^2$ (5.4)

Hugoniot: $(P_d + \beta)(\nu_d - \beta) = (1 + \beta)(\nu_f - \beta)$ (5.5)

where

$$\nu_f = \left(\frac{a_o}{a_3}\right)^2 \nu_F + \left[1 - \left(\frac{a_o}{a_3}\right)^2\right]\left[\frac{\gamma_3}{\gamma_4}\;\frac{(\gamma_4 - 1)}{\gamma_3 - 1} - \frac{\bar{M}_3}{\bar{M}_4} + 1\right]$$

$$\beta = \frac{\gamma_4 - 1}{\gamma_4 + 1}$$

where p, v, a, and u represent the pressure, specific volume, speed of sound, and particle velocity respectively, S - the burning speed, $\bar{M}$ - the mean molecular weight, and γ - the ratio of specific heats. The Hugoniot equation has been written in terms of the hyperbolic relation where the heat liberated during combustion is expressed in terms of ν_F, the ratio of specific volumes at the reference pressure.

When conditions ahead of the deflagration affect conditions behind, the forward jump equations, obtained by substituting Eq. (5.3) into Eq. (5.4) and solving for specific volume, are used. For flame burning speeds less than the Chapman-Jouguet value, these equations are

$$S = S_o\left(\frac{p_3}{p_o}\right)^{\delta+\theta}\left(\frac{v_3}{v_o}\right)^{\delta} \tag{5.6}$$

$$v_4 = v_3 \, \frac{\xi - \sqrt{\zeta}}{\eta} \tag{5.7}$$

where

$$\eta = 2\gamma_3 S^2/a_3^2 \qquad \text{and} \qquad \xi = (1 + \beta)(1 + \eta/2)$$

$$\zeta = \xi^2 - 2\eta[\beta\eta/2 + (1 + \beta)\ \nu_f]$$

$$p_4 = p_3(1 + \eta(1 - \nu_d)/2) \tag{5.8}$$

$$u_4 = u_3 - S(\nu_d - 1) \tag{5.9}$$

A Chapman-Jouguet deflagration corresponds to $\zeta = 0$. In the Chapman-Jouguet case, the burning speed law becomes

$$S_{CJ} = \frac{\sqrt{(1 + \beta)\ p_3 v_3}}{(1 - \beta)} \left[\sqrt{\nu_f - \beta} - \sqrt{\nu_f - 1}\right] \tag{5.10}$$

while the forward jump Eqs. (5.7) to (5.9) remain unchanged with the exception that S_{CJ} replaces S. When conditions behind the deflagration influence conditions ahead, the backward jump equations are employed. As the burning speed represents an additional unknown, these equations no longer have a closed form, rather the equations require Newton's method for their solution. The algorithm for the backward jump equations begins by guessing a value for the density, ρ_3, and iteratively solving for pressure, p_3, from Eq. (5.4) as follows

$$p_3 = p_3 - G_p/(\partial G_p/\partial p_3) \tag{5.11}$$

where

$$S = S_o\left(\frac{p_3}{p_o}\right)^{\delta+\theta}\left(\frac{\rho_o}{\rho_3}\right)^{\delta}$$

$$G_p = p_3 - p_4 + S^2\rho_3(1 - \nu_d)$$

$$\frac{\partial G}{\partial p_3} = 1 + 2S^2(\delta + \theta)(1 - \nu_d)\ \rho_3/p_3$$

Then by substituting Eq. (5.4) into Eq. (5.5) and using p_3 from Eq. (5.11), the value of ρ_3 is obtained by the following iteration

$$\rho_3 = \rho_3 - G_\rho/(\partial G_\rho/\partial\rho_3) \tag{5.12}$$

where

$$S = S_o\left(\frac{p_3}{p_o}\right)^{\delta+\theta}\left(\frac{\rho_o}{\rho_3}\right)^{\delta}$$

$$G_\rho = p_4(\beta - \nu_d)(1 + \beta) + (1 + \beta)(\nu_F - \chi)\ \rho_3 + p_4(1 + \beta)(\chi - \beta)$$
$$+ S^2\rho_3(1 - \nu_d)[\beta(\nu_d - \beta) - (1 + \beta)(\chi - \beta)]$$

$$\chi = \gamma_3(\gamma_4 - 1)/\gamma_4/(\gamma_3 - 1) - \bar{M}_3/\bar{M}_4 + 1$$

$$\frac{\partial G}{\partial\rho_3} = S^2\rho_3(1 - \nu_d)[\beta(\nu_d - \beta)$$
$$- (1 + \beta)(\chi - \beta)]\left[\frac{2}{S}\ \frac{\partial S}{\partial\rho_3} + \frac{(1 - 2\nu_d)}{\rho_3(1 - \nu_d)}\right]$$
$$- (1 + \beta)\ p_4/\rho_4 + (1 + \beta)(\nu_F - \chi)$$

$$\frac{\partial S}{\partial\rho_3} = -\ \delta S/\rho_3 + (\delta + \theta)\ S\ \frac{\partial p_3}{\partial\rho_3}/p_3$$

$$\frac{\partial p_3}{\partial\rho_3} = \frac{[(2\delta - 1)(1 - \nu_d) + \nu_d]\ S^2}{1 + 2\rho_3(\delta + \theta)\ S^2(1 - \nu_d)/p_3}$$

Using the momentum equation, Eq. (5.4), to determine $(\partial p_3/\partial\rho_3)$ proved a better choice than the Hugoniot equation, Eq. (5.5), for flame burning speeds approaching the Chapman-Jouguet value. A new value of p_3, based on the value of ρ_3 from Eq. (5.12), is obtained from Eq. (5.11). The iterative procedure ends when the changes in p_3 and ρ_3 fall below 10^{-N}, where N is a large integer. The particle velocity ahead of the deflagration is evaluated from

$$u_3 = u_4 + S(\nu_d - 1) \tag{5.13}$$

This completes the calculation of state 3.

In the case of a Chapman-Jouguet deflagration, Eq. (5.7) yields

$$S_{CJ}^2 = \frac{(1 + \beta)\ p_3}{\rho_3[2\nu_d - (1 + \beta)]} \tag{5.14}$$

Substituting Eq. (5.14) into Eq. (5.4) gives

$$\nu_d = \frac{P_d(1 + \beta)}{2P_d - (1 - \beta)} \tag{5.15}$$

Substituting Eq. (5.15) into Eq. (5.5) yields the following relation

$$p_3 = \frac{-\zeta + \sqrt{\zeta^2 - 4\xi\eta}}{2\xi} \tag{5.16}$$

where

$$\xi = (1 + \beta)^2 \frac{\rho_4}{P_4} (\nu_F - \chi) - (1 - \beta)$$

$$\zeta = 2\ [(1 + \beta)\chi - 2\beta]$$

$$\eta = (1 - \beta)\ [\beta - \chi\ (1 + \beta)]$$

and χ is defined as Eq. (5.12). The values of ρ_3, S_{CJ}, and u_3 are obtained from Eqs. (5.15), (5.14), and (5.13) respectively. This completes the calculation of state 3 in the Chapman-Jouguet case.

Figure 23 presents the difference scheme in symbolic form used in the floating flame fitting technique. The semi-circular backward and forward arrowed arcs represent application of the forward and backward jump equations. The algorithm in Fig. 23 consists of the following five steps:

1) forward predictor at i+1
2) forward jump equations at i+1
3) forward predictor at i-1
4) backwards corrector at i
5) backward jump equations at i

An algorithm applying the backward jump equations prior to the forward jump equations gives the same results.

5.4. Floating Detonation Fitting

Floating detonation fitting is based on the classical treatment of a detonation front as a shock front followed by a deflagration. Figure 24 describes this thermodynamic process on the pressure-specific volume plane for the case of a Chapman-Jouguet detonation. Symbol u refers to the upstream conditions, N corresponds to conditions at the von Neumann spike, and J denotes conditions associated with the Chapman-Jouguet deflagration. Not shown are overdriven detonations which correspond to the upper interaction point of the Rayleigh line with the Hugoniot curve. Rayleigh lines associated with overdriven detonations have slopes in the pressure-specific volume plane greater than the Rayleigh line corresponding to Chapman-Jouguet detonation. This occurs because overdriven detonations have wave speeds greater than the Chapman-Jouguet detonation wave speed.

The condition of tangency of the Rayleigh line to the Hugoniot at state J is expressed by

$$u_J + a_J = D \tag{5.17}$$

where u_J and a_J represent the particle and sound speed respectively at state J, and D refers to the detonation speed. The physical interpretation of Eq. (5.17) is that detonation states N and J only depend on the upstream conditions, state u, as information carried by downstream disturbances cannot penetrate the detonation front. This condition is reflected in the floating detonation fitting technique by using the following relations[59] to evaluate states N and J

$$M_D^2 = M_J^2 = M_N^2 = \frac{2P_{G_u} - (1 - \beta) + 2\sqrt{P_{G_u}^2 - (1 - \beta)\,P_{G_u} - \beta}}{(1 - \beta)\,\gamma_u} \tag{5.18}$$

where

$$P_{G_u} = 1 + \frac{P_{G_o} - 1}{(a_o/a_u)^2} + \gamma_J\left[1 - (a_o/a_u)^2\right](B\gamma_u/\gamma_J - \bar{M}_u/\bar{M}_J)$$

$$P_{G_o} = \frac{(1 + \beta)(\nu_F - \beta)}{1 - \beta} - \beta$$

$$B = \frac{\gamma_J - 1}{\gamma_i - 1} \quad \text{and} \quad \beta = \frac{\gamma_J - 1}{\gamma_J + 1}$$

and η_F represents the heat liberated during combustion in terms of the ratio of specific volumes at the undisturbed pressure.

$$P_J = \frac{(1 - \beta)}{2}\left(1 + \gamma_u M_D^2\right) \qquad P_N = (1 + \beta)\, M_N^2 - \beta \tag{5.19a,b}$$

where

$$p_J = P_J \cdot p_u \qquad p_N = P_N \cdot p_u$$

$$\rho_J = \rho_u \frac{P_J + \beta}{(P_J + \beta) - (1 - \beta)(P_J - P_{G_u})} \tag{5.20a}$$

$$\rho_N = \rho_u \frac{M_N^2}{\beta M_N^2 + (1 - \beta)} \tag{5.20b}$$

$$U_J = \left[\frac{(1 - \beta)(P_J - P_{G_u})(P_J - 1)}{\gamma_u (P_J + \beta)}\right]^{1/2} a_u + u_u \tag{5.21}$$

$$U_N = \left[(1 - \beta)\left(1 - \frac{1}{M_n^2}\right) Mn\right] a_u + u_u \tag{5.22}$$

Relations (5.18) to (5.22) are derived by applying the Chapman-Jouguet condition, Eq. (5.17), to the basic relations (5.3) to (5.5).

For overdriven detonations, a greater than inequality replaces the equality in relation (5.17). Under this circumstance, rarefaction waves can interact with the leading overdriven detonation front causing the overdriven detonation front to decay to Chapman-Jouguet detonation. The relations for evaluating the gasdynamic states associated with the overdriven detonation remain relations (5.19) to (5.22) with the exception that (5.19a) is replaced by[59]

$$P_J = \frac{(1 - \beta)}{2} \left(1 + \gamma_u M_D^2\right) + \Delta \tag{5.23}$$

where

$$\Delta^2 = \frac{(1 - \beta)^2}{2} \left(1 + \gamma_u M_D^2\right)^2 - \gamma_u (1 - \beta) M_D^2 P_{G_u} + \beta$$

The overdriven detonation Mach number is obtained from the analysis given in Section 3.2.2.

In the floating detonation fitting technique, the difference mesh remains at rest. Figure 25 shows the appropriate differencing scheme in symbolic form used in the two identifiable cases of detonation motion relative to the grid. The detonation trajectory is denoted by a thick solid line.

A detonation wave is modelled as a detonation front followed by a rarefaction wave. Even though the gasdynamic parameters are continuous across a rarefaction wave, MacCormack's scheme, a combination of two first order difference steps, can only accurately model a rarefaction of sufficient smoothness. Numerical experiments performed by the author indicate that the rarefaction associated with Chapman-Jouguet detonation must span at least five node spacings to avoid instability induced by truncation error. The number of node spacings depends on the strength of the rarefaction wave. When modelling the early stages of a detonation wave, in addition to using floating detonation fitting, the rarefaction is treated analytically. For a perfect gas the equations for the distribution of the gasdynamic parameters in a right running rarefaction are[59]

$$u_x = u_h + (u_h - u_t)\,\frac{(x_h - x_x)}{(x_h - x_t)} \tag{5.24}$$

$$a_x = \left(\frac{U_x}{\alpha} + 1\right) a_h \tag{5.25}$$

where

$$U_x = \frac{u_x - u_h}{a_h} \qquad \text{and} \qquad \alpha = \frac{2}{\gamma_h - 1}$$

$$\rho_x = \left(\frac{a_x}{a_h}\right)^{\alpha} \rho_h \tag{5.26}$$

$$p_x = \left[\frac{\alpha\gamma_h}{2 + \alpha}\left(\rho_x^{\frac{2+\alpha}{\alpha}} - 1\right) + 1\right] p_h \tag{5.27}$$

where subscripts h, t, and x represent the conditions at the head, tail, and intermediate point in the rarefaction wave respectively, and p, ρ, u, a, and x represent the pressure, density, particle velocity, speed of sound, and the position respectively. The velocity is distributed linearly in a rarefaction.

Figure 26 compares the pressure-, density-, and particle velocity-space profiles computed by 139 cycles of MacCormack's scheme with floating detonation fitting to the exact solution[18] for the case of a Chapman-Jouguet detonation in a closed end tube. The results are for a typical hydrocarbon-air mixture initially at atmospheric conditions. The profiles compare very well.

6. APPROACH AND SEPARATION OF DISCONTINUITIES

The Courant-Friedrichs-Lewy stability condition states that a necessary condition for stability is that the numerical domain of dependence at every point in the flow field include all of the partial differential equation's domain of dependence at least as the time step and mesh spacing approach zero. The floating discontinuity techniques were designed to satisfy this condition thereby ensuring that the computed motion of the discontinuities depended on all the data physically influencing the motion. Generally, the numerical domains of dependence exceed the analytical domains. These larger numerical domains of dependence do not increase the stability as the numerical scheme inherently neglects all but the physically meaningful data associated with the node under consideration. However, the accuracy of the solution can be upgraded if the numerical scheme is designed for that purpose.[60] The larger numerical domains arise in the attempt to include the physically pertinent data when the only data available exists at the equally spaced nodes.

The four discontinuities in the flow field explicitly treated in the numerical scheme are shock waves, contact surfaces, deflagrations, and detonations. When these discontinuities exist alone or are well separated from each other and points of symmetry, such as solid walls, Figs. 19, 22, 23, and 25 show the difference scheme to be applied in the immediate neighborhood of the discontinuity for each of the identifiable cases of the discontinuity's relative motion to the difference grid. The inner and outermost nodes specifically involved in the floating discontinuities computations define the left and right hand boundaries respectively of the numerical domain of dependence.

Deflagration has the smallest domain spanning only one mesh width on either side of the deflagration. Values at six nodal positions are included in the numerical domain of dependence for a right running shock or detonation crossing a node. A right running contact surface crossing a node has the largest numerical domain of dependence involving values at eight nodal positions and stretching over five cell spacings. Two discontinuities are well separated when the left hand boundary of the numerical domain of dependence of the leading discontinuity lies to the right of the trailing discontinuity's position during the time step Δt and the right hand boundary of the trailing discontinuity's numerical domain of dependence lies to the left of the leading discontinuity's initial location. When the discontinuities are not well separated from each other or points of symmetry, one or more of the predictor and/or corrector steps, indicated in Figs. 19, 22, 23, and 25, involve differentiation across a discontinuity. Therefore, special differencing algorithms about the discontinuities are necessary. Procedures for handling of discontinuities which are not well separated fall into two categories: (A) procedures involving only two discontinuities, (B) procedures involving three or more discontinuities. Generally, category (A) represents the approach of discontinuities prior to an interaction, while category (B) is representative of the flow field soon after the interaction process. The details of the interaction process are set forth in Chapter 3. The following interaction systems are included in category (A):

(1) shock-shock

(2) shock-contact discontinuity

(3) shock-deflagration

(4) shock-detonation

(5) contact discontinuity-shock

(6) contact discontinuity-deflagration

(7) contact discontinuity-rarefaction (interface)

(8) deflagration-shock

(9) deflagration-contact discontinuity

(10) detonation-shock

(11) detonation-contact discontinuity

(12) plane of symmetry-shock

(13) rarefaction-CJ detonation

Associated with each situation are the many ways in which the discontinuities are not well separated. In the actual computations, it proved helpful to print a code number identifying the particular way in which the two discontinuities were not well separated. The code is calculated as follows

$$\mathrm{NSC} = 100(\mathrm{NDISCL1} - \mathrm{NDISCL2} - \Delta) + 10\ \mathrm{NXS2} + \mathrm{NXS1} \tag{6.1}$$

where NDISCL1 and NDISCL2 are the mesh node numbers immediately behind the leading and trailing discontinuities respectively at time instant n. NXS1 and NXS2 indicate that the leading and trailing discontinuities cross over the node on the right, left or not at all during time interval n to n + 1 by assuming values 1, -1, or 0. Interaction systems involving a contact discontinuity assign Δ a value of unity, otherwise Δ equals zero. The appropriate numerical scheme and identifying code for all the ways for each of the first twelve interaction systems are presented in symbolic form in Fig. 27 to 38. Interaction system (13) has been discussed in Section 5.4. The chain dotted line in Fig. 33 denotes the trajectory of the tail of the rarefaction wave in interaction system (7). Interactions in which the identifying code points to another identifying code indicate that the difference scheme to be applied corresponds to the latter identifying code number. Asterisk superscripts denote that an interaction between discontinuities can occur. The interaction systems included in category B are:

(1) shock-shock-shock
(2) shock-shock-detonation
(3) shock-contact discontinuity-shock
(4) shock-contact discontinuity-deflagration
(5) shock-detonation-shock
(6) contact discontinuity-deflagration-shock
(7) contact discontinuity-rarefaction-CJ detonation
(8) detonation-shock-shock
(9) shock-contact discontinuity-deflagration-shock
(10) shock-contact discontinuity-rarefaction-CJ detonation

The first eight situations were treated by decomposing each situation into two appropriate category (A) situations and properly matching values at the interface. Situation (9) was decomposed into situation (6), category (B), and situation (2), category A, while situation (10) was decomposed into situation (7), category (B) and situation (2), category (A).

7. APPLICATION OF THE NUMERICAL METHOD

The flow diagram presented in Fig. 39 describes the executive instructions controlling the whole of the computational process. The block entitled FIDIF (Finite Difference) controls the application of the numerical method. The flow diagram of FIDIF is presented in Fig. 40.

The strategy in applying the numerical method is to first treat the flow field as it were devoid of discontinuities, and apply MacCormack's predictor corrector bilevel scheme at all points. By predicting values at nodes 1 to N and applying the corrector step in the reverse nodal order, that is, nodes N to 1, considerable savings in the required computer storage is realized.

The next step is to locate and identify the type of discontinuity nearest to the origin. A determination of whether the discontinuity is well separated in the flow field can be made after the next nearest discontinuity to the origin is located and identified. If the first discontinuity is well separated, then the flow field about the discontinuity is calculated using the appropriate floating discontinuity technique. Then the next nearest discontinuity to the origin is located and the type identified. Now the adequacy of the separation of the second discontinuity can be determined. This procedure continues until two discontinuities are not well separated. Again the location and type of the next nearest discontinuity is made. The adequacy of the separation between this discontinuity and the previous discontinuity is checked. If they are well separated, then the previous two discontinuities correspond to a situation in category (A), described in Chapter 6. The particular way in which the two discontinuities are not well separated is noted and the appropriate difference scheme is used. If the discontinuities undergo an interaction, the appropriate solution to the interaction is obtained by the polar method discussed in Chapter 3. However, if the test fails, all three discontinuities are inadequately separated and a check for the possibility of a fourth or more is made. In any event, the situation belongs in category (B), described in Chapter 6. If the three or more discontinuities undergo an interaction, the vector polar method provides the solution. This step terminates when all the discontinuities have been explicitly handled.

The next step is to reorder the discontinuities in the flow field by location. The purpose is to add and/or terminate in the appropriate positions those discontinuities which, as a result of an interaction process, have been introduced or eliminated into the flow field. Prior to applying the polar method to interactions which occur in the burnt medium and involve shock waves and contact discontinuities, an evaluation of their strengths is made. Shocks with Mach number less than 1.001 and contact surfaces with density differences less than 2% are terminated in the flow field. This avoids needlessly computing infinitessimal disturbances as discontinuities and unclutters the flow field. The last two steps are enclosed by a dashed line in Fig. 40. The remaining steps are to calculate particle paths and characteristic trajectories along which information is transmitted and to print out and write on tape a description of the flow field at desired intervals. This completes the description of the numerical method used in solving the problem. Appendix C contains a listing of the computational code FLAME.

8. RESULTS

8.1. Stable and Unstable Cases

Of particular interest in the study reported here was the evaluation of the non-steady effects caused by an abrupt change in the burning speed of an initially steady deflagration propagating in an unconfined combustible gas mixture. The numerical solutions presented here were obtained for the particular case of hydrocarbon-air mixtures initially at NTP conditions. In the computations, the burned and unburned media were considered to behave as thermally and calorically perfect gases with constant isentropic indices, $\gamma_u = 1.3$ and $\gamma_b = 1.2$ respectively. The energy deposited by combustion was expressed in terms of the specific volume ratio corresponding to a change of state at constant pressure, $\nu_F = 7$, and the ratio of molecular weights, M, set equal to one.

Time-space wave diagrams for two cases corresponding to the same initial burning speed of 9.6 meters per second but experiencing different increments in burning speed, ΔS, of 14.6 and 19.0 meters per second respectively, are illustrated in Figs. 41 and 42. The thick dashed lines in the wave diagrams denote deflagration, the solid lines represent shock fronts, the thick solid line denotes detonation, and thin dashed lines are the representative particle paths. In the first case, the disturbances associated with an increment in the burning speed of 14.6 meters per second have, after experiencing eight interactions with the deflagration, been reduced to sound waves. Meanwhile the deflagration's burning speed has increased to the final value of 37.2 meters per second. This process has taken approximately 240 x_f' milliseconds and the deflagration has travelled a distance equal to

53 x_f' meters, that is, 53 times the distance, expressed in meters, traversed by the deflagration prior to the actuation of the initial disturbance. The strength of the front of the leading blast wave has increased from a Mach number of 1.1 to 1.4. In this case, the non-steady disturbances have died down and the flow field that is established corresponds to another steady state self-similar solution as it was at the start.

In contrast, in the second case, the disturbances associated with the increment in the burning speed of 19.0 meters per second have been of sufficient intensity to trigger transition to detonation. Associated with the establishment of a detonation wave travelling at 2250 meters per second into the unburned medium is the occurrence of a retonation wave, Mach number 1.29, travelling at 1250 meters per second into the burned medium. these wave speeds are quite typical of those reported in the literature.[34] Detonation occurred at 76 x_f' milliseconds and at a distance of 20.3 x_f' meters from the origin. The Chapman-Jouguet detonation speed in the undisturbed medium is 1690 meters per second. The effect of the transition to detonation is manifested dramatically by the trajectories of the representative particle paths. Details of the transition to detonation, in the form of the trajectories of the forward and backward running characteristics denoted by thin solid lines, are illustrated in the time-space wave diagram in Fig. 43. The basis of their importance is that it is along these trajectories that gasdynamic information is transmitted from one part of the flow field to another. The time and space intervals used in Fig. 43 correspond to the area enclosed by the rectangle in Fig. 42.

At the establishment of detontation, the generation of a strong centered rarefaction wave attached to the back of the detonation front is evidenced. The strength of this rarefaction wave is qualitatively shown by the curvature of the backward running, u-a, characteristic behind the detonation front. The rapidly increasing extent of this centered rarefaction wave is indicated in Fig. 43 by the trajectory of the tail of the rarefaction, denoted by the chain double dotted line. Also the effect of the detonation-shock merging interaction, resulting in a non-steady overdriven detonation which decayed to the appropriate Chapman-Jouguet level, is aptly indicated by the separation of the trajectories of the backward running characteristics eminating from a position immediately ahead of and behind the point of interaction. The effect of the rarefaction fans on the particle path initiated at R = 18 is noted.

In addition, pressure-space profiles at various instants in time for the two cases are presented in Figs. 44 and 45. The vertical dashed lines denote a pressure change due to the deflagration process. The horizontal dashed lines, used only as a drawing aid to distinguish the individual profiles, indicate that over that portion of the profile the pressure equals that of the adjacent solid line. The pressure profile at T = 4.5 correspond to the spatial pressure distributions at the instant prior to the increment in the burning speed. In Fig. 45, the rapid pressure change which first appears at T = 22.3 and progresses into a rarefaction wave at T = 24.2 is attributed to the effect of the gasdynamics of overdriven detonation on the flow field in the immediate neighborhood of the point of interaction. The detonation-shock interaction occurred at T = 22.2. The last pressure profile at T = 25.8 shows that the flow field is rapidly approaching that given by a steady state self-similar Chapman-Jouguet analysis.[18]

For completeness, pressure histories at various locations are presented in Figs. 46 and 47 for the stable and unstable solutions respectively. The vertical and horizontal dashed lines maintain their meaning as previously explained. In Fig. 46 all the pressure histories can be approximated by a single pressure rise followed by a pressure drop, due to the deflagration process. Such a pressure signature is indicative of a flow field corresponding to a steady state self-similar solution.[20] However, in Fig. 47, the evidence of transition to detonation is provided for in the pressure signature by the rapid exponential-like decrease of pressure with distance.[16]

In this section the differences in the time-space wave diagrams and the differences in the spatial and time distributions of the most significant physical parameter, pressure, have been presented for two cases of increase in burning speed leading to steady state deflagration and transition to detonation. A significant similarity of the time-space wave diagram in the case of transition to detonation to that of Oppenheim and Stern's[34] time-space wave analysis of Schmidt et al.'s[7,8] experimental record of transition to detonation is noted.

8.2. Critical Curve

The most important conclusion reached on the basis of the results of the two cases presented in Section 8.1 is that for a given initial flame burning speed there exists a critical increment in the burning speed above which the acceleration process is capable of triggering the transition to detonation. Below this limit the flow field reaches a new steady state corresponding to a self-similar solution for the final deflagration velocity. Figure 48 presents the limit line between the steady, denoted by S, and non-steady unstable, denoted by U, regimes of solution in the case of plane-symmetrical flow. For example, with an initial flame burning speed of 6, 21, and 40 metes per second the critical increments in burning speed necessary to promote transition are 17.5, 10.0, and 2.5 meters per second respectively. The two cases considered in Section 8.1 are indicated in the stability plane by the "+" signs adjacent to the symbols S and U. The critical curves corresponding to flow geometries with line and point symmetry are expected to lie above the plane-symmetrical flow curve, that is, for each initial flame burning speed a larger increment in the burning speed would be required for transition than in the plane-symmetrical case. The expected greater stability with increasing geometrical symmetry is attributed to the decreasing maximum value of the thermodynamic variables with increasing geometrical symmetry for the same normal flame burning speed. For example, see Figs. 4 and 5.

Included in the stability plane are three data points obtained from the experimental results of Dörge, Pangritz and Wagner.[17] They reported that large spherical stoichiometric hydrocarbon-air mixtures, after being centrally ignited, burned at constant velocity. Then upon passing through a screen the flames experienced a significant increase in burning speed, but did not transist to detonation. Rather they developed a new thermodynamic and gasdynamic state commensurate with the higher flame burning speed. The largest such increases in burning rates, factors of six and twelve for single and triple screens respectively, occurred in the case of stoichiometric C_2H_2-air. These conditions are denoted by points W_{S_1} and W_{S_3} respectively located in the stable lower portion of the stability plane. However, they reported that in the case of larger initial flame propagation rates, on the order of 50 meters per second, it was indeed possible to trigger transition to detonation. In fact, their results indicate that the higher the initial flame propagation speed, the fewer number of screens were required to promote transition. These higher burning speeds were achieved in C_2H_2-air mixtures by reducing the N_2 content to 36%. For the case of a single screen, the data point is indicated by W_{U_1}.

The experimental results of Dörge et al.[17] corroborate the predicted critical stability curve.

9. SUMMARY AND CONCLUSIONS

9.1. Summary

A brief summary of the more significant aspects of this work follow:

1) Transition to detonation is dependent on the interaction processes between waves generated by the accelerating flame.
2) Proper account of the progress of shock waves, contact discontinuities, detonations, and flames in the flow field can be achieved by treating these elementary wave processes as plane finite waves.
3) The flame burning speed law governing the motion of the deflagration was expressed in the form of a power law dependence on the pressure and temperature immediately ahead of the deflagration.
4) The vector polar method provided a facility for solving the problem of multiple wave interactions involving non-reactive and reactive waves.
5) The conservation equations, governing the non-steady motion of the flow field, formed a system of non-linear hyperbolic equations which had to be solved numerically.
6) MacCormack's algorithm, an explicit two step second order accurate finite difference technique utilizing implicit artificial viscosity, was employed to integrate the governing equations expressed in divergence form. Time steps were computed using the Courant-Freidrichs-Lewy stability condition times the factor 0.7.

7) Shock fronts, contact discontinuities, detonations, and deflagrations were computationally maintained as sharp discontinuities through the technique of floating discontinuity fitting. A technique based on the solution of the Riemann problem was proposed for the treatment of contact discontinuities. Detonation fronts were modelled as a shock front followed by a deflagration, thereby allowing for the possibility of obtaining overdriven as well as Chapman-Jouguet detonations. The most important principle in the floating flame fitting technique was that at each time interval the Eulerian computational grid is shifted throughout the whole flow field with a speed equal to that of the deflagration. When detonation is established the computational mesh is stationary.

8) Associated with each of the floating discontinuity techniques was a zone of influence. When two or more zones of influence overlaped, more elaborate difference algorithms were employed.

9) The results of the computations for two cases, one leading to the transition to detonation and the other to a steady state self-similar flow field were presented in the form of time-space wave diagrams, pressure-space profiles and pressure-time signatures.

10) Presented in the stability plane, for the case of plane-symmetrical flow is a limit curve, a line of demarcation between the steady and non-steady regimes of solution. Agreement with experiments was noted.

9.2. Conclusions

A numerical technique for the analysis of non-steady flow fields generated by accelerating flames in gaseous media has been introduced. This versatile technique is not limited in scope to flow fields containing only blast waves, nor are the combustion processes restricted to the case of detonations or Chapman-Jouguet deflagrations.[25,26] Rather the numerical technique is capable of responding to and tracing the flow field processes which occur during the development of detonation. Contrary to many currently proposed numerical schemes,[24,25] this technique is devoid of explicit artificial viscosity. The influence of the existing implicit artificial viscosity has been minimized through the judicious choice of the basic algoritm of MacCormack and the implementation of floating discontinuity fitting techniques. In addition, provisions for the evaluation of interaction phenomena involving reactive and non-reactive waves, necessary for the transition to detonation, are included.

The most significant conclusion derived from the results of this work is that in the plane of increment in flame burning speed and its initial value, the possible solutions to the problem of an initially steady flame experiencing an instantaneous increment in burning speed can be divided into steady and non-steady regimes. That is, for each and every initial burning speed there exists a critical burning speed increment above which the acceleration process is capable of triggering transition to detonation. Below this limit the flow field reaches a new steady state commensurate with a final higher flame burning speed. the significance of this result is that in so far as it has been well established[6] that if a sufficient amount of energy is deposited at a sufficiently high rate in an explosive gas a detonation wave is directly initiated, while below a certain critical value of the initiation energy and power there ensues a flame front that gradually recedes from that of the leading blast wave, no indication as to the level of disturbance necessary to trigger transition from deflagration to detonation has ever been reported. The importance of the existence of a critical level of disturbance necessary to cause acceleration to detonation is that in the event of an accidental fuel spill,[12-15] the potential danger resulting from the blast waves generated by the exploding cloud practically depends on the combustion process[16] by which the cloud is consummed. Detonative combustion of the cloud produces a higher level of damage than in the case of deflagrative combustion. However, it is noted that by time the leading front reaches a distance of five initial cloud radii, these individual differences fade away. Therefore, knowledge of this critical level of disturbance should make it possible to minimize the occurrence of detonation by prohibiting design configurations, including the location of energy sources and turbulence sources which could directly or indirectly trigger detonation.

ACKNOWLEDGEMENTS

The author expresses his gratitude to Professors H. Dwyer and A. J. Chorin for their expert advice on numerical analysis and enlightening discussions throughout the course of this research. He also wishes to thank his wife, Anne, and his family for their continued support and encouragement. Finally, the author acknowledges his deepest appreciation to his research advisor, Professor A. K. Oppenheim, whose foresight, inspiration and guidance over the past years have provided the author both the opportunity and the ability to meet future challenges.

This work was done with support from the U. S. Department of Energy.

APPENDIX A. CHAPMAN-JOUGUET FLAME BURNING SPEED LAW

For the case of a thermally and calorically perfect gas, the conservation equations governing the motion of the deflagration can be expressed as[59]

Continuity: $\nu_d = v_4/v_3 = (S + u_3 + u_4)/S$ (A.1)

Momentum: $P_d = p_4/p_3 = 1 + \gamma_3 S^2(1 - \nu_d)/a_3^2$ (A.2)

Hugoniot: $(P_d + \beta)(\nu_d - \beta) = (1 + \beta)(\nu_f - \beta)$ (A.3)

where

$$\nu_f = \left(\frac{a_o}{a_3}\right)^2 \nu_F + \left[1 - \left(\frac{a_o}{a_3}\right)^2\right]\left[\frac{\gamma_3}{\gamma_4}\;\frac{\gamma_4 - 1}{\gamma_3 - 1} - \frac{\bar{M}_3}{\bar{M}_4} + 1\right]$$

$$\beta = \frac{\gamma_4 - 1}{\gamma_4 + 1}$$

where p, v, a, and u are the gasdynamic parameters pressure, specific volume, speed of sound, and particle velocity respectively, S is the deflagration burning speed, $\bar{M}$ is the mean molecular weight and the ratio of specific heats. Subscripts 0, 3 and 4 denote the undisturbed reference state, and the uncombusted and combusted states immediately ahead of and behind the deflagration respectively. The Hugoniot relation has been written in terms of a hyperbolic relation where the heat liberated during combustion is expressed in terms of ν_F, the ratio of specific volumes at the undisturbed pressure.

Substituting Eqs. (A.2) into (A.3) and solving for the specific volume ratio yields

$$\left[(1+\beta) + \frac{\gamma_3 s^2}{a_3^2}(1-\gamma_d)\right](\nu_d - \beta) - (1+\beta)\,\nu_f - \beta(1+\beta) = 0 \tag{A.4}$$

$$\left(\frac{\gamma_3 s^2}{a_3^2}\right)\nu_d^2 - \left[(1+\beta)\left(1+\frac{\gamma_3 s^2}{a_3^2}\right)\right]\nu_d + \left[(1+\beta)\,\nu_f + \frac{\beta\gamma_3 s^2}{a_3^2}\right] = 0 \tag{A.5}$$

Therefore

$$\nu_d = \frac{\xi \pm \sqrt{\zeta}}{\eta} \tag{A.6}$$

where

$$\eta = \frac{2\gamma_3 s^2}{a_3^2} \qquad \text{and} \qquad \xi = (1+\beta)\left(1+\frac{\eta}{2}\right)$$

$$\zeta = \xi^2 - 2\eta\left[\beta\frac{\eta}{2} + (1+\beta)\,\nu_f\right]$$

The two roots correspond to strong and weak deflagration. Chapman-Jouguet correspond to $\zeta = 0$. That is

$$(1+\beta)^2\left(1+\frac{\gamma_3 S^2}{a_3^2}\right)^2 - \frac{4\gamma_3 S^2}{a_3^2}\left(\frac{\beta\gamma_3 S^2}{a_3^2}+(1+\beta)\ \nu_f\right)=0 \tag{A.7}$$

Expanding in powers of S^2 yields

$$\left[\frac{\gamma_3}{a_3^2}(1-\beta)\right]^2 S^4 + \left[\frac{2\gamma_3}{a_3^2}(1+\beta)(1+\beta-2\nu_f)\right] S^2 + (1+\beta)^2 = 0 \tag{A.8}$$

Solving for S^2 gives

$$S^2 = \frac{(1+\beta)\ a_3^2}{\gamma_3(1-\beta)^2} \cdot \left[(2\nu_f-\beta-1) \pm 2\sqrt{\beta-\nu_f-\beta\nu_f+\nu_f^2}\right] \tag{A.9}$$

$$S^2 = \frac{(1+\beta)\ a_3^2}{\gamma_3(1-\beta)^2} \cdot \left[(\nu_f-1)+(\nu_f-\beta) \pm 2\sqrt{(\nu_f-\beta)(\nu_f-1)}\right] \tag{A.10}$$

$$S^2 = \frac{(1+\beta)\ a_3^2}{\gamma_3(1-\beta)^2}\left[\sqrt{\nu_f-1} \pm \sqrt{\nu_f-\beta}\right]^2 \tag{A.11}$$

Therefore

$$S = \frac{a_3}{1 - \beta} \sqrt{\frac{1 + \beta}{\gamma_3}} \left[\sqrt{\nu_f - \beta} - \sqrt{\nu_f - 1}\right] \qquad \text{(A.12)}$$

where the positive subsonic character of the flame burning speed has been used to eliminate the extraneous solutions. In terms of temperature, the Chapman-Jouguet flame burning speed law becomes

$$S_{CJ} = \frac{\sqrt{(1 + \beta)\ R_3 T_3}}{1 - \beta} \left[\sqrt{\nu_f - \beta} - \sqrt{\nu_f - 1}\right] \qquad \text{(A.13)}$$

where

$$\nu_f = \left(\frac{T_o}{T_3}\right) \nu_F + \left(1 - \frac{T_o}{T_3}\right)\left(\frac{\gamma_o}{\gamma_4} \ \frac{\gamma_4 - 1}{\gamma_o - 1} - \frac{\bar{M}_3}{\bar{M}_4} + 1\right)$$

$$\beta = \frac{\gamma_4 - 1}{\gamma_4 + 1}$$

where subscript 0, 3, and 4 refer to the reference, uncombusted, and combusted states respectively.

APPENDIX B. PROGRAMS FOR THE INITIAL CONDITIONS

```
      PROGRAM ZF(INPUT,OUTPUT,PUNCH)
C-    PHASE PLANE INTEG IN FL, FOR CONST VEL UNIF INITIAL DENSITY DEFLAG
C
      DZLDFL(AJ,G,FL,ZL)=(2.*(EXP(ZL)-(1.-EXP(FL))**2)+AJ*(G-1.)*(1.-EXP
     *(FL))*EXP(FL))/((AJ+1.)*EXP(ZL)-(1.-EXP(FL))**2)
      DXLDFL(AJ,G,FL,ZL)=-(EXP(ZL)-(1.-EXP(FL))**2)/((AJ+1.)*EXP(ZL)-
     *(1.-EXP(FL))**2)
C
C-    DATA INPUT
C     NJOBS=NO. OF CASES TO BE RUN
C     Z0=Z(PISTON)
C     F0=F(PISTON)
C     X0=INITIAL VALUE OF X/XP
C     S0=INITIAL STEP SIZE(INPUT AS A NEGATIVE NO.)
C     DELO=REQD ACC IN (Z2-Z) TO TERMINATE THIS CASE
C     J=0,1,2 SYMMETRY FACTOR
C     M=MAX NO. OF ITERATIONS ALLOWED TO ACHIEVE DELO
      READ 1000,NJOBS
 1000 FORMAT(I5)
      DO 1 JOB=1,NJOBS
      READ 1001,G,Z0,F0,X0,S0,DELO,J,M
 1001 FORMAT(F10.8,F10.4,2F10.8,2F10.5,2I5)
C
C-    PRINT INPUT DATA
      PRINT 1002,J,G,Z0,F0,X0,S0,DELO,M
 1002 FORMAT(1H1,2/,20X,*SELF SIMILAR, CONST FRONT VEL, UNIF INITIAL DEN
     *SITY, EULERIAN SPACE BLAST WAVE*,2/,5X,*J=*,I1,5X,*G=*,F5.3,5X,
     **Z0=*,F15.8,5X,*F0=*,F10.8,5X,*X0=*,F 10.8,5X,*S0=*,F7.5,5X,
     **DELO=*,F10.8,5X,*M=*,I4,3/,16X,*Z*,22X,*F*,16X,*X/XP*,16X,*Z2-Z*
     *,/)
C
C-    INITIAL CONDITIONS
      F=F0
      Z=Z0
      X=X0
      FL=ALOG(F)
      ZL=ALOG(Z)
      XL=ALOG(X)
      AJ=J
      S=S0
      PRINT 1003,Z,F,X
 1003 FORMAT(1H ,4(5X,F16.10))
C
      DO 2 KK=1,M
      IF(F.LT.0.0.OR.F.GT.1.0) GO TO 2
C     INTEGRATION  RUNGE KUTTA
      A0=S*DZLDFL(AJ,G,FL,ZL)
      A1=S*DZLDFL(AJ,G,FL+.5*S,ZL+.5*A0)
      A2=S*DZLDFL(AJ,G,FL+.5*S,ZL+.5*A1)
      A3=S*DZLDFL(AJ,G,FL+S,ZL+A2)
      B0=S*DXLDFL(AJ,G,FL,ZL)
      B1=S*DXLDFL(AJ,G,FL+.5*S,ZL+.5*A0)
      B2=S*DXLDFL(AJ,G,FL+.5*S,ZL+.5*A1)
      B3=S*DXLDFL(AJ,G,FL+S,ZL+A2)
      ZL=ZL+(A0+2.*A1+2.*A2+A3)/6.
      XL=XL+(B0+2.*B1+2.*B2+B3)/6.
```

```
      FL=FL+S
      Z=EXP(ZL)
      F=EXP(FL)
      X=EXP(XL)
C
C-    BC AT SHOCK FRONT
      Z2=(G-1.)*(1.-F)*(2./(G-1.)+F)/2.
      DEL=Z2-Z
C
C     PRINT RESULTS
C     PRINT 1003,Z,F,X,DEL
      IF(DEL) 3,5,5
 3    FLSAV=FL
      ZLSAV=ZL
      XLSAV=XL
      GO TO 2
 5    IF(DELO-DEL) 4,6,6
 4    FL=FLSAV
      ZL=ZLSAV
      XL=XLSAV
      S=S/2.
 2    CONTINUE
C
 6    F2=F
      Z2=Z
      Y=1.-(G+1.)*F2/2.
      XF=1./X
      AMACH=1./SQRT(Y)
      PRINT 1004,J,G,ZO,FO,Y,XF,Z2,F2,AMACH
 1004 FORMAT(1H ,3/,1X,*J=*,I1,3X,*G=*,F6.4,3X,*ZO=*,F16.10,3X,*FO=*,
     *F12.10,3X,*Y=*,F15.12,3X,*XF=*,F15.12,/,1X,*Z2=*,F15.12,5X,
     **F2=*,F15.12,5X,*AMACH=*,F15.10)
      PRINT 3000,KK
 3000 FORMAT(1H ,5X,*KK=*,I5)
C-    PUNCH DATA
      PUNCH 1005,J,G,DELO,ZO,FO,Y,XF,Z2,F2
 1005 FORMAT(I1,F7.4,E12.5,2E20.12,/,4E20.12)
 1    CONTINUE
C
      STOP
      END
```

```
     1
1.3         4.855      1.0         1.0         -0.006    .000001        1 2000
*******************************END OF DATA*******************************
```

```
      PROGRAM DFLAG(INPUT,OUTPUT,PUNCH)
C-    PHASE PLANE INTEG IN FL, FOR CONST VEL UNIF DEN DFLAG OR PIST PBM
C-    SELFSIMILAR EULERIAN SPACE PROFILES
      DIMENSION XUPV(4,110)
C
      DZLDFL(AJ,G,FL,ZL)=(2.*(EXP(ZL)-(1.-EXP(FL))**2)+AJ*(G-1.)*(1.-EXP
     *(FL))*EXP(FL))/((AJ+1.)*EXP(ZL)-(1.-EXP(FL))**2)
      DXLDFL(AJ,G,FL,ZL)=-(EXP(ZL)-(1.-EXP(FL))**2)/((AJ+1.)*EXP(ZL)-
     *(1.-EXP(FL))**2)
C
C
C-    DATA INPUT
C     NJOBS=NO. OF CASES TO BE RUN
C-    FOLLOWING 9 PIECES OF DATA OBTAINED FROM PROGRAM ZF
C     J=0,1,2 SYMMETRY FACTOR
C     G=UNCOMBUSTED VALUE OF THE SPECIFIC HEAT RATIO
C     DELO=ACC TO WHICH Z2 WAS CALC IN PROG  ZF-KURYLO
C     Z0=INITIAL VALUE OF Z (I.E. Z-PISTON)
C     F0=INITIAL VALUE OF F (I.E. F-PISTON)
C     Y=1/(MACH NO. **2)
C     XP=INITIAL VALUE OF X (I.E. X-PISTON)
C     Z2=VALUE OF Z AT SHOCK FRONT
C     F2=VALUE OF F AT SHOCK FRONT
C-
C     G4=COMBUSTED VALUE OF THE SPECIFIC HEAT RATIO
C     AA=UNDISTURBED VELOCITY OF SOUND IN M/SEC
C     R4R1=RATIO OF MOLECULAR WTS. (STATE1/STATE4)
C     S0=INITIAL STEP SIZE
C     ANUF0=SPEC VOL RATIO FOR CONST PRESS DEFLAG (WRT. STATE1)
C     DELNU=ACC REQD IN CALC ANUF (THIS FIXES THE FLAME POSITION)
C     M=MAX NO. OF DO LOOP TO GENERATE FLOW PROFILE
C     NPUNCH=(0,1),(NO PUNCHING REQD THIS CASE,PUNCHING REQD)
C     NPIST=(0,1) (THIS IS A DFLAG RUN, THIS IS A PISTON PBM RUN)
C     NC=NO. OF CELLS BETWEEN X=0.0 AND X=1.0
C     WWN=SHOCK FRONT VELOCITY IN M/SEC
C     AMACH=FRONT MACH NO
      READ 1000,NJOBS
 1000 FORMAT(I5)
      DO 1 JOB=1,NJOBS
      READ 1001,J,G,DELO,Z0,F0,Y,XP,Z2,F2
 1001 FORMAT(I1,F7.4,F12.5,2E20.12,/,4E20.12)
      READ 1002,G4,AA,R4R1,S0,ANUF0,DELNU,M,NPUNCH,NPIST,NC
 1002 FORMAT(F5.3,5F10.7,I5,I2,I3,I5)
      AMACH=1./SQRT(Y)
      WWN=AA*AMACH
      Z1=Y
      T2T1=Z2/Z1
      P2P1=(G-1.)/(G+1.)*(2.*G/(G-1.)/Y-1.)
      V2V1=(G-1.+2.*Y)/(G+1.)
      D2D1=1./V2V1
C
C-    PRINT INPUT DATA
      IF(NPIST.EQ.0) PRINT 1010
      IF(NPIST.EQ.1) PRINT 1011
 1010 FORMAT(1H1,/,25X,*SELFSIMILAR EULERIAN SPACE PROFILES FOR CONST VE
     *L UNIF INITIAL DENSITY DEFLAGRATION*,2/)
```

```
 1011 FORMAT(1H1,/,25X,*SELFSIMILAR EULERIAN SPACE PROFILES FOR CONST VE
     *L UNIF INITIAL DENSITY PISTON PROBLEM*,2/)
      PRINT 1003,Z0,F0,XP,Y,Z2,F2,DEL0,J,G,G4,AA,R4R1,AMACH,WWN,S0,
     *ANUF0,DELNU,M,NPUNCH,NPIST,NC,T2T1,P2P1,D2D1
 1003 FORMAT(1H ,1X,*Z0=*,F14.7,3X,*F0=*,
     *F9.7,5X,*XP=*,F12.10,5X,*Y=*,F12.10,5X,*Z2=*,F12.10,5X,*F2=*,
     *F12.10,4X,*DEL0=*,F11.9,/,* J=*,I1,5X,*G=*,F5.3,5X,*G4=*,F5.3,5X,
     **AA=*,F10.6,5X,*R4R1=*,F7.5,5X,*AMACH=*,F12.9,5X,*WWN=*,F15.10,/,
     *1X,*S0=*,F7.5,5X,*ANUF0=*,F7.5,5X,*DELNU=*,F10.7,5X,*M=*,I4,5X,
     **NPUNCH=*,I1,5X,*NPIST=*,I1,5X,*NC=*,I3,
     *                     2/,5X,*T2/T1=*,F12.8,5X,*P2/P1=*,F12.8,5X,
     **D2/D1=*,F12.8,3/, 5X,*F*,15X,*Z*,16X,*X*,12X,*U/SQRT(PV)*,10X,
     **P/P1*,13X,*D/D1*,12X,*T/T1*,12X,*ANUF*, /)
C
C
C-    INITIAL PROFILE VALUES
C
C-    SYMBOL REPRESENTATION
C     H=DENSITY/DENSITY2
C     UPV=PARTICLE VEL/SQRT(P1/D1)
C     ANUF=CALCULATED SPECIFIC VOL RATIO (VF/V1) FOR THE R.L.
C     SF=FLAME VEL IN M/SEC RELATIVE TO PARTICLES IN STATE3
C     KP=PRINT ONLY 53 LINES PER PAGE
C     KGC=0,1,2
      UAA=XP*WWN*F0/AA
      UPV=UAA*SQRT(G)
      TT2=Z0*XP**2/Z2
      TT1=TT2*T2T1
      H=TT2**(1./(G-1.))
      VV1=V2V1/H
      DD1=1./VV1
      PP2=H**G
      PP1=PP2*P2P1
C
C-    PRINT INITIAL PROFILE VALUES
      PRINT 1004,F0,Z0,XP,UPV,PP1,DD1,TT1
 1004 FORMAT(1H ,  1X,F14.12,5E17.10,E16.9,5X,*INFINITY*)
C
C-    INITIAL CONDITIONS
      F=F0
      Z=Z0
      X=XP
      ZL=ALOG(Z)
      FL=ALOG(F)
      XL=ALOG(X)
      S=S0
      AJ=J
      KP=10
      KGC=0
C
      IF(NPIST.EQ.0) GO TO 15
      KGC=1
      IF(NPUNCH.LE.0) GO TO 15
      XM1=XUPV(1,1)=X
      UM1=XUPV(2,1)=UPV
      PM1=XUPV(3,1)=PP1
```

```
      DM1=XUPV(4,1)=DD1
      DX=1./FLOAT(NC)
      NC=2
      XUPV(2,1)=X+DX
C
 15   DO 2 MM=1,M
      IF(F.LT.0.0.OR.F.GT.1.0) GO TO 1
C
C     INTEGRATION  RUNGE KUTTA
      A0=S*DZLDFL(AJ,G,FL,ZL)
      A1=S*DZLDFL(AJ,G,FL+.5*S,ZL+.5*A0)
      A2=S*DZLDFL(AJ,G,FL+.5*S,ZL+.5*A1)
      A3=S*DZLDFL(AJ,G,FL+S,ZL+A2)
      B0=S*DXLDFL(AJ,G,FL,ZL)
      B1=S*DXLDFL(AJ,G,FL+.5*S,ZL+.5*A0)
      B2=S*DXLDFL(AJ,G,FL+.5*S,ZL+.5*A1)
      B3=S*DXLDFL(AJ,G,FL+S,ZL+A2)
      ZL=ZL+(A0+2.*A1+2.*A2+A3)/6.
      XL=XL+(B0+2.*B1+2.*B2+B3)/6.
      FL=FL+S
      Z=EXP(ZL)
      F=EXP(FL)
      X=EXP(XL)
C
      UAA=X*WWN*F/AA
      UPV=UAA*SQRT(G)
      TT2=Z*X**2/Z2
      TT1=TT2*T2T1
      H=TT2**(1./(G-1.))
      VV1=V2V1/H
      DD1=1./VV1
      PP2=H**G
      PP1=PP2*P2P1
      ANUF=X*X*Z/Z1*(1./(1.-F)-G/G4*F/Z*(1.+(G4-1.)/2.*F))-
     *(G*(G4-1.)/G4/(G-1.)-R4R1+1.)*(X*X*Z/Z1-1.)
C
C-    PRINT RESULTS OF THIS STEP
      KP=KP+1
      IF(KP-53.LT.0) GO TO 3
      PRINT 1005
 1005 FORMAT(1H1,/, 9X,*F*,15X,*Z*,16X,*X*,12X,*U/SQRT(PV)*,10X,*P/P1*,
     *13X,*D/D1*,12X,*T/T1*,12X,*ANUF*,/)
      KP=0
 1009 FORMAT(1H ,1X,F14.12,5E17.10,2E16.5)
 3    PRINT 1009,F,Z,X,UPV,PP1,DD1,TT1,ANUF
C
      IF(KGO) 4,4,6
 4    DEL=ANUFO-ANUF
      IF(DEL) 7,8,8
 8    IF(DELNU-DEL) 9,10,10
C
 10   V4V3=1./(1.-F)
      V4V1=V4V3*VV1
      D4D1=1./V4V1
      P4P3=1.+G/Z*(1.-V4V3)/V4V3/V4V3
      P4P1=P4P3*PP1
```

```
      Z4=G4/G*P4P3*V4V3*Z
      T4T3=Z4*G/Z/G4/R4R1
      SF=X*WWN*(1.-F)
      SFPV=SF/AA*SQRT(G)
      PRINT 1009,F,Z,X,UPV,PP1,DD1,TT1,ANUF
      PRINT 1008,P4P3,V4V3,T4T3,P4P1,D4D1,ANUF0,SF,SFPV
 1008 FORMAT(1H ,/,1X,*P4/P3=*,E16.10,3X,*V4/V3=*,E16.10,3X,*T4T3=*,E
     *16.10,3X,*P4/P1=*,E16.10,3X,*D4/D1=*,E16.10,
     *2/,10X,      *THE LAST TWO LINES ABOVE CORRES TO THE FLAME FRONT
     *(STATE3) FOR WHICH ANUF0=*,F15.10,/,10X,*FLAME SPEED=*,F17.12,
     *1X,*M/SEC*,5X,*SF/PV= *,F16.12)
      PRINT 3010,MM
 3010 FORMAT(1H ,5X,*MM *,I5)
C
C-    PUNCHING
      IF(NPUNCH.LE.0) GO TO 11
      DX=1./FLOAT(NC)
      NCFLM=NC=X/DX+2
      PRINT 1006,X,NC
 1006 FORMAT(1H ,2/,10X,*FLAME POSITION *,F13.10,* CORRES TO MESH PT NO
     **,I4)
      DO 17 L=1,NC
      XUPV(1,L)=X-FLOAT(NC-L)*DX
      XUPV(2,L)=0.0
      XUPV(3,L)=P4P1
 17   XUPV(4,L)=D4D1
      XUPV(1,NC+1)=X
      XUPV(2,NC+1)=UPV
      XUPV(3,NC+1)=PP1
      XUPV(4,NC+1)=DD1
      NC=NC+2
      XUPV(1,NC)=X+DX
C
 11   FLM1=FL
      ZLM1=ZL
      XLM1=XL
      KGO=1
      S=S0
      KF=53
      GO TO 50
 7    FLM1=FL
      ZLM1=ZL
      XLM1=XL
      GO TO 2
 9    S=S/2.
 14   FL=FLM1
      ZL=ZLM1
      XL=XLM1
      GO TO 2
C
 6    IF(KGO.EQ.2) GO TO 12
      IF(F.LT.F2) GO TO 13
      IF(ABS(PP1-1.).GT.5.E-6.OR.ABS(UPV).GT.5.E-6.OR.ABS(DD1-1.).GT.
     *5.E-6) GO TO 53
      DP=PP1-1. $ KGO=2
      PRINT 1022,DP,X
```

```
 1022 FORMAT(1H ,2/,1X,*SHOCK TREATED AS SND WVE AS SHK STRNGTH= *,E15.7
     *,* AT X= *,E15.7)
      GO TO 54
53    FLM1=FL
      ZLM1=ZL
      XLM1=XL
      GO TO 50
 13   S=ALOG(F2)-FLM1
C-    REMEMBER S IS A NEG QTY
      KGO=2
      GO TO 14
C
 12   ZZ=.5*(G-1.)*(1.-F)*(2./(G-1.)+F)
      DEL=ZZ-Z
      PRINT 1007,DEL,DELO
 1007 FORMAT(1H ,/,10X,*ACC IN EVAL SHOCK FRONT POSITION ON THIS RUN=*,
     *E11.4 ,9X,*ACC IN EVAL IT IN PROG ZF-KURYLO=*,E11.4 ,/)
54    IF(NPUNCH.LE.0) GO TO 1
      GO TO 52
C
 50   IF(NPUNCH.LE.0) GO TO 2
 52   IF(X.GT.XUPV(1,NC)) GO TO 51
      IF(KGO.EQ.2) GO TO 1
      XM1=X $ UM1=UPV $ PM1=PP1 $ DM1=DD1
      GO TO 2
 51   XUPV(2,NC)=UM1+(XUPV(1,NC)-XM1)/(X-XM1)*(UPV-UM1)
      XUPV(3,NC)=PM1+(XUPV(1,NC)-XM1)/(X-XM1)*(PP1-PM1)
      XUPV(4,NC)=DM1+(XUPV(1,NC)-XM1)/(X-XM1)*(DD1-DM1)
      NC=NC+1
      XUPV(1,NC)=XUPV(1,NC-1)+DX
      XM1=X $ UM1=UPV $ PM1=PP1 $ DM1=DD1
      IF(KGO.EQ.2) GO TO 1
 2    CONTINUE
C
 1    PRINT 3010,MM
      IF(NPUNCH.LE.0) STOP
C
      NCP5=NC+5
      DO 60 N=NC,NCP5
      XUPV(1,N)=XUPV(1,N-1)+DX
      XUPV(2,N)=0.0 $ XUPV(3,N)=1.0
 60   XUPV(4,N)=1.0
      PRINT 1020
 1020 FORMAT(1H1,2/,* CELL NO*,7X,*X*,13X,*U/SQRT(P1/D1)*,12X,*P*,17X,
     **D/D1*,15X,*E*,/)
      DO 61 M=1,NCP5
      GE=G4 $ IF(M.GT.NCFLM) GE=G
      E=XUPV(3,M)/XUPV(4,M)/(GE-1.)+XUPV(2,M)**2/2.
      PRINT 1021,M,(XUPV(I,M),I=1,4),E
 1021 FORMAT(1H ,I4,2X,E16.9,4(3X,E15.9))
 61   PUNCH 2000,M,XUPV(1,M),XUPV(2,M) , M,XUPV(3,M),XUPV(4,M),E
 2000 FORMAT(I5,2E25.17,/,I5,3E25.17)
C
      STOP
      END
```

```
    1
1 1.3000 1.00000E-06  4.855CCCCCC0C0E+0C  1.CCCCCCCCCCCCE+C0
  8.€27472€4S4S3E-01  4.3913€3€83572E-01  8.9€41S4825352E-01  1.19350187000€E-01
1.20 330.€50873 1.0         -C.CC4  7.C        C.CCCCC1   5600 0  0  100
********************************END CF CATA************************************
```

APPENDIX C. PROGRAM FLAME

```
      PROGRAM FLAME(INPUT,OUTPUT,PUNCH,TAPE1)
      COMMON/UBDCND/NUBC
      COMMON/PARAM/N,J,AJ,G,GF,DELR,NFLM
      COMMON/TIME/T,DT,DTL,TWRITE,DELT,DTDX,AT
      COMMON/FIRFFTC/INDEX,NCYCLE,NN,NNN,NSTORE,NS,NITRCTN
      COMMON/FIR/LSTART,TERMIN,NCELL,NSTEPS,NDISC,NPUNCH,PF,DF,UF
      COMMON/ARRAYS/R(501),U(2,501),P(2,501),D(2,501),E(2,501),RR(501)
      COMMON/DISCS/NTDISC,NDISCNO(51),NTYPE(51),NDISCL(51),SE(51),SL(51)
     *,RD(51),NTDISCT,NTDISCS
C
C-----RESULTS FOR PLANE-SYMMETRIC FLOW FIELDS
C     LSTART=NO OF RUNS FOR THIS CASE THUS FAR
C     NCYCLE=NO OF CYCLES ASOF LSTART=0
C     NPUNCH=(0,1) (NO PUNCH,PUNCH)
C     NSTORE=(0,1) (DO NOT WRITE ON TAPE,WRITE ON TAPE)
C     NSTEPS=MAX NO OF CYCLES FOR THIS RUN
C     NCELL=MAX NO OF CELLS FORMING THE FLOW FIELD
C     NDISC=MAX NO OF DISCONTINUITIES IN THE FLOW FIELD
C     NN=PRINT ON PAPER EVERYTH NN CYCLE (MUST BE .GT. 0)
C     NNN=PRINT ON PAPER EVERYTH NNN CELL IN CYCLE NN  (MUST BE .GT. 0)
C     NS=WRITE ON TAPE EVERYTH NS CYCLE (MUST BE .GT. 0)
C     TERMIN=MAX COMPUTING TIME IN DECIMAL SECONDS
C     N=NO OF CELL BOUNDARIES FORMING THE FLOW FIELD
C     J(0,1,2)=GEOMETRY INDEX NO. (PLANE,LINE,POINT-SYMMETRY)
C     G,GF=UNCMBSTD,CMBSTD RTIO OF SPCFC HTS (NFLM,NFLM+1 BDRS FOR G,GF)
C     DELR=CELL SPACING
C     VCAPF=CONSTANT PRESSURE SPECIFIC VOLUME RATIO AT UNDIST CONDITS
C     R4R1=RATIO OF BURNT TO UNBURNT GAS CONST. I.E. (MWUNBURNT/MWBURNT)
C     PDPOWER,PPOWER=PWER IN FLME SPD LAW - S=(P/D)**PDPOWER * P**PPOWER
C     P,D,U(0,1,F)=PRESS,DENSTY,PRTCL VL(CMESTD,CMPRSD,UNDSTURBD)
C     XF=INITIAL FLM POSITION
C     DELT=TIME BETWEEN STEPS ON TAPE
C     SFLMNEW=SS FLM SPD REL PRTCLE AHD FLM WRT UNDST STTE, AFT INIT ACC
C     NTDISCT=NO OF DISC IN THE FLOW FIELD AT TIME T
C     NDISCNO=DISC NO WRT THE INITIAL FLOW FIELD
C     NTYPE(1,2,3,4,5,6,7,8)=(FLM,SHK,CD,PARECD,DET,CDTRD,TRARE,CDRARE)
C     NDISCL=CELL NO CONTAINING THE DISC ( .LT. 0 . TERMINATE DISC)
C     SL=SHK-MCH NO WRT STTE AHD,FLM-SPD REL PRTCLE AHD WRT UNDST
C     RD=POSITION OF THE DISCONTINUITY
C     NPPTH=NO. OF INITIAL PARTICLE PATHS
C     PTHNEXT=POSITION OF THE NEXT PARTICLE PATH
C     DELPPTH=DST BET SCCSSVE PRTCL PTHS(MST BE > 0. TO HVE ADDTLN PTHS)
C     NUMA=NO. OF INITIAL NEGATIVE CHARACTERISTIC TRAJECTORIES
C     RUMANXT=POSITION OF THE NEXT NEGATIVE CHARACTERISTIC TRAJECTORY
C     DELUMA=DST BET SCCSSVE NG CHR TRJS(MST BE > 0. TO HVE ADDTLN TRJS)
C     NUPA=NO. OF INITIAL POSITIVE CHARACTERISTIC TRAJECTORIES
C     TUPANXT=POSITION OF THE NEXT POSITIVE CHARACTERISTIC TRAJECTORY
C     DELUPA=DST BET SCCSSVE PS CHR TRJS(MST BE > 0. TO HVE ADDTLN TRJS)
C     RPPTH,RUMA,RUPA=PRTCLE, NEGTVE CHRACT, POSTVE CHARCT PSTN
C
C     INDEX=NO. OF TIME STEPS FOR THIS VALUE OF LSTART
C     T,DT,DTL=CURRENT TIME , LAST TIME STEP , PREVIOUS TIME STEP
C     TWRITE=NEXT TIME THAT SHOULD BE ON TAPE
C     CND=NONDIM CONST USED IN THE FLAME SPD LAW
C     IFLM,NFLM=DISC NO OF FLM WRT INTL FLWFLD, CEL NO. CNT FLM OR DET
C     NTDISC=NO OF DISC THT ARE OR HAVE BEEN IN THE FLW FLD AS OF TIME T
```

```
C      QCAPF=HEAT RELEASED AT UNDIST CONDITIONS FOR CONST PRESS COMB
C      SE=EULERIAN DISCONTINUITY SPEED
C      NTDISCS=NO DISC IN FLWFLD AFT INTERCTN USD IN DET INCRSG SPCL ORDR
C      DTDX=RATIO OF THE TIME STEP TO THE CELL SPACING
C      AT=ADDTNL TME DERVE DUE TO CHNGE IN INDP CRDS FRM (X,T) + (X-ST,T)
C      SFE,SFFOLD=EULERIAN FLM SPEED AT THE CURRENT, PREVIOUS TIME STEP
C      NUBC( NO,YES)=(STD, MODIFD L.H.S. VELOCITY CNDT)
C      NCJ( NO,YES)=(FLM SPD LT CJ VALUE,CJ FLM SPD)
C      PLTR,DLTR,ULTR,ELTR=PRSS,DNSTY,PRTCL SPD, ENRGY AT TAIL OF CJ RARE
C      OVERDNM=OVERDRIVEN DETONATION MACH NUMBER
C      NOVERDN=NO. OF TIME STEPS DET HAS BEEN OVERDRIVEN
C      NUMAFST=NO. OF THE FRST NEG CHARACT TRJ WTH PSTN > 0.0
C      NCLPPTH,NCLUMA,NCLUPA=CL NO OF PRTCL PTH,NG CHRCT TRJ,PS CHRCT TRJ
C      NITRCTN(YES, NO)=(DO,DO NT)WRITE ON TAPE AS INTRCTN BET DISCS OCRD
C
 1000 FORMAT(4I5)
 2000 FORMAT(1H0,*COMPUTER TIME IS APPROACHING DESIGNATED MAXIMUM*)
 2001 FORMAT(1H0,*TIME STEPS EQUAL DESIGNATED MAXIMUM*)
 2002 FORMAT(1H0* MESH EXPANSION(N) OR DISC(NTDISC) LIMIT*,2I6)
 2003 FORMAT(1H0,*NON-POSITIVE TIME STEP *,E13.6)
C
C
C-----DETERMINE INITIAL OR RESTART CONDITIONS
      CALL SECOND(TA)
      INDEX=0
      READ 1000,LSTART,NCYCLE,NPUNCH,NSTORE
      IF(LSTART.EQ.0) CALL INITIAL $ IF(LSTART.NE.0) CALL RESTART
C
C
C-----SET LSTART
      LSTART=LSTART+1
 1    INDEX=INDEX+1 $ NCYCLE=NCYCLE+1
C-----CHECK CENTRAL PROCESSOR ELAPSED TIME
      CALL SECOND(TB) $ TC=5.*(TB-TA) $ TD=TERMIN-TB $ TA=TB
      IF(TC.GE.TD) PRINT 2000
C-----CHECK NUMBER OF TIME STEPS
      IF(INDEX.GT.NSTEPS) PRINT 2001
C
C-----CHECK FOR MESH EXPANSION
      IF(ABS(D(2,N-6)-DF).LT.1.E-5) GO TO 2
      N=N+1 $ R2(N)=R2(N-1)+DELR
      P(2,N)=PF $ D(2,N)=DF $ U(2,N)=UF*DF
      E(2,N)=DF*(PF/DF/(G-1.)+UF**2/2.)
 2    IF(R2(1).LT.0.0) GO TO 5
      N=N+1
      DO 3 M=2,N
      L=N+2-M
      D(2,L)=D(2,L-1) $ U(2,L)=U(2,L-1) $ P(2,L)=P(2,L-1) $R2(L)=R2(L-1)
 3    E(2,L)=E(2,L-1)
      NFLM=NFLM+1 $ R2(1)=R2(2)-DELR $ U(2,2)=-U(2,2)
      IF(NUBC.EQ.3HYES) U(2,2)=-U(2,2)
      DO 4 II=1,NTDISCT
 4    NDISCL(II)=NDISCL(II)+1
C
C-----CHECK FOR MESH EXPANSION OR DISCONTINUITY LIMIT
 5    IF(N.GT.NCELL.OR.NTDISCT.GT.NDISC) PRINT 2002,N,NTDISCT
```

```
C-----DETERMINE TIME STEP,TIME AND REINITIAL PROPERTIES
      CALL TSTEP
      IF(DT.LE.0.0) PRINT 2003,DT
      IF(TC.GT.TQ.OR.INDEX.GT.NSTEPS.OR.N.GE.NCELL.OR.NTDISCT.GE.NDISC
     *.OR.DT.LE.0.0) GO TO 20
C-----DETERMINE PROPERTIES AT NEW TIME
      CALL FIDIF
C-----RECYCLE
      GO TO 1
C
C
C-----PUNCH RESTART AND TERMINATE
 20   CALL RESTART
C
      STOP $ END
```

```
      SUBROUTINE INITIAL
      COMMON/UBDCND/NUBC
      COMMON/PARAM/N,J,AJ,G,GF,DELR,NFLM
      COMMON/TIME/T,DT,DTL,TWRITE,DELT,DTDX,AT
      COMMON/TPDT/PLTR,DLTR,ULTR,ELTR,OVERDNM,NOVERDN
      COMMON/FIRSETC/INDEX,NCYCLE,NN,NNN,NSTORE,NS,NITRCTN
      COMMON/FIR/LSTART,TERMIN,NCELL,NSTEPS,NDISC,NPUNCH,PF,DF,UF
      COMMON/POWER/VCAPF,R4R1,PDPOWER,PPOWER,CNC,CCAPF,SFEOLD,SFE,NCJ
      COMMON/ARRAYS/R(501),U(2,501),P(2,501),D(2,501),E(2,501),R2(501)
      COMMON/DISCS/NTDISC,NDISCNO(51),NTYPE(51),NDISCL(51),SE(51),SL(51)
     *,RD(51),NTDISCT,NTDISCS
      COMMON/PPCHR/NPPTH,NUMA,NUMAFST,NUPA,RPPTH(24),RUMA(150),RUPA(150)
     *,NCLPPTH(24),NCLUMA(150),NCLUPA(150),PTHNEXT,RUMANXT,TUPANXT,
     *DELPPTH,DELUMA,DELUPA
C
 2000 FORMAT(1H1,*RESTART NO.*,I3,/,* MAX-CPTIME*,F7.1,/,* MAX-STEPS*,
     *I6,/,* GEOMETRY*,I2,/,* GAMMA UNCOMB*,E13.6,7X,*GAMMA COMB*,E13.
     *6,/,* CELL SPACING*,F13.6,/,* TIME BETWEEN TAPE WRITES*,E13.6,/,
     ** FREE FIELD P,D,U *,3E20.10)
 2001 FORMAT(1H ,/,* CONST PRESS COMB SPEC VOL RATIO(AT UNDIST STATE CON
     *D)*,E13.6,/,* RATIO OF GAS CONST, R4R1*,E13.6,/,* HEAT RELEASE AT
     *UNDIST CONDIT*,E13.6,/,* FLAME ACCELERATION LAW      PDPOWER*,F7.3,
     *3X,*PPOWER*,F7.3,/,* LAGRANGE FLAME SPEEDS BEFORE, AFTER ACCL*,
     *2E20.10,/,* CELL CONTAINING THE FLAME*,I5,/,* INITIAL FLAME POSITI
     *ON *,E20.10)
 2002 FORMAT(1H ,/,* NO. OF INTL PRTCLE PTHS *,I3,5X,*NEXT PRTCLE PTH *,
     *F10.5,5X,*SPCING BET SCCSSVE PRTCLE PTHS *,F10.5,/,* FRST AND TOT
     *NO. OF INTL NEG CHRCT TRAJS *,2I4,5X,*NEXT TRAJ AT *,F10.5,5X,*SPC
     *ING BET SCCSSVE TRAJS *,F10.5,/,* NO. OF INTL POS CHRCT TRAJS *,I4
     *,5X,*NEXT TRAJ AT *,F10.5,5X,* SPCING BET SCCSSVE TRAJS *,F10.5)
C
C
C------READ INPUT (DISC MUST BE INPUT IN INCREASING SPATIAL ORDER)
      READ 1000,NSTEPS,NCELL,NDISC,NN,NNN,NS,TERMIN, N,J,G,GF,DELR,
     *VCAPF,R4R1,PDPOWER,PPOWER, P0,D0,U0, P1,D1,U1, PF,DF,UF,XF,
     *DELT,SFLMNEW, NTDISCT,
     *(NDISCNO(I),NTYPE(I),NDISCL(I),SL(I),RD(I),I=1,NTDISCT)
 1000 FORMAT(6I5,F10.4,/,2I5,7F10.8,/,3F25.17,/,3E25.17,/,
     *2F20.15,E30.17,F10.7,/,2F20.15,/,I5,/,(3I5,2E25.17))
      READ 1001,NPPTH,PTHNEXT,DELPPTH,NUMA,RUMANXT,DELUMA,NUPA,TUPANXT,
     *DELUPA
 1001 FORMAT(3(I5,2F10.5))
      IF(NPPTH.GT.0) READ 1002,(RPPTH(I),I=1,NPPTH)
      IF(NUMA.GT.0) READ 1002,(RUMA(I),I=1,NUMA)
      IF(NUPA.GT.0) READ 1002,(RUPA(I),I=1,NUPA)
 1002 FORMAT(8F10.5)
C
C-    INITIALIZE PROGRAM VARIABLES
      AJ=FLOAT(J) $ T=DT=0.0 $ TWRITE=T+DELT $ CNC=0.0 $ NTDISC=NTDISCT
      NUBC=3H NO $ NCJ=3H NO $ PLTR=DLTR=ULTR=ELTR=0.0 $ NUMAFST=1
      OVERDNM=0.0 $ NOVERDN=100 $ NITRCTN=3H NO
C
C-----DET IFLM AND NFLM
      DO 1 I=1,NTDISCT
      IF(NTYPE(I).EQ.1) GO TO 2
 1    CONTINUE
```

```
      NFLM=IFLM=0 $ SFE=0.0 $ GO TO 3
 2    NFLM=NDISCL(I) $ IFLM=I
C
C-----DEFINE INITIAL MESH POINTS AND PARAMETERS
C-----SPECIAL PROCEDURE FOR J=0, IFLM=1, NTDISCT=2
      IF((J.NE.0).OR.(IFLM.NE.1).OR.(NTDISCT.NE.2)) GO TO 3
      NFLM=NDISCL(1)=((RD(1)-1.E-12)/DELR+2.)
      R2(1)=RD(1)-FLOAT(NFLM-1)*DELR
      NDISCL(2)=((RD(2)-1.E-12)/DELR+3.) $ N=NDISCL(2)+6
      DO 10 M=1,N
      IF(M.NE.1) R2(M)=R2(M-1)+DELR
      IF(M.EQ.NFLM+1) R2(M)=R2(M-1)
      IF(M.GT.NDISCL(2)) GO TO 12
      IF(M.GT.NDISCL(1)) GO TO 11
      P(2,M)=P0 $ D(2,M)=D0 $ U(2,M)=U0
      GO TO 13
 11   P(2,M)=P1 $ D(2,M)=D1 $ U(2,M)=U1
      GO TO 13
 12   P(2,M)=PF $ D(2,M)=DF $ U(2,M)=UF
 13   GE=GF $ IF(M.GT.NFLM) GE=G
 10   E(2,M)=P(2,M)/D(2,M)/(GE-1.)+U(2,M)**2/2.0
      GO TO 15
C
C-----PROCEDURE FOR ALL OTHER CASES
 3    READ 1003,(K,R2(M),U(2,M), K,P(2,M),D(2,M),E(2,M),M=1,N)
 1003 FORMAT(1001(I5,2E25.17,/,I5,3E25.17,/))
C
C-----SET REMAINING GASDYNAMIC VALUES
 15   DO 16 M=1,N
      R(M)=R2(M) $ U(1,M)=U(2,M) $ P(1,M)=P(2,M) $ D(1,M)=D(2,M)
 16   E(1,M)=E(2,M)
C
C-----CALCULATE HEAT RELEASE AT UNDIST CONDITS FOR CONST PRESS COMB
      QCAPF=(VCAPF-1.)*GF/(GF-1.)
C
C-----CALC LAGRANGIAN FLAME SPEED CONST, EULERIAN DISCS SPDS
      IF(IFLM.NE.0) CNC=SL(IFLM)/(P(2,NFLM+1)/D(2,NFLM+1))**PDPOWER
     */P(2,NFLM+1)**PPOWER
      GE=GF
      DO 20 I=1,NTDISCT
      NDISCLS=NDISCL(I)
      L=1 $ IF(SL(I).LT.0.0) L=0
      IF(NDISCLS.GT.NFLM) GE=G
      SE(I)=SL(I)*SQRT(GE*P(2,NDISCLS+L)/D(2,NDISCLS+L))+U(2,NDISCLS+L)
      IF(NTYPE(I).EQ.1) SE(I)=SL(I)+U(2,NDISCLS+1)
 20   CONTINUE
C
C-----PARTICLE PATHS AND CHARACTERISTIC TRAJECTORIES AT THE INITIAL TIME
      CALL CHARDIR(NCYCLE)
C
C-----PRINTING INSTRUCTIONS
      PRINT 2000,LSTART,TERMIN;NSTEPS,J,G,GF,DELR,DELT,PF,DF,UF
      IF(IFLM.NE.0) PRINT 2001,VCAPF,RAR1,CCAPF,PDPOWER,PPOWER,SL(IFLM),
     *SFLMNEW,NFLM,XF
      PRINT 2002,NPPTH,PTHNEXT,DELPPTH,NUMAFST,NUMA,RUMANXT,DELUMA,
     *NUPA,TUPANXT,DELUPA
```

```
      NCODE=10HINITIAL     $ CALL PRNTFF(NCODE)
C
C-----WRITING INSTRUCTIONS
      IF(NSTORE.LE.0) GO TO 30
      WRITE(1) J,G,GF,DELR,VCAPF,R4R1,GCAPF,PDPOWER,PPOWER,DELT,XF,
     *PTHNEXT,DELPPTH,RUMANXT,DELUMA,TUPANXT,DELUPA
      WRITE(1) NCYCLE,T,NTDISCT,NFLM,NPPTH,NUMAFST,NUMA,NUPA,(NDISCNO(I)
     *,NTYPE(I),NDISCL(I),SE(I),SL(I),RD(I),I=1,NTDISCT),N,(R2(M),U(2,M)
     *,P(2,M),D(2,M),M=1,N)
      IF(NPPTH.GT.0) WRITE(1) (RPPTH(I),I=1,NPPTH)
      IF(NUMA.GT.0) WRITE(1) (RUMA(I),I=NUMAFST,NUMA)
      IF(NUPA.GT.0) WRITE(1) (RUPA(I),I=1,NUPA)
C
C
C-----FOR SUDDEN FLM ACC- DET S.S. CONDITS, PRINT AND WRITE
 30   IF(IFLM.EQ.0) GO TO 40
C-----LGRNGE FLM SPD BEF+AFT ACC MST DFFR 1E-6 IF SDDN FLM ACC IS CNSDRD
      IF(ABS(SFLMNEW-SL(IFLM)).LT.1.E-6) GO TO 33
      CALL FLMACCL(SFLMNEW,IFLM,P30,D30)
      DT=T=1.E-10 $ NCYCLE=NCYCLE+1 $ INDEX=INDEX+1
      CALL DISC(1)
      CALL CHARDIR(NCYCLE)
      DO 31 IFLM=1,NTDISCT
      IF(NTYPE(IFLM).EQ.1) GO TO 32
 31   CONTINUE
 32   CND=SL(IFLM)/(P30/D30)**PDPOWER/P30**PPOWER
      NCODE=10HINITIALACC $ CALL PRNTFF(NCODE)
      IF(NSTORE.LE.0) GO TO 33
      WRITE(1) NCYCLE,T,NTDISCT,NFLM,NPPTH,NUMAFST,NUMA,NUPA,(NDISCNO(I)
     *,NTYPE(I),NDISCL(I),SE(I),SL(I),RD(I),I=1,NTDISCT),N,(R2(M),U(2,M)
     *,P(2,M),D(2,M),M=1,N)
      IF(NPPTH.GT.0) WRITE(1) (RPPTH(I),I=1,NPPTH)
      IF(NUMA.GT.0) WRITE(1) (RUMA(I),I=NUMAFST,NUMA)
      IF(NUPA.GT.0) WRITE(1) (RUPA(I),I=1,NUPA)
 33   SFEOLD=SFE=SE(IFLM)
C
C
 40   DO 41 M=1,N
      U(2,M)=U(2,M)*D(2,M)
 41   E(2,M)=E(2,M)*D(2,M)
C
      AT=-SFE
C
      RETURN $ END
```

```
      SUBROUTINE RESTART
      COMMON/UBDCND/NUBC
      COMMON/PARAM/N,J,AJ,C,GF,DELR,NFLM
      COMMON/TIME/T,DT,DTL,TWRITE,DELT,DTDX,AT
      COMMON/TRDT/PLTR,DLTR,ULTR,ELTR,CVERDNM,NOVERDN
      COMMON/FIRFETC/INDEX,NCYCLE,NN,NNN,NSTORE,NS,NITRCTN
      COMMON/FIR/LSTART,TERMIN,NCELL,NSTEPS,NDISC,NPUNCH,PF,DF,UF
      COMMON/POWER/VCAPF,R4R1,PDPOWER,PPOWER,CND,QCAPF,SFEOLD,SFE,NCJ
      COMMON/ARRAYS/R(501),U(2,501),P(2,501),D(2,501),E(2,501),R2(501)
      COMMON/DISCS/NTDISC,NDISCNO(51),NTYPE(51),NDISCL(51),SE(51),SL(51)
     *,RD(51),NTDISCT,NTDISCS
      COMMON/PPCHR/NPPTH,NUMA,NUMAFST,NUPA,RPPTH(24),RUMA(150),RUPA(150)
     *,NCLPPTH(24),NCLUMA(150),NCLUPA(150),PTHNEXT,RUMANXT,TUPANXT,
     *DELPPTH,DELUMA,DELUPA
C
 1000 FORMAT(4I5)
 1001 FORMAT(6I5,E10.4,/,3E25.17)
 1002 FORMAT(2I5,7F10.8,/,4F20.17,/,E25.17,3E17.9,/,2E25.17,I10,/,
     *3I5,2X,A3,2X,A3,/,3E25.17,/,(I3,I1,I4,E25.17,F22.18,E25.17))
 1003 FORMAT(1001(I5,E25.17,/,I5,3E25.17,/))
 1004 FORMAT(5X,2I5,5X,2I5,10X,/,3E25.17)
 1005 FORMAT(I5,2F10.5,2I5,2F10.5,I5,2F10.5)
 1006 FORMAT(40(4(F10.5,I5,5X),/))
 2000 FORMAT(1H1,*RESTART NO.*,I3,/,* MAX-CPTIME*,F7.1,/,* MAX-STEPS*,
     *I6,/,* GEOMETRY*,I2,/,* GAMMA UNCOMB*,E13.6,7X,*GAMMA COMB*,E13.
     *6,/,* CELL SPACING*,E13.6,/,* TIME BETWEEN TAPE WRITES*,E13.6,/,
     ** FREE FIELD P,D,U *,3E20.10)
 2001 FORMAT(1H ,/,* CONST PRESS COMB SPEC VOL RATIO(AT UNDIST STATE CON
     *D)*,E13.6,/,* RATIO OF GAS CONST, R4R1*,E13.6,/,* HEAT RELEASE AT
     *UNDIST CONDIT*,E13.6,/,* FLAME ACCELERATION LAW      PDPOWER*,F7.3,
     *3X,*PPOWER*,F7.3,/,* EULERIAN FLAME SPD*,E20.10,/,* CELL CONTAININ
     *G THE FLAME*,I5)
 2002 FORMAT(1H ,/,* NO. OF PRTCLE PTHS *,I3,5X,*NXT PTH *,F10.5,5X,
     **SPACING *,F10.5,/,* FST AND TOT NO. NEG CHARCT TRAJS *,2I4,5X,
     **NEXT TRAJ *,F10.5,5X,*SPACING *,F10.5,/,* NO. OF POS CHARCT TRAJS
     * *,I4,5X,*NEXT TRAJ *,F10.5,5X,*SPACING *,F10.5)
C
C
      IF(INDEX.NE.0) GO TO 10
C
C-----READ IN DATA CARDS
      READ 1001,NSTEPS,NCELL,NDISC,NN,NNN,NS,TERMIN, PF,DF,UF
      READ 1002,N,J,G,GF,DELR,VCAPF,R4R1,PDPOWER,PPOWER,
     *DELT,QCAPF,SFE,SFEOLD, CND,T,DT,TWRITE, AT,CVERDNM,NOVERDN,
     *NFLM,NTDISC,NTDISCT,NCJ,NUBC, PLTR,DLTR,ULTR,
     *(NDISCNO(I),NTYPE(I),NDISCL(I),SL(I),SE(I),RD(I),I=1,NTDISCT)
      READ 1003,(K,R2(I), K,U(2,I),P(2,I),D(2,I),I=1,N)
      READ 1005,NPPTH,PTHNEXT,DELPPTH,NUMAFST,NUMA,RUMANXT,DELUMA,NUPA,
     *TUPANXT,DELUPA
      IF(NPPTH.GT.0) READ 1006,(RPPTH(I),NCLPPTH(I),I=1,NPPTH)
      IF(NUMA.GT.0) READ 1006,(RUMA(I),NCLUMA(I),I=NUMAFST,NUMA)
      IF(NUPA.GT.0) READ 1006,(RUPA(I),NCLUPA(I),I=1,NUPA)
      NITRCTN=3H NO $ ELTR=0.0
      IF(DLTR.GT.1.E-10)ELTR=DLTR*(PLTR/DLTR/(GF-1.)+ULTR**2/DLTR**2/2.)
      GE=GF
      DO 1 I=1,N
```

```
      IF(I.GT.NFLM) GE=G
 1    E(2,I)=P(2,I)/D(2,I)/(GE-1.)+U(2,I)**2/2.
      AJ=FLOAT(J) $ GO TO 20
C
C
C-----PUNCH DATA CARDS
 10   NCYCLE=NCYCLE-1 $ INDEX=INDEX-1 $ T=T-DT $ DT=DTL
      DO 11 I=1,N
      U(2,I)=U(2,I)/D(2,I)
 11   E(2,I)=E(2,I)/D(2,I)
      IF(NPUNCH.NE.1) GO TO 20
      PUNCH 1000,LSTART,NCYCLE,NPUNCH,NSTORE
      PUNCH 1004,        NCELL,NDISC,   NNN,NS,       PF,DF,UF
      PUNCH 1002,N,J,G,GF,DELR,VCAPF,R4R1,PCPOWER,PPOWER,
     *DELT,CCAPF,SFE,SFEOLD, CND,T,DT,TWRITE, AT,OVERDNW,NOVERDN,
     *NFLM,NTDISC,NTDISCT,NCJ,NUBC, PLTR,DLTR,ULTR,
     *(NDISCND(I),NTYPE(I),NDISCL(I),SL(I),SE(I),RD(I),I=1,NTDISCT)
      PUNCH 1003,(I,R2(I), I,U(2,I),P(2,I),D(2,I),I=1,N)
      PUNCH 1005,NPPTH,PTHNEXT,DELPPTH,NUMAFST,NUMA,RUMANXT,DELUMA,NUPA,
     *TUPANXT,DELUPA
      IF(NPPTH.GT.0) PUNCH 1006,(RPPTH(I),NCLPPTH(I),I=1,NPPTH)
      IF(NUMA.GT.0) PUNCH 1006,(RUMA(I),NCLUMA(I),I=NUMAFST,NUMA)
      IF(NUPA.GT.0) PUNCH 1006,(RUPA(I),NCLUPA(I),I=1,NUPA)
C
C
 20   PRINT 2000,LSTART,TERMIN,NSTEPS,J,G,GF,DELR,DELT, PF,DF,UF
      IF(NFLM.NE.0) PRINT 2001,VCAPF,R4R1,CCAPF,PCPOWER,PPOWER,SFE,NFLM
      PRINT 2002,NPPTH,PTHNEXT,DELPPTH,NUMAFST,NUMA,RUMANXT,DELUMA,
     *NUPA,TUPANXT,DELUPA
      NCODE=10HRESTART     $ CALL PRNTFF(NCODE)
C
      IF(INDEX.NE.0) RETURN
      DO 21 I=1,N
      U(2,I)=U(2,I)*D(2,I)
 21   E(2,I)=E(2,I)*D(2,I)
C
      RETURN $ END
```

```
      SUBROUTINE FIDIF
      COMMON/UBDCND/NUBC
      COMMON/PARAM/N,J,AJ,G,GF,DELR,NFLM
      COMMON/TIME/T,DT,DTL,TWRITE,DELT,DTDX,AT
      COMMON/FIRFETC/INDEX,NCYCLE,NN,NNN,NSTORE,NS,NITRCTN
      COMMON/POWER/VCAPF,R4R1,FDPOWER,FPOWER,CND,CCAPF,SFEOLD,SFE,NCJ
      COMMON/ARRAYS/R(501),U(2,501),P(2,501),D(2,501),E(2,501),R2(501)
      COMMON/DISCS/NTDISC,NDISCNO(51),NTYPE(51),NDISCL(51),SE(51),SL(51)
     *,RD(51),NTDISCT,NTDISCS
      COMMON/PPCHR/NPPTH,NUMA,NUMAFST,NUPA,RPPTH(24),RUMA(150),RUPA(150)
     *,NCLPPTH(24),NCLUMA(150),NCLUPA(150),PTHNEXT,RUMANXT,TUPANXT,
     *DELPPTH,DELUMA,DELUPA
C
C----ADVANCE MESH POSITIONS
      SDT=SFE*DT
      DO 1 I=1,N
 1    R2(I)=R(I)+SDT
C
C-----DIFFERENCE SCHEME-----
      AT=-SFE $ IF(NCYCLE.GT.1) AT=-SFE-(T-DT)*(SFE-SFEOLD)/DTL
      DTDX=DT/DELR $ NL=N-1 $ GE=GF
C-----PREDICTOR
      DO 5 I=2,NL
      D(2,I)=D(1,I)-DTDX*(U(1,I+1)-U(1,I))-AT*DTDX*(D(1,I+1)-D(1,I))
      U(2,I)=U(1,I)-DTDX*(U(1,I+1)**2/D(1,I+1)+P(1,I+1)-U(1,I)**2/
     *D(1,I)-P(1,I))-AT*DTDX*(U(1,I+1)-U(1,I))
      E(2,I)=E(1,I)-DTDX*(U(1,I+1)/D(1,I+1)*(E(1,I+1)+P(1,I+1))
     *-U(1,I)/D(1,I)*(E(1,I)+P(1,I)))-AT*DTDX*(E(1,I+1)-E(1,I))
      IF(I.GT.NFLM) GE=G
 5    P(2,I)=(E(2,I)/D(2,I)-U(2,I)**2/D(2,I)**2/2.0)*D(2,I)*(GE-1.)
C-----PREDICTOR B.C.
      U(2,1)=-U(2,2) $ P(2,1)=P(2,2) $ D(2,1)=D(2,2) $ E(2,1)=E(2,2)
      IF(NUBC.EQ.3HYES) U(2,1)=U(1,1)
C
C-----CORRECTOR
      GE=G
      DO 6 I=2,NL
      II=N-I+1
      D2II=(D(1,II)+D(2,II)-DTDX*(U(2,II)-U(2,II-1))-AT*DTDX*
     *(D(2,II)-D(2,II-1)))/2.
      U2II=(U(1,II)+U(2,II)-DTDX*(U(2,II)**2/D(2,II)+P(2,II)-U(2,II-1)**
     *2/D(2,II-1)-P(2,II-1))-AT*DTDX*(U(2,II)-U(2,II-1)))/2.
      E(2,II)=(E(2,II)+E(1,II)-DTDX*(U(2,II)/D(2,II)*(E(2,II)+P(2,II))
     *-U(2,II-1)/D(2,II-1)*(E(2,II-1)+P(2,II-1)))
     *-AT*DTDX*(E(2,II)-E(2,II-1)))/2.
      D(2,II)=D2II $ U(2,II)=U2II
      IF(II.LE.NFLM) GE=GF
 6    P(2,II)=(E(2,II)/D(2,II)-U(2,II)**2/D(2,II)**2/2.)*D(2,II)*(GE-1.)
C-----CORRECTOR B.C.
      U(2,1)=-U(2,2) $ P(2,1)=P(2,2) $ D(2,1)=D(2,2) $ E(2,1)=E(2,2)
      IF(NUBC.EQ.3HYES) U(2,1)=U(1,1)
C
C
      SFEOLD=SFE
C
C-----DISCONTINUITY DYNAMICS
```

```
      CALL DISC(0)
C
C-----PRINT GASDYNAMIC PARAMETERS ABOUT EACH DISC AT EVERY NCY-TH CYCLE
      NCY=5000
      IF(NCYCLE/NCY*NCY.NE.NCYCLE) GO TO 20
      DO 10 II=1,NTDISCT
      NF=NDISCL(II) $ IF(NF.LT.5) NF=5 $ K=NF-4 $ L=NF+4 $ NFM3=NF-3
      DO 11 M=K,L
      U(2,M)=U(2,M)/D(2,M)
 11   E(2,M)=E(2,M)/D(2,M)
      PRINT 2000,NCYCLE,NDISCNC(II),NTYPE(II),NDISCL(II),SE(II),SL(II),
     *RD(II),R2(1),(R2(I),I=NFM3,L),T,DT, (U(2,I),I=K,L),(P(2,I),I=K,L),
     *(D(2,I),I=K,L), (E(2,I),I=K,L)
 2000 FORMAT(1H ,1X,I4,I3,I2,I4,2F14.9,10F7.4,2F10.7,4(/,1X,9F14.10))
      DO 12 M=K,L
      U(2,M)=U(2,M)*D(2,M)
 12   E(2,M)=E(2,M)*D(2,M)
 10   CONTINUE
      PRINT 2001
 2001 FORMAT(1H ,2/)
C
C-----PARTICLE PATHS AND CHARACTERISTIC TRAJECTORIES
 20   CALL CHARDIR(NCYCLE)
C
C-----PRINT AND WRITE INSTRUCTIONS
      NX=NSX=0
      IF(NCYCLE/NN*NN.EQ.NCYCLE) NX=1 $ IF(NCYCLE/NS*NS.EQ.NCYCLE) NSX=1
      IF((NX.EQ.0).AND.((NSTORE.LE.0).OR.(NSX.EQ.0)).AND.((NSTORE.LE.0).
     *OR.(T.LT.TWRITE)).AND.((NITRCTN.EQ.3H NO).OR.(NSTORE.LE.0)))
     *GO TO 50
C
C-----CALCULATE U,E
      DO 30 M=1,N
      U(2,M)=U(2,M)/D(2,M)
 30   E(2,M)=E(2,M)/D(2,M)
C
C-----WRITE
      IF((NSTORE.GT.0).AND.((NSX.EQ.1).OR.(T.GE.TWRITE).OR.(NITRCTN.EQ.
     *3HYES))) GO TO 35
      GO TO 36
 35   WRITE(1) NCYCLE,T,NTDISCT,NFLM,NPPTH,NUMAFST,NUMA,NUPA,(NDISCNC(I)
     *,NTYPE(I),NDISCL(I),SE(I),SL(I),RC(I),I=1,NTDISCT),N,(R2(M),U(2,M)
     *,P(2,M),D(2,M),M=1,N)
      IF(NPPTH.GT.0) WRITE(1) (RPPTH(I),I=1,NPPTH)
      IF(NUMA.GT.0) WRITE(1) (RUMA(I),I=NUMAFST,NUMA)
      IF(NUPA.GT.0) WRITE(1) (RUPA(I),I=1,NUPA)
C
 36   NITRCTN=3H NO
C
C-----PRINT HEADERS
      IF(NX.EQ.0) GO TO 40
      NCODE=10HFIDIF      $ CALL PRNTFF(NCODE)
C
 40   DO 41 M=1,N
      U(2,M)=U(2,M)*D(2,M)
 41   E(2,M)=E(2,M)*D(2,M)
```

```
C
 50    IF(T.GE.TWRITE) TWRITE=T+DELT
C
       RETURN $ END
```

```
      SUBROUTINE DISC(NINITL)
      COMMON/PARAM/N,J,AJ,G,GF,DELR,NFLM
      COMMON/TIME/T,DT,DTL,TWRITE,DELT,DTDX,AT
      COMMON/FIRFETC/INDEX,NCYCLE,NN,NNN,NSTOPE,NS,NITRCTN
      COMMON/DISCSKF/RDSAV(51),NLHS(51),NRHS(51),NCROSS(51)
      COMMON/ARRAYS/R(501),U(2,501),F(2,501),D(2,501),E(2,501),R2(501)
      COMMON/DISCS/NTDISC,NDISCNO(51),NTYPE(51),NDISCL(51),SE(51),SL(51)
     *,RD(51),NTDISCT,NTDISCS
C
C     NLHS,NRHS=LEFT,RIGHT HAND CELL NO. INFLUENCED BY THE DISC
C     NI=NO. OF DISCS INTERACTING
C     NFIRST,NLAST=FIRST,LAST DISC IN THE INTERACTION
C     NUMBER=INTERACTION CODE NO.
C     NINITL(0,1)=(DOESNT,DOES COME FROM SUB-INITIAL SO ONLY RESET DISC)
C
C
C-----DET IF CALLED FROM SUB-INITIAL
      IF(NINITL.EQ.1) GO TO 100
C
C
C-----LOOP 1 DETS AND EXECUTES ALL DISC(FLM,SHK,CD,DET,TRARE) INTRCTNS
      NTDISCS=NTDISCT $ NI=0 $ NFIRST=1
      DO 1 I=1,NTDISCS
      NI=NI+1
C
C-----DETERMINE CELL RANGE INFLUENCE OF THE DISC
      NDISCLS=NDISCL(I)
      RDSAV(I)=RD(I)+SE(I)*DT
      IF((NTYPE(I).EQ.2).OR.(NTYPE(I).EQ.5)) GO TO 2
      IF((NTYPE(I).EQ.3).OR.(NTYPE(I).EQ.4).OR.(NTYPE(I).EQ.6).OR.
     *(NTYPE(I).EQ.8)) GO TO 4
      IF(NTYPE(I).EQ.7) GO TO 7
C
      NCROSS(I)=0
      NLHS(I)=NDISCLS-1 $ NRHS(I)=NDISCLS+2
      GO TO 5
C
 2    IF(((RDSAV(I).LT.R2(NDISCLS)).AND.(SL(I).LT.0.0)).OR.
     *((RDSAV(I).GT.R2(NDISCLS+1)).AND.(SL(I).GT.0.0)).OR.((RDSAV(I).GT.
     *R2(NDISCLS)).AND.(RDSAV(I).LT.R2(NDISCLS+1)))) GO TO 3
      PRINT1000,I,NDISCNO(I),NDISCLS,NTYPE(I),RD(I),RDSAV(I),R2(NDISCLS)
     *,R2(NDISCLS+1),SL(I),SE(I),DT
 1000 FORMAT(1H ,/,1X,* SPECIAL DISC X/C *,4I5,7E13.6)
 3    NCROSS(I)=0
      IF(RDSAV(I).GT.R2(NDISCLS+1)) NCROSS(I)=1
      IF(RDSAV(I).LT.R2(NDISCLS)) NCROSS(I)=-1
      NLHS(I)=NDISCLS-1-NCROSS(I)*(NCROSS(I)-1)/2
      NRHS(I)=NDISCLS+3+NCROSS(I)
      GO TO 5
 4    NCROSS(I)=0
      IF(RDSAV(I).GT.R2(NDISCLS+2)) NCROSS(I)=1
      IF(RDSAV(I).LT.R2(NDISCLS-1)) NCROSS(I)=-1
      NLHS(I)=NDISCLS-2+NCROSS(I)*(NCROSS(I)-1)/2
      NRHS(I)=NDISCLS+4+NCROSS(I)
      GO TO 5
 7    NCROSS(I)=0
```

```
      IF(RDSAV(I).GT.R2(NDISCLS+1)) NCROSS(I)=1
      IF(RDSAV(I).LT.R2(NDISCLS)) NCROSS(I)=-1
      NLHS(I)=NDISCLS $ NRHS(I)=NDISCLS+12
C
C-----LOOP 10 DETS IF THE DISC IS SUFF CLOSE FOR INTERACTION
 5    IF(NLHS(I).LT.1) NLHS(I)=1
      IF(NTDISCS.EQ.1) GO TO 20
      IF(NI.EQ.1) GO TO 1
      MLHS=0
      IF((NTYPE(I)*NCROSS(I).EQ.-3).OR.(NTYPE(I)*NCROSS(I).EQ.-4).OR.
     *(NTYPE(I)*NCROSS(I).EQ.-6).OR.(NTYPE(I).EQ.7)) MLHS=-1
      IF((NTYPE(I)*NCROSS(I).EQ.-2).AND.(NTYPE(I-1).NE.1).AND.
     *(NCROSS(I-1).NE.-1)) MLHS=1
      IF(NRHS(I-1).LE.NLHS(I)+1+MLHS) GO TO 10 $ GO TO 21
C
C------NO INTERACTION
 10   NI=NI-1
      GO TO 20
C-----INTERACTION
 21   IF((NI.EQ.2).AND.(NTYPE(I-1).EQ.2).AND.(NTYPE(I)*NCROSS(I).EQ.-2)
     *.AND.(((NDISCL(I)-NDISCL(I-1).EQ.0).AND.(NCROSS(I-1).NE.-1)).OR.
     *((NDISCL(I)-NDISCL(I-1).EQ.1).AND.(NCROSS(I-1).EQ.1)))) GO TO 22
      GO TO 25
 22   IF(NDISCL(I-1).EQ.NDISCL(I)-1) GO TO 23
      NCROSS(I-1)=0 $ NRHS(I-1)=NDISCL(I-1)+3
 24   NCROSS(I)=0 $ NRHS(I)=NDISCL(I)+3 $ NLHS(I)=NDISCL(I)-2
      GO TO 25
 23   M=NDISCL(I)
      IF(RD(I)+(RD(I)-RD(I-1))/(SE(I-1)-SE(I))*SE(I).LT.R2(M)) GO TO 25
      GO TO 24
 25   IF(I.NE.NTDISCS) GO TO 1
C
C-----DETERMINE INTERACTION CODE NO.
 20   NUMBER=0
      DO 30 II=1,NI
 30   NUMBER=NUMBER+10**(4-II)*NTYPE(NFIRST-1+II)
      NI=NFIRST+NI-1
C-----PRINT THE INTERACTION NO. AT EVERY NCY-TH CYCLE
      NCY=5000
      IF(NCYCLE/NCY*NCY.EQ.NCYCLE) PRINT 1002,NUMBER,(M,NCROSS(M),NDISCL
     *(M),NTYPE(M),M=NFIRST,NI)
 1002 FORMAT(1H ,17I5)
C------CHECK IF INTERACTION BELONGS TO A KNOWN MOTION
      IF((NUMBER.EQ.1000).OR.(NUMBER.EQ.1200).OR.(NUMBER.EQ.1400).OR.
     *   (NUMBER.EQ.2000).OR.(NUMBER.EQ.2100).OR.((NUMBER.EQ.2200).AND.
     *(SL(NFIRST).GT.0.)).OR.(NUMBER.EQ.2220).OR.(NUMBER.EQ.2250).OR.
     *   (NUMBER.EQ.2300).OR.(NUMBER.EQ.2310).OR.(NUMBER.EQ.2312).OR.
     *   (NUMBER.EQ.2320).OR.(NUMBER.EQ.2400).OR.(NUMBER.EQ.2500).OR.
     *   (NUMBER.EQ.2520).OR.(NUMBER.EQ.2600).OR.(NUMBER.EQ.2670).OR.
     *   (NUMBER.EQ.2675).OR.(NUMBER.EQ.2800).OR.(NUMBER.EQ.3000).OR.
     *   (NUMBER.EQ.3100).OR.(NUMBER.EQ.3120).OR.(NUMBER.EQ.3200).OR.
     *   (NUMBER.EQ.4000).OR.(NUMBER.EQ.4200).OR.(NUMBER.EQ.5000).OR.
     *   (NUMBER.EQ.5200).OR.(NUMBER.EQ.5220).OR.(NUMBER.EQ.5300).OR.
     *   (NUMBER.EQ.5400).OR.(NUMBER.EQ.6000).OR.(NUMBER.EQ.6700).OR.
     *   (NUMBER.EQ.6750).OR.(NUMBER.EQ.7000).OR.(NUMBER.EQ.7500).OR.
     *   (NUMBER.EQ.8000).OR.(NUMBER.EQ.8200)) GO TO 35
```

```
      DELR=-ABS(DELR) $ PRINT 1001,NUMBER $ RETURN
 1001 FORMAT(1H ,5/,* NUMBER= *,I5)
C-----CARRY OUT THE INTERACTION
 35   IF(NUMBER.EQ.1000) CALL FLM(RD(N1),SE(N1),SL(N1),6H      )
      IF(NUMBER.EQ.1200) CALL FLMSHK(4HZERO,NFIRST,N1)
      IF(NUMBER.EQ.1400) CALL FLRCD(NFIRST,N1,NCROSS(N1))
      IF(NUMBER.EQ.2000) CALL SHK(RD(N1),SE(N1),SL(N1),NDISCL(N1))
      IF(NUMBER.EQ.2100) CALL SHKFLM(NFIRST,N1)
      IF((NUMBER.EQ.2200).AND.(SL(N1).GT.0.0).AND.(SL(NFIRST).GT.0.0))
     *CALL SKSKPP(NFIRST,N1,NFIRST,7H DISCSS)
      IF((NUMBER.EQ.2200).AND.(SL(N1).LE.0.0).AND.(SL(NFIRST).GT.0.0))
     *CALL SKSKPN(NFIRST,N1)
      IF(NUMBER.EQ.2220) CALL SKSKSK(NFIRST,NFIRST+1,N1)
      IF(NUMBER.EQ.2250) CALL DTSKSK(NFIRST,N1-1,N1)
      IF(NUMBER.EQ.2300) CALL SHKCD(NFIRST,N1,8H    ZERO)
      IF(NUMBER.EQ.2310) CALL SCDFLM(NFIRST,NFIRST+1,N1)
      IF(NUMBER.EQ.2312) CALL SCDFLMS(NFIRST,NFIRST+1,NFIRST+2,N1)
      IF(NUMBER.EQ.2320) CALL SKCDSHK(NFIRST,NFIRST+1,N1)
      IF(NUMBER.EQ.2400) CALL SHKCD(NFIRST,N1,8HRARCD   )
      IF(NUMBER.EQ.2500) CALL DSCRSD(NFIRST,N1,N1,7H DISCSD)
      IF(NUMBER.EQ.2520) CALL DTSKSK(NFIRST,N1-1,N1)
      IF(NUMBER.EQ.2600) CALL SHKCD(NFIRST,N1,8HSKCDDT  )
      IF(NUMBER.EQ.2670) CALL SCTRDTR(NFIRST,NFIRST+1,N1,NDISCL(N1))
      IF(NUMBER.EQ.2675) CALL SCDTDET(NFIRST,NCROSS(NFIRST),N1-2,
     *NCROSS(N1-2),N1-1,NCROSS(N1-1),N1,NCROSS(N1),4H2675)
      IF(NUMBER.EQ.2800) CALL SHKCD(NFIRST,N1,8HSHKCDRAR)
      IF(NUMBER.EQ.3000) CALL CD(RD(N1),SE(N1),NDISCL(N1),5H     )
      IF(NUMBER.EQ.3100) CALL CDFLM(4HZERO,NFIRST,N1)
      IF(NUMBER.EQ.3120) CALL CDFLMSK(NFIRST,NFIRST+1,N1)
      IF(NUMBER.EQ.3200) CALL CDSHK(NFIRST,N1,6H  ZERO)
      IF(NUMBER.EQ.4000) CALL CD(RD(N1),SE(N1),NDISCL(N1),5HRARCD)
      IF(NUMBER.EQ.4200) CALL CDSHK(NFIRST,N1,6H RARCD)
      IF(NUMBER.EQ.5000) CALL DET(RD(N1),SE(N1),SL(N1),NDISCL(N1),NCROSS
     *(N1),7H    DET)
      IF(NUMBER.EQ.5200) CALL DSCRSD(NFIRST,N1,N1,7H DISCDS)
      IF(NUMBER.EQ.5220) CALL DTSKSK(NFIRST,N1-1,N1)
      IF(NUMBER.EQ.5300) CALL SHKCD(NFIRST,N1,8HDETCD   )
      IF(NUMBER.EQ.5400) CALL SHKCD(NFIRST,N1,8HDETRARCD)
      IF(NUMBER.EQ.6000) CALL CD(RD(N1),SE(N1),NDISCL(N1),5HCDTRD)
      IF(NUMBER.EQ.6000) NTYPE(NFIRST)=3
      IF(NUMBER.EQ.6700) CALL CDTRDTR(NFIRST,N1,NDISCL(N1))
      IF(NUMBER.EQ.6750) CALL SCDTDET(NFIRST,NCROSS(NFIRST),N1-2,
     *NCROSS(N1-2),N1-1,NCROSS(N1-1),N1,NCROSS(N1),4H6750)
      IF(NUMBER.EQ.7000) CALL TRARE(NDISCL(NFIRST))
      IF(NUMBER.EQ.7500) CALL TRDET(NFIRST,NCROSS(NFIRST),N1,NCROSS(N1),
     *8HDISC  TD)
      IF(NUMBER.EQ.8000) CALL CD(RD(N1),SE(N1),NDISCL(N1),5HCDRAR)
      IF(NUMBER.EQ.8200) CALL CDSHK(NFIRST,N1,6HCDRSHK)
C-----CHECK IF ALL DISC HAVE BEEN HANDLED
      NFIRST=I
      N1=1
      IF((N1.EQ.NTDISCS-1).AND.(I.EQ.NTDISCS)) GO TO 20
C
 1    CONTINUE
C
C
```

```
C
C
C-----LOOP 200 RESETS DISC IN AN ASCENDING SPACIAL ORDER
 100  NT=0
      DO 200 I=1,NTDISCT
      IF(NDISCL(I).LT.0) GO TO 200
      NT=NT+1
      IF(NT.NE.1) GO TO 210
C
C-----SET 1ST DISC POSITION AND PARAMETERS
      NTYPE(NT)=NTYPE(I) $ NDISCL(NT)=NDISCL(I) $ NDISCNO(NT)=NDISCNO(I)
      SL(NT)=SL(I) $ SE(NT)=SE(I) $ RD(NT)=RD(I)
      GO TO 200
C
C-----SET DISC RELATIVE TO THE OTHER DISC
 210  IILAST=NT-1
      DO 220 II=1,IILAST
      IF(RD(I).GT.RD(II)) GO TO 220
C
C-----SET DISC BETWEEN THE OTHERS
      NTYPE(NTDISCT+1)=NTYPE(I) $ NDISCL(NTDISCT+1)=NDISCL(I)
      NDISCNO(NTDISCT+1)=NDISCNO(I)
      SL(NTDISCT+1)=SL(I) $ SE(NTDISCT+1)=SE(I) $ RD(NTDISCT+1)=RD(I)
      DO 221 III=II,IILAST
      L=NT-III+II
      NTYPE(L)=NTYPE(L-1) $ NDISCL(L)=NDISCL(L-1)
      NDISCNO(L)=NDISCNO(L-1) $ SL(L)=SL(L-1 ) $SE(L)=SE(L-1)
 221  RD(L)=RD(L-1)
      NTYPE(II)=NTYPE(NTDISCT+1) $ NDISCL(II)=NDISCL(NTDISCT+1)
      NDISCNO(II)=NDISCNO(NTDISCT+1)
      SL(II)=SL(NTDISCT+1) $ SE(II)=SE(NTDISCT+1) $ RD(II)=RD(NTDISCT+1)
      GO TO 200
 220  CONTINUE
C
C-----SET DISC AS LAST DISC
      NTYPE(NT)=NTYPE(I) $ NDISCL(NT)=NDISCL(I) $ NDISCNO(NT)=NDISCNO(I)
      SL(NT)=SL(I) $ SE(NT)=SE(I) $ RD(NT)=RD(I)
C
 200  CONTINUE
C
      NTDISCT=NT
      RETURN $ END
```

```
      SUBROUTINE FLMACCL(SFLMNEW,IFLM,P30,D30)
      COMMON/PARAM/N,J,AJ,G,GF,DELR,NFLM
      COMMON/TIME/T,DT,DTL,TWRITE,DELT,DTDX,AT
      COMMON/POWER/VCAPF,R4R1,PDPOWER,PPOWER,CND,CCAPF,SFEOLD,SFE,NCJ
      COMMON/ARRAYS/R(501),U(2,501),P(2,501),D(2,501),E(2,501),R2(501)
      COMMON/DISCS/NTDISC,NDISCND(51),NTYPE(51),NDISCL(51),SE(51),SL(51)
     *,RD(51),NTDISCT,NTDISCS
C
C-----DET SS FLW FLD (STTES 3,4,5) DUE TO SUDDEN FLM ACC BY POLAR METHOD
C     P43=TO BE READ AS PRESS IN STATE4 REL TO AND NONDIM BY STATE 3
C     U520=READ AS PRTCLE VEL IN STATE5 REL TO STATE2, NONDIM WRT STATE0
C
C     -STATE-     ---2---SHK--5--CD--4--FLM--3--SHK----1---SHK---0---
C                         *      *       *       *          *
C                          *     *             *          *
C                                *      *     *          *
C                           *    *     *    *          *
C                            *   *    *   *          *
C                             *  *   *   *         *
C                              * *  *  *         *
C                              * * *  *        *
C                               **  *         *
C                                **         *
C                                *         *
C                               FLM      *
C                                *     *
C                                    *
C                               *   *
C                                 *
C                                *
C                         ---2---*---0---
C
C-----SET STATES 1 AND 2
      P10=P(2,NFLM+1) $ D10=D(2,NFLM+1) $ U10=U(2,NFLM+1)
      P20=P(2,NFLM) $ D20=D(2,NFLM) $ U20=U(2,NFLM)
      A10=SQRT(P10/D10) $ A20=SQRT(P20/D20*GF/G)
C-----SET SUBROUTINE CONSTS
      BETA=(GF-1.)/(GF+1.) $ B=(GF-1.)/(G-1.) $ BETI=(G-1.)/(G+1.)
      PG0=(1.+BETA)*(VCAPF-BETA)/(1.-BETA)-BETA
      DELP30=0.1 $ EPS=1.E-14
C-----INITIALIZE ITERATION COUNTER AND PRESSURE GUESS
      NSIGN=0 $ P30=FLOAT(IFIX(P10*10.0))/10.0 $ NITER=0
C
C
C-----PERFORM ITERATION OF THE POLAR SOLN.
      PRINT 1002
 1002 FORMAT(1H1,20X,*SUDDEN FLM ACCL*,2/)
 1    NITER=NITER+1
      IF(NITER.LT.71) GO TO 2
      IF(NITER.EQ.71) DELUSAV=DELU
      IF(NITER.LT.100) GO TO 2
      IF((DELUSAV.LT.1.E-12).AND.(DELU.LT.1.E-12)) GO TO 10
      PRINT 1001
 1001 FORMAT(1H ,5/,10X,*NO. OF ITERATIONS HAS REACHED MAXIMUM OF 100*)
      DELR=-ABS(DELR) $ RETURN
C-----GUESS P30
```

```
 2     P30=P30+DELP30
       P31=P30/P10
       SSL31=SQRT((P31+BETI)/(1.+BETI))
       SSE30=SSL31*A10*SQRT(G)+U10
       U310=(1.-BETI)*(1.-1./SSL31**2)*SSL31*A10*SQRT(G)
       U30=U310+U10
       D31=1./(BETI+(1.-BETI)/SSL31**2) $ D30=D31*D10
       A31=SQRT((1.-BETI)**2*(SSL31**2-BETI/(1.+BETI))*(1./SSL31**2+
      *BETI/(1.-BETI))*G)
       A30=A31*A10
C
       PG3=1.+(PG0-1.)/A30**2+GF*(1.-1./A30**2)*(G/GF*P-P4R1)
       SSL43=SFLMNEW/SQRT(G)/A30
       SSE40=SFLMNEW+U30
       P43=(-(BETA-1.-SSL43**2*(1.-BETA)*G)+SQRT((BETA-1.-SSL43**2*(1.-
      *BETA)*G)**2+4.*(BETA-SSL43**2*(1.-BETA)*PG3*G)))/2.
       P40=P43*P30
       D43=1./(1.-(1.-BETA)*(P43-PG3)/(P43+BETA)) $ D40=D43*D30
       U430=-SQRT((1.-BETA)*(P43-PG3)*(P43-1.)/G/(P43+BETA))*A30*SQRT(G)
       U40=U430+U30
       A43=SQRT(((PG3+BETA)+BETA*(P43-PG3))/(P43+BETA)*GF/G*P43)
       A40=A43*A30
C
C-----ASSUME  P50=P40
       P50=P40
       P52=P50/P20
       SSL52=-SQRT((P52+BETA)/(1.+BETA))
       SSE50=SSL52*A20*SQRT(G)+U20
       U520=(1.-BETA)*(1.-1./SSL52**2)*SSL52*A20*SQRT(G)
       U50=U520+U20
       D52=1./(BETA+(1.-BETA)/SSL52**2) $ D50=D52*D20
       A52=SQRT((1.-BETA)**2*(SSL52**2-BETA/(1.+BETA))*(1./SSL52**2+
      *BETA/(1.-BETA))*GF)
       A50=A52*A20
C
       DELU=ABS(U40-U50)
C
       PRINT 1000,NITER,J,DELP30,PG0,PG3,DELU,
      *P10,D10,U10,A10      ,P20,D20,U20,A20,SL(IFLM),
      *P31,P30,D31,D30,U310,U30,A31,A30,SSL31,SSE30,
      *P43,P40,D43,D40,U430,U40,A43,A40,SSL43,SSE40,
      *P52,P50,D52,D50,U520,U50,A52,A50,SSL52,SSE50
 1000 FORMAT(1H ,1X,2I5,4E16.8,/,1X,4F13.8,13X,5F13.8,3(/,1X,10F13.8),/)
C
C
       IF(DELU.LE.EPS) GO TO 10
       IF((U40.LT.U50).AND.(NSIGN.EQ.1)) DELP30=ABS(DELP30)/2.0
       IF(U40.LT.U50) GO TO 1
       DELP30=-ABS(DELP30)/2.0
       NSIGN=1 $ GO TO 1
C
C
C-----ADD THE DISCS TO THE FLOW FIELD
 10    NT=NTDISCT
       NDISCNO(NT+1)=NT+1 $ NDISCNO(NT+2)=NT+2 $ NDISCNO(NT+3)=NT+3
       NTYPE(NT+1)=2 $ NTYPE(NT+2)=3 $ NTYPE(NT+3)=2
```

```
      SL(NT+1)=SSL52 $ SL(NT+2)=0.0 $ SL(IFLM)=SFLMNEW $ SL(NT+3)=SSL31
      SE(NT+1)=SSE50 $ SE(NT+2)=U40 $ SE(IFLM)=SSE40 $ SE(NT+3)=SSE30
      RD(NT+1)=RD(IFLM)*(1.-1.E-10) $ RD(NT+2)=RD(IFLM)*(1.-.5E-10)
      RD(NT+3)=RD(IFLM)*(1.+1.E-10)
C
      DO 100 I=NFLM,N
      M=N+NFLM-I+2
      D(2,M)=D(2,M-2) $ U(2,M)=U(2,M-2)
      E(2,M)=E(2,M-2) $ P(2,M)=P(2,M-2)
 100  R2(M)=R2(M-2)
      N=N+2
      R2(NFLM)=R2(NFLM+1)=RD(NT+2)
      DO 101 I=IFLM,NT
 101  NDISCL(I)=NDISCL(I)+2
      NFLM=NFLM+2
      NDISCL(NT+1)=NFLM-3 $ NDISCL(NT+2)=NFLM-2 $ NDISCL(NT+3)=NFLM+1
      NTDISCT=NTDISC=NT+3
C-----SET THERMO AND GAS PARAMETERS IN CELLS NFLM-2,-1,0,+1
      D(2,NFLM-2)=D50 $ U(2,NFLM-2)=U50 $ P(2,NFLM-2)=P50
      D(2,NFLM-1)=D40 $ U(2,NFLM-1)=U40 $ P(2,NFLM-1)=P40
      D(2,NFLM)=D40 $ U(2,NFLM)=U40 $ P(2,NFLM)=P40
      D(2,NFLM+1)=D30 $ U(2,NFLM+1)=U30 $ P(2,NFLM+1)=P30
      E(2,NFLM-2)=P50/D50/(GF-1.)+U50**2/2.
      E(2,NFLM-1)=E(2,NFLM)=P40/D40/(GF-1.)+U40**2/2.
      E(2,NFLM+1)=P30/D30/(G-1.)+U30**2/2.
C
      RETURN $ END
```

```
      SUBROUTINE FLM(RF,SFES,SFL,NAME)
      COMMON/PARAM/N,J,AJ,G,GF,DELR,NFLM
      COMMON/TIME/T,DT,DTL,TWRITE,DELT,DTDX,AT
      COMMON/PREDCOR/DPC(2,13),UPC(2,13),EPC(2,13),PPC(2,13)
      COMMON/POWER/VCAPF,R4R1,PDPOWER,EPOWER,CND,GCAPF,SFEOLD,SFE,NCJ
      COMMON/ARRAYS/R(501),U(2,501),P(2,501),D(2,501),E(2,501),R2(501)
C
C     PD,VD=PRESS,SPECIFIC VOL RATIO ACROSS THE FLAME
C
      IF(NAME.EQ.6H FLRCD) GO TO 1
C-----PREDICTOR FOR NFLM+1 = 3
      M=NFLM+1
      DPC(1,3)=DP(M,0) $ UPC(1,3)=UP(M,0)
      EPC(1,3)=EP(M,0) $ PPC(1,3)=PP(3,0)
C-----PREDICTOR FOR NFLM = 2
      CALL SFVDPD(G,GF,PPC(1,3),DPC(1,3),SFL,VD,PD,DELR)
      DPC(1,2)=DPC(1,3)/VD $ PPC(1,2)=PPC(1,3)*PD
      UPC(1,2)=(SFL*(1.-VD)+UPC(1,3)/DPC(1,3))*DPC(1,2)
      EPC(1,2)=DPC(1,2)*(PPC(1,2)/DPC(1,2)/(GF-1.)+UPC(1,2)**2/DPC(1,2)
     ***2/2.0)
C-----PREDICTOR FOR NFLM-1 = 1
 1    M=NFLM-1
      DPC(1,1)=DP(M,0) $ UPC(1,1)=UP(M,0)
      EPC(1,1)=EP(M,0) $ PPC(1,1)=PP(1,GF)
C-----CORRECTOR FOR NFLM = 2
      DPC(2,2)=D(2,NFLM)=DC(2,NFLM,0) $ UPC(2,2)=U(2,NFLM)=UC(2,NFLM,0)
      EPC(2,2)=E(2,NFLM)=EC(2,NFLM,0) $ PPC(2,2)=P(2,NFLM)=PC(2,GF)
C-----CORRECTOR FOR NFLM+1 = 3
      VD=DPC(1,3)/DPC(1,2)
      CALL FLM43(2,VD,SFL)
      M=NFLM+1
      D(2,M)=DPC(2,3) $ U(2,M)=UPC(2,3) $E(2,M)=EPC(2,3)$P(2,M)=PPC(2,3)
C-----SET FLAME POSITION AND SPEED
      RF=SFES*DT+RF $ SFES=SFE
C
      RETURN $ END
```

```
      SUBROUTINE SFVDPD(G,GF,PAVG,DAVG,SF,VD,PD,DELR)
      COMMON/POWER/VCAPF,R4R1,PDPOWER,PPOWER,CND,CCAPF,SFEOLD,SFE,NCJ
C
      FLMSPD1(PAVG,DAVG)=CND*(PAVG/DAVG)**PDPOWER*PAVG**PPOWER
C
C     VF=CONST PRESS SPECIFIC VOL RATIO AT FLAME STATE CONDITIONS
C
      SFSAV=SF
      BETA=(GF-1.)/(GF+1.)
      VF=VCAPF/PAVG*DAVG+(1.-DAVG/PAVG)*(G*(GF-1.)/GF/(G-1.)-R4R1+1.)
      SF=FLMSPD1(PAVG,DAVG)
      IF(ABS(SFSAV-SF).LT.1.E-08) SF=SFSAV
      C=2.*G*SF**2/G/PAVG*DAVG $ A=(1.+BETA)*(1.+C/2.)
      B=A*A-2.*C*(C/2.*BETA+(1.+BETA)*VF)
      IF(B.GE.0.0) GO TO 1
      SFSAV1=SF $ IF(NCJ.EQ.3H NO) DELR=-ABS(DELR)
      SF=SQRT((1.+BETA)*PAVG/DAVG)/(1.-BETA)*(SQRT(VF-BETA)-SQRT(VF-1.))
      IF(ABS(SFSAV-SF).LT.1.E-08) SF=SFSAV
      C=2.*G*SF**2/G/PAVG*DAVG $ A=(1.+BETA)*(1.+C/2.)
      PRINT 1000,B,PAVG,DAVG,SFSAV1,SF,NCJ
 1000 FORMAT(1H ,5X,*E.LT.0*,E17.9,1X,4F20.12,5X,A3)
      VD=A/C
      GO TO 2
 1    IF(NCJ.EQ.3HYES) DELR=-ABS(DELR)
      IF(NCJ.EQ.3HYES) PRINT 1000,B,PAVG,DAVG,SFSAV,SF,NCJ
      VD=(A-SQRT(B))/C
 2    PD=1.+G*SF**2/G/PAVG*DAVG*(1.-VD)
C
      RETURN $ END
```

```
      SUBROUTINE FLM43(K,VD,SFL)
      COMMON/PARAM/N,J,AJ,G,GF,DELR,NFLM
      COMMON/PREDCOR/DPC(2,13),UPC(2,13),EPC(2,13),PPC(2,13)
      COMMON/POWER/VCAPF,R4R1,PDPOWER,PPOWER,CND,CCAPF,SFEOLD,SFE,NCJ
C
      FLMSPD2(PAVG,DAVG)=CND*(PAVG/DAVG)**PDPOWER*PAVG**PPOWER
      L=K+1 $ CONST=G*(GF-1.)/GF/(G-1.)-R4R1+1. $ BETA=(GF-1.)/(GF+1.)
C
C
      SFLSAV=SFL
      IF(NCJ.EQ.3HYES) GO TO 15
C
      NITER=0 $ EPS=1.E-14
      DPC(2,L)=DPC(2,K)*VD $ PPC(2,L)=PPC(1,L)
      DO 1 ITER=1,45
      SFL=FLMSPD2(PPC(2,L),DPC(2,L))
      G1=-PPC(2,K)+PPC(2,L)+SFL**2*DPC(2,L)*(1.-VD)
      DG1DP3=1.+2.*SFL**2*(PDPOWER+PPOWER)*(1.-VD)*DPC(2,L)/PPC(2,L)
      DP3=-G1/DG1DP3 $ PPC(2,L)=PPC(2,L)+DP3
      IF(ABS(DP3).LT.EPS*PPC(1,L)) GO TO 2
 1    CONTINUE
      PRINT 2000,NITER,PPC(1,L),PPC(2,L),DP3,G1,DG1DP3,SFL,SFL
 2000 FORMAT(1H  1,1X,*FLM43*,I4,7E17.10)
      IF(ABS(DP3).LE.2000.*EPS*PPC(1,L)) GO TO 2
      DELR=-ABS(DELR) $ RETURN
 2    SFL=FLMSPD2(PPC(2,L),DPC(2,L))
C
 5    NITER=NITER+1
      IF(NITER.LT.35) GO TO 6
      IF(NITER.EQ.35) DV3SAV=DV3
C     PRINT 2000,NITER,G3,DV3,PPC(2,L),PPC(2,K),DPC(2,L),DPC(2,K),SFL
      IF(NITER.LT.40) GO TO 6
      IF((ABS(DV3SAV).LT.500.*EPS*DPC(1,L)).AND.(ABS(DV3).LT.500.*EPS*
     *DPC(1,L))) GO TO 16
      PRINT 2000,NITER,G3,DV3,PPC(2,L),PPC(2,K),DPC(2,L),DPC(2,K),SFL
      PRINT 2001
 2001 FORMAT(1H ,5/,1X,*ITERATION EQUALS MAXIMUM IN FLM43*)
      DELR=-ABS(DELR) $ RETURN
 6    D3D4=DPC(2,L)/DPC(2,K)
      G3=PPC(2,K)*(BETA-D3D4)+(1.+BETA)*(VCAPF-CONST)*DPC(2,L)+
     *(SFL**2*DPC(2,L)*(1.-D3D4)-PPC(2,K))*(BETA*D3D4-BETA**2-(1.+BETA)*
     *(CONST-BETA))
      DPDV3=(D3D4/(1.-D3D4)+2.*PDPOWER-1.)/(1./SFL**2/(1.-D3D4)+
     *2.*DPC(2,L)/PPC(2,L)*(PDPOWER+PPOWER))
      DSDV3=-PDPOWER*SFL/DPC(2,L)+(PDPOWER+PPOWER)*SFL/PPC(2,L)*DPDV3
      DG3DV3=-(1.+BETA)*PPC(2,K)/DPC(2,K)+(1.+BETA)*(VCAPF-CONST)+
     *SFL**2*DPC(2,L)*(1.-D3D4)*(BETA*(D3D4-BETA)-(1.+BETA)*(CONST-BETA)
     *)*(2./SFL*DSDV3+(1.-2.*D3D4)/DPC(2,L)/(1.-D3D4)+BETA/DPC(2,K)/
     *(BETA*(D3D4-BETA)-(1.+BETA)*(CONST-BETA)))
      DV3=-G3/DG3DV3
      DPC(2,L)=DPC(2,L)+DV3
C
      DO 10 ITER=1,45
      G1=-PPC(2,K)+PPC(2,L)+SFL**2*DPC(2,L)*(1.-VD)
      DG1DP3=1.+2.*SFL**2*(PDPOWER+PPOWER)*(1.-VD)*DPC(2,L)/PPC(2,L)
      DP3=-G1/DG1DP3 $ PPC(2,L)=PPC(2,L)+DP3
```

```
      SFL=FLWSPD2(PPC(2,L),DPC(2,L))
      IF(ABS(DP3).LT.EPS*PPC(1,L)) GO TO 11
 10   CONTINUE
      PRINT 2000,NITER,PPC(1,L),PPC(2,L),DP3,G1,DG1DP3,SFL
      IF(ABS(DP3).LE.2000.*EPS*PPC(1,L)) GO TO 11
      DELR=-ABS(DELR) $ RETURN
 11   IF(ABS(DV3).GT.EPS*DPC(1,L)) GO TO 5
C     PRINT 2003,NITER
 2003 FORMAT(1H ,1X,*NITER *,I5)
      GO TO 16
C
C
 15   CONST=G*(GF-1.)/GF/(G-1.)-R4R1+1.
      A=(1.+BETA)**2*DPC(2,K)/PPC(2,K)*(VCAFF-CONST)-1.+BETA
      B=2.*((1.+BETA)*CONST-2.*BETA) $C=(1.-BETA)*(BETA-CONST*(1.+BETA))
      PD=(-B+SQRT(B*B-4.*A*C))/2./A $ VD=PD*(1.+BETA)/(2.*PD-1.+BETA)
      PPC(2,L)=PPC(2,K)/PD $ DPC(2,L)=DPC(2,K)*VD
      SFL=SQRT((1.+BETA)*PPC(2,L)/DPC(2,L)/(2.*VD-1.-BETA))
C
C
 16   UPC(2,L)=(DPC(2,L)/DPC(2,K)*SFL-SFL+UPC(2,K)/DPC(2,K))*DPC(2,L)
      EPC(2,L)=DPC(2,L)*(PPC(2,L)/DPC(2,L)/(G-1.)+UPC(2,L)**2/DPC(2,L)**
     *2/2.)
C
      IF(ABS(SFLSAV-SFL).GT.1.E-08) GO TO 17
      SFL=SFLSAV
      DPC(2,L)=DPC(1,L) $ UPC(2,L)=UPC(1,L)
      EPC(2,L)=EPC(1,L) $ PPC(2,L)=PPC(1,L)
 17   CONTINUE
      SFE=SFL+UPC(2,L)/DPC(2,L)
C
      RETURN $ END
```

```
      SUBROUTINE DET(RS,SSE,SSL,NSHK,NXSS,NAME)
      COMMON/PARAM/N,J,AJ,G,GF,DELR,NFLM
      COMMON/TIME/T,DT,DTL,TWRITE,DELT,DTDX,AT
      COMMON/TRDT/PLTR,DLTR,ULTR,ELTR,OVERDNM,NOVERDN
      COMMON/PREDCOR/DPC(2,13),UPC(2,13),EPC(2,13),PPC(2,13)
      COMMON/POWER/VCAPF,R4R1,PCPOWER,PPOWER,CND,CCAPF,SPEOLD,SPE,NOJ
      COMMON/ARRAYS/R(501),U(2,501),P(2,501),D(2,501),E(2,501),R2(501)
C
      PG1(A)=1.+(PG0-1.)/A**2+GF*(1.-1./A**2)*(G/GF*B-R4R1)
C
C     OVERDNM=OVERDRIVEN DETONATION MACH NUMBER
C     NOVERDN=NO. OF TIME STEPS DET HAS BEEN OVERDRIVEN
C
      BETA=(GF-1.)/(GF+1.) $ B=(GF-1.)/(G-1.)
      PG0=(1.+BETA)*(VCAPF-BETA)/(1.-BETA)-BETA
C
C
C-----PREDICTOR FOR NSHK = 2
      A=SQRT(P(1,NSHK+1)/D(1,NSHK+1)) $ PG=PG1(A)
      PS1=(1.-BETA)/2.*(1.+G*SSL**2)
      IF(NOVERDN.LT.20) PS1=PS1+SQRT(PS1**2-(1.-BETA)*G*SSL**2*PG+BETA)
      PS=PS1*P(1,NSHK+1)
      DS=D(1,NSHK+1)/(1.-(1.-BETA)*(PS1-PG)/(PS1+BETA))
      US=DS*(SQRT(G)*A*SQRT((1.-BETA)*(PS1-PG)*(PS1-1.)/G/
     *(PS1+BETA))+U(1,NSHK+1)/D(1,NSHK+1))
      ES=(PS/DS/(GF-1.)+US**2/DS**2/2.)*DS
      IF(NAME.EQ.7HTRDET  ) GO TO 10
      EPS=(RS-R(NSHK))/DELR
      C1=2.*(2.-EPS)/(1.+EPS)
      C2=2.*EPS-3. $ C3=(1.-EPS)*(2.*EPS-1.)/(1.+EPS)
      DPC(1,2)=D(1,NSHK)-DTDX*(C1*US+C2*U(1,NSHK)+C3*U(1,NSHK-1))
      UPC(1,2)=U(1,NSHK)-DTDX*(C1*(US*US/DS+PS*RS**J)+C2*PM(NSHK)+
     *C3*PM(NSHK-1))
      EPC(1,2)=E(1,NSHK)-DTDX*(C1*US/DS*(ES+PS*RS**J)+C2*PE(NSHK)+C3*
     *PE(NSHK-1))
      PPC(1,2)=PP(2,GF)
      IF(NAME.EQ.7HSKCD DT) GO TO 10
C-----PREDICTOR FOR NSHK-1 = 1
      M=NSHK-1
      DPC(1,1)=DP(M,0) $ UPC(1,1)=UP(M,0)
      EPC(1,1)=EP(M,0) $ PPC(1,1)=PP(1,GF)
C-----CORRECTOR FOR NSHK = 2
      D(2,NSHK)=DPC(2,2)=DC(2,NSHK,0) $ U(2,NSHK)=UPC(2,2)=UC(2,NSHK,0)
      E(2,NSHK)=EPC(2,2)=EC(2,NSHK,0) $ P(2,NSHK)=PPC(2,2)=PC(2,GF)
C-----PREDICTOR FOR NSHK+1+NXSS = 3+NXSS
 10   M=NSHK+1+NXSS $ MM=3+NXSS
      DPC(1,MM)=DP(M,0) $ UPC(1,MM)=UP(M,0)
      EPC(1,MM)=EP(M,0) $ PPC(1,MM)=PP(MM,G)
C-----PREDICTOR FOR NSHK+2+NXSS = 4+NXSS
      M=NSHK+2+NXSS $ MM=4+NXSS
      DPC(1,MM)=DP(M,0) $ UPC(1,MM)=UP(M,0)
      EPC(1,MM)=EP(M,0) $ PPC(1,MM)=PP(MM,G)
C-----CORRECTOR FOR NSHK+1+NXSS = 3+NXSS
      M=NSHK+1+NXSS $ MM=3+NXSS
      D(2,M)=DPC(2,MM)=DC(MM,M,1) $ U(2,M)=UPC(2,MM)=UC(MM,M,1)
      E(2,M)=EPC(2,MM)=EC(MM,M,1) $ P(2,M)=PPC(2,MM)=PC(MM,G)
```

```
C
      IF(NXSS.EQ.0) GO TO 1
C-----PREDICTOR AND CORRECTOR FOR NSHK+1 = 3
      M=NSHK+1
      D(2,M)=DPC(2,3)=DPC(1,3)=DS $ U(2,M)=UPC(2,3)=UPC(1,3)=US
      E(2,M)=EPC(2,3)=EPC(1,3)=ES $ P(2,M)=PPC(2,3)=PPC(1,3)=PS
      SSESAV=SSE
      IF(NOVERDN.GE.20) GO TO 2
      A=SQRT(P(1,M)/D(1,M)) $ PG=PG1(A)
      SSLSAV=SQRT((2.*PG-(1.-BETA)+2.*SQRT(PG**2-(1.-BETA)*PG-BETA))/
     *(1.-BETA)/G)
      IF(OVERDNM.GT.SSLSAV) SSLSAV=(SSL+SSLSAV+(OVERDNM-SSLSAV)/20.*
     *FLOAT(20-NOVERDN-1))/2.
      IF(OVERDNM.LE.SSLSAV) SSLSAV=(SSL+SSLSAV)/2.
      SSESAV=SSLSAV*A*SQRT(G)+U(1,M)/D(1,M)
      P(2,M)=(1.-BETA)/2.*(1.+G*SSLSAV**2)
      P(2,M)=(P(2,M)+SQRT(P(2,M)**2-G*(1.-BETA)*SSLSAV**2*PG+BETA))*
     *P(1,M)
      PS1=P(2,M)/P(1,M)
      D(2,M)=D(1,NSHK+1)/(1.-(1.-BETA)*(PS1-PG)/(PS1+BETA))
      U(2,M)=D(2,M)*(SQRT(G)*A*SQRT((1.-BETA)*(PS1-PG)*(PS1-1.)/G/
     *(PS1+BETA))+U(1,NSHK+1)/D(1,NSHK+1))
      E(2,M)=D(2,M)*(P(2,M)/D(2,M)/(GF-1.)+U(2,M)**2/D(2,M)**2/2.)
      GO TO 2
C-----CALCULATE DETONATION POSITION
 1    A=SQRT(PPC(1,3)/DPC(1,3)) $ PG=PG1(A)
      SSLSAV=SQRT((2.*PG-(1.-BETA)+2.*SQRT(PG**2-(1.-BETA)*PG
     *-BETA))/(1.-BETA)/G)
      SSESAV=A*SQRT(G)*SSLSAV+UPC(1,3)/DPC(1,3)
      IF(OVERDNM.LE.SSLSAV) GO TO 2
      IF(NOVERDN+1.GE.20) GO TO 2
      SSL=SSLSAV+(OVERDNM-SSLSAV)/20.*FLOAT(20-NOVERDN-1)
      SSESAV=SSL*A*SQRT(G)+UPC(1,3)/DPC(1,3)
 2    RS=RS+(SSE+SSESAV)/2.*DT
      IF(((RS.GT.R2(NSHK+1)).AND.(NXSS.EQ.0)).OR.((RS.LT.R2(NSHK+1)).
     *AND.(NXSS.EQ.1))) RS=R2(NSHK+1)
C
C-----ADVANCE DETONATION INDEX
      NFLM=NSHK=NSHK+NXSS
C
C-----CALCULATE DETONATION SPEED
      A=SQRT(PPC(2,3+NXSS)/DPC(2,3+NXSS)) $ PG=PG1(A)
      SSL=SQRT((2.*PG-(1.-BETA)+2.*SQRT(PG**2-(1.-BETA)*PG-BETA))/
     *(1.-BETA)/G)
      SSE=SSL*SQRT(G)*A+UPC(2,3+NXSS)/DPC(2,3+NXSS)
      IF(NOVERDN.GE.20) RETURN
      IF(OVERDNM.LE.SSL) NOVERDN=100
      IF(OVERDNM.LE.SSL) RETURN
      SSLSAV=SSL $ SSESAV=SSE
      NOVERDN=NOVERDN+1
      SSL=SSL+(OVERDNM-SSL)/20.*FLOAT(20-NOVERDN)
      SSE=SSL*A*SQRT(G)+UPC(2,3+NXSS)/DPC(2,3+NXSS)
      PRINT 1000,NOVERDN,OVERDNM,SSL,SSLSAV,SSE,SSESAV
 1000 FORMAT(1H ,I6,3E20.10,10X,2E20.10)
C
      RETURN $ END
```

```
      SUBROUTINE SHK(RS,SSE,SSL,NSHK)
      COMMON/UBDCND/NUBC
      COMMON/PARAM/N,J,AJ,G,GF,DELR,NFLM
      COMMON/TIME/T,DT,DTL,TWRITE,DELT,DTDX,AT
      COMMON/PREDCOR/DPC(2,13),UPC(2,13),EPC(2,13),PPC(2,13)
      COMMON/ARRAYS/R(501),U(2,501),P(2,501),D(2,501),E(2,501),R2(501)
C
C
      GE=GF $ IF(NSHK.GT.NFLM) GE=G
C------CHECK SHK DIRN AND SHK X/O
      NSHKSGN=0 $ IF(SSL.GT.0.0) NSHKSGN=1
      NCROSS=0 $ IF(RS+SSE*DT.GE.R2(NSHK+1)) NCROSS=1
      IF(NSHK.EQ.1) GO TO 1
      IF(RS+SSE*DT.LT.R2(NSHK)) GO TO 10
C
C
C------PREDICTOR FOR NSHK-1 = 1
      M=NSHK-1
      DPC(1,1)=DP(M,0) $ UPC(1,1)=UP(M,0)
      EPC(1,1)=EP(M,0) $ PPC(1,1)=PP(1,GE)
C------PREDICTOR FOR NSHK = 2
      IF(NSHKSGN.EQ.1) GO TO 3
      DPC(1,2)=DP(NSHK,-1) $ UPC(1,2)=UP(NSHK,-1)
      EPC(1,2)=EP(NSHK,-1) $ PPC(1,2)=PP(2,GE)
      GO TO 4
 3    PS=P(1,NSHK+1)*(2.*GE/(GE+1.)*SSL**2-(GE-1.)/(GE+1.))
      DS=D(1,NSHK+1)/((GE-1.)/(GE+1.)+2./(GE+1.)/SSL**2)
      US=DS*(U(1,NSHK+1)/D(1,NSHK+1)+SQRT(GE*P(1,NSHK+1)/D(1,NSHK+1))*2.
     */(GE+1.)*SSL*(1.-1./SSL**2))
      ES=(PS/DS/(GE-1.)+US**2/DS**2/2.)*DS
      EPS=(RS-R(NSHK))/DELR
      C1=2.*(2.-EPS)/(1.+EPS)
      C2=2.*EPS-3. $ C3=(1.-EPS)*(2.*EPS-1.)/(1.+EPS)
      DPC(1,2)=D(1,NSHK)-DTDX*(C1*US+C2*U(1,NSHK)+C3*U(1,NSHK-1))
     *-AT*DTDX*(C1*DS+C2*D(1,NSHK)+C3*D(1,NSHK-1))
      UPC(1,2)=U(1,NSHK)-DTDX*(C1*(US*US/DS+PS*RS**J)+C2*PM(NSHK)+
     *C3*PM(NSHK-1))-AT*DTDX*(C1*US+C2*U(1,NSHK)+C3*U(1,NSHK-1))
      EPC(1,2)=E(1,NSHK)-DTDX*(C1*US/DS*(ES+PS*RS**J)+C2*PE(NSHK)+C3*
     *PE(NSHK-1))-AT*DTDX*(C1*ES+C2*E(1,NSHK)+C3*E(1,NSHK-1))
      PPC(1,2)=PP(2,GE)
C------CORRECTOR FOR NSHK = 2
 4    D(2,NSHK)=DPC(2,2)=DC(2,NSHK,0) $ U(2,NSHK)=UPC(2,2)=UC(2,NSHK,0)
      E(2,NSHK)=EPC(2,2)=EC(2,NSHK,0) $ P(2,NSHK)=PPC(2,2)=PC(2,GE)
C
      IF(NSHK.NE.2) GO TO 2
C
C------SET D,U,E,P FOR NSHK(=2)-1 = 1
      D(2,1)=D(2,2) $ U(2,1)=-U(2,2) $ E(2,1)=E(2,2) $ P(2,1)=P(2,2)
      IF(NUBC.EQ.3HYES) U(2,1)=U(1,1)
      GO TO 2
C
C------SET D,U,E,P FOR NSHK(=1) = 2
 1    D(2,1)=DPC(1,2)=D(1,1) $ U(2,1)=UPC(1,2)=U(1,1)
      E(2,1)=EPC(1,2)=E(1,1) $ P(2,1)=PPC(1,2)=P(1,1)
C
C
```

```
C-----SHK CROSS OVER DECISION
 2    IF(NCROSS.EQ.1) GO TO 5
C
C-----PREDICTOR FOR NSHK+2 = 4
      M=NSHK+2
      DPC(1,4)=DP(M,0) $ UPC(1,4)=UP(M,0)
      EPC(1,4)=EP(M,0) $ PPC(1,4)=PP(4,GE)
C-----PREDICTOR FOR NSHK+1 = 3
      M=NSHK+1
      DPC(1,3)=DP(M,0) $ UPC(1,3)=UP(M,0)
      EPC(1,3)=EP(M,0) $ PPC(1,3)=PP(3,GE)
C-----CALC SHK POSITION
      SSLSAV=SQRT((PPC(1,3-NSHKSGN)/PPC(1,2+NSHKSGN)+(GE-1.)/(GE+1.))*
     *(GE+1.)/2./GE) * SSL/ABS(SSL)
      RS=(SSLSAV*SQRT(GE*PPC(1,2+NSHKSGN)/DPC(1,2+NSHKSGN))+
     *UPC(1,2+NSHKSGN)/DPC(1,2+NSHKSGN)+SSE)/2.*DT+RS
      IF((NSHK.EQ.1).AND.(NSHKSGN.EQ.0).AND.(NUBC.EQ.3H NO)) RS=RS-
     *2.*UPC(1,2+NSHKSGN)/DPC(1,2+NSHKSGN)/2.*DT
      IF(RS.GT.R2(NSHK+1)) RS=R2(NSHK+1)
      IF((NSHK.EQ.1).AND.(NSHKSGN.EQ.0)) GO TO 9
      IF(RS.LT.R2(NSHK)) RS=R2(NSHK)
C-----CORRECTOR FOR NSHK+1 = 3
 9    IF(NSHKSGN.EQ.0) GO TO 6
      D(2,M)=DPC(2,3)=DC(3,M,1) $ U(2,M)=UPC(2,3)=UC(3,M,1)
      E(2,M)=EPC(2,3)=EC(3,M,1) $ P(2,M)=PPC(2,3)=PC(3,GE)
      GO TO 7
 6    SSL=(SSL+SSLSAV)/2.
      PS=P(1,NSHK)*(2.*GE/(GE+1.)*SSL**2-(GE-1.)/(GE+1.))
      DS=D(1,NSHK)/((GE-1.)/(GE+1.)+2./(GE+1.)/SSL**2)
      US=DS*(U(1,NSHK)/D(1,NSHK)+SQRT(GE*P(1,NSHK)/D(1,NSHK))*2.
     */(GE+1.)*SSL*(1.-1./SSL**2))
      IF((NSHK.EQ.1).AND.(NUBC.EQ.3H NO))US=US-2.*U(1,NSHK)/D(1,NSHK)*DS
      ES=(PS/DS/(GE-1.)+US**2/DS**2/2.)*DS
      EPS=(R2(NSHK+1)-RS)/DELR
      C1=2.*(2.-EPS)/(1.+EPS)
      C2=2.*EPS-3. $ C3=(1.-EPS)*(2.*EPS-1.)/(1.+EPS)
      D(2,M)=DPC(2,3)=0.5*(D(1,M)+DPC(1,3)+DTDX*(C1*US+C2*UPC(1,3)+C3*
     *UPC(1,4))+AT*DTDX*(C1*DS+C2*DPC(1,3)+C3*DPC(1,4)))
      U(2,M)=UPC(2,3)=0.5*(U(1,M)+UPC(1,3)+DTDX*(C1*(US*US/DS+PS*RS**J)+
     *C2*CM(3)+C3*CM(4))+AT*DTDX*(C1*US+C2*UPC(1,3)+C3*UPC(1,4)))
      E(2,M)=EPC(2,3)=0.5*(E(1,M)+EPC(1,3)+DTDX*(C1*US/DS*(ES+PS*RS**J)+
     *C2*CE(3)+C3*CE(4))+AT*DTDX*(C1*ES+C2*EPC(1,3)+C3*EPC(1,4)))
      P(2,M)=PPC(2,3)=PC(3,GE)
C-----CALC NEW SHK SPD
 7    SSL=SQRT((P(2,NSHK+1-NSHKSGN)/P(2,NSHK+NSHKSGN)+(GE-1.)/(GE+1.))*
     *(GE+1.)/2./GE) * SSL/ABS(SSL)
      SSE=SSL*SQRT(GE*P(2,NSHK+NSHKSGN)/D(2,NSHK+NSHKSGN))+U(2,NSHK+
     *NSHKSGN)/D(2,NSHK+NSHKSGN)
      IF((NSHK.EQ.1).AND.(NSHKSGN.EQ.0).AND.(NUBC.EQ.3H NO)) SSE=SSE-
     *2.*U(2,NSHK+NSHKSGN)/D(2,NSHK+NSHKSGN)
      IF(RS.LT.0.0) GO TO 20
C
      RETURN
C
C
C
```

```
C-----POSITIVE SHK CROSSES MESH AT NSHK+1
C
C-----PREDICTOR FOR NSHK+1 = 3
 5    P(2,NSHK+1)=P(1,NSHK+1)*(2.*GE/(GE+1.)*SSL**2-(GE-1.)/(GE+1.))
C-----PREDICTOR FOR NSHK+3 = 5
      M=NSHK+3
      DPC(1,5)=DP(M,0) $ UPC(1,5)=UP(M,0)
      EPC(1,5)=EP(M,0) $ PPC(1,5)=PP(5,GE)
C-----PREDICTOR FOR NSHK+2 = 4
      M=NSHK+2
      DPC(1,4)=DP(M,0) $ UPC(1,4)=UP(M,0)
      EPC(1,4)=EP(M,0) $ PPC(1,4)=PP(4,GE)
C-----CORRECTOR FOR NSHK+2 = 4
      D(2,M)=DPC(2,4)=DC(4,M,1) $ U(2,M)=UPC(2,4)=UC(4,M,1)
      E(2,M)=EPC(2,4)=EC(4,M,1) $ P(2,M)=PPC(2,4)=PC(4,GE)
C-----CALC NEW SHK POSITION
      SSLSAV=SQRT((P(2,NSHK+1)/PPC(1,4)+(GE-1.)/(GE+1.))*(GE+1.)/2./GE)
      RS=(SSLSAV*SQRT(GE*PPC(1,4)/DPC(1,4))+UPC(1,4)/DPC(1,4)+SSE)/2.*DT
     *+RS
      IF(RS.LT.R2(NSHK+1)) RS=R2(NSHK+1)
C-----SET D,U,E,P FOR NSHK+1
      SSLSAV=(SSL+SSLSAV)/2.0
      D(2,NSHK+1)=D(1,NSHK+1)/((GE-1.)/(GE+1.)+2./(GE+1.)/SSLSAV**2)
      U(2,NSHK+1)=D(2,NSHK+1)*(U(1,NSHK+1)/D(1,NSHK+1)+SQRT(GE*P(1,NSHK+
     *1)/D(1,NSHK+1))*(2./(GE+1.)*SSLSAV*(1.-1./SSLSAV**2)))
      P(2,NSHK+1)=P(1,NSHK+1)*(2.*GE/(GE+1.)*SSLSAV**2-(GE-1.)/(GE+1.))
      E(2,NSHK+1)=D(2,NSHK+1)*(P(2,NSHK+1)/D(2,NSHK+1)/(GE-1.)+
     *U(2,NSHK+1)**2/D(2,NSHK+1)**2/2.0)
C-----CALC NEW SHK SPD
      SSL=SQRT((P(2,NSHK+1)/P(2,NSHK+2)+(GE-1.)/(GE+1.))*(GE+1.)/2./GE)
      SSE=SSL*SQRT(GE*P(2,NSHK+2)/D(2,NSHK+2))+U(2,NSHK+2)/D(2,NSHK+2)
C
      IF(NSHK.NE.1) GO TO 8
C-----SET D,U,E,P FOR NSHK(=1) = 2
      D(2,1)=D(2,2) $ U(2,1)=-U(2,2) $ E(2,1)=E(2,2) $ P(2,1)=P(2,2)
      IF(NUBC.EQ.3HYES) U(2,1)=U(1,1)
C
C-----ADVANCE SHK INDEX
 8    NSHK=NSHK+1
C
      RETURN
C
C
C
C-----NEGATIVE SHK CROSSES OVER MESH AT NSHK
C
 10   IF(NSHK.EQ.2) GO TO 11
C-----PREDICTOR FOR NSHK-2 = 1
      M=NSHK-2
      DPC(1,1)=DP(M,0) $ UPC(1,1)=UP(M,0)
      EPC(1,1)=EP(M,0) $ PPC(1,1)=PP(1,GE)
C-----PREDICTOR FOR NSHK-1 = 2
      M=NSHK-1
      DPC(1,2)=DP(M,0) $ UPC(1,2)=UP(M,0)
      EPC(1,2)=EP(M,0) $ PPC(1,2)=PP(2,GE)
C-----CORRECTOR FOR NSHK-1 = 2
```

```
      D(2,M)=DPC(2,2)=DC(2,M,0) $ U(2,M)=UPC(2,2)=UC(2,M,0)
      E(2,M)=EPC(2,2)=EC(2,M,0) $ P(2,M)=PPC(2,2)=PC(2,GE)
C
      IF(NSHK.NE.3) GO TO 12
C------SET D,U,E,P FOR NSHK(=3)-2 = 1     ***J=0 ONLY***
      D(2,1)=D(2,2) $ U(2,1)=-U(2,2) $ E(2,1)=E(2,2) $ P(2,1)=P(2,2)
      IF(NUBC.EQ.3HYES) U(2,1)=U(1,1)
      GO TO 12
C
C------SET D,U,E,P FOR NSHK(=2)-1 = 2
 11   D(2,1)=DPC(1,2)=DPC(2,2)=D(1,1) $ U(2,1)=UPC(1,2)=UPC(2,2)=U(1,1)
      E(2,1)=EPC(1,2)=EPC(2,2)=E(1,1) $ P(2,1)=PPC(1,2)=PPC(2,2)=P(1,1)
C
C------PREDICTOR FOR NSHK = 3
 12   NP=1-2*NSHKSGN
      PPC(1,3)=P(1,NSHK)*(2.*GE/(GE+1.)*SSL**2-(GE-1.)/(GE+1.))**NP
      DPC(1,3)=D(1,NSHK)/((GE-1.)/(GE+1.)+2./(GE+1.)/SSL**2)**NP
      UPC(1,3)=DPC(1,3)*(U(1,NSHK)/D(1,NSHK)+FLOAT(NP)*SQRT(GE*P(1,NSHK)
     */D(1,NSHK)*(PPC(1,3)/P(1,NSHK)/DPC(1,3)*D(1,NSHK))**NSHKSGN)
     **(2./(GE+1.)*SSL*(1.-1./SSL**2)))
      EPC(1,3)=DPC(1,3)*(PPC(1,3)/DPC(1,3)/(GE-1.)+UPC(1,3)**2/DPC(1,3)
     ***2/2.)
C------PREDICTOR FOR NSHK+1 = 4
      M=NSHK+1
      DPC(1,4)=DP(M,0) $ UPC(1,4)=UP(M,0)
      EPC(1,4)=EP(M,0) $ PPC(1,4)=PP(4,GE)
C------CORRECTOR FOR NSHK+1 = 4
      D(2,M)=DPC(2,4)=DC(4,M,0) $ U(2,M)=UPC(2,4)=UC(4,M,0)
      E(2,M)=EPC(2,4)=EC(4,M,0) $ P(2,M)=PPC(2,4)=PC(4,GE)
C------CALC SHK POSITION
      SSLSAV=SQRT((PPC(1,3-NSHKSGN)/PPC(1,2+NSHKSGN)+(GE-1.)/(GE+1.))*(
     *GE+1.)/2./GE)*SSL/ABS(SSL)
      RS=(SSLSAV*SQRT(GE*PPC(1,2+NSHKSGN)/DPC(1,2+NSHKSGN))+UPC(1,2+
     *NSHKSGN)/DPC(1,2+NSHKSGN)+SSE)/2.*DT+RS
      IF((NSHK.EQ.2).AND.(NUBC.EQ.3H NO).AND.(NSHKSGN.EQ.0)) RS=RS-2.*
     *UPC(1,2)/DPC(1,2)/2.*DT
      IF(RS.GT.R2(NSHK)) RS=R2(NSHK)
C------SET D,U,P,E, FOR NSHK
      SSLSAV=(SSL+SSLSAV)/2.0
      D(2,NSHK)=D(1,NSHK)/((GE-1.)/(GE+1.)+2./(GE+1.)/SSLSAV**2)**NP
      U(2,NSHK)=D(2,NSHK)*(U(1,NSHK)/D(1,NSHK)+FLOAT(NP)*SQRT(GE*
     *P(1,NSHK)/D(1,NSHK)*(PPC(1,3)/P(1,NSHK)/DPC(1,3)*D(1,NSHK))**
     *NSHKSGN ) *(2./(GE+1.)*SSLSAV*(1.-1./SSLSAV**2)))
      P(2,NSHK)=P(1,NSHK)*(2.*GE/(GE+1.)*SSLSAV**2-(GE-1.)/(GE+1.))**NP
      E(2,NSHK)=D(2,NSHK)*(P(2,NSHK)/D(2,NSHK)/(GE-1.)+
     *U(2,NSHK)**2/D(2,NSHK)**2/2.0)
C------CALC NEW SHK SPD
      SSL=SQRT((P(2,NSHK-NSHKSGN)/P(2,NSHK-1+NSHKSGN)+(GE-1.)/(GE+1.))*
     *(GE+1.)/2./GE)*SSL/ABS(SSL)
      SSE=SSL*SQRT(GE*P(2,NSHK-1+NSHKSGN)/D(2,NSHK-1+NSHKSGN))+U(2,NSHK
     *-1+NSHKSGN)/D(2,NSHK-1+NSHKSGN)
      IF((NSHK.EQ.2).AND.(NUBC.EQ.3H NO).AND.(NSHKSGN.EQ.0)) SSE=SSE-
     *2.*U(2,NSHK-1)/D(2,NSHK-1)
C------ADVANCE SHK INDEX
      NSHK=NSHK-1
      IF(RS.LT.0.0) GO TO 20
```

```
C
      RETURN
C
C
C
C-----SHK REFLECTION OFF OF THE PT OF SYMMETRY
 20   U(2,1)=0.0
      XI=(U(2,1)/D(2,1)-U(2,2)/D(2,2))/SQRT(GF*P(2,2)/D(2,2))*(GF+1.)/2.
      SSL=(XI+SQRT(XI**2+4.))/2.
      P(2,1)=P(2,2)*(2.*GF/(GF+1.)*SSL**2-(GF-1.)/(GF+1.))
      D(2,1)=D(2,2)/((GF-1.)/(GF+1.)+2./(GF+1.)/SSL**2)
      E(2,1)=D(2,1)*(P(2,1)/D(2,1)/(GF-1.)+U(2,1)**2/D(2,1)**2/2.)
      TREFLCT=RS/SSE
      SSE=SSL*SQRT(GF*P(2,2)/D(2,2))+U(2,2)/D(2,2)
      RS=SSE*TREFLCT
      IF(RS.GT.R2(2)) RS=R2(2)
      NUBC=3H NC
C
      RETURN $ END
```

```
      SUBROUTINE CD(RCD,SECD,NCD,NAME)
      COMMON/PARAM/N,J,AJ,G,GF,DELR,NFLM
      COMMON/TIME/T,DT,DTL,TWRITE,DELT,DTDX,AT
      COMMON/PREDCOR/DPC(2,13),UPC(2,13),EPC(2,13),PPC(2,13)
      COMMON/ARRAYS/R(501),U(2,501),P(2,501),D(2,501),E(2,501),R2(501)
C
C
      RCDSAV=RCD $ RCD=RCD+SECD*DT
      NCROSS=0
      IF(RCD.GT.R2(NCD+2)) NCROSS=1 $ IF(RCD.LT.R2(NCD-1)) NCROSS=-1
C
      GE=GF $ IF(NCD.GT.NFLM) GE=G
C-----PREDICTOR AND CORRECTOR FOR L AND R OF CD = 4,5
      M=NCD+2 $ IF(NAME.EQ.5HCDTRD) M=NCD+1
      CALL GLIM(NCD-1,M,4,NCD,SECD)
C
      RCD=RCDSAV+SECD*DT
      IF(((RCD.GT.R2(NCD+2)).AND.(NCROSS.EQ.0)).OR.((RCD.LT.R2(NCD+2)).
     *AND.(NCROSS.EQ.1))) RCD=R2(NCD+2)
      IF(((RCD.LT.R2(NCD-1)).AND.(NCROSS.EQ.0)).OR.((RCD.GT.R2(NCD-1)).
     *AND.(NCROSS.EQ.-1))) RCD=R2(NCD-1)
      R2(NCD)=R2(NCD+1)=RCD
C
      IF(NCROSS.EQ.0) GO TO 10
C-----PREDICTOR AND CORRECTOR FOR NCD+(3*NCROSS+1)/2 = 6+(NCROSS-1)/2*3
      MM=5-(NCROSS+1)/2 $ M=6+(NCROSS-1)/2*3 $ MMM=NCD+(3*NCROSS+1)/2
      U(2,MMM)=UPC(1,M)=UPC(2,M)=UPC(1,MM)
      D(2,MMM)=DPC(1,M)=DPC(2,M)=DPC(1,MM)
      P(2,MMM)=PPC(1,M)=PPC(2,M)=PPC(1,MM)
      E(2,MMM)=EPC(1,M)=EPC(2,M)=EPC(1,MM)
      IF(NCROSS.EQ.-1) GO TO 11
C-----PREDICTOR FOR NCD-2 = 2
 10   IF(NAME.EQ.5HFLRCD) GO TO 9
      M=NCD-2
      DPC(1,2)=DP(M,0) $ UPC(1,2)=UP(M,C)
      EPC(1,2)=EP(M,0) $ PPC(1,2)=PP(2,CE)
C-----PREDICTOR FOR NCD+3 = 7
 9    M=NCD+3
      DPC(1,7)=DP(M,0) $ UPC(1,7)=UP(M,C)
      EPC(1,7)=EP(M,0) $ PPC(1,7)=PP(7,CE)
      IF(NCROSS.EQ.0) GO TO 11
C-----PREDICTOR FOR NCD+4 = 8
      M=NCD+4
      DPC(1,8)=DP(M,0) $ UPC(1,8)=UP(M,C)
      EPC(1,8)=EP(M,0) $ PPC(1,8)=PP(8,CE)
      GO TO 12
C-----PREDICTOR FOR NCD+2 = 6
 11   M=NCD+2
      DPC(1,6)=DP(M,0) $ UPC(1,6)=UP(M,C)
      EPC(1,6)=EP(M,0) $ PPC(1,6)=PP(6,CE)
      IF(NCROSS.EQ.-1) GO TO 13
 12   IF(NAME.EQ.5HFLRCD) GO TO 17
C-----PREDICTOR FOR NCD-1 = 3
      EPS=(RCDSAV-R(NCD-1))/DELR
      C1=2.*(2.-EPS)/(1.+EPS)
      C2=2.*EPS-3. $ C3=(1.-EPS)*(2.*EPS-1.)/(1.+EPS)
```

```
       DPC(1,3)=D(1,NCD-1)-DTDX*(C1*U(1,NCD)+C2*U(1,NCD-1)+C3*U(1,NCD-2))
      *-AT*DTDX*(C1*D(1,NCD)+C2*D(1,NCD-1)+C3*D(1,NCD-2))
       UPC(1,3)=U(1,NCD-1)-DTDX*(C1*PM(NCD)+C2*PM(NCD-1)+C3*PM(NCD-2))
      *-AT*DTDX*(C1*U(1,NCD)+C2*U(1,NCD-1)+C3*U(1,NCD-2))
       EPC(1,3)=E(1,NCD-1)-DTDX*(C1*PE(NCD)+C2*PE(NCD-1)+C3*PE(NCD-2))
      *-AT*DTDX*(C1*E(1,NCD)+C2*E(1,NCD-1)+C3*E(1,NCD-2))
       PPC(1,3)=PP(3,GE)
C-----CORRECTOR FOR NCD-1 = 3
      M=NCD-1
      D(2,M)=DPC(2,3)=DC(3,M,0) $ U(2,M)=UPC(2,3)=UC(3,M,0)
      E(2,M)=EPC(2,3)=EC(3,M,0) $ P(2,M)=PPC(2,3)=PC(3,GE)
C-----CORRECTOR FOR NCD+2+NCROSS = 6+NCROSS
 17   M=NCD+NCROSS+2 $ MMM=6+NCROSS
      EPS=(R2(M)-RCD)/DELR
      C1=2.*(2.-EPS)/(1.+EPS)
      C2=2.*EPS-3. $ C3=(1.-EPS)*(2.*EPS-1.)/(1.+EPS)
      D(2,M)=DPC(2,MMM)=(D(1,M)+DPC(1,MMM)+DTDX*(C1*UPC(1,5)+
     *C2*UPC(1,MMM)+C3*UPC(1,MMM+1))+AT*DTDX*(C1*DPC(1,5)+C2*DPC(1,MMM)
     *+C3*DPC(1,MMM+1)))/2.
      U(2,M)=UPC(2,MMM)=(U(1,M)+UPC(1,MMM)+DTDX*(C1*CM(5)+C2*CM(MMM)+
     *C3*CM(MMM+1))+AT*DTDX*(C1*UPC(1,5)+C2*UPC(1,MMM)+C3*UPC(1,MMM+1)))
     */2.
      E(2,M)=EPC(2,MMM)=(E(1,M)+EPC(1,MMM)+DTDX*(C1*CE(5)+C2*CE(MMM)+
     *C3*CE(MMM+1))+AT*DTDX*(C1*EPC(1,5)+C2*EPC(1,MMM)+C3*EPC(1,MMM+1)))
     */2.
      P(2,M)=PPC(2,MMM)=PC(MMM,GE)
      GO TO 14
C-----CORRECTOR FOR NCD+2 = 6
C     IF J .NE. 0 FIX UP AREA TERM
 13   M=NCD+2
      D(2,M)=DPC(2,6)=DC(6,M,0) $ U(2,M)=UPC(2,6)=UC(6,M,0)
      E(2,M)=EPC(2,6)=EC(6,M,0) $ P(2,M)=PPC(2,6)=PC(6,GE)
C
C-----ADVANCE CD CELL POSITION IF NECCESSARY
 14   IF(NCROSS.EQ.0) GO TO 16
      M=NCD+2+3*(NCROSS-1)/2
      PS=P(2,M) $ DS=D(2,M) $ US=U(2,M) $ ES=E(2,M) $ RS=R2(M)
      IF(NCROSS.EQ.-1) GO TO 15
      P(2,NCD+2)=P(2,NCD+1) $ P(2,NCD+1)=P(2,NCD) $ P(2,NCD)=PS
      U(2,NCD+2)=U(2,NCD+1) $ U(2,NCD+1)=U(2,NCD) $ U(2,NCD)=US
      E(2,NCD+2)=E(2,NCD+1) $ E(2,NCD+1)=E(2,NCD) $ E(2,NCD)=ES
      D(2,NCD+2)=D(2,NCD+1) $ D(2,NCD+1)=D(2,NCD) $ D(2,NCD)=DS
      R2(NCD+2)=R2(NCD+1) $ R2(NCD+1)=R2(NCD) $ R2(NCD)=RS
      GO TO 16
 15   IF(NAME.EQ.5HFLRCD) RETURN
      P(2,NCD-1)=P(2,NCD) $ P(2,NCD)=P(2,NCD+1) $ P(2,NCD+1)=PS
      U(2,NCD-1)=U(2,NCD) $ U(2,NCD)=U(2,NCD+1) $ U(2,NCD+1)=US
      D(2,NCD-1)=D(2,NCD) $ D(2,NCD)=D(2,NCD+1) $ D(2,NCD+1)=DS
      E(2,NCD-1)=E(2,NCD) $ E(2,NCD)=E(2,NCD+1) $ E(2,NCD+1)=ES
      R2(NCD-1)=R2(NCD) $ R2(NCD)=R2(NCD+1) $ R2(NCD+1)=RS
C
 16   NCD=NCD+NCROSS
C
      IF(NAME.NE.5HRARCD) GO TO 30
      M=NCD-1
      IF(NCROSS.NE.1) GO TO 20
```

```
      D(2,M)=D(2,M-1) $U(2,M)=U(2,M-1) $E(2,M)=E(2,M-1) $P(2,M)=P(2,M-1)
 20   D(2,NCD)=D(2,M) $U(2,NCD)=U(2,M) $E(2,NCD)=E(2,M) $P(2,NCD)=P(2,M)
      RETURN
 30   IF(NAME.NE.5HCDRAR) RETURN
      M=NCD+2
      IF(NCROSS.NE.-1) GO TO 31
      D(2,M)=D(2,M+1) $U(2,M)=U(2,M+1) $E(2,M)=E(2,M+1) $P(2,M)=P(2,M+1)
 31   D(2,M-1)=D(2,M) $U(2,M-1)=U(2,M) $E(2,M-1)=E(2,M) $P(2,M-1)=P(2,M)
C
      RETURN $ END
```

```
      SUBROUTINE GLIM(ML,MR,MPC,NCD,SECD)
      COMMON/PARAM/N,J,AJ,G,GF,DELR,NFLM
      COMMON/FIRFETC/INDEX,NCYCLE,NN,NNN,NSTORE,NS,NITRCTN
      COMMON/PREDCOR/DPC(2,13),UPC(2,13),EPC(2,13),PPC(2,13)
      COMMON/ARRAYS/R(501),U(2,501),P(2,501),D(2,501),E(2,501),R2(501)
C
C
C-    ML,MR=NODE NO REPRSNTING LEFT AND RIGHT STTES SATSFD BY THE CD
C-    MPC=PREDCOR NO FR THE STTES TO THE LEFT AND RIGHT OF THE CD
C-    NCD=CELL NO OF THE CD
C-    AML,AMR=MACH NO OF THE LEFT AND RIGHT DISTRBANCE (1.0 * RARE)
C
      PL=P(1,ML) $ PR=P(1,MR) $ RL=D(1,ML) $ RR=D(1,MR)
      UL=U(1,ML)/RL $ UR=U(1,MR)/RR
C
      GE=GF $ IF(NCD.GT.NFLM) GE=G
C
      A=0.0 $ AM=1.0
C-----SETTING UP THE RIEMANN PROBLEM
      EPS=1.E-14 $ ITER=0 $ ITMM=38 $ GG=(GE-1.)/2./GE
      PST=(PL+PR)/2.
      CR=SQRT(PR*RR) $ CL=SQRT(PL*RL)
      FL=FR=100.0
C-----BEGINNING OF THE ITERAION
 1    ITER=ITER+1
      IF(PST.LT.EPS) PST=EPS
      X=PST/PR $ DX=1.-X
      IF(X.LT.0.9999) FRT=CR*GG*SQRT(GE)*DX/(1.-X**GG)
      IF((X.LT.1.).AND.(X.GT.0.9999)) FRT=CR*GG*SQRT(GE)/GG/(1.+(GG-1.)
     */2.*DX*(-1.+(GG-2.)/3.*DX*(1.+(GG-3.)/4.*DX*(-1.+(GG-4.)/5.*DX))))
      IF(X.GE.1.0) FRT=CR*SQRT((GE+1.)/2.*X+(GE-1.)/2.)
      X=PST/PL $ DX=1.-X
      IF(X.LT.0.9999) FLT=CL*GG*SQRT(GE)*DX/(1.-X**GG)
      IF((X.LT.1.).AND.(X.GT.0.9999)) FLT=CL*GG*SQRT(GE)/GG/(1.+(GG-1.)
     */2.*DX*(-1.+(GG-2.)/3.*DX*(1.+(GG-3.)/4.*DX*(-1.+(GG-4.)/5.*DX))))
      IF(X.GE.1.0) FLT=CL*SQRT((GE+1.)/2.*X+(GE-1.)/2.)
      DFR=ABS(FR-FRT) $ FR=FRT
      DFL=ABS(FL-FLT) $ FL=FLT
      PS=PST
      PST=(UL-UR+PR/FR+PL/FL)/(1./FR+1./FL)
      PST=AM*PST+A*PS
C
      IF(ITER.LT.ITMM) GO TO 2
      IF(ABS(PS-PST).LT.EPS) GO TO 3
      A=A+0.5*(1.-A) $ AM=1.-A $ ITER=0
      PRINT 1000,A,AM,FL,FR,PS
 1000 FORMAT(1H ,1X,*CONVERGENCE FACTOR*,5E15.7)
      IF(AM.GT.1./33.) GO TO 2
      PRINT 1001
 1001 FORMAT(1H ,/,* CONVERGENCE FACTOR STOP*)
      DELR=-ABS(DELR) $ RETURN
C
 2    IF((DFL.GT.EPS).OR.(DFR.GT.EPS)) GO TO 1
C-----END OF THE ITERATION
 3    SECD=US=(PL-PR+FR*UR+FL*UL)/(FL+FR)
      M=MPC $ MM=MPC+1
```

```
      P(2,NCD)=P(2,NCD+1)=PPC(1,M)=PPC(2,M)=PPC(1,MM)=PPC(2,MM)=PST
      BETA=(GE-1.)/(GE+1.) $ ALPHA=2./(GE-1.)
      IF(PST/PL.GE.1.0) D(2,NCD)=DPC(1,M)=DPC(2,M)=RL*(PST/PL+BETA)/
     *(1.+PST/PL*BETA)
      IF(PST/PL.LT.1.0) D(2,NCD)=DPC(1,M)=DPC(2,M)=RL*((2.+ALPHA)/ALPHA/
     *GE*(PST/PL-1.)+1.)**(ALPHA/(2.+ALPHA))
      IF(PST/PR.GE.1.0) D(2,NCD+1)=DPC(1,MM)=DPC(2,MM)=RR*(PST/PR+BETA)/
     *(1.+PST/PR*BETA)
      IF(PST/PR.LT.1.0) D(2,NCD+1)=DPC(1,MM)=DPC(2,MM)=RR*((2.+ALPHA)/
     *ALPHA/GE*(PST/PR-1.)+1.)**(ALPHA/(2.+ALPHA))
      U(2,NCD)=UPC(1,M)=UPC(2,M)=US*DPC(1,M)
      U(2,NCD+1)=UPC(1,MM)=UPC(2,MM)=US*DPC(1,MM)
      E(2,NCD)=EPC(1,M)=EPC(2,M)=DPC(2,M)*(PPC(2,M)/DPC(2,M)/(GE-1.)+
     *US**2/2.)
      E(2,NCD+1)=EPC(1,MM)=EPC(2,MM)=DPC(2,MM)*(PPC(2,MM)/DPC(2,MM)/
     *(GE-1.)+US**2/2.)
      AML=-1.0 $ AMR=1.0
      IF(PST/PL.GE.1.0)AML=-SQRT((GE+1.)/2./GE*(PST/PL+(GE-1.)/(GE+1.)))
      IF(PST/PR.GE.1.0) AMR=SQRT((GE+1.)/2./GE*(PST/PR+(GE-1.)/(GE+1.)))
C-----PRINT RIEMANN SOLUTION AT EVERY NCY-TH CYCLE
      NCY=5000
      IF(NCYCLE/NCY*NCY.EQ.NCYCLE)
     *PRINT 1002,ITER,PST,PR,PL,DPC(1,M),DPC(1,MM),UL,UR,US,AML,AMR
 1002 FORMAT(1H ,I5,5E20.12,/,5X,5E20.12)
C
      RETURN $ END
```

```
      SUBROUTINE FLMSHK(NAME,NFIRST,N1)
      COMMON/PARAM/N,J,AJ,G,GF,DELR,NFLM
      COMMON/TIME/T,DT,DTL,TWRITE,DELT,DTDX,AT
      COMMON/PREDCOR/DPC(2,13),UPC(2,13),EPC(2,13),PPC(2,13)
      COMMON/ARRAYS/R(501),U(2,501),P(2,501),D(2,501),E(2,501),R2(501)
      COMMON/DISCS/NTDISC,NDISCNO(51),NTYPE(51),NDISCL(51),SE(51),SL(51)
     *,RD(51),NTDISCT,NTDISCS
C
      PRINT 1001,NAME
 1001 FORMAT(1H ,1X,A4,3I5)
      SSL=SL(N1) $ SSE=SE(N1) $ RS=RD(N1) $ NSHK=NDISCL(N1)
      SFE=SE(NFIRST) $ SFL=SL(NFIRST) $ RF=RD(NFIRST)
C
C
      IF((NAME.EQ.4HCFS2).OR.(NAME.EQ.4HSCFK)) GO TO 9
      NSHKSGN=0 $ IF(SSL.GT.0.0) NSHKSGN=1
      RSSAV=RS $ RS=RS+SSE*DT
      NCROSS=0
      IF(RS.GT.R2(NSHK+1)) NCROSS=1 $ IF(RS.LT.R2(NSHK)) NCROSS=-1
      NSC=100*(NSHK-NFLM)+NCROSS
      PRINT 1001,NAME,NSHKSGN,NCROSS,NSC
      IF(RS.GT.R2(NFLM)) GO TO 1
      PRINT 1000,RSSAV,DT,RS,R(NFLM),R2(NFLM),SSL,SSE,SFL,SFE,NFLM,NSHK
 1000 FORMAT(1H ,1X,9E13.6,2I5,/,* SUBROUTINE FLMSHK *)
      DELR=-ABS(DELR) $ RETURN
C
C-----PREDICTOR FOR NSHK+NCROSS*(NCROSS+1)/2+1 = 10
 1    M=NSHK+NCROSS*(NCROSS+1)/2+1
      DPC(1,10)=DP(M,0) $ UPC(1,10)=UP(M,0)
      EPC(1,10)=EP(M,0) $ PPC(1,10)=PP(10,G)
      IF(NCROSS.EQ.-1) GO TO 10
C-----PREDICTOR FOR NSHK+NCROSS*(NCROSS+1)/2+2 = 11
      M=M+1
      DPC(1,11)=DP(M,0) $ UPC(1,11)=UP(M,0)
      EPC(1,11)=EP(M,0) $ PPC(1,11)=PP(11,G)
      IF(NCROSS.EQ.0) GO TO 2
C-----PREDICTOR FOR NSHK+(NCROSS+1)/2 = 9
 10   M=NSHK+(NCROSS+1)/2
      DPC(1,9)=D(1,M)/((G-1.)/(G+1.)+2./(G+1.)/SSL**2)
      UPC(1,9)=DPC(1,9)*(U(1,M)/D(1,M)+SQRT(G*P(1,M)/D(1,M))*
     *2./(G+1.)*SSL*(1.-1./SSL**2))
      PPC(1,9)=P(1,M)*(2.*G/(G+1.)*SSL**2-(G-1.)/(G+1.))
      EPC(1,9)=DPC(1,9)*(PPC(1,9)/DPC(1,9)/(G-1.)+UPC(1,9)**2/
     *DPC(1,9)**2/2.)
 2    IF((NCROSS.EQ.0).AND.(NSHKSGN.EQ.0)) GO TO 3
C-----CORRECTOR FOR NSHK+NCROSS*(NCROSS+1)/2+1 = 10
      M=NSHK+NCROSS*(NCROSS+1)/2+1
      MM=NCROSS*(NCROSS+1)/2-(NCROSS+1)*(NCROSS-1)
      D(2,M)=DPC(2,10)=DC(10,M,MM) $ U(2,M)=UPC(2,10)=UC(10,M,MM)
      E(2,M)=EPC(2,10)=EC(10,M,MM) $ P(2,M)=UPC(2,10)=PC(10,G)
      GO TO 4
C-----CORRECTOR FOR NSHK+1 = 10
 3    M=NSHK+1
      SSLSAV=(SSL+SQRT((PPC(1,10)/P(1,NSHK)+(G-1.)/(G+1.))*(G+1.)/2./G)
     **SSL/ABS(SSL))/2.
      PPC(1,9)=P(1,NSHK)*(2.*G/(G+1.)*SSLSAV**2-(G-1.)/(G+1.))
```

```
      DPC(1,9)=D(1,NSHK)/((G-1.)/(G+1.)+2./(G+1.)/SSLSAV**2)
      UPC(1,9)=DPC(1,9)*(U(1,NSHK)/D(1,NSHK)+SQRT(G*P(1,NSHK)/D(1,NSHK))
     **2./(G+1.)*SSLSAV*(1.-1./SSLSAV**2))
      EPC(1,9)=DPC(1,9)*(PPC(1,9)/DPC(1,9)/(G-1.)+UPC(1,9)**2
     */DPC(1,9)**2/2.)
      RS=RSSAV+((SSLSAV*2.-SSL)*SQRT(P(1,NSHK)/D(1,NSHK)*G)+U(1,NSHK)/
     *D(1,NSHK)+SSE)/2.*DT
      IF(RS.LT.R2(NFLM)) RS=R2(NFLM)
      C1=2.*(2.-EPS)/(1.+EPS)
      C2=2.*EPS-3. $ C3=(1.-EPS)*(2.*EPS-1.)/(1.+EPS)
      D(2,M)=DPC(2,10)=(D(1,M)+DPC(1,10)+DTDX*(C1*UPC(1,9)+C2*UPC(1,10)
     *+C3*UPC(1,11))-AT*DTDX*(C1*DPC(1,9)+C2*DPC(1,10)+C3*DPC(1,11)))/2.
      U(2,M)=UPC(2,10)=(U(1,M)+UPC(1,10)+DTDX*(C1*CM(9)+C2*CM(10)+
     *C3*CM(11))-AT*DTDX*(C1*UPC(1,9)+C2*UPC(1,10)+C3*UPC(1,11)))/2.
      E(2,M)=EPC(2,10)=(E(1,M)+EPC(1,10)+DTDX*(C1*CE(9)+C2*CE(10)+
     *C3*CE(11))-AT*DTDX*(C1*EPC(1,9)+C2*EPC(1,10)+C3*EPC(1,11)))/2.
      P(2,M)=PPC(2,10)=PC(10,G)
 4    IF(NCROSS.NE.-1) GO TO 5
C-----PREDICTOR FOR NFLM+1 = 8
      M=NFLM+1
      DPC(1,8)=DP(M,0) $ UPC(1,8)=UP(M,0)
      EPC(1,8)=EP(M,0) $ PPC(1,8)=PP(8,G)
C-----PREDICTOR FOR NFLM = 7
      CALL SFVDPD(G,GF,PPC(1,8),DPC(1,8),SFL,VD,PD,DELR)
      DPC(1,7)=DPC(1,8)/VD $ PPC(1,7)=PPC(1,8)*PD
      UPC(1,7)=(SFL*(1.-VD)+UPC(1,8)/DPC(1,8))*DPC(1,7)
      EPC(1,7)=DPC(1,7)*(PPC(1,7)/DPC(1,7)/(GF-1.)+UPC(1,7)**2/DPC(1,7)*
     **2/2.)
      GO TO 6
C-----PREDICTOR FOR NFLM+1,NFLM = 8,7
 5    M=NFLM+1
      DPC(1,8)=D(1,M) $UPC(1,8)=U(1,M) $EPC(1,8)=E(1,M) $PPC(1,8)=P(1,M)
      M=NFLM
      DPC(1,7)=D(1,M) $UPC(1,7)=U(1,M) $EPC(1,7)=E(1,M) $PPC(1,7)=P(1,M)
C-----CORRECTOR FOR NFLM,NFLM+1 = 7,8
 6    M=NFLM
      D(2,M)=DPC(2,7)=DPC(1,7) $ U(2,M)=UPC(2,7)=UPC(1,7)
      E(2,M)=EPC(2,7)=EPC(1,7) $ P(2,M)=PPC(2,7)=PPC(1,7)
      M=NFLM+1
      D(2,M)=DPC(2,8)=DPC(1,8) $ U(2,M)=UPC(2,8)=UPC(1,8)
      E(2,M)=EPC(2,8)=EPC(1,8) $ P(2,M)=PPC(2,8)=PPC(1,8)
      IF(NAME.EQ.4HSCFS) GO TO 7
      RD(NFIRST)=RF=R2(NFLM)=R2(NFLM+1)=RF+(SFL+UPC(1,8)/DPC(1,8))*DT
      IF(NAME.EQ.4HCFS1) GO TO 7
C-----PREDICTOR FOR NFLM-1 = 6
      M=NFLM-1
      DPC(1,6)=DP(M,0) $ UPC(1,6)=UP(M,0)
      EPC(1,6)=EP(M,0) $ PPC(1,6)=PP(6,GF)
C-----CORRECTOR FOR NFLM = 7
      D(2,NFLM)=DPC(2,7)=DC(7,NFLM,0) $ U(2,NFLM)=UPC(2,7)=UC(7,NFLM,0)
      E(2,NFLM)=EPC(2,7)=EC(7,NFLM,0) $ P(2,NFLM)=PPC(2,7)=PC(7,GF)
C-----CORRECTOR FOR NFLM+1 = 8
      VD=DPC(1,8)/DPC(1,7)
      CALL FLM43(7,VD,SFL)
      M=NFLM+1
      D(2,M)=DPC(2,8) $ U(2,M)=UPC(2,8) $E(2,M)=EPC(2,8)$P(2,M)=PPC(2,8)
```

```
      SL(NFIRST)=SFL $ SF(NFIRST)=SFL+U(2,M)/D(2,M)
C-----PREDICTOR FOR SHK POS AND VEL
 7    MH=10-2*NSHKSGN+NCROSS $ ML=8+2*NSHKSGN
      SSLSAV=SQRT((PPC(1,MH)/PPC(1,ML)+(G-1.)/(G+1.))*(G+1.)/2./G
     *)*SSL/ABS(SSL)
      RS=RSSAV+(SSLSAV*SQRT(PPC(1,ML)/DPC(1,ML)*G)+UPC(1,ML)/DPC(1,ML)+
     *SSE)/2.*DT
      IF(((RS.LT.R2(NSHK+1)).AND.(NCROSS.EQ.1)).OR.((RS.GT.R2(NSHK+1)).
     *AND.(NCROSS.EQ.0))) RS=R2(NSHK+1)
      IF(((RS.GT.R2(NSHK)).AND.(NCROSS.EQ.-1)).OR.((RS.LT.R2(NSHK)).AND.
     *(NCROSS.EQ.0))) RS=R2(NSHK)
      RD(N1)=RS
      EPS=(R2(NSHK+1)-RS)/DELR
      IF(NCROSS.EQ.0) GO TO 8
C-----CORRECTOR FOR NSHK+(NCROSS+1)/2 = 9
      M=NSHK+(NCROSS+1)/2
      SSL=(SSL+SSLSAV)/2.
      D(2,M)=DPC(2,9)=D(1,M)/((G-1.)/(G+1.)+2./(G+1.)/SSL**2)
      U(2,M)=UPC(2,9)=D(2,M)*(U(1,M)/D(1,M)+SQRT(G*P(1,M)/D(1,M))*2./
     *(G+1.)*SSL*(1.-1./SSL**2))
      P(2,M)=PPC(2,9)=P(1,M)*(2.*G/(G+1.)*SSL**2-(G-1.)/(G+1.))
      E(2,M)=EPC(2,9)=D(2,M)*(P(2,M)/D(2,M)/(G-1.)+U(2,M)**2/D(2,M)**2
     */2.)
 8    IF((NAME.EQ.4HCFS1).OR.(NAME.EQ.4HSCFS)) RETURN
C-----CORRECTOR FOR SHK VEL
 9    M=NFLM+NSHKSGN+NCROSS*(NCROSS+1)/2+1
      SL(N1)=SSL=SQRT((P(2,NFLM+2-NSHKSGN+NCROSS*(NCROSS+1)/2)/P(2,M)+
     *(G-1.)/(G+1.))*(G+1.)/2./G)*SSL/ABS(SSL)
      SF(N1)=SSF=SSL*SQRT(G*P(2,M)/D(2,M))+U(2,M)/D(2,M)
      NDISCL(N1)=NSHK=NSHK+NCROSS
C
      RETURN $ END
```

```
      SUBROUTINE FLRCD(NFIRST,N1,NXCD)
      COMMON/TIME/T,DT,DTL,TWRITE,DELT,DTDX,AT
      COMMON/PREDCOR/DPC(2,13),UPC(2,13),EPC(2,13),PPC(2,13)
      COMMON/POWER/VCAPF,R4R1,PDPOWER,PPOWER,CND,CCAPF,SFEOLD,SFE,NCJ
      COMMON/ARRAYS/R(501),U(2,501),P(2,501),D(2,501),E(2,501),R2(501)
      COMMON/DISCS/NTDISC,NDISCNO(51),NTYPE(51),NDISCL(51),SE(51),SL(51)
     *,RC(51),NTDISCT,NTDISCS
C
C
      RCDSAV=RC(N1) $ SFESAV=SE(NFIRST) $ RFSAV=RC(NFIRST)
C
      NSC=100*(NDISCL(N1)-NDISCL(NFIRST)-1)+NXCD
      PRINT 1000,NSC
 1000 FORMAT(1H ,I5)
C
C-----PREDICTOR FOR NCD(0,1,2,3+NXCD) = 4,5,6,7+NXCD
      CALL CD(RD(N1),SE(N1),NDISCL(N1),5HFLRCD)
C-----PREDICTOR FOR NFLM,1 = 2,3
      NFLM=NDISCL(NFIRST)
      DPC(1,2)=D(1,NFLM) $ DPC(1,3)=D(1,NFLM+1)
      UPC(1,2)=U(1,NFLM) $ UPC(1,3)=U(1,NFLM+1)
      EPC(1,2)=E(1,NFLM) $ EPC(1,3)=E(1,NFLM+1)
      PPC(1,2)=P(1,NFLM) $ PPC(1,3)=P(1,NFLM+1)
C-----PREDICTOR FOR NFLM-1 = 1 CORRECTOR FOR NFLM+(0,1) = 2,3
      CALL FLM(RD(NFIRST),SE(NFIRST),SL(NFIRST),6H FLRCD)
C
      M=NDISCL(N1)-NXCD*(NXCD+1)/2
      D(2,M)=D(2,M-1) $U(2,M)=U(2,M-1) $E(2,M)=E(2,M-1) $P(2,M)=P(2,M-1)
      IF(NXCD) 1,2,3
 3    D(2,M+1)=D(2,M) $U(2,M+1)=U(2,M) $E(2,M+1)=E(2,M) $P(2,M+1)=P(2,M)
      RETURN
 1    TRFLCT=(RCDSAV-RFSAV)/(SFESAV-SE(N1))
      IF(TRFLCT.GT.DT*(1.-1.E-8)) TRFLCT=DT*(1.-1.E-8)
      RD(NFIRST)=RD(N1)=RCDSAV+TRFLCT*SE(N1)
      IF(NCJ.EQ.3HYES) CALL FCJRCDA(NFIRST,N1,TRFLCT)
      IF(NCJ.EQ.3H NO) CALL FRCDA(NFIRST,N1,TRFLCT)
C
 2    RETURN $ END
```

```
      SUBROUTINE FRCDA(II,NI,IRFLCT)
      COMMON/PARAM/N,J,AJ,G,GF,DELR,NFLM
      COMMON/TIME/T,DT,DTL,TWRITE,DELT,DTCX,AT
      COMMON/FIREFTC/INDEX,NCYCLE,NN,NNN,NSTORE,NS,NITRCTN
      COMMON/POWER/VCAPF,R4R1,PCPOWER,FPOWER,CND,CCAPF,SFEOLD,SFE,NCJ
      COMMON/ARRAYS/R(501),U(2,501),P(2,501),D(2,501),E(2,501),R2(501)
      COMMON/DISCS/NTDISC,NDISCNO(51),NTYPE(51),NDISCL(51),SE(51),SL(51)
     *,RD(51),NTDISCT,NTDISCS
C
C-----DET SS FLW FLD (STTES 4,5,6) DUE TO FLM-RCD INTERACTN BY PLR MTHD
C     SOLN REQUIRES FINAL FLM SPD OF FLM SPD LAW
C     SET STATE 0 AS THE REF STATE FOR POLOR AND OVERALL SOLN
C     VEL REF WRT SQRT(G*P0/D0)
C     P54=TO BE READ AS PRESS IN STATE4 REL TO AND NONDIM BY STATE 4
C     U540=READ AS PRTCLE VEL IN STATE5 REL TO STATE4, NONDIM WRT STATE0
C
C     -STATE-     ----3---SHK--6--CD---5---FLM----4----SHK---1----
C                          *      *         *           *
C                           *     *       *         *
C                            *    *      *       *
C                             *   *     *     *
C                              *  *   *    *
C                               * *  *  *
C                               ***  *
C                                 *
C                               **
C                             * *
C                          *   *
C                        *    *
C                      *     *
C                   *       *
C              -3--FLM--2--RCD--1-
C
C-----SET STATES 1 AND 3
      SG=SQRT(G) $ MNI=NDISCL(NI)+1
      P10=P(2,MNI) $ D10=D(2,MNI) $ U10=U(2,MNI)/D10/SG
      P20=P(2,MNI-1) $ D20=D(2,MNI-1) $ U20=U(2,MNI-1)/D20/SG
      P30=P(2,NFLM) $ D30=D(2,NFLM) $ U30=U(2,NFLM)/D30/SQRT(G)
      A10=SQRT(P10/D10) $ A20=SQRT(P20/D20) $ A30=SQRT(P30/D30*GF/G)
C
C-----SET SUBROUTINE CONSTS, ITER COUNTER AND PRESS GUESS
      BETA=(GF-1.)/(GF+1.) $ B=(GF-1.)/(G-1.) $ BETI=(G-1.)/(G+1.)
      PG0=(1.+BETA)*(VCAPF-BETA)/(1.-BETA)-BETA
C
C-----PERFORM ITERATION OF THE POLOR SOLN
      DELP40=P10/30.
 1    NSIGN=-1 $ EPS=1.E-14 $ NITER=0 $ P40=P10-DELP40
      DELU1=DELU2=DELU3=DELU4=DELU5=1.E+300
      PRINT 1000,NCJ,P10,D10,A10,U10, P20,D20,A20,U20,
     *SL(II),SE(II),P30,D30,A30,U30
 1000 FORMAT(1H1,/,10X,*FLM-RCD INTERACTION *,A3,/,2(/,25X,4F12.8),/,1X,
     *6F12.8,2/)
 10   NITER=NITER+1
      IF(NITER.LT.90) GO TO 11
      IF(NITER.EQ.90) DELUSAV=DELU
      IF(NITER.LT.100) GO TO 11
```

```
      IF((DELUSAV.LT.1.E-12).AND.(DELU.LT.1.E-12)) GO TO 100
 5    PRINT 1001,NCJ,NITER,DELU,DELUSAV,DELU1,DELU2,DELU3,DELU4,DELU5
 1001 FORMAT(1H ,5/,2X,*ITER HAS RCHD MAX IN FLM-FCD-A  NCJ= *,A3,/,
     *1X,I5,7E17.9)
      IF(NCJ.EQ.3HYES) GO TO 2
 6    NCJ=3HYES $ DELP40=P10/7.5 $ GO TO 1
 2    DELR=-ABS(DELR) $ RETURN
C-----GUESS P40
 11   P40=P40+DELP40
      P41=P40/P10
      SL41=SQRT((P41+BETI)/(1.+BETI))
      SE40=SL41*A10+U10
      D41=1./(BETI+(1.-BETI)/SL41**2)
      D40=D41*D10
      A41=SQRT((1.-BETI)/(1.+BETI)*(1.+BETI+BETI*(P41-1.))/(P41+BETI)*G*
     *P41)
      A40=A41*A10
      U410=(1.-BETI)*(1.-1./SL41**2)*SL41*A10
      U40=U410+U10
C
      PG4=1.+(PG0-1.)/A40**2+GF*(1.-1./A40**2)*(G/GF*B-R4R1)
      IF(NCJ.EQ.3HYES) GO TO 3
C-----FLAME BURNING SPD BELOW CJ VALUE
      SL54=CND*(P40/D40)**PDPOWER*P40**PPOWER/SQRT(G)/A40
      XI=(1.-BETA)*SL54**2*G
      P54=(1.-BETA+XI+SQRT((BETA-1.-XI)**2+4.*(BETA-PG4*XI)))/2.
      GO TO 4
C
C-----CJ FLAME BURNING SPD
 3    P54=PG4-SQRT(PG4**2-(1.-BETA)*PG4-BETA)
      SL54=SQRT((2.*P54-(1.-BETA))/(1.-BETA)/G)
C
 4    SE50=SL54*A40+U40
      D54=1./(1.-(1.-BETA)*(P54-PG4)/(P54+BETA))
      P50=P54*P40
      D50=D54*D40
      A54=SQRT((PG4+BETA+BETA*(P54-PG4))/(P54+BETA)*GF/G*P54)
      A50=A54*A40
      U540=-SQRT((1.-BETA)/G*(P54-PG4)*(P54-1.)/(P54+BETA))*A40
      U50=U540+U40
      P60=P50
      P63=P60/P30
      SL63=-SQRT((P63+BETA)/(1.+BETA))
      SE60=SL63*A30+U30
      D63=1./(BETA+(1.-BETA)/SL63**2)
      D60=D63*D30
      A63=SQRT((1.-BETA)/(1.+BETA)*(1.+BETA+BETA*(P63-1.))/(P63+BETA)*GF
     **P63)
      A60=A63*A30
      U630=(1.-BETA)*(1.-1./SL63**2)*SL63*A30
      U60=U630+U30
C
      DELU=ABS(U50-U60)
C
      PRINT 1002,NITER,P41,P40,D41,D40,A41,A40,U410,U40,SL41,SE40,
     *DELU,P54,P50,D54,D50,A54,A50,U540,U50,SL54,SE50,
```

```
     *DELP40,P63,P60,D63,D60,A63,A60,U63C,U60,SL63,SE60
 1002 FORMAT(1H ,1X,I5,5X,10F12.8,/,2(1X,E11.4,10F12.8,/))
C
      IF(DELU.LE.EPS) GO TO 100
      DELU1=DELU2 $ DELU2=DELU3 $ DELU3=DELU4 $ DELU4=DELU5 $ DELU5=DELU
      IF((DELU5.GT.DELU4).AND.(DELU4.GT.DELU3).AND.(DELU3.GT.DELU2).AND.
     *(DELU2.GT.DELU1)) GO TO 5
C
      IF((U60.LT.U50).AND.(NSIGN.LE.0)) DELP40=-ABS(DELP40)
      IF((U60.LT.U50).AND.(NSIGN.EQ.-1)) NSIGN=0
      IF((U60.LT.U50).AND.(NSIGN.EQ.1)) DELP40=-ABS(DELP40)/2.
      IF((U50.LT.U60).AND.(NSIGN.EQ.-1)) DELP40=ABS(DELP40)
      IF((U50.LT.U60).AND.(NSIGN.GE.0)) DELP40=ABS(DELP40)/2.
      IF((U50.LT.U60).AND.(NSIGN.EQ.0)) NSIGN=1
      GO TO 10
C
C
C-----ADD AND TERMINATE DISCS IN THE FLOW FIELD
 100  TARFLCT=DT-TRFLCT $ IF(TARFLCT.LT.1.E-9) TARFLCT=1.E-9
      NT=NTDISCT
      NDISCNC(NT+1)=NTDISC+1 $ NDISCNC(NT+2)=NTDISC+2
      NTYPE(NT+1)=2 $ NTYPE(NT+2)=2 $ NTYPE(NI)=3
      SL(NT+1)=SL63 $ SL(NT+2)=SL41 $ SL(II)=SL54*SG*A40
      SE(NT+1)=SE60*SG $ SE(NT+2)=SE40*SG $ SE(NI)=U50*SG
      SEOLD=SFE=SE(II)=SE50*SG
      RD(II)=R2(NFLM) $ RD(NT+1)=RD(NI)+SE(NT+1)*TARFLCT
      RD(NT+2)=RD(NI)+SE(NT+2)*TARFLCT $ RD(NI)=RD(NI)+SE(NI)*TARFLCT
      IF(RD(NT+1).LT.R2(NFLM-1)) RD(NT+1)=R2(NFLM-1)
      IF(RD(NT+2).GT.R2(NFLM+4)) RD(NT+2)=R2(NFLM+4)
      IF(RD(NI).LT.RD(NT+1)) RD(NI)=(RD(NT+1)+RD(II))/2.
C
      NDISCL(II)=NFLM=NFLM+2
      R2(NFLM-2)=R2(NFLM-1)=RD(NI)
      NDISCL(NI)=NFLM-2 $ NDISCL(NT+1)=NFLM-3 $ NDISCL(NT+2)=NFLM+1
C-----SET THERMO AND GAS PARAMETERS IN CELLS NFLM-2,-1,0,+1
      D(2,NFLM-2)=D60 $ U(2,NFLM-2)=U60*D60*SG $ P(2,NFLM-2)=P60
      D(2,NFLM-1)=D50 $ U(2,NFLM-1)=U50*D50*SG $ P(2,NFLM-1)=P50
      D(2,NFLM)=D50 $ U(2,NFLM)=U50*D50*SG $ P(2,NFLM)=P50
      D(2,NFLM+1)=D40 $ U(2,NFLM+1)=U40*D40*SG $ P(2,NFLM+1)=P40
      E(2,NFLM-2)=(P60/D60/(GF-1.)+U60**2*G/2.)*D60
      E(2,NFLM-1)=E(2,NFLM)=(P50/D50/(GF-1.)+U50**2*G/2.)*D50
      E(2,NFLM+1)=(P40/D40/(G-1.)+U40**2*G/2.)*D40
      NTDISCT=NTDISCT+2 $ NTDISC=NTDISC+2
C
C
      NCODE=10HFRCDA       $ CALL PRNTFF(NCODE)
      NITRCTN=3HYES

      RETURN $ END
```

```
      SUBROUTINE FOURCDA(N3,N1,TRFLCT)
      COMMON/PARAM/N,J,AJ,G,GF,DELR,NFLM
      COMMON/TIME/T,DT,DTL,TWRITE,DELT,DTDX,AT
      COMMON/TRDT/PLTR,DLTR,ULTR,ELTR,OVERCNW,NOVERCN
      COMMON/FIRFETC/INDEX,NCYCLE,NN,NNN,NSTORE,NS,NITRCTN
      COMMON/DISCSKF/RDSAV(51),NLHS(51),NRHS(51),NCROSS(51)
      COMMON/POWER/VCAPF,R4R1,RDPOWER,RPOWER,CND,CCAPF,SFEOLD,SFE,NCJ
      COMMON/ARRAYS/R(501),U(2,501),P(2,501),D(2,501),E(2,501),R2(501)
      COMMON/DISCS/NTDISC,NDISCNC(51),NTYPE(51),NDISCL(51),SE(51),SL(51)
     *,RD(51),NTDISCT,NTDISCS
      COMMON/PPCHR/NPPTH,NUMA,NUMAFST,NUPA,RPPTH(24),RUMA(150),RUPA(150)
     *,NCLPPTH(24),NCLUMA(150),NCLUPA(150),PTHNEXT,RUMANXT,TUPANXT,
     *DELPPTH,DELUMA,DELUPA
C
C-----DET SS FLWFLD-STTES 5,6,7 DUETO FLM(CJ)-RARE-CD INTRCTN BY PLRMTHD
C      SOLN REQUIRES FINAL FLM SPD OF CJ FLM OR CJ DET
C      SET STATE 0 AS THE REF STATE FOR FOLOW AND OVERALL SOLN
C      VEL REF WRT SQRT(G*P0/D0)
C      P65=TO BE READ AS PRESS IN STATE6 REL TO AND NONDIM BY STATE 5
C      U540=READ AS PRTCLE VEL IN STATE5 REL TO STATE4, NONDIM WRT STATE0
C
C      -STATE-      ---3---SHK--7--CD--6--RAREFACTION-5-DETONATION---1---
C
C
C                -3---FLM--2--CD---1---
C
C
C-----SET STATES 1,2 AND 3
      M3=NDISCL(N3) $ M1=NDISCL(N1)+1 $ SG=SQRT(G)
      P10=P(2,M1) $ D10=D(2,M1) $ U10=U(2,M1)/D10/SG
      P20=P(2,M3+1) $ D20=D(2,M3+1) $ U20=U(2,M3+1)/D20/SG
      P30=P(2,M3) $ D30=D(2,M3) $ U30=U(2,M3)/D30/SG
      A10=SQRT(P10/D10) $ A20=SQRT(P20/D20) $ A30=SQRT(P30/D30*GF/G)
C
C-----SET SUBROUTINE CONSTANTS
      BETA=(GF-1.)/(GF+1.) $ B=(GF-1.)/(G-1.) $ ALPHA=2./(GF-1.)
      PG0=(1.+BETA)*(VCAPF-BETA)/(1.-BETA)-BETA
C
C-----DET STATE 5 DUE TO A CJ DETONATION
      PG1=1.+(PG0-1.)/A10**2+GF*(1.-1./A10**2)*(G/GF*P-R4R1)
      SL51=SQRT((2.*PG1-(1.-BETA)+2.*SQRT(PG1**2-(1.-BETA)*PG1-BETA))/
     *(1.-BETA)/G)
      SE50=SL51*A10+U10
      P51=(1.-BETA)/2.*(1.+G*SL51**2) $ P50=P51*P10
      D51=1./(1.-(1.-BETA)*(P51-PG1)/(P51+BETA)) $ D50=D51*D10
      U510=SQRT((1.-BETA)*(P51-PG1)*(P51-1.)/G/(P51+BETA))*A10
      U50=U510+U10
      A51=SQRT((PG1+BETA+BETA*(P51-PG1))/(P51+BETA)*GF/G*P51)
      A50=A51*A10
C
      U50PA50=U50+A50
      PRINT 1000,NCJ, U50PA50,SE50, PG0,PG1,
     *P10,D10,A10,U10,SL(N1),SE(N1),
     *P20,D20,A20,U20, P30,D30,A30,U30,SL(N3),SE(N3),
     *P51,P50,D51,D50,A51,A50,U510,U50,SL51,SE50
 1000 FORMAT(1H1,2/,24X,*FLAME(*,A3,*)-RARE-CD INTERACTION - DETONATION(
```

```
     *DET)*,2/,40X,*U50 + A50 = SE50 *,
     */,37X,2F10.6,4/,1X,2F13.9,/,1X,4(13X,F13.9),2F13.9,/,
     *1X,4(13X,F13.9),/,1X,4(13X,F13.9),2F13.9,/,1X,10F13.9,3/)
C
C
C-----SET UP ITER COUNTER AND PRESSURE GUESS
      NSIGN=-1 $ NITER=0 $ DELP60=(P50-P30)/15. $ EPS=1.E-14
      P60=P30-DELP60
C-----PERFORM ITERATION OF THE POLAR SOLN
 1    NITER=NITER+1
      IF(NITER.LT.51) GO TO 2
      IF(NITER.EQ.51) DELUSAV=DELU
      IF(NITER.LT.75) GO TO 2
      IF((DELUSAV.LT.1.E-12).AND.(DELU.LT.1.E-12)) GO TO 100
      PRINT 1001,NCJ,DELUSAV,DELU
 1001 FORMAT(1H ,5/,10X,*NO. OF ITERATIONS HAS REACHED MAX OF 75 IN FCJR
     *CDA(*,A3,*)*,5X,2E15.7)
      IF(DELU.LT.1.E-10) GO TO 100
      DELR=-ABS(DELR) $ RETURN
C-----GUESS P60
 2    P60=P60+DELP60
      P65=P60/P50
      D65=((2.+ALPHA)/ALPHA/GF*(P65-1.)+1.)**(ALPHA/(2.+ALPHA))
      D60=D65*D50
      A65=D65**(1./ALPHA) $ A60=A65*A50
      U650=ALPHA*(A65-1.)*A50 $ U60=U650+U50
      SLH=SLT=1. $ SEH50=SLH*A50+U50 $ SET60=SLT*A60+U60
C
      P70=P60 $ P73=P70/P30
      SL73=-SQRT((P73+BETA)/(1.+BETA)) $ SE70=SL73*A30+U30
      D73=1./(BETA+(1.-BETA)/SL73**2) $ D70=D73*D30
      A73=SQRT((1.-BETA)/(1.+BETA)*(1.+BETA+BETA*(P73-1.))/(P73+BETA)*GF
     **P73)
      A70=A73*A30
      U730=(1.-BETA)*(1.-1./SL73**2)*SL73*A30 $ U70=U730+U30
      U70=U730+U30
C
      DELU=ABS(U60-U70)
C
      PRINT 1002,NITER,DELU,DELP60,
     *SLH,SEH50,
     *P65,P60,D65,D60,A65,A60,U650,U60,SLT,SET60,
     *P73,P70,D73,D70,A73,A70,U730,U70,SL73,SE70
 1002 FORMAT(1H ,1X,I5,1X,2E15.7,60X,2F12.8,/,2(1X,10F12.8,/))
C
      IF(DELU.LE.EPS) GO TO 100
      IF((U70.LT.U60).AND.(NSIGN.LE.0)) DELP60=-ABS(DELP60)
      IF((U70.LT.U60).AND.(NSIGN.EQ.-1)) NSIGN=0
      IF((U70.LT.U60).AND.(NSIGN.EQ.1)) DELP60=-ABS(DELP60)/2.
      IF((U60.LT.U70).AND.(NSIGN.EQ.-1)) DELP60=ABS(DELP60)
      IF((U60.LT.U70).AND.(NSIGN.GE.0)) DELP60=ABS(DELP60)/2.
      IF((U60.LT.U70).AND.(NSIGN.EQ.0)) NSIGN=1
C
      GO TO 1
C
 100  NT=NTDISCT
```

```
      NDISCNC(NT+1)=NTDISC+1 $ NDISCNC(NT+2)=NTDISC+2
      NDISCNC(NT+3)=NTDISC+3 $ NDISCNC(NT+4)=NTDISC+4
      NTYPE(NT+1)=2 $ NTYPE(NT+2)=6 $ NTYPE(NT+3)=7 $ NTYPE(NT+4)=5
      SL(NT+1)=SL73 $ SL(NT+2)=0.0 $ SL(NT+3)=1. $ SL(NT+4)=SL51
      SE(NT+1)=SE70*SG $ SE(NT+2)=U60*SG $ SE(NT+3)=SE60*SG
      SE(NT+4)=SE50*SG
      RD(NT+1)=RD(N1)+SE(NT+1)*(DT-TRFLCT)
      RD(NT+2)=RD(N1)+SE(NT+2)*(DT-TRFLCT)
      RD(NT+3)=RD(N1)+SE(NT+3)*(DT-TRFLCT)
      RD(NT+4)=RD(N1)+SE(NT+4)*(DT-TRFLCT)
      IF(RD(NT+4).GT.R2(M1+1)) RD(NT+4)=R2(M1+1)
      IF(RD(NT+1).LT.R2(M3-1)) RD(NT+1)=R2(M3-1)
      IF(RD(NT+1).LT.R2(M3)) GO TO 103
      PRINT 1003,NT,R2(M3),RD(NT+1),RD(N1),DT,TRFLCT,SE(NT+1),SL(NT+1)
 1003 FORMAT(1H ,/,1X,*TROUBLE IN FOURCDA*,I5,3F12.8,2E15.7,2F12.8)
      DELR=-ABS(DELR)
 103  IF(RD(NT+2).LT.RD(NT+1)) RD(NT+2)=RD(NT+1)+DELR*1.E-8
      IF(RD(NT+2).GT.RD(NT+4)) RD(NT+2)=RD(NT+4)-DELR*1.E-8
      IF(RD(NT+3).LT.RD(NT+2)) RD(NT+3)=RD(NT+2)+DELR*5.E-9
      IF(RD(NT+3).GT.RD(NT+4)) RD(NT+3)=RD(NT+4)-DELR*5.E-9
C
      N=N-1
      DO 101 I=M1,N
      IP1=I+1
      D(2,I)=D(2,IP1) $U(2,I)=U(2,IP1) $E(2,I)=E(2,IP1) $P(2,I)=P(2,IP1)
      D(1,I)=D(1,IP1) $U(1,I)=U(1,IP1) $E(1,I)=E(1,IP1) $P(1,I)=P(1,IP1)
      R(I)=R(IP1)
 101  R2(I)=R2(IP1)
      DO 102 I=1,NTDISCT
      IF(NDISCL(I).GT.M1) NDISCL(I)=NDISCL(I)-1
 102  CONTINUE
      IF(N1+1.LE.NTDISCS) NRHS(N1+1)=NRHS(N1+1)-1
      IF(N1+1.LE.NTDISCS) NLHS(N1+1)=NLHS(N1+1)-1
      NDISCL(N3)=NDISCL(N1)=-1
      NTDISCT=NTDISCT+4 $ NTDISC=NTDISC+4
      NCJ=3HDET $ SFE=SFECLD=0.0
C-----SET THERMO AND GAS PARAMETERS IN CELLS NDET-2,-1,0
      NDISCL(NT+1)=M3-1 $ NDISCL(NT+4)=M3+2
      NDISCL(NT+3)=NDISCL(NT+4) $ NFLM=NDISCL(NT+4)
      IF(RD(NT+3).LT.R2(M3)) NDISCL(NT+3)=NDISCL(NT+4)-1
      D(2,M3)=D70 $ U(2,M3)=U70=U70*SG*D70 $ P(2,M3)=P70
      E(2,M3)=D70*(P70/D70/(GF-1.)+U70**2/D70**2/2.)
      IF(RD(NT+2).LT.R2(M3)) GO TO 104
      D(2,M3+1)=D70 $ U(2,M3+1)=U70 $ P(2,M3+1)=P70 $ E(2,M3+1)=E(2,M3)
      D(2,M3+2)=D60 $ U(2,M3+2)=U60=U60*SG*D60 $ P(2,M3+2)=P60
      D(2,M3+2)=D60*(P60/D60/(GF-1.)+U60**2/D60**2/2.)
      R2(M3+1)=R2(M3+2)=RD(NT+2)
      NDISCL(NT+2)=NDISCL(NT+1)+2
      GO TO 200
 104  D(2,M3+1)=D60 $ U(2,M3+1)=U60=U60*SG*D60 $ P(2,M3+1)=P60
      E(2,M3+1)=D60*(P60/D60/(GF-1.)+U60**2/D60**2/2.)
      IF(RD(NT+3).LT.R2(M3)) GO TO 105
      D(2,M3+2)=D60 $ U(2,M3+2)=U60 $ P(2,M3+2)=P60 $E(2,M3+2)=E(2,M3+1)
 106  R2(M3+2)=R2([illegible] $ R2(M3)=R2(M3+1)=RD(NT+2)
      NDISCL(NT+2)=NDISCL(NT+1)+1
      GO TO 200
```

```
 105  U(2,M3+2)=U50*SG-(U50*SG-U60/D60)*(RD(NT+4)-R2(M3))/(RD(NT+4)-
     *RD(NT+3))
      AI=SQRT(GF*P50/D50) $ UI=(U(2,M3+2)-U50*SG)/AI $ AI=UI/ALPHA+1.0
      DI=AI**ALPHA $ D(2,M3+2)=DI*D50
      PI=ALPHA*GF/(2.+ALPHA)*(DI**((2.+ALPHA)/ALPHA)-1.)+1.
      P(2,M3+2)=PI*P50 $ U(2,M3+2)=U(2,M3+2)*D(2,M3+2)
      E(2,M3+2)=D(2,M3+2)*(P(2,M3+2)/D(2,M3+2)/(GF-1.)+U(2,M3+2)**2/
     *D(2,M3+2)**2/2.)
      GO TO 106
C
 200  PLTR=P60 $ DLTR=D60 $ ULTR=U60
      ELTR=D60*(P60/D60/(GF-1)+U60**2/D60**2/2.)
C
C-----ACCT FOR PRTCLE PTH AND NEG AND POS CHRCT TRAJ CELL LOCTNS CHANGES
      IF(NPPTH.LE.0) GO TO 110
      DO 111 I=1,NPPTH
      IF(NCLPPTH(I).GT.M1) NCLPPTH(I)=NCLPPTH(I)-1
 111  CONTINUE
 110  IF(NUMA.LE.0) GO TO 112
      DO 113 I=NUMAFST,NUMA
      IF(NCLUMA(I).GT.M1) NCLUMA(I)=NCLUMA(I)-1
 113  CONTINUE
 112  IF(NUPA.LE.0) GO TO 114
      DO 115 I=1,NUPA
      IF(NCLUPA(I).GT.M1) NCLUPA(I)=NCLUPA(I)-1
 115  CONTINUE
C
 114  NCODE=10HFCJFCDA     $ CALL FRNTFF(NCODE)
      NITRCTN=3HYES
C
      RETURN $ END
```

```
      SUBROUTINE SHKFLM(II,NI)
      COMMON/PARAM/N,J,AJ,G,GF,DELR,NFLM
      COMMON/TIME/T,DT,DTL,TWRITE,DELT,DTDX,AT
      COMMON/DISCSKF/RDSAV(51),NLHS(51),NRHS(51),NCROSS(51)
      COMMON/PREDCOR/DPC(2,13),UPC(2,13),EPC(2,13),PPC(2,13)
      COMMON/POWER/VCAPF,R4R1,PDPOWER,PPOWER,CND,CCAPF,SFECLD,SFE,NCJ
      COMMON/ARRAYS/R(501),U(2,501),P(2,501),D(2,501),E(2,501),R2(501)
      COMMON/DISCS/NTDISC,NDISCNO(51),NTYPE(51),NDISCL(51),SF(51),SL(51)
     *,RD(51),NTDISCT,NTDISCS
C
C-----SUB HNDLS INCMG SHK UPTO AND INCLDG INTERACTN WTH FLM
C     PD,VD=PRESS,SPECIFIC VOL RATIO ACROSS THE FLAME
C
C
C-----DET CLOSENESS OF THE INCOMING SHK TO THE FLM
      NSC=100*(NDISCL(NI)-NDISCL(II))+10*NCROSS(II)
      PRINT 1001,II,NI,NRHS(II),NFLM,NCROSS(II),NSC
 1001 FORMAT(1H ,1X,6I5)
      IF(NRHS(II).GT.NFLM+1) GO TO 10
C
C-----PREDCTR FR NFLM-1 = 1, PREDCTR AND CORRECTOR FR NFLM,NFLM+1 = 2,3
      CALL FLM(RD(NI),SE(NI),SL(NI),6HSHKFLM)
C-----CORRECTOR FOR NFLM-1 = 1
      M=NFLM-1
      D(2,M)=DPC(2,1)=DC(1,M,1) $ U(2,M)=UPC(2,1)=UC(1,M,1)
      E(2,M)=EPC(2,1)=EC(1,M,1) $ P(2,M)=PPC(2,1)=PC(1,GF)
C-----PREDICTOR FOR NSHK-1 = 3
      M=NDISCL(II)-1
      DPC(1,3)=DP(M,0) $ UPC(1,3)=UP(M,0)
      EPC(1,3)=EP(M,0) $ PPC(1,3)=PP(3,GF)
C-----PREDICTOR FOR NSHK = 4
      M=NDISCL(II)
      PS=P(1,M+1)*(2.*GF/(GF+1.)*SL(II)**2-(GF-1.)/(GF+1.))
      DS=D(1,M+1)/((GF-1.)/(GF+1.)+2./(GF+1.)/SL(II)**2)
      US=DS*(U(1,M+1)/D(1,M+1)+SQRT(GF*P(1,M+1)/D(1,M+1))*2./(GF+1.)*
     *SL(II)*(1.-1./SL(II)**2))
      ES=DS*(PS/DS/(GF-1.)+US**2/DS**2/2.)
      EPS=(RD(II)-R(M))/DELR
      C1=2.*(2.-EPS)/(1.+EPS)
      C2=2.*EPS-3. $ C3=(1.-EPS)*(2.*EPS-1.)/(1.+EPS)
      DPC(1,4)=D(1,M)-DTDX*(C1*US+C2*U(1,M)+C3*U(1,M-1))
     *-AT*DTDX*(C1*DS+C2*D(1,M)+C3*D(1,M-1))
      UPC(1,4)=U(1,M)-DTDX*(C1*(US*US/DS+PS*RD(II)**J)+C2*PM(M)+
     *C3*PM(M-1))-AT*DTDX*(C1*US+C2*U(1,M)+C3*U(1,M-1))
      EPC(1,4)=E(1,M)-DTDX*(C1*US/DS*(ES+PS*RD(II)**J)+C2*PE(M)+C3*
     *PE(M-1))-AT*DTDX*(C1*ES+C2*E(1,M)+C3*E(1,M-1))
      PPC(1,4)=PP(4,GF)
C-----CORRECTOR FOR NSHK = 4
      D(2,M)=DPC(2,4)=DC(4,M,0) $ U(2,M)=UPC(2,4)=UC(4,M,0)
      E(2,M)=EPC(2,4)=EC(4,M,0) $ P(2,M)=PPC(2,4)=PC(4,GF)
C
C-----DET IF SHK CROSSES A MESH PT OR NOT
      L=0
      IF(NCROSS(II)) 2,3,4
 2    PRINT 1000,II,NTYPE(II),SE(II),SL(II),RD(II)
 1000 FORMAT(1H ,/,1X,*NEG X/O OF DISC NO.*,I3,* OF TYPE*,I2,* WTH SE,SL
```

```
     * DF*,2F14.4,* AT POSITION*,E13.4)
       DELR=-ABS(DELR) $ RETURN
 3     SSL=SQRT((PPC(1,4)/PPC(1,1)+(GF-1.)/(GF+1.))*(GF+1.)/2./GF)
       RD(II)=RD(II)+(SSL*SQRT(PPC(1,1)/DPC(1,1)*GF)+UPC(1,1)/DPC(1,1)+
     *SE(II))/2.0*DT
       IF(((NRHS(II).LE.NFLM+2).AND.(RD(II).GT.R2(NFLM-1+L))).OR.((NRHS(I
     *I).GT.NFLM+2).AND.(RD(II).LT.R2(NFLM-1+L)))) RD(II)=R2(NFLM-1+L)
       SL(II)=SQRT((PPC(2,4)/PPC(2,1)+(GF-1.)/(GF+1.))*(GF+1.)/2./GF)
       SE(II)=SL(II)*SQRT(PPC(2,1)/DPC(2,1)*GF)+UPC(2,1)/DPC(2,1)
       IF(NRHS(II).GT.NFLM+2) GO TO 20
       RETURN
 4     PPC(1,5)=P(1,NFLM-2+L)*(2.*GF/(GF+1.)*SL(II)**2-(GF-1.)/(GF+1.))
       SSL=SQRT((PPC(1,5)/PPC(1,1)+(GF-1.)/(GF+1.))*(GF+1.)/2./GF)
       RD(II)=RD(II)+(SSL*SQRT(PPC(1,1)/DPC(1,1)*GF)+UPC(1,1)/DPC(1,1)
     *+SE(II))/2.0*DT
       IF(RD(II).LT.R2(NFLM-2+L)) RD(II)=R2(NFLM-2+L)
       NDISCL(II)=NDISCL(II)+1
       SSL=(SSL+SL(II))/2.0
       D(2,NFLM-2+L)=D(1,NFLM-2+L)/((GF-1.)/(GF+1.)+2./(GF+1.)/SSL**2)
       U(2,NFLM-2+L)=D(2,NFLM-2+L)*(U(1,NFLM-2+L)/D(1,NFLM-2+L)+SQRT(GF*
     *P(1,NFLM-2+L)/D(1,NFLM-2+L))*(2./(GF+1.)*SSL*(1.-1./SSL**2)))
       P(2,NFLM-2+L)=P(1,NFLM-2+L)*(2.*GF/(GF+1.)*SSL**2-(GF-1.)/(GF+1.))
       E(2,NFLM-2+L)=D(2,NFLM-2+L)*(P(2,NFLM-2+L)/D(2,NFLM-2+L)/(GF-1.)+
     *U(2,NFLM-2+L)**2/D(2,NFLM-2+L)**2/2.0)
       SL(II)=SQRT((P(2,NFLM-2+L)/PPC(2,1)+(GF-1.)/(GF+1.))*(GF+1.)/2./GF
     *)
       SE(II)=SL(II)*SQRT(PPC(2,1)/DPC(2,1)*GF)+UPC(2,1)/DPC(2,1)
       RETURN
C
 10    IF(NRHS(II).GT.NFLM+2) NCROSS(II)=0
C-----PREDICTOR AND CORRECTOR FOR NFLM+1 = 2
       M=NFLM+1
       D(2,M)=DPC(1,2)=DP(M,0) $ U(2,M)=UPC(1,2)=UP(M,0)
       E(2,M)=EPC(1,2)=EP(M,0) $ P(2,M)=PPC(1,2)=PP(2,G)
C-----PREDICTOR AND CORRECTOR FOR NFLM = 1
       SFL=SL(NI)
       CALL SFVDPD(G,GF,P(2,M),D(2,M),SFL,VD,PD,DELR)
       D(2,NFLM)=DPC(1,1)=DPC(2,1)=D(2,M)/VD
       P(2,NFLM)=PPC(1,1)=PPC(2,1)=P(2,M)*PD
       U(2,NFLM)=UPC(1,1)=UPC(2,1)=(SFL*(1.-VD)+U(2,M)/D(2,M))*D(2,NFLM)
       E(2,NFLM)=EPC(1,1)=EPC(2,1)=D(2,NFLM)*(P(2,NFLM)/D(2,NFLM)/(GF-1.)
     *+U(2,NFLM)**2/D(2,NFLM)**2/2.)
C-----CALC NEW FLM SPD AND POSITION
       RD(NI)=RD(NI)+SE(NI)*DT $ SL(NI)=SFL $SFE=SE(NI)=SFL+U(2,M)/D(2,M)
C-----PREDICTOR FOR NSHK-1 = 3
       M=NDISCL(II)-1
       DPC(1,3)=DP(M,0) $ UPC(1,3)=UP(M,0)
       EPC(1,3)=EP(M,0) $ PPC(1,3)=PP(3,GF)
C-----PREDICTOR FOR NSHK = 4
       M=NDISCL(II)
       PS=P(1,M+1)*(2.*GF/(GF+1.)*SL(II)**2-(GF-1.)/(GF+1.))
       DS=D(1,M+1)/((GF-1.)/(GF+1.)+2./(GF+1.)/SL(II)**2)
       US=DS*(U(1,M+1)/D(1,M+1)+SQRT(GF*P(1,M+1)/D(1,M+1))*2./(GF+1.)*
     *SL(II)*(1.-1./SL(II)**2))
       ES=DS*(PS/DS/(GF-1.)+US**2/DS**2/2.)
       EFS=(RD(II)-R(M))/DELR
```

```
      C1=2.*(2.-EPS)/(1.+EPS)
      C2=2.*EPS-3. $ C3=(1.-EPS)*(2.*EPS-1.)/(1.+EPS)
      DPC(1,4)=D(1,M)-DTDX*(C1*US+C2*U(1,M)+C3*U(1,M-1))
     *-AT*DTDX*(C1*DS+C2*D(1,M)+C3*D(1,M-1))
      UPC(1,4)=U(1,M)-DTDX*(C1*(US*US/DS+PS*RD(II)**J)+C2*PM(M)+
     *C3*PM(M-1))-AT*DTDX*(C1*US+C2*U(1,M)+C3*U(1,M-1))
      EPC(1,4)=E(1,M)-DTDX*(C1*US/DS*(ES+PS*RD(II)**J)+C2*PE(M +C3*
     *PE(M-1))-AT*DTDX*(C1*ES+C2*E(1,M)+C3*E(1,M-1))
      PPC(1,4)=PP(4,GF)
C-----CORRECTOR FOR NSHK = 4
      D(2,M)=DPC(2,4)=DC(4,M,0) $ U(2,M)=UPC(2,4)=UC(4,M,0)
      E(2,M)=EPC(2,4)=EC(4,M,0) $ P(2,M)=PPC(2,4)=PC(4,GF)
C
      L=1
      IF(NCROSS(II)) 2,3,4
C
C
C-----SHK AND FLM INTERACT
 20   CALL SHKFLMA(II,NI)
C
      RETURN $ END
```

```
      SUBROUTINE SHKFLMA(II,NI)
      COMMON/PARAM/N,J,AJ,G,GF,DELR,NFLM
      COMMON/TIME/T,DT,DTL,TWRITE,DELT,DTDX,AT
      COMMON/FIRFETC/INDEX,NCYCLE,NN,NNN,NSTOPE,NS,NITRCTN
      COMMON/DISCSKF/RDSAV(51),NLHS(51),NRHS(51),NCROSS(51)
      COMMON/POWER/VCAPF,R4R1,PDPOWER,PPOWER,CND,CCAPF,SFEOLD,SFE,NCJ
      COMMON/ARRAYS/R(501),U(2,501),P(2,501),D(2,501),E(2,501),R2(501)
      COMMON/DISCS/NTDISC,NDISCNC(51),NTYPE(51),NDISCL(51),SE(51),SL(51)
     *,RD(51),NTDISCT,NTDISCS
      COMMON/PPCHR/NPPTH,NUMA,NUMAFST,NUPA,RPPTH(24),RUMA(150),RUPA(150)
     *,NCLPPTH(24),NCLUMA(150),NCLUPA(150),FTHNEXT,RUMANXT,TUPANXT,
     *DELPPTH,DELUMA,DELUPA
C
C-----DET SS FLW FLD (STTES 4,5,6) DUE TO SHK-FLM INTERACTN BY PLR MTHD
C      SOLN REQUIRES FINAL FLM SPD OF FLM SPD LAW
C      SET STATE 0 AS THE REF STATE FOR POLAR AND OVERALL SOLN
C      VEL REF WRT SQRT(G*P0/D0)
C      P54=TO BE READ AS PRESS IN STATE4 REL TO AND NONDIM BY STATE 4
C      U540=READ AS PRTCLE VEL IN STATE5 REL TO STATE4, NONDIM WRT STATE0
C
C     -STATE-     ----3---SHK--6--CD---5---FLM----4----SHK---1----
C                          *      *         *           *
C                           *     *       *         *
C                            *    *      *       *
C                             *   *     *     *
C                              *  *   *    *
C                               * *  *  *
C                                *** *
C                                 *
C                               **
C                             * *
C                          *   *
C                        *    *
C                      *     *
C                   *       *
C              -3--SHK--2--FLM--1-
C
C-----SET STATES 1 AND 3
      P10=P(2,NFLM+1) $ D10=D(2,NFLM+1) $ U10=U(2,NFLM+1)/SQRT(G)/D10
      A10=SQRT(P10/D10)
      SL210=SL(NI)/SQRT(G) $ SL32=SL(II) $ NDISCLS=NDISCL(II)
      P30=P(2,NDISCLS) $ D30=D(2,NDISCLS) $ U30=U(2,NDISCLS)/SQRT(G)/D30
      A30=SQRT(P30/D30*GF/G)
C
C-----SET SUBROUTINE CONSTS, ITER COUNTER AND PRESS GUESS
      BETA=(GF-1.)/(GF+1.) $ B=(GF-1.)/(G-1.) $ BETI=(G-1.)/(G+1.)
      PG0=(1.+BETA)*(VCAPF-BETA)/(1.-BETA)-BETA
C
C-----PERFORM ITERATION OF THE POLAR SOLN
      DELP40=P30/30. $ IF(NCJ.EQ.3HYES) DELP40=P30/7.5
 1    NSIGN=-1 $ EPS=1.E-14 $ NITER=0 $ P40=P30-DELP40
      DELU1=DELU2=DELU3=DELU4=DELU5=1.E+30
      PRINT 1000,NCJ,P10,D10,U10,A10,SL210,P30,D30,U30,A30,SL32
 1000 FORMAT(1H1,X,*SHK-FLM INTRACTN *,A3,1X,10F11.7,2/)
 10   NITER=NITER+1
      IF(NITER.LT.90) GO TO 11
```

```
      IF(NITER.EQ.90) DELUSAV=DELU
      IF(NITER.LT.100) GO TO 11
      IF((DELUSAV.LT.1.E-12).AND.(DELU.LT.1.E-12)) GO TO 100
 5    PRINT 1001,NCJ, NITER,DELU,DELUSAV,DELU1,DELU2,DELU3,DELU4,DELU5
 1001 FORMAT(1H ,5/,2X,*ITER HAS RCHD MAX IN SHK-FLM-A  NCJ= *,A3,/,1X,
     *I5,7E17.9)
      IF(NCJ.EQ.3HYES) GO TO 2
 6    NCJ=3HYES $ DELP40=P30/7.5 $ GO TO 1
 2    DELR=-ABS(DELR) $ RETURN
C-----GUESS P40
 11   P40=P40+DELP40
      P41=P40/P10
      SL41=SQRT((P41+BETI)/(1.+BETI))
      SE40=SL41*A10+U10
      D41=1./(BETI+(1.-BETI)/SL41**2)
      D40=D41*D10
      A41=SQRT((1.-BETI)/(1.+BETI)*(1.+BETI+BETI*(P41-1.))/(P41+BETI)*G*
     *P41)
      A40=A41*A10
      U410=(1.-BETI)*(1.-1./SL41**2)*SL41*A10
      U40=U410+U10
C
      PG4=1.+(PG0-1.)/A40**2+GF*(1.-1./A40**2)*(G/GF*B-R4R1)
      IF(NCJ.EQ.3HYES) GO TO 3
C-----FLAME BURNING SPD BELOW CJ VALUE
      SL54=CND*(P40/D40)**PDPOWER*P40**PPOWER/SQRT(G)/A40
      XI=(1.-BETA)*SL54**2*G
      P54=(1.-BETA+XI+SQRT((BETA-1.-XI)**2+4.*(BETA-PG4*XI)))/2.
      GO TO 4
C
C-----CJ FLAME BURNING SPD
 3    P54=PG4-SQRT(PG4**2-(1.-BETA)*PG4-BETA)
      SL54=SQRT((2.*P54-(1.-BETA))/(1.-BETA)/G)
C
 4    SE50=SL54*A40+U40
      D54=1./(1.-(1.-BETA)*(P54-PG4)/(P54+BETA))
      P50=P54*P40
      D50=D54*D40
      A54=SQRT((PG4+BETA+BETA*(P54-PG4))/(P54+BETA)*GF/G*P54)
      A50=A54*A40
      U540=-SQRT((1.-BETA)/G*(P54-PG4)*(P54-1.)/(P54+BETA))*A40
      U50=U540+U40
      P60=P50
      P63=P60/P30
      SL63=-SQRT((P63+BETA)/(1.+BETA))
      SE60=SL63*A30+U30
      D63=1./(BETA+(1.-BETA)/SL63**2)
      D60=D63*D30
      A63=SQRT((1.-BETA)/(1.+BETA)*(1.+BETA+BETA*(P63-1.))/(P63+BETA)*GF
     **P63)
      A60=A63*A30
      U630=(1.-BETA)*(1.-1./SL63**2)*SL63*A30
      U60=U630+U30
C
      DELU=ABS(U50-U60)
C
```

```
      PRINT 1002,NITER,P41,P40,D41,D40,A41,A40,U410,U40,SL41,SE40,
     *DELU,P54,P50,D54,D50,A54,A50,U540,U50,SL54,SE50,
     *DELP40,P63,P60,D63,D60,A63,A60,U630,U60,SL63,SE60
 1002 FORMAT(1H ,1X,I5,5X,10F12.8,/,2(1X,E11.4,10F12.8,/))
C
      IF(DELU.LE.EPS) GO TO 100
      DELU1=DELU2 $ DELU2=DELU3 $ DELU3=DELU4 $ DELU4=DELU5 $ DELU5=DELU
      IF((DELU5.GT.DELU4).AND.(DELU4.GT.DELU3).AND.(DELU3.GT.DELU2).AND.
     *(DELU2.GT.DELU1)) GO TO 5
C
      IF((U60.LT.U50).AND.(NSIGN.LE.0)) DELP40=-ABS(DELP40)
      IF((U60.LT.U50).AND.(NSIGN.EQ.-1)) NSIGN=0
      IF((U60.LT.U50).AND.(NSIGN.EQ.1)) DELP40=-ABS(DELP40)/2.
      IF((U50.LT.U60).AND.(NSIGN.EQ.-1)) DELP40=ABS(DELP40)
      IF((U50.LT.U60).AND.(NSIGN.GE.0)) DELP40=ABS(DELP40)/2.
      IF((U50.LT.U60).AND.(NSIGN.EQ.0)) NSIGN=1
      GO TO 10
C
C
C-----ADD AND TERMINATE DISCS IN THE FLOW FIELD
 100  TARFLCT=(RD(II)-RD(NI))/(SE(II)-SE(NI))
      NT=NTDISCT
      NDISCNO(NT+1)=NTDISC+1 $ NDISCNO(NT+2)=NTDISC+2
      NDISCNO(NT+3)=NTDISC+3
      NTYPE(NT+1)=2 $ NTYPE(NT+2)=3 $ NTYPE(NT+3)=2
      SL(NT+1)=SL63 $ SL(NT+2)=0. $ SL(NI)=SL54*SQRT(G)*A40
      SL(NT+3)=SL41
      SE(NT+1)=SE60*SQRT(G) $ SE(NT+2)=U50*SQRT(G)
      SE(NT+3)=SE40*SQRT(G)
      SEEOLD=SEE=SE(NI)=SE50*SQRT(G)
      RD(NI)=RD(II)-SE(II)*TARFLCT $ IF(TARFLCT.LT.1.E-10)TARFLCT=1.E-10
      RD(NT+1)=RD(II)+SE(NT+1)*TARFLCT
      RD(NT+2)=RD(II)+SE(NT+2)*TARFLCT $RD(NT+3)=RD(II)+SE(NT+3)*TARFLCT
      IF(RD(NT+1).LT.R2(NFLM-1)) RD(NT+1)=R2(NFLM-1)+1.E-10
      IF(RD(NT+2).LT.RD(NT+1)) RD(NT+2)=(RD(NT+1)+RD(II))/2.
      IF(RD(NT+3).GT.R2(NFLM+2)) RD(NT+3)=R2(NFLM+2)-1.E-10
C
      DO 101 I=NFLM,N
      M=N+2+NFLM-I
      D(2,M)=D(2,M-2) $ U(2,M)=U(2,M-2) $E(2,M)=E(2,M-2)$P(2,M)=P(2,M-2)
      D(1,M)=D(1,M-2) $ U(1,M)=U(1,M-2) $E(1,M)=E(1,M-2)$P(1,M)=P(1,M-2)
      R(M)=R(M-2)
 101  R2(M)=R2(M-2)
      N=N+2
      R2(NFLM)=R2(NFLM+1)=RD(NT+2)
      DO 102 I=1,NTDISCT
      IF(NDISCL(I).GE.NFLM) NDISCL(I)=NDISCL(I)+2
 102  CONTINUE
      IF(NI+1.LE.NTDISCS) NRHS(NI+1)=NRHS(NI+1)+2
      IF(NI+1.LE.NTDISCS) NLHS(NI+1)=NLHS(NI+1)+2
      NFLM=NFLM+2
      NDISCL(II)=-1 $ NDISCL(NT+1)=NFLM-3 $ NDISCL(NT+2)=NFLM-2
      NDISCL(NT+3)=NFLM+1
C-----SET THERMO AND GAS PARAMETERS IN CELLS NFLM-2,-1,0,+1
      D(2,NFLM-2)=D60 $ U(2,NFLM-2)=U60*D60*SQRT(G) $ P(2,NFLM-2)=P60
      D(2,NFLM-1)=D50 $ U(2,NFLM-1)=U50*D50*SQRT(G) $ P(2,NFLM-1)=P50
```

```
      D(2,NFLM)=D50 $ U(2,NFLM)=U50*D50*SQRT(G) $ P(2,NFLM)=P50
      D(2,NFLM+1)=D40 $ U(2,NFLM+1)=U40*D40*SQRT(G) $ P(2,NFLM+1)=P40
      E(2,NFLM-2)=(P60/D60/(GF-1.)+U60**2*G/2.)*D60
      E(2,NFLM-1)=E(2,NFLM)=(P50/D50/(GF-1.)+U50**2*G/2.)*D50
      E(2,NFLM+1)=(P40/D40/(G-1.)+U40**2*G/2.)*D40
      NTDISCT=NTDISCT+3 $ NTDISC=NTDISC+3
C
C-----ACCT FOR PRTCLE PTH AND NEG AND POS CHRCT TRAJ CELL LOCTNS CHANGES
      IF(NPPTH.LE.0) GO TO 110
      DO 111 I=1,NPPTH
      IF(NCLPPTH(I).GE.NFLM-2) NCLPPTH(I)=NCLPPTH(I)+2
  111 CONTINUE
  110 IF(NUMA.LE.0) GO TO 112
      DO 113 I=NUMAFST,NUMA
      IF(NCLUMA(I).GE.NFLM-2) NCLUMA(I)=NCLUMA(I)+2
  113 CONTINUE
  112 IF(NUPA.LE.0) GO TO 114
      DO 115 I=1,NUPA
      IF(NCLUPA(I).GE.NFLM-2) NCLUPA(I)=NCLUPA(I)+2
  115 CONTINUE
C
  114 NCODE=10HSHKFLMA     $ CALL PRNTFF(NCODE)
      NITRCTN=3HYES
C
      RETURN $ END
```

```
      SUBROUTINE SKSKPP(II,NI,NJ,NAME)
      COMMON/PARAM/N,J,AJ,G,GF,DELR,NFLM
      COMMON/TIME/T,DT,DTL,TWRITE,DELT,DTDX,AT
      COMMON/DISCSKF/RDSAV(51),NLHS(51),NRHS(51),NCROSS(51)
      COMMON/PREDCOR/DPC(2,13),UPC(2,13),EPC(2,13),PPC(2,13)
      COMMON/ARRAYS/R(501),U(2,501),P(2,501),D(2,501),E(2,501),R2(501)
      COMMON/DISCS/NTDISC,NDISCNO(51),NTYPE(51),NDISCL(51),SE(51),SL(51)
     *,RD(51),NTDISCT,NTDISCS
C
C-----SL(II) AND SL(NI) MUST BE GT 0.0
C
C
      D2M(SSL,GE,M)=D(1,M)/((GE-1.)/(GE+1.)+2./(GE+1.)/SSL**2)
      U2M(SSL,GE,M)=D(2,M)*(U(1,M)/D(1,M)+SQRT(GE*P(1,M)/D(1,M))*(2./
     *(GE+1.)*SSL*(1.-1./SSL**2)))
      U2MM(SSL,GE,M)=D(2,M)*(U(1,M)/D(1,M)-SQRT(GE*P(1,M+1)/D(1,M+1))*(
     *2./(GE+1.)*SSL*(1.-1./SSL**2)))
      P2M(SSL,GE,M)=P(1,M)*(2.*GE/(GE+1.)*SSL**2-(GE-1.)/(GE+1.))
      E2M(GE,M)=D(2,M)*(P(2,M)/D(2,M)/(GE-1.)+U(2,M)**2/D(2,M)**2/2.)
C
      SSLS(RAHD,RBHD,GE)=SQRT((RBHD/RAHD+(GE-1.)/(GE+1.))*(GE+1.)/2./GE)
      SSFS(SSL,GE,K,L)=SSL*SQRT(GE*PPC(K,L)/DPC(K,L))+UPC(K,L)/DPC(K,L)
C
C
      GE=G $ IF(NDISCL(NI).LE.NFLM) GE=GF
      NCRSII=NCROSS(II) $ MII=NDISCL(II)+1 $ NPII=(NCRSII-1)/2
      NCRSNI=NCROSS(NI) $ MNI=NDISCL(NI)+1 $ NPNI=(NCRSNI-1)/2
      NSC=100*(NDISCL(NI)-NDISCL(II))+10*NCRSII+NCRSNI
      PRINT 2000,NAME,NCRSII,NCRSNI,MII,MNI,II,NI,NSC
 2000 FORMAT(1H ,1X,A7,1X,7I5)
      NSC=0 $ SLNISAV=SL(NI)
      IF((MII.EQ.MNI).AND.(NCRSII.EQ.1).AND.(NCRSNI.EQ.0)) GO TO 72
      IF((MII.EQ.MNI).AND.(NCRSII.EQ.0).AND.(NCRSNI.EQ.-1)) GO TO 73
      IF((MII.EQ.MNI).AND.(NCRSNI.EQ.-1).AND.(NCRSII.EQ.1)) GO TO 74
      GO TO 71
 72   NCRSII=NCROSS(II)=0 $ NRHS(II)=NRHS(II)-1 $ GO TO 71
 73   NCRSNI=NCROSS(NI)=NPNI=0 $ NRHS(NI)=NRHS(NI)+1 $ GO TO 71
 74   NCRSNI=NCROSS(NI)=NPNI=0 $ NRHS(NI)=NRHS(NI)+1
      NCRSII=NCROSS(II)=0 $ NRHS(II)=NRHS(II)-1
 71   IF((NRHS(II).NE.MNI).OR.(NCRSII.EQ.-1).OR.(NCRSNI.EQ.-1)) GO TO 1
C
C-----SHKNI(NCRS=0) PRDTR FR NSHK-1,0,1,2=1,2,3,4 - CRRCTR FR NSHK,1=2,3
C-----SHKNI(NCRS=1) PRDTR FR NSHK-1,0,1,2,3 AND CRRCTR FR NSHK,1,2=2,3,4
      IF(NAME.EQ.7HSKSKSK0) GO TO 2
      IF((NAME.EQ.7HSPECIAL).AND.(NDISCL(NI)-NDISCL(II).EQ.3)) GO TO 2
      IF((NAME.EQ.7HSPECIAL).AND.(NDISCL(NI)-NDISCL(II).EQ.2).AND.
     *(NCRSII.EQ.0)) GO TO 2
      CALL SHK(RD(NI),SE(NI),SL(NI),NDISCL(NI))
      GO TO 2
C-----SHKNI PREDICTOR FOR NSHK+2+NCRSNI = 4
 1    IF((NAME.EQ.7HSKSKSKM).OR.(NAME.EQ.7HSKSKSKC).OR.(NAME.EQ.7HSPECIA
     *L)) GO TO 2
      M=NDISCL(NI)+2+NCRSNI
      DPC(1,4)=DP(M,0) $ UPC(1,4)=UP(M,0)
      EPC(1,4)=EP(M,0) $ PPC(1,4)=PP(4,GE)
      IF(NCRSNI.EQ.-1) GO TO 2
```

```
C-----SHKNI PREDICTOR FOR NSHK+1+NCRSNI = 3
      M=NDISCL(NI)+1+NCRSNI
      DPC(1,3)=DP(M,0) $ UPC(1,3)=UP(M,0)
      EPC(1,3)=EP(M,0) $ PPC(1,3)=PP(3,GE)
C-----SHKNI CORRECTOR FOR NSHK+1+NCRSNI = 3
      D(2,M)=DPC(2,3)=DC(3,M,1) $ U(2,M)=UPC(2,3)=UC(3,M,1)
      E(2,M)=EPC(2,3)=EC(3,M,1) $ P(2,M)=PPC(2,3)=PC(3,GE)
C-----SHKII PREDICTOR FOR NSHK-1 = 6
 2    IF(NAME.EQ.7H SKSKSK) GO TO 7
      M=NDISCL(II)-1
      DPC(1,6)=DP(M,0) $ UPC(1,6)=UP(M,0)
      EPC(1,6)=EP(M,0) $ PPC(1,6)=PP(6,GE)
      IF(NCRSII.NE.-1) GO TO 8
C-----SHKII PREDICTOR FOR NSHK-2 = 5
      M=MII-3
      DPC(1,5)=DP(M,0) $ UPC(1,5)=UP(M,0)
      EPC(1,5)=EP(M,0) $ PPC(1,5)=PP(5,GE)
C-----SHKII CORRECTOR FOR NSHK-2 = 6
      M=MII-2
      D(2,M)=DPC(2,6)=DC(6,M,0) $ U(2,M)=UPC(2,6)=UC(6,M,0)
      E(2,M)=EPC(2,6)=EC(6,M,0) $ P(2,M)=PPC(2,6)=PC(6,GE)
      GO TO 7
C-----SHKII PREDICTOR FOR NSHK = 7
 8    M=NDISCL(II)
      IF(NDISCL(II).EQ.NDISCL(NI)) GO TO 5
      PS=P2M(SL(II),GE,MII) $ DS=D2M(SL(II),GE,MII)
      US=U2M(SL(II),GE,MII)/D(2,MII)*DS
      GO TO 6
 5    IF((NAME.EQ.7HSKSKSKO).AND.(NDISCL(NJ)+1.EQ.MNI)) GO TO 96
      IF((NAME.EQ.7HSPECIAL).AND.(NDISCL(NJ)+1.EQ.MNI)) GO TO 102
      PS=P2M(SL(NI),GE,MNI) $ DS=D2M(SL(NI),GE,MNI)
      US=U2M(SL(NI),GE,MNI)/D(2,MNI)*DS
      GO TO 97
 102  PS=PPC(1,1) $ DS=DPC(1,1) $ US=UPC(1,1)
      GO TO 97
 96   PS=P2M(SL(NJ),GE,MNI) $ DS=D2M(SL(NJ),GE,MNI)
      US=U2M(SL(NJ),GE,MNI)/D(2,MNI)*DS
      US=US/DS+SQRT(GE*PS/DS)*2./(GE+1.)*SL(NI)*(1.-1./SL(NI)**2)
      DS1=DS=D2M(SL(NI),GE,MNI)/D(1,MNI)*DS
      US1=US=US*DS $ PS1=PS=P2M(SL(NI),GE,MNI)/P(1,MNI)*PS
 97   US=US/DS+SQRT(GE*PS/DS)*2./(GE+1.)*SL(II)*(1.-1./SL(II)**2)
      DS=D2M(SL(II),GE,MNI)/D(1,MNI)*DS
      US=US*DS
      PS=P2M(SL(II),GE,MNI)/P(1,MNI)*PS
 6    ES=(PS/DS/(GE-1.)+US**2/DS**2/2.)*DS
      EPS=(RD(II)-R(M))/DELR
      C1=2.*(2.-EPS)/(1.+EPS)
      C2=2.*EPS-3. $ C3=(1.-EPS)*(2.*EPS-1.)/(1.+EPS)
      DPC(1,7)=D(1,M)-DTDX*(C1*US+C2*U(1,M)+C3*U(1,M-1))
     *-AT*DTDX*(C1*DS+C2*D(1,M)+C3*D(1,M-1))
      UPC(1,7)=U(1,M)-DTDX*(C1*(US*US/DS+PS*RD(II)**J)+C2*PM(M)+
     *C3*PM(M-1))-AT*DTDX*(C1*US+C2*U(1,M)+C3*U(1,M-1))
      EPC(1,7)=E(1,M)-DTDX*(C1*US/DS*(ES+PS*RD(II)**J)+C2*PE(M)+C3*
     *PE(M-1))-AT*DTDX*(C1*ES+C2*E(1,M)+C3*E(1,M-1))
      PPC(1,7)=PP(7,GE)
C-----SHKII CORRECTOR FOR NSHK = 7
```

```
      D(2,M)=DPC(2,7)=DC(7,M,0) $ U(2,M)=UPC(2,7)=UC(7,M,0)
      E(2,M)=EPC(2,7)=EC(7,M,0) $ P(2,M)=PPC(2,7)=PC(7,GE)
 7    IF(NCROSS(II)**2.NE.1) GO TO 3
      IF((NAME.EQ.7H SKSKSK).AND.(NCRSII.EQ.-1)) GO TO 3
C-----SHKII PREDICTOR FOR NSHK+1+NPII = 8
      P(2,MII+NPII)=PPC(1,8)=(F2M(SL(II),GE,MII)*(P(1,MII)*P(1,MII-1))
     ***NPII)**NCRSII
      IF(NCRSII.NE.1) GO TO 3
      DPC(1,8)=D2M(SL(II),GE,MII)
      UPC(1,8)=U2M(SL(II),GE,MII)/D(2,MII)*DPC(1,8)
      EPC(1,8)=DPC(1,8)*(PPC(1,8)/DPC(1,8)/(GE-1.)+UPC(1,8)**2/DPC(1,8)
     ***2/2.)
 3    IF(NCROSS(NI)**2.NE.1) GO TO 4
      IF((NAME.EQ.7HSKSKSK0).OR.(NAME.EQ.7HSPECIAL)) GO TO 4
      IF((NRHS(II).EQ.MNI).AND.(NCRSII.NE.-1).AND.(NCRSNI.NE.-1))GO TO 4
C-----SHKNI PREDICTOR FOR NSHK+1+NPNI = 2+NPNI**2
      P(2,MNI+NPNI)=PPC(1,2-NPNI)=(F2M(SL(NI),GE,MNI)*(P(1,MNI)*
     *P(1,MNI-1))**NPNI)**NCRSNI
      IF(NCRSNI.NE.-1) GO TO 4
C-----SHKNI COMPLETE PREDICTOR FOR NSHK = 2+NPNI**2
      D(2,MNI-1)=DPC(1,3)=D2M(SL(NI),GE,MNI)**NCRSNI/(D(1,MNI)*D(1,MNI-1
     *))**NPNI
      U(2,MNI-1)=UPC(1,3)=U2MM(SL(NI),GE,MNI-1)
      IF((NAME.EQ.7HSKSKSKM).AND.(NDISCL(NJ)+1.EQ.MNI)) GO TO 82
      GO TO 83
 82   U(2,MNI-1)=UPC(1,3)=DPC(1,3)*(U(1,MNI-1)/D(1,MNI-1)-SQRT(GE*
     *PPC(1,3)/DPC(1,3))*2./(GE+1.)*SL(NI)*(1.-1./SL(NI)**2))
 83   EPC(1,3)=E2M(GE,MNI-1)
      IF(NAME.EQ.7HSKSKSKM) GO TO 4
C-----SHKNI CORRECTOR FOR NSHK+1 = 4
      D(2,MNI)=DPC(2,4)=DC(4,MNI,0) $ U(2,MNI)=UPC(2,4)=UC(4,MNI,0)
      E(2,MNI)=EPC(2,4)=EC(4,MNI,0) $ P(2,MNI)=PPC(2,4)=PC(4,GE)
C
C
C
 4    IF((NCRSII.EQ.-1).OR.(NCRSNI.EQ.-1)) GO TO 80
      IF(NRHS(II).EQ.MNI) GO TO 10
      IF(NRHS(II).EQ.MNI+1) GO TO 20
      IF(NRHS(II).EQ.MNI+2) GO TO 30
      IF(NRHS(II).EQ.MNI+3) GO TO 24
C
 10   IF(NAME.NE.7HSKSKSK0) GO TO 80
      GO TO 22
C-----SHKNI PREDICTOR FOR NSHK = 2(=1)
 81   DPC(1,2)=DPC(1,1) $ UPC(1,2)=UPC(1,1)
      EPC(1,2)=EPC(1,1) $ PPC(1,2)=PPC(1,1)
C-----SHKNI PREDICTOR FOR NSHK-1 = 1
      M=NDISCL(NI)-1
      DPC(1,1)=DP(M,0) $ UPC(1,1)=UP(M,0)
      EPC(1,1)=EP(M,0) $ PPC(1,1)=PP(1,GE)
C-----SHKNI CORRECTOR FOR NSHK = 2
      M=NDISCL(NI)
      D(2,M)=DPC(2,2)=DC(2,M,0) $ U(2,M)=UPC(2,2)=UC(2,M,0)
      E(2,M)=EPC(2,2)=EC(2,M,0) $ P(2,M)=PPC(2,2)=PC(2,GE)
C-----SHKII CORRECTOR FOR NSHK+1+NCRSII = 1
 80   D(2,MII+NCRSII)=DPC(2,1)=DC(1,MII+NCRSII,1)
```

```
      U(2,MII+NCRSII)=UPC(2,1)=UC(1,MII+NCRSII,1)
      E(2,MII+NCRSII)=EPC(2,1)=EC(1,MII+NCRSII,1)
      P(2,MII+NCRSII)=PPC(2,1)=PC(1,GE)
 12   IF((NAME.EQ.7H SKSKSK).AND.(NCRSII.EQ.-1)) GO TO 75
      ML=1+7*NCRSII*(NCRSII-1)/2
      SSL=SSLS(PPC(1,ML),PPC(1,7+NCRSII),GE) $ SSE=SSES(SSL,GE,1,ML)
      IF(NCRSII**2.NE.1) GO TO 11
C-----SHKII CORRECTOR FOR NSHK+1+NPII = 8
      NDISCL(II)=NDISCL(II)+NCRSII
      SSL=(SSL+SL(II))/2.0
      D(2,MII+NPII)=DPC(2,8)=D2M(SSL,GE,MII)**NCRSII/(D(1,MII)*D(1,MII
     *-1))**NPII
      IF(NCRSII.EQ.1) U(2,MII+NPII)=UPC(2,8)=U2M(SSL,GE,MII)
      IF(NCRSII.EQ.-1) U(2,MII+NPII)=UPC(2,8)=U2MM(SSL,GE,MII-1)
      IF((NSC.EQ.-9).OR.(NSC.EQ.-10).OR.(NSC.EQ.-11)) U(2,MII+NPII)=
     *UPC(2,8)=DPC(2,8)*(U(1,MII-1)/D(1,MII-1)-SQRT(GE*PPC(1,8)/DPC(1,8)
     *)*(2./(GE+1.)*SSL*(1.-1./SSL**2)))
      P(2,MII+NPII)=PPC(2,8)=(P2M(SSL,GE,MII)*(P(1,MII)*P(1,MII-1))
     ***NPII)**NCRSII
      E(2,MII+NPII)=EPC(2,8)=E2M(GE,MII+NPII)
 11   RD(II)=RD(II)+(SSE+SE(II))/2.*DT
      SL(II)=SSLS(PPC(2,ML),PPC(2,7+NCRSII),GE)
      SE(II)=SSES(SL(II),GE,2,ML)
      IF((RD(II).GT.R2(MII)).AND.(NCRSII.EQ.0)) RD(II)=R2(MII)-DELR*
     *1.E-10
      IF((RD(II).LT.R2(MII)).AND.(NCRSII.EQ.1)) RD(II)=R2(MII)+DELR*
     *1.E-10
      IF((RD(II).GT.R2(MII-1)).AND.(NCRSII.EQ.-1)) RD(II)=R2(MII-1)
     *-DELR*1.E-10
      IF((RD(II).LT.R2(MII-1)).AND.(NCRSII.EQ.0)) RD(II)=R2(MII-1)
     *+DELR*1.E-10
 75   IF(NSC.EQ.-11) GO TO 61
 9    IF((RD(II).GT.RD(NI)).AND.(NAME.EQ.7H DISCSS)) GO TO 100
      RETURN
C
 20   IF((NCRSII.EQ.0).AND.(NCRSNI.EQ.0)) GO TO 21
      IF((NCRSII.EQ.1).AND.(NCRSNI.EQ.0)) GO TO 22
C-----HERE NCRSII=1 AND NCRSNI=1 OR NCRSII=0 AND NCRSNI=1
C-----SHKNI COMPLETE PREDICTOR FOR NSHK+1 = 2
      IF(NAME.EQ.7HSPECIAL) GO TO 27
      D(2,MNI)=DPC(1,2)=D2M(SL(NI),GE,MNI)
      U(2,MNI)=UPC(1,2)=U2M(SL(NI),GE,MNI) $ EPC(1,2)=E2M(GE,MNI)
 27   IF((NCRSII.EQ.0).AND.(NCRSNI.EQ.1)) GO TO 21
C-----SHKNI PREDICTOR FOR NSHK = 1
 22   IF((NAME.EQ.7HSPECIAL).AND.(MNI-MII.EQ.2).AND.(NCRSII.EQ.1))
     *GO TO 76
      IF((NAME.EQ.7HSKSKSKO).AND.(NDISCL(NJ)+1.EQ.MNI)) GO TO 77
      GO TO 78
 77   PS=P2M(SL(NJ),GE,MNI) $ DS=D2M(SL(NJ),GE,MNI)
      US=U2M(SL(NJ),GE,MNI)/D(2,MNI)*DS
      US=US/DS+SQRT(GE*PS/DS)*2./(GE+1.)*SL(NI)*(1.-1./SL(NI)**2)
      DS=D2M(SL(NI),GE,MNI)/D(1,MNI)*DS
      US=US*DS
      PS=P2M(SL(NI),GE,MNI)/P(1,MNI)*PS
      GO TO 79
 78   PS=P2M(SL(NI),GE,MNI) $ DS=D2M(SL(NI),GE,MNI)
```

```
        US=U2M(SL(NI),GE,MNI)/D(2,MNI)*DS
 79     ES=(PS/DS/(GE-1.)+US**2/DS**2/2.)*DS
        M=NDISCL(NI)
        EPS=(RD(NI)-R(M))/DELR
        C1=2.*(2.-EPS)/(1.+EPS)
        C2=2.*EPS-3. $ C3=(1.-EPS)*(2.*EPS-1.)/(1.+EPS)
        D(2,M)=DPC(1,1)=DPC(2,1)=D(1,M)-DTDX*(C1*US+C2*U(1,M)+C3*U(1,M-1))
       *-AT*DTDX*(C1*DS+C2*D(1,M)+C3*D(1,M-1))
        U(2,M)=UPC(1,1)=UPC(2,1)=U(1,M)-DTDX*(C1*(US*US/DS+PS*RD(II)**J)+
       *C2*PM(M)+C3*PM(M-1))-AT*DTDX*(C1*US+C2*U(1,M)+C3*U(1,M-1))
        E(2,M)=EPC(1,1)=EPC(2,1)=E(1,M)-DTDX*(C1*US/DS*(ES+PS*RD(II)**J)+
       *C2*PE(M)+C3*PE(M-1))-AT*DTDX*(C1*ES+C2*E(1,M)+C3*E(1,M-1))
        P(2,M)=PPC(1,1)=PPC(2,1)=PP(1,GE)
        IF((NAME.EQ.7HSKSKSKO).AND.(NRHS(II).EQ.MNI)) GO TO 81
 76     IF((NCRSII.EQ.1).AND.(NCRSNI.EQ.0)) GO TO 24
C-----SHKNI CORRECTOR FOR NSHK = 1
 23     M=NDISCL(NI)
        D(2,M)=DPC(2,1)=DC(1,M,1) $ U(2,M)=UPC(2,1)=UC(1,M,1)
        E(2,M)=EPC(2,1)=EC(1,M,1) $ P(2,M)=PPC(2,1)=PC(1,GE)
        GO TO 24
C-----SHKII PREDICTOR AND CORRECTOR FOR NSHK = 1
 21     D(2,MII)=DPC(1,1)=DPC(2,1)=D(1,MII)
        U(2,MII)=UPC(1,1)=UPC(2,1)=U(1,MII)
        E(2,MII)=EPC(1,1)=EPC(2,1)=E(1,MII)
        P(2,MII)=PPC(1,1)=PPC(2,1)=P(1,MII)
        IF((NCRSII.EQ.0).AND.(NCRSNI.EQ.0)) GO TO 24
        GO TO 23
C------CALC SHKNI(HERE) AND SHKII(AT 12) SPDS AND POSITION
 24     IF(NAME.EQ.7HSKSKSKO) GO TO 12
        IF(NAME.EQ.7HSPECIAL) GO TO 26
        MH=1+NCRSNI+NCRSNI*(NCRSNI-1)/2
        SSL=SSLS(PPC(1,3),PPC(1,MH),GE)
        SSE=SSES(SSL,GE,1,3)
        IF(NCRSNI**2.NE.1) GO TO 25
C-----SHKNI CORRECTOR FOR NSHK+1+NPNI = 2+NPNI**2
        NDISCL(NI)=NDISCL(NI)+NCRSNI
        SSL=(SSL+SL(NI))/2.
        D(2,MNI+NPNI)=DPC(2,2-NPNI)=D2M(SSL,GE,MNI)**NCRSNI/(D(1,MNI)*
       *D(1,MNI-1))**NPNI
        P(2,MNI+NPNI)=PPC(2,2-NPNI)=(P2M(SSL,GE,MNI)*(P(1,MNI)*P(1,
       *MNI-1))**NPNI)**NCRSNI
        IF(NCRSNI.EQ.1) U(2,MNI+NPNI)=UPC(2,2-NPNI)=U2M(SSL,GE,MNI)
        IF(NCRSNI.EQ.1) GO TO 85
        U(2,MNI+NPNI)=UPC(2,2-NPNI)=U2MM(SSL,GE,MNI-1)
        IF((NAME.EQ.7HSKSKSKM).AND.(NDISCL(NJ)+1.EQ.MNI)) GO TO 84
        GO TO 85
 84     U(2,MNI-1)=UPC(2,3)=DPC(2,3)*(U(1,MNI-1)/D(1,MNI-1)-SQRT(GE*
       *PPC(1,3)/DPC(1,3))*2./(GE+1.)*SSL*(1.-1./SSL**2))
 85     E(2,MNI+NPNI)=EPC(2,2-NPNI)=E2M(GE,MNI+NPNI)
 25     RD(NI)=RD(NI)+(SSE+SE(NI))/2.*DT
        SL(NI)=SSLS(PPC(2,3),PPC(2,MH),GE)
        SE(NI)=SSES(SL(NI),GE,2,3)
        IF((RD(NI).GT.R2(MNI)).AND.(NCRSNI.EQ.0)) RD(NI)=R2(MNI)-DELR*
       *1.E-10
        IF((RD(NI).LT.R2(MNI)).AND.(NCRSNI.EQ.1)) RD(NI)=R2(MNI)+DELR*
       *1.E-10
```

```
      IF((RD(NI).GT.R2(MNI-1)).AND.(NCRSNI.EQ.-1)) RD(NI)=R2(MNI-1)
     *-DELR*1.E-10
      IF((RD(NI).LT.R2(MNI-1)).AND.(NCRSNI.EQ.0)) RD(NI)=R2(MNI-1)
     *+DELR*1.E-10
 26   IF(NSC.EQ.-11) GO TO 9
      IF(NRHS(II).EQ.MNI+3) GO TO 40
      GO TO 12
C
C
 30   IF((NCRSII.EQ.0).AND.(NCRSNI.EQ.0)) GO TO 31
      IF((NCRSII.EQ.1).AND.(NCRSNI.EQ.0)) GO TO 32
C-----HERE NCRSII=1 AND NCRSNI=1 OR NCRSII=0 AND NCRSNI=1
C-----PRDCTR AND CRRCTR FOR FCTTOUS NDE =1
      IF(NAME.EQ.7HSPECIAL) GO TO 24
      D(2,MNI)=DPC(1,1)=D2M(SL(NI),GE,MNI)
      UPC(1,1)=U2M(SL(NI),GE,MNI) $ PPC(1,1)=PPC(1,2)
      SSL=(SSLS(PPC(1,3),PPC(1,2),GE)+SL(NI))/2.
      DPC(2,1)=D2M(SSL,GE,MNI) $ PPC(2,1)=P2M(SSL,GE,MNI)
      UPC(2,1)=U2M(SSL,GE,MNI)/D(2,MNI)*DPC(2,1)
      GO TO 24
C-----SHKII PRDCTR AND CRRCTR FOR FCTTOUS NDE =1
 31   DPC(2,1)=DPC(1,1)=D2M(SL(NI),GE,MNI)
      PPC(2,1)=PPC(1,1)=P2M(SL(NI),GE,MNI)
      UPC(2,1)=UPC(1,1)=U2M(SL(NI),GE,MNI)/D(2,MNI)*DPC(1,1)
      IF((NAME.EQ.7HSKSKSKO).AND.(NDISCL(NJ)+1.EQ.MNI)) GO TO 101
      GO TO 24
 101  DPC(2,1)=DPC(1,1)=DS1 $ UPC(2,1)=UPC(1,1)=US1
      PPC(2,1)=PPC(1,1)=PS1
      GO TO 24
C-----SHKII PREDICTOR AND CORRECTOR FOR NSHK+1 = 1
 32   DPC(2,1)=DPC(1,1)=D(1,MII) $ UPC(2,1)=UPC(1,1)=U(1,MII)
      PPC(2,1)=PPC(1,1)=P(1,MII)
      GO TO 24
C
C-----SHKII PREDICTOR FOR NSHK+1 = 8
 40   IF(NAME.EQ.7HSPECIAL) GO TO 41
      PPC(1,1)=PPC(1,2) $ D(2,MNI)=DPC(1,1)=D2M(SLNISAV,GE,MNI)
      UPC(1,1)=U2M(SLNISAV,GE,MNI)/D(2,MNI)*DPC(1,1)
 41   PPC(1,8)=P2M(SL(II),GE,MII)/P(1,MII)*PPC(1,1)
      DPC(1,8)=D2M(SL(II),GE,MII)/D(1,MII)*DPC(1,1)
      UPC(1,8)=DPC(1,8)*(UPC(1,1)/DPC(1,1)+SQRT(GE*PPC(1,1)/DPC(1,1))*
     *(2./(GE+1.)*SL(II)*(1.-1./SL(II)**2)))
      EPC(1,8)=DPC(1,8)*(PPC(1,8)/DPC(1,8)/(GE-1.)+UPC(1,8)**2/DPC(1,8)
     ***2/2.)
      SSL=SSLS(PPC(1,1),PPC(1,8),GE) $ SSE=SSES(SSL,GE,1,1)
      NDISCL(II)=NDISCL(II)+NCRSII
      SSL=(SSL+SL(II))/2.
      D(2,MII)=DPC(2,8)=D2M(SSL,GE,MII)/D(1,MII)*DPC(1,1)
      P(2,MII)=PPC(2,8)=P2M(SSL,GE,MII)/P(1,MII)*PPC(1,1)
      U(2,MII)=UPC(2,8)=DPC(2,8)*(UPC(1,1)/DPC(1,1)+SQRT(GE*PPC(1,1)/
     *DPC(1,1))*(2./(GE+1.)*SSL*(1.-1./SSL**2)))
      E(2,MII)=EPC(2,8)=E2M(GE,MII)
      DPC(2,1)=DPC(2,2) $ UPC(2,1)=UPC(2,2)
      EPC(2,1)=EPC(2,2) $ PPC(2,1)=PPC(2,2)
      ML=1
      GO TO 11
```

```
C
 50    NSC=100*(MNI-MII)+10*NCRSII+NCRSNI
       PRINT 2002,NSC,MNI,MII,NCRSII,NCRSNI
 2002 FORMAT(1H ,1X,5I5)
       IF(NSC.EQ.-9) GO TO 60
       IF((NSC.NE.90).AND.(NSC.NE.91)) GO TO 51
C-----SHKII PREDICTOR FOR NSHK+1 = 9,1
       DPC(1,9)=DPC(1,1)=D(1,MII) $ UPC(1,9)=UPC(1,1)=U(1,MII)
       EPC(1,9)=EPC(1,1)=E(1,MII) $ PPC(1,9)=PPC(1,1)=P(1,MII)
       GO TO 60
 51    IF((NSC.NE.199).AND.(NSC.NE.309).AND.(NSC.NE.189)) GO TO 57
C-----SHKNI PREDICTOR AND CORRECTOR FOR NSHK-1 = 9,1
       D(2,MNI-2)=DPC(1,9)=DPC(1,1)=DPC(2,1)=DP(MNI-2,0)
       U(2,MNI-2)=UPC(1,9)=UPC(1,1)=UPC(2,1)=UP(MNI-2,0)
       E(2,MNI-2)=EPC(1,9)=EPC(1,1)=EPC(2,1)=EP(MNI-2,0)
       P(2,MNI-2)=PPC(1,9)=PPC(1,1)=PPC(2,1)=PP(1,GE)
       IF(NSC.NE.189) GO TO 24
C-----SHKII COMPLETE CORRECTOR FOR NSHK = 8
 60    IF((NAME.EQ.7H SKSKSK).AND.(NCRSII.EQ.-1)) GO TO 53
       D(2,MII-1)=DPC(1,8)=D2M(SL(II),GE,MII)**NCRSII/(D(1,MII)*D(1,MII-1
      *))**NPII
       U(2,MII-1)=UPC(1,8)=U2MM(SL(II),GE,MII-1)
       IF((NSC.EQ.-11).OR.(NSC.EQ.-10).OR.(NSC.EQ.-9)) GO TO 93
       GO TO 88
 93    U(2,MNI-1)=UPC(1,8)=DPC(1,8)*(U(1,MNI-1)/D(1,MNI-1)-SQRT(GE*
      *PPC(1,8)/DPC(1,8))*2./(GE+1.)*SL(II)*(1.-1/SL(II)**2))
 88    EPC(1,8)=E2M(GE,MII-1)
 53    IF((NSC.EQ.-9).OR.(NSC.EQ.-10).OR.(NSC.EQ.89)) GO TO 24
       IF(NSC.EQ.-11) GO TO 12
C-----SHKII CORRECTOR FOR NSHK+1 = 1
       D(2,MII)=DPC(2,1)=DC(9,MII,0) $ U(2,MII)=UPC(2,1)=UC(9,MII,0)
       E(2,MII)=EPC(2,1)=EC(9,MII,0) $ P(2,MII)=PPC(2,1)=PC(1,GE)
       GO TO 24
 57    IF((NSC.NE.-10).AND.(NSC.NE.89)) GO TO 54
C-----SHKII PREDICTOR AND CORRECTOR FOR NSHK = 1
       PPC(1,1)=PPC(1,8)
       SSL1=SSLS(PPC(1,8),PPC(1,6),GE) $ SSL1=(SSL1+SL(II))/2.
       PPC(2,1)=(P2M(SSL1,GE,MII)*(P(1,MII)*P(1,MII-1))**NPII)**NCRSII
       IF((NAME.EQ.7H SKSKSK).AND.(NCRSII.EQ.-1)) PPC(2,1)=PPC(2,8)
       GO TO 60
 54    IF(NSC.NE.209) GO TO 55
C-----SHKII PREDICTOR AND CORRECTOR FOR NSHK+1 = 1
       DPC(1,1)=DPC(2,1)=(D(1,MII)+D(1,MII+1))/2.
       UPC(1,1)=UPC(2,1)=(U(1,MII)/D(1,MII)+U(1,MII+1)/D(1,MII+1))*
      *DPC(1,1)/2.
       PPC(1,1)=PPC(2,1)=(P(1,MII)+P(1,MII+1))/2.
       GO TO 24
 55    IF(NSC.NE.99) GO TO 56
C-----SHKII PREDICTOR AND CORRECTOR FOR NSHK+1 = 1
 63    DPC(1,1)=DPC(2,1)=D(1,MII) $ PPC(1,1)=PPC(2,1)=P(1,MII)
       UPC(1,1)=UPC(2,1)=U(1,MII)
       GO TO 24
 56    IF(NSC.NE.-11) GO TO 58
       GO TO 60
C-----SHKNI PREDICTOR FOR NSHK = 3
 61    P(2,MNI-1)=PPC(1,3)=(P2M(SL(NI),GE,MNI)*(P(1,MNI)*PPC(1,8))**NPNI)
```

```
     ***NCRSNI
      D(2,MNI-1)=DPC(1,3)=D2M(SL(NI),GE,MNI)**NCRSNI/(D(1,MNI)*DPC(1,8)
     *)**NPNI
      U(2,MNI-1)=UPC(1,3)=DPC(1,3)*(UPC(1,8)/DPC(1,8)-SQRT(GE*P(1,MNI)/
     *D(1,MNI))*(2./(GE+1.)*SL(NI)*(1.-1./SL(NI)**2)))
      IF((NAME.EQ.7HSKSKSKM).AND.(NDISCL(NJ)+1.EQ.MNI)) GO TO 89
      GO TO 90
 89   U(2,MNI-1)=UPC(1,3)=DPC(1,3)*(UPC(1,8)/DPC(1,8)-SQRT(GE*PPC(1,3)/
     *DPC(1,3))*2./(GE+1)*SL(NI)*(1.-1/SL(NI)**2))
 90   EPC(1,3)=E2M(GE,MNI-1)
      SSL=SSLS(PPC(1,3),PPC(1,8),GE) $ SSE=SSES(SSL,GE,1,3)
      NDISCL(NI)=NDISCL(NI)+NCRSNI
C-----SHKNI CORRECTOR FOR NSHK = 3
      SSL=(SSL+SL(NI))/2.
      D(2,MNI-1)=DPC(2,3)=D2M(SSL,GE,MNI)**NCRSNI/(D(1,MNI)*DPC(1,8))
     **NPNI
      P(2,MNI-1)=PPC(2,3)=(P2M(SSL,GE,MNI)*(P(1,MNI)*PPC(1,8))**NPNI)
     ***NCRSNI
      U(2,MNI-1)=UPC(2,3)=DPC(2,3)*(UPC(1,8)/DPC(1,8)-SQRT(GE*P(1,MNI)
     */D(1,MNI))*(2./(GE+1.)*SSL*(1.-1./SSL**2)))
      IF((NAME.EQ.7HSKSKSKM).AND.(NDISCL(NJ)+1.EQ.MNI))GO TO 91
      GO TO 92
 91   U(2,MNI-1)=UPC(2,3)=DPC(2,3)*(UPC(1,8)/DPC(1,8)-SQRT(GE*PPC(1,3)
     */DPC(1,3))*2./(GE+1.)*SSL*(1.-1./SSL**2))
 92   E(2,MNI-1)=EPC(2,3)=E2M(GE,MNI-1)
C-----SHKNI CORRECTOR FOR NSHK+1 = 4
      IF(NAME.EQ.7HSKSKSKM) GO TO 59
      D(2,MNI)=DPC(2,4)=DC(4,MNI,0) $ U(2,MNI)=UPC(2,4)=UC(4,MNI,0)
      E(2,MNI)=EPC(2,4)=EC(4,MNI,0) $ P(2,MNI)=PPC(2,4)=PC(4,GE)
 59   MH=8
      GO TO 25
 58   IF(NSC.NE.109) GO TO 62
      TRFLCT=(RD(NI)-RD(II))/(SE(II)-SE(NI))
      MM=NCD+1 $ IF(NSHK-NCD-1.GE.1) MM=NCD+2
      IF(RD(NI)+TRFLCT*SE(NI).GT.R2(MII)) GO TO 64
      NPII=NCROSS(II)=NCRSII=0 $ NRHS(II)=NRHS(II)-1 $ GO TO 63
 64   NCROSS(NI)=NCRSNI=NPNI=0 $ NRHS(NI)=NRHS(NI)+1 $ GO TO 71
 62   PRINT 1000,NSC
 1000 FORMAT(1H ,3/,* TROUBLE IN SKSKPF, NSC= *,I5)
      DELR=-ABS(DELR) $ RETURN
C
 100  TARFLCT=(RD(II)-RD(NI))/(SE(II)-SE(NI))
      RD(NI)=RD(II)=RD(NI)-SE(NI)*TARFLCT
      CALL SKSKPPA(II,NI,6HSKSKPF,TARFLCT)
C
      RETURN $ END
```

```
      SUBROUTINE SKSKPPA(II,NI,NAME,TARFLCT)
      COMMON/PARAM/N,J,AJ,G,GF,DELR,NFLM
      COMMON/TIME/T,DT,DTL,TWRITE,DELT,DTDX,AT
      COMMON/FIRFETC/INDEX,NCYCLE,NN,NNN,NSTORE,NS,NITRCTN
      COMMON/DISCSKF/RDSAV(51),NLHS(51),NRHS(51),NCROSS(51)
      COMMON/ARRAYS/R(501),U(2,501),P(2,501),D(2,501),E(2,501),R2(501)
      COMMON/DISCS/NTDISC,NDISCNC(51),NTYPE(51),NDISCL(51),SE(51),SL(51)
     *,RD(51),NTDISCT,NTDISCS
      COMMON/PPCHR/NPPTH,NUMA,NUMAFST,NUPA,RPPTH(24),RUMA(150),RUPA(150)
     *,NCLPPTH(24),NCLUMA(150),NCLUPA(150),PTHNEXT,RUMANXT,TUPANXT,
     *DELPPTH,DELUMA,DELUPA
C
C-----DET SS FLW FLD (STTES 4,5,6) DUE TO SHK-SHK INTRCTN BY TH PLR MTHD
C-----SET STATE1,0 AS REF STATE FOR POLAR SOLN, AS OVERALL REF STATE
C-----VEL REF WRT SQRT(G*P1/D1) IN PLR SOLN
C     P41=TO BE READ AS PRESS IN STATE4 REL TO AND NONDIM BY STATE 1
C     U531=READ AS PRTCLE VEL IN STATE5 REL TO STATE3, NONDIM WRT STATE1
C
C
C     -STATE- -3-+++6+++RAREFACTION+++6+++-----5---CD----4-----SHK-1-
C                H    *     *     *      T         D           *
C                  H    *    *     *               C          *
C                    H    *    *    *     T        D
C                      H   *    *    *             C         *
C                        H   *    *   *    T       D        *
C                          H   *   *   *           C
C                            H  *   *   *   T      D       *
C                              H  *  *   *         C      *
C                                H  *  *  *   T    D
C                                  H  * *  *       C     *
C                                    H  * * *  T   D    *
C                                      H * * *     C
C                                        H ** * T  D   *
C                                          H ***   C  *
C                                            H***T D *
C                                              H**TC
C                                                H*T*
C                                                  *
C                                                **
C                                            *   *
C                                        *      *
C                                     *        *
C                                 *           *
C                              *             *
C                        -3---SHK----2-----SHK---1-
C
C
C-----SET STATES 1 AND 3
      GE=G $ IF(NDISCL(NI).LE.NFLM) GE=GF
      MNI=NDISCL(NI)+1 $ MII=NDISCL(II)
      P10=P(2,MNI) $ D10=D(2,MNI) $ U10=U(2,MNI)/D10
      A10=SQRT(P10/D10*GE/G)
      P11=D11=A11=1.0 $ U11=0.0
      P31=P(2,MII)/P(2,MNI) $ D31=D(2,MII)/D(2,MNI)
      U31=(U(2,MII)/D(2,MII)-U(2,MNI)/D(2,MNI))/SQRT(G)/A10
      A31=SQRT(P31/D31) $ SL21=SL(NI) $ SL32=SL(II)
```

```
C
C-----SET SUB CONSTS AND ITERATION COUNTER
      BETA=(GE-1.)/(GE+1.) $ ALPHA=2./(GE-1.)
      NSIGN=-1 $ EPS=1.E-14 $ NITER=0 $ DELP541=1.0
C-----GUESS P51=P41 AS AVG OF P31 AND P FOR U=U31 AT A=A11
      XI=(U31-0.)**2/A11**2*GE/(1.-BETA)
      P41=(2.+XI+SQRT(XI**2+4.*XI*(1.+BETA)))/2.
      P51=P41=(P41+P31)/2.0-1.0
C
C-----PERFORM ITER OF THE PLR SOLN
      PRINT 1002,NAME,II,NI,SL(II),SL(NI)
 1002 FORMAT(1H1,5X,*SHK-SHK INTERACTION (*,A6,*)*,5X,2I3,10X,*MACH NOS
     *= *,2F12.8)
 1    NITER=NITER+1
      IF(NITER.LT.51) GO TO 2
      IF(NITER.EQ.51) DELUSAV=DELU
      IF(NITER.LT.70) GO TO 2
      IF((DELUSAV.LT.1.E-12).AND.(DELU.LT.1.E-11)) GO TO 10
      PRINT 1000,DELUSAV,DELU
 1000 FORMAT(1H ,5/,5X,*ITERATIONS HAVE REACHED MAX IN SHK-SHK (*,A6,
     **)*,5X,2E11.4,5/)
      DELP=-ABS(DELP) $ RETURN
 2    P51=P41=P41+DELP541
      SL41=SQRT((P41+BETA)/(1.+BETA)) $ SE41=SL41*A11+U11
      D41=1./(BETA+(1.-BETA)/SL41**2)
      U41=(1.-BETA)*(1.-1./SL41**2)*SL41*A11
      A41=SQRT((1.-BETA)/(1.+BETA)*(1.+BETA+BETA*(P41-1.))/(P41+BETA)*GE
     **P41)
      P53=P51/P31
      D53=((2.+ALPHA)/GE/ALPHA*(P53-1.)+1.)**(ALPHA/(2.+ALPHA))
      D51=D53*D31
      A53=D53**(1./ALPHA) $ A51=A53*A31
      U531=-ALPHA*(A53-1.)*A31 $ U51=U531+U31
      SLH31=SLT51=-1.0 $ SEH31=SLH31*A31+U31 $ SET51=SLT51*A51+U51
C
      DELU=ABS(U41-U51)
C
      PRINT 1001,NITER,DELP541,DELU,P10,D10,A10,U10,SL21,
     *P31,D31,A31,U31,SL32,
     *P41,D41,A41,U41,SL41,SE41,
     *P53,P51,D53,D51,A53,A51,U531,U51,SLH31,SEH31,
     *                                        SLT51,SET51
 1001 FORMAT(1H ,1X,I5,2E13.5,5F12.8,/,1X,4(12X,F12.8),F12.8,/,1X,
     *4(12X,F12.8),2F12.8,/,1X,10F12.8,/,97X,2F12.8,/)
C
      IF(DELU.LE.EPS) GO TO 10
C
      IF((U51.LT.U41).AND.(NSIGN.LE.0)) DELP541=-ABS(DELP541)
      IF((U51.LT.U41).AND.(NSIGN.EQ.-1)) NSIGN=0
      IF((U51.LT.U41).AND.(NSIGN.EQ.1)) DELP541=-ABS(DELP541)/2.0
      IF((U41.LT.U51).AND.(NSIGN.EQ.-1)) DELP541=ABS(DELP541)
      IF((U41.LT.U51).AND.(NSIGN.GE.0)) DELP541=ABS(DELP541)/2.0
      IF((U41.LT.U51).AND.(NSIGN.EQ.0)) NSIGN=1
      GO TO 1
C
C
```

```
 10    NT=NTDISCT
       NDISCNC(NT+1)=NTDISC+1 $ NDISCNC(NT+2)=NTDISC+2
       NTYPE(NT+1)=4 $ NTYPE(NT+2)=2
       SL(NT+1)=0.0 $ SL(NT+2)=SL41
       SE(NT+1)=U41*A10*SQRT(G)+U10 $ SE(NT+2)=SE41*A10*SQRT(G)+U10
       IF(RD(II).GT.R2(MII)) GO TO 11
       DTRFLCT=((RD(II)+SE(II)*TARFLCT+R2(MII))/2.-RD(II))/SE(II)
       RD(II)=RD(NI)=RD(II)+SE(II)*DTRFLCT
       IF(NAME.EQ.6HSKSKSK) RD(II+1)=RD(II)
       TARFLCT=TARFLCT-DTRFLCT
 11    RD(NT+1)=RD(II)+SE(NT+1)*TARFLCT $RD(NT+2)=RD(II)+SE(NT+2)*TARFLCT
       IF(RD(NT+2).GT.R2(MNI)) RD(NT+2)=R2(MNI)
       IF(RD(NT+1).LT.R2(MII)) RD(NT+1)=R2(MII)
       IF(RD(NT+1).GE.RD(NT+2)) RD(NT+1)=(RD(II)+RD(NT+2))/2.
C
       DO 12 I=MNI,N
       M=N+2+MNI-I
       D(2,M)=D(2,M-2) $U(2,M)=U(2,M-2) $E(2,M)=E(2,M-2) $P(2,M)=P(2,M-2)
       D(1,M)=D(1,M-2) $U(1,M)=U(1,M-2) $E(1,M)=E(1,M-2) $P(1,M)=P(1,M-2)
       R(M)=R(M-2)
 12    R2(M)=R2(M-2)
       N=N+2
       R2(MII+1)=R2(MII+2)=RD(NT+1)
       DO 13 I=1,NTDISCT
       IF(NDISCL(I).GE.MNI) NDISCL(I)=NDISCL(I)+2
 13    CONTINUE
       IF(NI+1.LE.NTDISCS) NRHS(NI+1)=NRHS(NI+1)+2
       IF(NI+1.LE.NTDISCS) NLHS(NI+1)=NLHS(NI+1)+2
       NDISCL(II)=NDISCL(NI)=-1
       IF(NAME.EQ.6HSKSKSK) NDISCL(II+1)=-1
       NDISCL(NT+1)=MII+1 $ NDISCL(NT+2)=MII+2
C-----SET GAS AND THERMC PARAMETERS IN CELLS MII+1,2
       D(2,MII+1)=D(2,MII) $ U(2,MII+1)=U(2,MII) $ E(2,MII+1)=E(2,MII)
       P(2,MII+1)=P(2,MII)
       D(2,MII+2)=D41*D10 $ U(2,MII+2)=(U41*A10*SQRT(G)+U10)*D(2,MII+2)
       P(2,MII+2)=P41*P10
       E(2,MII+2)=D(2,MII+2)*(P(2,MII+2)/D(2,MII+2)/(GE-1.)+U(2,MII+2)**2
      */D(2,MII+2)**2/2.)
       NTDISCT=NTDISCT+2 $ NTDISC=NTDISC+2
C
C-----ACCT FOR PRTCLE PTH AND NEG AND POS CHRCT TRAJ CELL LOCTNS CHANGES
       IF(NPPTH.LE.0) GO TO 110
       DO 111 I=1,NPPTH
       IF(NCLPPTH(I).GE.MNI) NCLPPTH(I)=NCLPPTH(I)+2
 111   CONTINUE
 110   IF(NUMA.LE.0) GO TO 112
       DO 113 I=NUMAFST,NUMA
       IF(NCLUMA(I).GE.MNI) NCLUMA(I)=NCLUMA(I)+2
 113   CONTINUE
 112   IF(NUPA.LE.0) GO TO 114
       DO 115 I=1,NUPA
       IF(NCLUPA(I).GE.MNI) NCLUPA(I)=NCLUPA(I)+2
 115   CONTINUE
C
 114   NCODE=10HSKSKFPA    $ CALL PRNTFF(NCODE)
       NITRCTN=3HYES
```

```
C
      RETURN $ END
```

```
      SUBROUTINE SKSKPN(N1,N2)
      COMMON/PARAM/N,J,AJ,G,GF,DELR,NFLM
      COMMON/TIME/T,DT,DTL,TWRITE,DELT,DTDX,AT
      COMMON/DISCSKP/RDSAV(51),NLFS(51),NRFS(51),NCROSS(51)
      COMMON/PREDCOR/DPC(2,13),UPC(2,13),EPC(2,13),PPC(2,13)
      COMMON/ARRAYS/R(501),U(2,501),P(2,501),D(2,501),E(2,501),R2(501)
      COMMON/DISCS/NTDISC,NDISCNC(51),NTYPE(51),NDISCL(51),SE(51),SL(51)
     *,RD(51),NTDISCT,NTDISCS
C
C-----SL1.GT.0.0 , SL2.LT.0.0
C
C
      D2M(SSL,GE,M)=D(1,M)/((GE-1.)/(GE+1.)+2./(GE+1.)/SSL**2)
      U2M(SSL,GE,M)=U(1,M)/D(1,M)+SQRT(GE*P(1,M)/D(1,M))*(2./
     *(GE+1.)*SSL*(1.-1./SSL**2))
      P2M(SSL,GE,M)=P(1,M)*(2.*GE/(GE+1.)*SSL**2-(GE-1.)/(GE+1.))
      E2M(GE,M,L)=DPC(M,L)*(PPC(M,L)/DPC(M,L)/(GE-1.)+UPC(M,L)**2/
     *DPC(M,L)**2/2.)
C
      SSLS(PHGH,PLOW,GE)=SQRT((PHGH/PLOW+(GE-1.)/(GE+1.))*(GE+1.)/2./GE)
      SSPS(SSL,GE,K,L)=SSL*SQRT(GE*PPC(K,L)/DPC(K,L))+UPC(K,L)/DPC(K,L)
C
C
      GE=G $ IF(NDISCL(N2).LE.NFLM) GE=GF
      NSHK1=NDISCL(N1) $ NSHK2=NDISCL(N2)
      NXS1=NCROSS(N1) $ NXS2=NCROSS(N2)
      NSKSKX=3H NO $ IF(RDSAV(N1).GT.RDSAV(N2)) NSKSKX=3HYES
      NSC=100*(NSHK2-NSHK1)+10*NXS1+NXS2
      PRINT 1000,N1,N2,NXS1,NXS2,NSHK1,NSHK2,NSC,NSKSKX
 1000 FORMAT(1H ,1X,*SKSKPN*,7I5,A3)
      TRFLCT=(RD(N2)-RD(N1))/(SE(N1)-SE(N2))
      RRFLCT=RD(N1)+SE(N1)*TRFLCT
      IF((NSC.NE.10).AND.(NSC.NE.-1).AND.(NSC.NE.9)) GO TO 7
      NXS1=NXS2=0 $ NSC=0 $ GO TO 9
 7    IF(NSC.NE.109) GO TO 9
      IF(RRFLCT.GT.R2(NSHK2)) GO TO 14
      NXS1=0 $ NSC=99 $ GO TO 9
 14   NXS2=0 $ NSC=110
C
 9    IF(NSC.NE.0) GO TO 4
C-----PREDICTOR AND CORRECTOR FOR NODES 4,5
      DPC(1,4)=DPC(2,4)=DPC(1,5)=DPC(2,5)=D(1,NSHK1)**2/D2M(SL(N1),GE,
     *NSHK1)
      PPC(1,4)=PPC(2,4)=PPC(1,5)=PPC(2,5)=P(1,NSHK1)**2/P2M(SL(N1),GE,
     *NSHK1)
      UPC(1,4)=UPC(2,4)=UPC(1,5)=UPC(2,5)=(U(1,NSHK1)/D(1,NSHK1)-SQRT(
     *GE*PPC(1,4)/DPC(1,4))*2./(GE+1.)*SL(N1)*(1.-1./SL(N1)**2))*DPC(1,4
     *)
C-----PREDICTOR FOR NSHK2+1 = 6
 4    M=NSHK2+1
      DPC(1,6)=DP(M,0) $ UPC(1,6)=UP(M,0)
      EPC(1,6)=EP(M,0) $ PPC(1,6)=PP(6,GE)
      IF(NXS2.EQ.-1) GO TO 1
C-----PREDICTOR FOR NSHK2+2 = 7
      M=NSHK2+2
      DPC(1,7)=DP(M,0) $ UPC(1,7)=UP(M,0)
```

```
      EPC(1,7)=EP(M,0) $ PPC(1,7)=PP(7,GE)
      GO TO 2
C-----PREDICTOR FOR NSHK2 = 5
 1    DPC(1,5)=D2M(SL(N2),GE,NSHK2) $ PPC(1,5)=P2M(SL(N2),GE,NSHK2)
      UPC(1,5)=U2M(SL(N2),GE,NSHK2)*DPC(1,5) $ EPC(1,5)=E2M(GE,1,5)
C-----CORRECTOR FOR NSHK2+1 = 6
      M=NSHK2+1
      D(2,M)=DPC(2,6)=DC(6,M,0) $ U(2,M)=UPC(2,6)=UC(6,M,0)
      E(2,M)=EPC(2,6)=EC(6,M,0) $ P(2,M)=PPC(2,6)=PC(6,GE)
C-----PREDICTOR FOR NSHK1-1 = 1
 2    M=NSHK1-1
      DPC(1,1)=DP(M,0) $ UPC(1,1)=UP(M,0)
      EPC(1,1)=EP(M,0) $ PPC(1,1)=PP(1,GE)
C-----PREDICTOR FOR NSHK1 = 2
      M=NSHK1+1
      DPC(1,13)=D2M(SL(N1),GE,M) $ PPC(1,13)=P2M(SL(N1),GE,M)
      UPC(1,13)=U2M(SL(N1),GE,M)*DPC(1,13) $ EPC(1,13)=E2M(GE,1,13)
      IF(NSC.NE.0) GO TO 18
      DPC(1,13)=DPC(1,13)/D(1,M)*DPC(1,4)
      PPC(1,13)=PPC(1,13)/P(1,M)*PPC(1,4)
      UPC(1,13)=DPC(1,13)*(UPC(1,4)/DPC(1,4)+SQRT(GE*PPC(1,4)/DPC(1,4))
     **2./(GE+1)*SL(N1)*(1.-1./SL(N1)**2))
      EPC(1,13)=E2M(GE,1,13)
 18   EPS=(RD(N1)-R(NSHK1))/DELR
      C1=2.*(2.-EPS)/(1.+EPS)
      C2=2.*EPS-3. $ C3=(1.-EPS)*(2.*EPS-1.)/(1.+EPS)
      DPC(1,2)=D(1,NSHK1)-DTDX*(C1*UPC(1,13)+C2*U(1,NSHK1)+C3*U(1,NSHK1-
     *1))-AT*DTDX*(C1*DPC(1,13)+C2*D(1,NSHK1)+C3*D(1,NSHK1-1))
      UPC(1,2)=U(1,NSHK1)-DTDX*(C1*CM(13)+C2*PM(NSHK1)+C3*PM(NSHK1-1))
     *-AT*DTDX*(C1*UPC(1,13)+C2*U(1,NSHK1)+C3*U(1,NSHK1-1))
      EPC(1,2)=E(1,NSHK1)-DTDX*(C1*CE(13)+C2*PE(NSHK1)+C3*PE(NSHK1-1))
     *-AT*DTDX*(C1*EPC(1,13)+C2*E(1,NSHK1)+C3*E(1,NSHK1-1))
      PPC(1,2)=PP(2,GE)
C-----CORRECTOR FOR NSHK1 = 2
      M=NSHK1
      D(2,M)=DPC(2,2)=DC(2,M,0) $ U(2,M)=UPC(2,2)=UC(2,M,0)
      E(2,M)=EPC(2,2)=EC(2,M,0) $ P(2,M)=PPC(2,2)=PC(2,GE)
      IF(NSC.EQ.0) GO TO 11
      IF(NXS1.NE.1) GO TO 3
C-----PREDICTOR FOR NSHK1+1 = 3
      M=NSHK1+1
      DPC(1,3)=D2M(SL(N1),GE,M) $ PPC(1,3)=P2M(SL(N1),GE,M)
      UPC(1,3)=U2M(SL(N1),GE,M)*DPC(1,3) $ EPC(1,3)=E2M(GE,1,3)
 3    IF((NSC.NE.110).AND.(NSC.NE.100).AND.(NSC.NE.99)) GO TO 6
      IF(NSC.EQ.99) GO TO 5
C-----PREDICTOR AND CORRECTOR FOR NSHK2 = 5
      D(2,NSHK2)=DPC(2,5)=DPC(1,5)=D(1,NSHK2)
      U(2,NSHK2)=UPC(2,5)=UPC(1,5)=U(1,NSHK2)
      E(2,NSHK2)=EPC(2,5)=EPC(1,5)=E(1,NSHK2)
      P(2,NSHK2)=PPC(2,5)=PPC(1,5)=P(1,NSHK2)
C-----PREDICTOR AND CORRECTOR FOR NODE 4
 5    DPC(2,4)=DPC(1,4)=D(1,NSHK2) $ UPC(1,4)=UPC(2,4)=U(1,NSHK2)
      EPC(2,4)=EPC(1,4)=E(1,NSHK2) $ PPC(1,4)=PPC(2,4)=P(1,NSHK2)
      GO TO 11
 6    IF(NSC.EQ.209) GO TO 10
      IF((NSC.NE.210).AND.(NSC.NE.200).AND.(NSC.NE.310)) GO TO 8
```

```
C-----PREDICTOR AND CORRECTOR FOR NSHK2 = 4,5
      D(2,NSHK2)=DPC(2,4)=DPC(1,4)=DPC(2,5)=DPC(1,5)=DP(NSHK2,-1)
      U(2,NSHK2)=UPC(2,4)=UPC(1,4)=UPC(2,5)=UPC(1,5)=UP(NSHK2,-1)
      E(2,NSHK2)=EPC(2,4)=EPC(1,4)=EPC(2,5)=EPC(1,5)=EP(NSHK2,-1)
      P(2,NSHK2)=PPC(2,4)=PPC(1,4)=PPC(2,5)=PPC(1,5)=PP(4,GE)
      IF(NSC.EQ.210) GO TO 11
C-----PREDICTOR AND CORRECTOR FOR NSHK2-1 = 4
 8    M=NSHK2-1
      D(2,M)=DPC(2,4)=DPC(1,4)=DP(M,0)
      U(2,M)=UPC(2,4)=UPC(1,4)=UP(M,0)
      E(2,M)=EPC(2,4)=EPC(1,4)=EP(M,0)
      P(2,M)=PPC(2,4)=PPC(1,4)=PP(4,GE)
      IF((NSC.NE.200).AND.(NSC.NE.310)) GO TO 11
C-----CORRECTOR FOR NSHK2-1 = 4
      M=NSHK2-1
      D(2,M)=DPC(2,4)=DC(4,M,1) $ U(2,M)=UPC(2,4)=UC(4,M,1)
      E(2,M)=EPC(2,4)=EC(4,M,1) $ P(2,M)=PPC(2,4)=PC(4,GE)
C-----CORRECTOR FOR NSHK2 = 5
      M=NSHK2
      D(2,M)=DPC(2,5)=DC(5,M,0) $ U(2,M)=UPC(2,5)=UC(5,M,0)
      E(2,M)=EPC(2,5)=EC(5,M,0) $ P(2,M)=PPC(2,5)=PC(5,GE)
      GO TO 11
C-----PREDICTOR AND CORRECTOR FOR NODE 4
 10   DPC(1,4)=DPC(2,4)=(D(1,NSHK1+1)+D(1,NSHK2))/2.
      UPC(1,4)=UPC(2,4)=(U(1,NSHK1+1)/D(1,NSHK1+1)+
     *U(1,NSHK2)/D(1,NSHK2))/2.*DPC(1,4)
      PPC(1,4)=PPC(2,4)=(P(1,NSHK1+1)+P(1,NSHK2))/2.
C-----CALCULATE SHK VEL AND POSITION
 11   MH1=2+NXS1 $ ML1=4 $ MH2=6+NXS2 $ ML2=5+NXS2
      SSL1=SSLS(PPC(1,MH1),PPC(1,ML1),GE)
      SSL2=-SSLS(PPC(1,MH2),PPC(1,ML2),GE)
      SSE1=SSES(SSL1,GE,1,ML1) $ SSE2=SSES(SSL2,GE,1,ML2)
      RD(N1)=RD(N1)+(SE(N1)+SSE1)/2.*DT
      RD(N2)=RD(N2)+(SE(N2)+SSE2)/2.*DT
      IF(((RD(N1).GT.R2(NSHK1+1)).AND.(NXS1.EQ.0)).OR.((RD(N1).LT.
     *R2(NSHK1+1)).AND.(NXS1.EQ.1))) RD(N1)=R2(NSHK1+1)
      IF(((RD(N2).LT.R2(NSHK2)).AND.(NXS2.EQ.0)).OR.((RD(N2).GT.R2(NSHK2
     *)).AND.(NXS2.EQ.-1))) RD(N2)=R2(NSHK2)
      IF(NXS1.NE.1) GO TO 12
C-----CORRECTOR FOR NSHK1+1 = 3
      M=NSHK1+1 $ SSL=(SL(N1)+SSL1)/2.
      D(2,M)=DPC(2,3)=D2M(SSL,GE,M) $ P(2,M)=PPC(2,3)=P2M(SSL,GE,M)
      U(2,M)=UPC(2,3)=U2M(SSL,GE,M)*DPC(2,3) $ E(2,M)=E2M(GE,2,3)
 12   IF(NXS2.NE.-1) GO TO 13
C-----CORRECTOR FOR NSHK2 = 5
      M=NSHK2 $ SSL=(SL(N2)+SSL2)/2.
      D(2,M)=DPC(2,5)=D2M(SSL,GE,M) $ P(2,M)=PPC(2,5)=P2M(SSL,GE,M)
      U(2,M)=UPC(2,5)=U2M(SSL,GE,M)*DPC(2,5) $ E(2,M)=E2M(GE,2,5)
      GO TO 20
C-----CORRECTOR FOR NSHK2+1 = 6
 13   M=NSHK2+1 $ SSL=(SL(N2)+SSL2)/2.
      DPC(1,9)=D2M(SSL,GE,NSHK2) $ PPC(1,9)=P2M(SSL,GE,NSHK2)
      UPC(1,9)=U2M(SSL,GE,NSHK2)*DPC(1,9) $ EPC(1,9)=E2M(GE,1,9)
      IF(NSC.NE.0) GO TO 16
      DPC(1,9)=DPC(1,9)/D(1,NSHK2)*DPC(1,4)
      UPC(1,9)=DPC(1,9)*(UPC(1,4)/DPC(1,4)+SQRT(GE*PPC(1,4)/DPC(1,4))*
```

```
     *2./(GE+1.)*SSL*(1.-1./SSL**2))
      PPC(1,9)=PPC(1,9)/P(1,NSHK2)*PPC(1,4)
      EPC(1,9)=E2M(GE,1,9)
 16   EPS=(R2(M)-RD(N2))/DELR
      C1=2.*(2.-EPS)/(1.+EPS)
      C2=2.*EPS-3. $ C3=(1.-EPS)*(2.*EPS-1.)/(1.+EPS)
      D(2,M)=DPC(2,6)=(D(1,M)+DPC(1,6)+DTDX*(C1*UPC(1,9)+C2*UPC(1,6)+
     *C3*UPC(1,7))+AT*DTDX*(C1*DPC(1,9)+C2*DPC(1,6)+C3*DPC(1,7)))/2.
      U(2,M)=UPC(2,6)=(U(1,M)+UPC(1,6)+DTDX*(C1*CM(9)+C2*CM(6)+
     *C3*CM(7))+AT*DTDX*(C1*UPC(1,9)+C2*UPC(1,6)+C3*UPC(1,7)))/2.
      E(2,M)=EPC(2,6)=(E(1,M)+EPC(1,6)+DTDX*(C1*CE(9)+C2*CE(6)+
     *C3*CE(7))+AT*DTDX*(C1*EPC(1,9)+C2*EPC(1,6)+C3*EPC(1,7)))/2.
      P(2,M)=PPC(2,6)=PC(6,GE)
C-----CALC FINAL SHK SPEED
 20   SL(N1)=SSLS(PPC(2,MH1),PPC(2,ML1),GE)
      SL(N2)=-SSLS(PPC(2,MH2),PPC(2,ML2),GE)
      SE(N1)=SSES(SL(N1),GE,2,ML1) $ SE(N2)=SSES(SL(N2),GE,2,ML2)
      NDISCL(N1)=NDISCL(N1)+NXS1 $ NDISCL(N2)=NDISCL(N2)+NXS2
C
      IF(NSKSKX.EQ.3HYES) CALL SKSKFNA(N1,N2,TRFLCT,RRFLCT)
C
      RETURN $ END
```

```
      SUBROUTINE SKSKRNA(N2,N3,TRFLCT,RRFLCT)
      COMMON/PARAM/N,J,AJ,G,GF,DFLR,NFLM
      COMMON/TIME/T,DT,DTL,TWRITE,DELT,DTDX,AT
      COMMON/FIRFETC/INDEX,NCYCLE,NN,NNN,NSTORE,NS,NITRCTN
      COMMON/DISCSKF/RDSAV(51),NLHS(51),NRHS(51),NCROSS(51)
      COMMON/ARRAYS/R(501),U(2,501),P(2,501),D(2,501),E(2,501),R2(501)
      COMMON/DISCS/NTDISC,NDISCNC(51),NTYPE(51),NDISCL(51),SE(51),SL(51)
     *,RD(51),NTDISCT,NTDISCS
      COMMON/PPCHR/NPPTH,NUMA,NUMAFST,NUPA,RPPTH(24),RUMA(150),RUPA(150)
     *,NCLPPTH(24),NCLUMA(150),NCLUPA(150),PTHNEXT,RUMANXT,TUPANXT,
     *DELPPTH,DELUMA,DELUPA
C
C-----SET STATE 0 AS POLAR AND OVERALL REFERENCE STATE
C-----VEL REF WRT SQRT(G*P0/D0)
C
C      ---2-SHK---5-CD-4---SHK--3---
C
C      ---2--SHK-----1-----SHK--3---
C
C
      GE=G $ IF(NDISCL(N3).LE.NFLM) GE=GF
C-----SET STATES 2 AND 3
      NS2=NDISCL(N2) $ NS3=NDISCL(N3)+1 $ SG=SQRT(G)
      P20=P(2,NS2) $ D20=D(2,NS2) $ U20=U(2,NS2)/D20/SG
      P30=P(2,NS3) $ D30=D(2,NS3) $ U30=U(2,NS3)/D30/SG
      A20=SQRT(P20/D20*GE/G) $ A30=SQRT(P30/D30*GE/G)
C-----GUESS P50=P40 AS AVG OF P50 AND P40 AT A30 FOR U=(U20+U30)/2
      BETA=(GE-1.)/(GE+1.)
      XI=((U30-U20)/2./SG)**2/A30**2*GE/(1.-BETA)
      P43=(2.+XI+SQRT(XI**2+4.*XI*(1.+BETA)))/2.
      XI=((U20-U30)/2./SG)**2/A20**2*GE/(1.-BETA)
      P52=(2.+XI+SQRT(XI**2+4.*XI*(1.+BETA)))/2.
      P50=P40=(P43*P30+P52*P20)/2.
C-----SET SUBROUTINE CONSTS AND ITER COUNTER
      NSIGN=-1 $ EPS=1.E-14 $ NITER=0
      DELP540=P50/5.0 $ P50=P50-DELP540
C
C-----PERFORM ITER OF THE POLAR SOLN
      PRINT 1002,SL(N2),SL(N3),N2,N3,NDISCL(N2),NDISCL(N3)
 1002 FORMAT(1H1,2/,5X,*SHK(P)-SHK(N) INTERATION   MACH NOS.= *,2F11.7,
     *5X,4I5,2/)
 1    NITER=NITER+1
      IF(NITER.LT.50) GO TO 2
      IF(NITER.EQ.50) DELUSAV=DELU
      IF(NITER.LT.75) GO TO 2
      IF((DELUSAV.LT.1.E-12).AND.(DELU.LT.1.E-12)) GO TO 10
      PRINT 1000,DELUSAV,DELU
 1000 FORMAT(1H ,5/,5X,*ITERTNS HAS RCHD MX IN SHK(P)-SHK(N)-A*,2E10.10)
      IF(DELU.LT.1.E-10) GO TO 10
      DELP=-ABS(DELR) $ RETURN
 2    P50=P40=P40+DELP540
      P43=P40/P30
      SL43=SQRT((P43+BETA)/(1.+BETA)) $ SE40=SL43*A30+U30
      D43=1./(BETA+(1.-BETA)/SL43**2) $ D40=D43*D30
      U430=(1.-BETA)*(1.-1./SL43**2)*SL43*A30 $ U40=U430+U30
      A43=SQRT((1.-BETA)/(1.+BETA)*(1.+BETA+BETA*(P43-1.))/(P43+BETA)*GE
```

```
     **P43)
      A40=A43*A30
      P52=P50/P20
      SL52=-SQRT((P52+BETA)/(1.+BETA)) $ SE50=SL52*A20+U20
      D52=1./(BETA+(1.-BETA)/SL52**2) $ D50=D52*D20
      U520=(1.-BETA)*(1.-1./SL52**2)*SL52*A20 $ U50=U520+U20
      A52=SQRT((1.-BETA)/(1.+BETA)*(1.+BETA+BETA*(P52-1.))/(P52+BETA)*GE
     **P52)
      A50=A52*A20
C
      DELU=ABS(U40-U50)
C
      PRINT 1001,NITER,DELP540,DELU,
     *P20,D20,A20,U20,SL(N2),SE(N2),
     *P30,D30,A30,U30,SL(N3),SE(N3),
     *P43,P40,D43,D40,A43,A40,U430,U40,SL43,SE40,
     *P52,P50,D52,D50,A52,A50,U520,U50,SL52,SE50
 1001 FORMAT(1H ,1X,I5,2E15.6,2(/,1X,4(12X,F12.9),2F12.9),2(/,1X,10F12.9
     *),/)
C
      IF(DELU.LE.EPS) GO TO 10
C
      IF((U50.LT.U40).AND.(NSIGN.LE.0)) DELP540=-ABS(DELP540)
      IF((U50.LT.U40).AND.(NSIGN.EQ.-1)) NSIGN=0
      IF((U50.LT.U40).AND.(NSIGN.EQ.1)) DELP540=-ABS(DELP540)/2.0
      IF((U40.LT.U50).AND.(NSIGN.EQ.-1)) DELP540=ABS(DELP540)
      IF((U40.LT.U50).AND.(NSIGN.GE.0)) DELP540=ABS(DELP540)/2.0
      IF((U40.LT.U50).AND.(NSIGN.EQ.0)) NSIGN=1
      GO TO 1
C
C
 10   NT=NTDISCT
      NDISCND(NT+1)=NTDISC+1 $ NTYPE(NT+1)=2
      SL(N2)=SL43 $ SL(NT+1)=0.0 $ SL(N3)=SL52
      SE(N2)=SE40*SG $ SE(NT+1)=U40*SG $ SE(N3)=SE50*SG
      RD(N2)=RRFLCT+SE(N2)*(DT-TRFLCT)
      RD(N3)=RRFLCT+SE(N3)*(DT-TRFLCT)
      RD(NT+1)=RRFLCT+SE(NT+1)*(DT-TRFLCT)
      IF(RD(N3).LT.R2(NS2)) RD(N3)=R2(NS2)
      IF(RD(N2).GT.R2(NS3)) RD(N2)=R2(NS3)
      IF(RD(NT+1).LE.R2(NS2)) RD(NT+1)=R2(NS2)+1.E-8*DELR
      IF(RD(NT+1).GE.R2(NS3)) RD(NT+1)=R2(NS3)-1.E-8*DELR
      DO 11 I=NS3,N
      M=N+2+NS3-I
      D(2,M)=D(2,M-2) $ U(2,M)=U(2,M-2) $E(2,M)=E(2,M-2)$P(2,M)=P(2,M-2)
      D(1,M)=D(1,M-2) $ U(1,M)=U(1,M-2) $E(1,M)=E(1,M-2)$P(1,M)=P(1,M-2)
      R(M)=R(M-2)
 11   R2(M)=R2(M-2)
      N=N+2
      R2(NS3)=R2(NS3+1)=RD(NT+1)
      DO 12 I=1,NTDISCT
      IF(NDISCL(I).GE.NS3) NDISCL(I)=NDISCL(I)+2
 12   CONTINUE
      IF(NFLM.GE.NS3) NFLM=NFLM+2
      IF(NS3.LE.NTDISCS) NRHS(NS3)=NRHS(NS3)+2
      IF(NS3.LE.NTDISCS) NLHS(NS3)=NLHS(NS3)+2
```

```
      NDISCL(N3)=NS2 $ NDISCL(NT+1)=NS2+1 $ NDISCL(N2)=NS2+2
C------SET THERMO AND GAS PARAMETERS IN CELLS NS2+1,2
      P(2,NS2+1)=P(2,NS2+2)=P50 $ D(2,NS2+1)=D50 $ D(2,NS2+2)=D40
      U(2,NS2+1)=U50=U50*SG*D50 $ U(2,NS2+2)=U40=U40*SG*D40
      E(2,NS2+1)=D50*(P50/D50/(GE-1.)+U50**2/D50**2/2.)
      E(2,NS2+2)=D40*(P40/D40/(GE-1.)+U40**2/D40**2/2.)
      NTDISCT=NTDISCT+1 $ NTDISC=NTDISC+1
C
C-----ACCT FOR PRTCLE PTH AND NEG AND POS CHRCT TRAJ CELL LOCTNS CHANGES
      IF(NPPTH.LE.0) GO TO 110
      DO 111 I=1,NPPTH
      IF(NCLPPTH(I).GE.NS3) NCLPPTH(I)=NCLPPTH(I)+2
 111  CONTINUE
 110  IF(NUMA.LE.0) GO TO 112
      DO 113 I=NUMAFST,NUMA
      IF(NCLUMA(I).GE.NS3) NCLUMA(I)=NCLUMA(I)+2
 113  CONTINUE
 112  IF(NUPA.LE.0) GO TO 114
      DO 115 I=1,NUPA
      IF(NCLUPA(I).GE.NS3) NCLUPA(I)=NCLUPA(I)+2
 115  CONTINUE
C
 114  NCODE=10HSKSKFNA    $ CALL PRNTFF(NCODE)
      NITRCTN=3HYES
C
      RETURN $ END
```

```
      SUBROUTINE SKSKSK(N1,N2,N3)
      COMMON/PARAM/N,J,AJ,G,GF,DELR,NFLM
      COMMON/DISCSKF/RDSAV(51),NLHS(51),NRHS(51),NCROSS(51)
      COMMON/PREDCOR/DPC(2,13),UPC(2,13),EPC(2,13),PPC(2,13)
      COMMON/ARRAYS/R(501),U(2,501),F(2,501),D(2,501),E(2,501),R2(501)
      COMMON/DISCS/NTDISC,NDISCNC(51),NTYPE(51),NDISCL(51),SE(51),SL(51)
     *,RD(51),NTDISCT,NTDISCS
C
      PRINT 1000,N1,N2,N3,RD(N1),RD(N2),RD(N3),NDISCL(N1),NDISCL(N2),
     *NDISCL(N3),NCROSS(N1),NCROSS(N2),NCROSS(N3),NRHS(N1),NRHS(N2),
     *NRHS(N3)
      IF(NCROSS(N2)) 1,2,3
C
 2    NADD=0
      IF((NDISCL(N2)-NDISCL(N1).EQ.3).OR.((NDISCL(N2)-NDISCL(N1).EQ.2)
     *.AND.(NCROSS(N1).EQ.0))) NADD=1
      CALL SKSKPP(N1,N2,N3,7HSKSKSKO)
C-----PREDICTOR AND CORRECTOR FOR NODE 7 = (1)
      DPC(1,7)=DPC(1,1+NADD) $ UPC(1,7)=UPC(1,1+NADD)
      EPC(1,7)=EPC(1,1+NADD) $ PPC(1,7)=PPC(1,1+NADD)
      DPC(2,7)=DPC(2,1+NADD) $ UPC(2,7)=UPC(2,1+NADD)
      EPC(2,7)=EPC(2,1+NADD) $ PPC(2,7)=PPC(2,1+NADD)
      CALL SKSKPP(N2,N3,N1,7H SKSKSK)
      GO TO 20
C
 1    IF((NDISCL(N1).EQ.NDISCL(N2)).AND.(NCROSS(N1).NE.-1)) GO TO 2
      IF((NDISCL(N1).EQ.NDISCL(N2)-1).AND.(NCROSS(N1).EQ.1)) GO TO 11
      GO TO 13
 11   TRFLCT=(RD(N2)-RD(N1))/(SE(N1)-SE(N2))
      M=NDISCL(N2)
      IF(RD(N2)+SE(N2)*TRFLCT.LT.R2(M)) GO TO 12
      NCROSS(N2)=0 $ NRHS(N2)=NRHS(N2)+1 $ GO TO 2
 12   NCROSS(N1)=0 $ NRHS(N1)=NRHS(N1)-1
 13   CALL SKSKPP(N1,N2,N3,7HSKSKSKM)
C-----PREDICTOR AND CORRECTOR FOR NODE 8 = (3)
      DPC(1,8)=DPC(1,3) $ UPC(1,8)=UPC(1,3)
      EPC(1,8)=EPC(1,3) $ PPC(1,8)=PPC(1,3)
      DPC(2,8)=DPC(2,3) $ UPC(2,8)=UPC(2,3)
      EPC(2,8)=EPC(2,3) $ PPC(2,8)=PPC(2,3)
      NDISCL(N2)=NDISCL(N2)-NCROSS(N2)
      CALL SKSKPP(N2,N3,N1,7H SKSKSK)
      NDISCL(N2)=NDISCL(N2)+NCROSS(N2)
      GO TO 20
C
 3    IF((NDISCL(N2).EQ.NDISCL(N3)).AND.(NCROSS(N3).NE.1)) GO TO 2
      IF((NDISCL(N2).EQ.NDISCL(N3)-1).AND.(NCROSS(N3).EQ.-1)) GO TO 5
      GO TO 6
 5    TRFLCT=(RD(N3)-RD(N2))/(SE(N2)-SE(N3))
      M=NDISCL(N3)
      IF(RD(N3)+TRFLCT*SE(N3).GT.R2(M)) GO TO 4
      NCROSS(N2)=0 $ NRHS(N2)=NRHS(N2)-1 $ GO TO 2
 4    NCROSS(N3)=0 $ NRHS(N3)=NRHS(N3)+1
 6    CALL SKSKPP(N2,N3,N1,7HSKSKSKF)
      NDISCL(N2)=NDISCL(N2)-NCROSS(N2) $NDISCL(N3)=NDISCL(N3)-NCROSS(N3)
      IF((NDISCL(N2)-NDISCL(N1).EQ.3).OR.((NDISCL(N2)-NDISCL(N1).EQ.2)
     *.AND.(NCROSS(N1).EQ.0))) GO TO 7
```

```
      GO TO 8
C-----PREDICTOR FOR NODE 1 = (6)
 7    DPC(1,1)=DPC(1,6) $ UPC(1,1)=UPC(1,6)
      EPC(1,1)=EPC(1,6) $ PPC(1,1)=PPC(1,6)
C-----PREDICTOR FOR NODE 2 = (7)
      DPC(1,2)=DPC(1,7) $ UPC(1,2)=UPC(1,7)
      EPC(1,2)=EPC(1,7) $ PPC(1,2)=PPC(1,7)
      GO TO 10
C-----PREDICTOR AND CORRECTOR FOR NODES 2,1 = (8)
 8    DPC(1,1)=DPC(1,2)=DPC(1,8) $ UPC(1,1)=UPC(1,2)=UPC(1,8)
      EPC(1,1)=EPC(1,2)=EPC(1,8) $ PPC(1,1)=PPC(1,2)=PPC(1,8)
      DPC(2,1)=DPC(2,2)=DPC(2,8) $ UPC(2,1)=UPC(2,2)=UPC(2,8)
      EPC(2,1)=EPC(2,2)=EPC(2,8) $ PPC(2,1)=PPC(2,2)=PPC(2,8)
      IF((NDISCL(N2)-NDISCL(N1).EQ.2).AND.(NCROSS(N1).EQ.1)) GO TO 9
      GO TO 10
C-----PREDICTOR FOR NODE 1 = (7)
 9    DPC(1,1)=DPC(1,7) $ UPC(1,1)=UPC(1,7)
      EPC(1,1)=EPC(1,7) $ PPC(1,1)=PPC(1,7)
 10   CALL SKSKPP(N1,N2,N3,7HSPECIAL)
      NDISCL(N2)=NDISCL(N2)+NCROSS(N2) $NDISCL(N3)=NDISCL(N3)+NCROSS(N3)
C
C
 20   CONTINUE
      PRINT 1000,N1,N2,N3,RD(N1),RD(N2),RD(N3),NDISCL(N1),NDISCL(N2),
     *NDISCL(N3),NCROSS(N1),NCROSS(N2),NCROSS(N3),NRHS(N1),NRHS(N2),
     *NRHS(N3)
 1000 FORMAT(1H ,1X,3I5,3F12.8,9I5)
      IF((RD(N1).LT.RD(N2)).AND.(RD(N2).LT.RD(N3)))GO TO 21
      IF((RD(N1).GT.RD(N2)).AND.(RD(N2).GT.RD(N3)))GO TO 31
      IF((RD(N1).LT.RD(N3)).AND.(RD(N2).GT.RD(N3)))GO TO 32
      IF((RD(N1).GT.RD(N2)).AND.(RD(N1).GT.RD(N3)))GO TO 31
      IF((RD(N1).GT.RD(N2)).AND.(RD(N1).LT.RD(N3)))GO TO 34
      IF((RD(N1).GT.RD(N3)).AND.(RD(N1).LT.RD(N2)))GO TO 35
      GO TO 21
 31   NXS=N3
      IF((RD(N1)-RD(N2))/(SE(N1)-SE(N2)).LT.(RD(N1)-RD(N3))/(SE(N1)-
     *SE(N3))) NXS=N2
      TARFLCT=(RD(N1)-RD(NXS))/(SE(N1)-SE(NXS))
      RD(N1)=RD(N2)=RD(N3)=RD(N1)-TARFLCT*SE(N1) $ GO TO 22
 32   RD(N3)=RD(N2)+DELR*.5E-10 $ GO TO 21
 34   RD(N2)=RD(N1)+DELR*.5E-10 $ GO TO 21
 35   RD(N3)=RD(N2)+DELR*.5E-10 $ GO TO 21
C
 22   PRINT 1001,N1,N2,N3,NDISCL(N1),NDISCL(N2),NDISCL(N3),NCROSS(N1),
     *NCROSS(N2),NCROSS(N3),NRHS(N1),NRHS(N2),NRHS(N3),RD(N1),RD(N2),
     *RD(N3),SL(N1),SL(N2),SL(N3),SE(N1),SE(N2),SE(N3)
 1001 FORMAT(1H ,5/,* SHK-SHK-SHK INTERACTION*,/,1X,12I5,/,1X,9F12.8)
      CALL SKSKPPA(N1,N3,6HSKSKSK,TARFLCT)
C
C
 21   RETURN $ END
```

```
      SUBROUTINE DTSKSK(N1,N2,N3)
      COMMON/PARAM/N,J,AJ,G,GF,DELR,NFLM
      COMMON/DISCSKF/RDSAV(51),NLHS(51),NRHS(51),NCROSS(51)
      COMMON/PREDCOR/DPC(2,13),UPC(2,13),EPC(2,13),PPC(2,13)
      COMMON/ARRAYS/R(501),U(2,501),P(2,501),D(2,501),E(2,501),R2(501)
      COMMON/DISCS/NTDISC,NDISCNC(51),NTYPE(51),NDISCL(51),SF(51),SL(51)
     *,RD(51),NTDISCT,NTDISCS
C
C
      PRINT 1000,N1,N2,N3,RD(N1),RD(N2),RD(N3),NDISCL(N1),NDISCL(N2),
     *NDISCL(N3),NCROSS(N1),NCROSS(N2),NCROSS(N3),NRHS(N1),NRHS(N2),
     *NRHS(N3),NTYPE(N1),NTYPE(N2),NTYPE(N3)
      IF(NTYPE(N3).EQ.5) GO TO 36
      IF(NTYPE(N2).EQ.5) GO TO 37
      IF((NDISCL(N2).NE.NDISCL(N3)).OR.(RDSAV(N2).LT.RDSAV(N3)).OR.
     *(RDSAV(N1).GT.RDSAV(N2)).OR.(NCROSS(N2).EQ.0).OR.(NCROSS(N3).EQ.
     *1)) GO TO 36
      IF(N3.EQ.NTDISCT) GO TO 38
      IF(NRHS(N2).GT.NDISCL(N3+1)) GO TO 36
      IF((NRHS(N2).EQ.NDISCL(N3+1)).AND.(NTYPE(N3+1).NE.2)) GO TO 36
 38   NCROSS(N3)=NCROSS(N2) $ NRHS(N3)=NRHS(N2)
      GO TO 36
 37   IF(((RDSAV(N1).GT.RDSAV(N3)).AND.(RDSAV(N2).GT.RDSAV(N3))).OR.
     *(NCROSS(N2).EQ.1).OR.(NCROSS(N1).EQ.0).OR.(RDSAV(N1).LT.RDSAV(N2))
     *.OR.(NDISCL(N1).NE.NDISCL(N2))) GO TO 36
      NCROSS(N2)=NCROSS(N1) $ NRHS(N2)=NRHS(N1)
C
 36   IF(NCROSS(N2).EQ.1) GO TO 3
C
 2    CALL DSORSD(N1,N2,N3,7HDTSKSKC)
C-----PREDICTOR AND CORRECTOR FOR NODE 7 = (1)
      M=1
      IF((NDISCL(N2)-NDISCL(N1).EQ.3).OR.((NDISCL(N2)-NDISCL(N1).EQ.2)
     *.AND.(NCROSS(N1).EQ.0))) M=2
      DPC(1,7)=DPC(1,M) $ UPC(1,7)=UPC(1,M)
      EPC(1,7)=EPC(1,M) $ PPC(1,7)=PPC(1,M)
      DPC(2,7)=DPC(2,M) $ UPC(2,7)=UPC(2,M)
      EPC(2,7)=EPC(2,M) $ PPC(2,7)=PPC(2,M)
      CALL DSORSD(N2,N3,N1,7H DTSKSK)
      GO TO 20
C
 3    IF((NDISCL(N2).EQ.NDISCL(N3)).AND.(NCROSS(N3).NE.1)) GO TO 2
      CALL DSORSD(N2,N3,N1,7HDTSKSKP)
      NDISCL(N2)=NDISCL(N2)-NCROSS(N2) $NDISCL(N3)=NDISCL(N3)-NCROSS(N3)
      IF(NTYPE(N2).EQ.5) NFLM=NDISCL(N2)
      IF(NTYPE(N3).EQ.5) NFLM=NDISCL(N3)
      IF((NDISCL(N2)-NDISCL(N1).EQ.3).OR.((NDISCL(N2)-NDISCL(N1).EQ.2)
     *.AND.(NCROSS(N1).EQ.0))) GO TO 7
      GO TO 8
C-----PREDICTOR FOR NODE 1 = (6)
 7    DPC(1,1)=DPC(1,6) $ UPC(1,1)=UPC(1,6)
      EPC(1,1)=EPC(1,6) $ PPC(1,1)=PPC(1,6)
C-----PREDICTOR FOR NODE 2 = (7)
      DPC(1,2)=DPC(1,7) $ UPC(1,2)=UPC(1,7)
      EPC(1,2)=EPC(1,7) $ PPC(1,2)=PPC(1,7)
      GO TO 10
```

```
C-----PREDICTOR AND CORRECTOR FOR NODES 2,1 = (8)
 8    DPC(1,1)=DPC(1,2)=DPC(1,8) $ UPC(1,1)=UPC(1,2)=UPC(1,8)
      EPC(1,1)=EPC(1,2)=EPC(1,8) $ PPC(1,1)=PPC(1,2)=PPC(1,8)
      DPC(2,1)=DPC(2,2)=DPC(2,8) $ UPC(2,1)=UPC(2,2)=UPC(2,8)
      EPC(2,1)=EPC(2,2)=EPC(2,8) $ PPC(2,1)=PPC(2,2)=PPC(2,8)
      IF((NDISCL(N2)-NDISCL(N1).EQ.2).AND.(NCROSS(N1).EQ.1)) GO TO 9
      GO TO 10
C-----PREDICTOR FOR NODE 1 = (7)
 9    DPC(1,1)=DPC(1,7) $ UPC(1,1)=UPC(1,7)
      EPC(1,1)=EPC(1,7) $ PPC(1,1)=PPC(1,7)
 10   CALL DSORSD(N1,N2,N3,7HSPECIAL)
      NDISCL(N2)=NDISCL(N2)+NCROSS(N2) $NDISCL(N3)=NDISCL(N3)+NCROSS(N3)
      IF(NTYPE(N2).EQ.5) NFLM=NDISCL(N2)
      IF(NTYPE(N3).EQ.5) NFLM=NDISCL(N3)
C
C
 20   PRINT 1000,N1,N2,N3,RD(N1),RD(N2),RD(N3),NDISCL(N1),NDISCL(N2),
     *NDISCL(N3),NCROSS(N1),NCROSS(N2),NCROSS(N3),NRHS(N1),NRHS(N2),
     *NRHS(N3),NTYPE(N1),NTYPE(N2),NTYPE(N3)
 1000 FORMAT(1H ,1X,3I5,3F12.8,12I5)
      IF((RD(N1).LT.RD(N2)).AND.(RD(N2).LT.RD(N3)))GO TO 21
      IF((RD(N1).GT.RD(N2)).AND.(RD(N2).GT.RD(N3)))GO TO 21
      IF((RD(N1).LT.RD(N3)).AND.(RD(N2).GT.RD(N3)))GO TO 32
      IF((RD(N1).GT.RD(N2)).AND.(RD(N1).GT.RD(N3)))GO TO 31
      IF((RD(N1).GT.RD(N2)).AND.(RD(N1).LT.RD(N3)))GO TO 34
      IF((RD(N1).GT.RD(N3)).AND.(RD(N1).LT.RD(N2)))GO TO 35
      GO TO 21
 31   NXS=N3
      IF((RD(N1)-RD(N2))/(SE(N1)-SE(N2)).LT.(RD(N1)-RD(N3))/(SE(N1)-
     *SE(N3))) NXS=N2
      TARFLCT=(RD(N1)-RD(NXS))/(SE(N1)-SE(NXS))
      RD(N1)=RD(N2)=RD(N3)=RD(N1)-TARFLCT*SE(N1) $ GO TO 22
 32   RD(N3)=RD(N2)+DELR*.5E-10 $ GO TO 21
 34   RD(N2)=RD(N1)+DELR*.5E-10 $ GO TO 21
 35   RD(N3)=RD(N2)+DELR*.5E-10 $ GO TO 21
C
 22   PRINT 1001,N1,N2,N3,NDISCL(N1),NDISCL(N2),NDISCL(N3),NCROSS(N1),
     *NCROSS(N2),NCROSS(N3),NRHS(N1),NRHS(N2),NRHS(N3),RD(N1),RD(N2),
     *RD(N3),SL(N1),SL(N2),SL(N3),SE(N1),SE(N2),SE(N3),NTYPE(N1),
     *NTYPE(N2),NTYPE(N3)
 1001 FORMAT(1H ,5/,* DET-SHK-SHK INTERACTION*,/,1X,12I5,/,1X,9F12.8,3I5
     *)
      CALL DTSKSKA(N1,N3,6HDTSKSK,TARFLCT)
C
C
 21   RETURN $ END
```

```
      SUBROUTINE DISKSKA(II,NI,NAME,IAFFLCT)
      COMMON/PARAM/N,J,AJ,G,GF,DELR,NFLM
C
      DELR=-ABS(DELR)
      PRINT 1000
 1000 FORMAT(1H ,2X,*DISKSKA STOP*)
C
      RETURN $ END
```

```
      SUBROUTINE SHKCD(NFIRST,NI,NAME)
      COMMON/UBDCND/NUBC
      COMMON/SCS/SECDSCS,RCDSCS,NXCDSCS
      COMMON/PARAM/N,J,AJ,G,GF,DELR,NFLM
      COMMON/TIME/T,DT,DTL,TWRITE,DELT,DTDX,AT
      COMMON/DISCSKF/RDSAV(51),NLHS(51),NRHS(51),NCROSS(51)
      COMMON/PREDCOR/DPC(2,13),UPC(2,13),EPC(2,13),PPC(2,13)
      COMMON/POWER/VCAPF,R4R1,PCPOWER,PPOWER,CND,CCAPF,SFEOLD,SFE,NCJ
      COMMON/ARRAYS/R(501),U(2,501),P(2,501),D(2,501),E(2,501),R2(501)
      COMMON/DISCS/NTDISC,NDISCNO(51),NTYPE(51),NDISCL(51),SE(51),SL(51)
     *,RD(51),NTDISCT,NTDISCS
      COMMON/PPCHR/NPPTH,NUMA,NUMAFST,NUPA,RPPTH(24),RUMA(150),RUPA(150)
     *,NCLPPTH(24),NCLUMA(150),NCLUPA(150),PTHNEXT,RUMANXT,TUPANXT,
     *DELPPTH,DELUMA,DELUPA
C
C
      RS=RD(NFIRST) $ SSL=SL(NFIRST) $ SSE=SE(NFIRST)
      NSHK=NDISCL(NFIRST) $ RCD=RD(NI) $ SECD=SE(NI) $ NCD=NDISCL(NI)
C
      GE=G $ IF(NCD.LE.NFLM) GE=GF $ GG=GE $ IF(NTYPE(NFIRST).EQ.5)GG=GF
      NSHKSGN=0 $ IF(SSL.GT.0.0) NSHKSGN=1
      SECDSAV=SECD $ SSESAV=SSE
      RSSAV=RS $ RS=RS+SSE*DT $ RCDSAV=RCD $ RCD=RCD+SECD*DT
      NXSS=NXSCD=0 $ MSC=0 $ IF(NSHK+1.EQ.NCD) MSC=2
      IF(RS.GT.R2(NSHK+1+MSC)) NXSS=1 $ IF(RS.LT.R2(NSHK)) NXSS=-1
      IF(RCD.GT.R2(NCD+2)) NXSCD=1 $ IF(RCD.LT.R2(NCD-1)) NXSCD=-1
      IF(NAME.EQ.8HSHKCDSHK) NXSCD=NXCDSCS
      NSHKCDX=0 $ IF(RS.GT.RCD) NSHKCDX=1
      NSC=100*(NCD-1-NSHK)+10*NXSS+NXSCD
      PRINT 9999,NAME,NXSS,NXSCD,NSHKCDX,NSC
 9999 FORMAT(1H ,1X,A8,4I5)
C
      IF(NSC.NE.10) GO TO 40
      NSC=NXSS=0
C
 40   IF(((NAME.EQ.8H    ZERO).OR.(NAME.EQ.8HSHKCDRAR).OR.(NAME.EQ.
     *8HRARCD   )).AND.(NSHKCDX.EQ.1).AND.(NFIRST.EQ.1).AND.
     *(SL(NFIRST).LT.1.001)) GO TO 100
      IF(NAME.EQ.8HSHKCDSHK) SECD=SECDSCS
      IF(NAME.EQ.8HSHKCDSHK) RCD=RCDSCS
C-----PREDICTOR FOR NSHK-1-NXSS*(NXSS-1)/2 = 1
      M=NSHK-1-NXSS*(NXSS-1)/2
      DPC(1,1)=DP(M,0) $ UPC(1,1)=UP(M,0)
      EPC(1,1)=EP(M,0) $ PPC(1,1)=PP(1,GG)
      IF(NXSS.NE.-1) GO TO 1
C-----PREDICTOR FOR NSHK-1 = 2
      M=NSHK-1
      DPC(1,2)=DP(M,0) $ UPC(1,2)=UP(M,0)
      EPC(1,2)=EP(M,0) $ PPC(1,2)=PP(2,GE)
      GO TO 2
 1    IF(NSHKSGN.EQ.1) GO TO 3
C-----PREDICTOR FOR NSHK = 2
      DPC(1,2)=DP(NSHK,-1) $ UPC(1,2)=UP(NSHK,-1)
      EPC(1,2)=EP(NSHK,-1) $ PPC(1,2)=PP(2,GE)
      GO TO 2
C-----PREDICTOR FOR NSHK = 2
```

```
3     IF(NTYPE(NFIRST).EQ.5) GO TO 200
      PPC(1,13)=P(1,NSHK+1)*(2.*GE/(GE+1.)*SSL**2-(GE-1.)/(GE+1.))
      DPC(1,13)=D(1,NSHK+1)/((GE-1.)/(GE+1.)+2./(GE+1.)/SSL**2)
      UPC(1,13)=DPC(1,13)*(U(1,NSHK+1)/D(1,NSHK+1)+SQRT(GE*P(1,NSHK+1)/
     *D(1,NSHK+1))*2./(GE+1.)*SSL*(1.-1./SSL**2))
      EPC(1,13)=DPC(1,13)*(PPC(1,13)/DPC(1,13)/(GE-1.)+UPC(1,13)**2/
     *DPC(1,13)**2/2.)
      GO TO 201
200   BETA=(GF-1.)/(GF+1.) $ B=(GF-1.)/(G-1.)
      PG0=(1.+BETA)*(VCAPF-BETA)/(1.-BETA)-BETA
      A=SQRT(P(1,NSHK+1)/D(1,NSHK+1))
      PG=1.+(PG0-1.)/A**2+GF*(1.-1./A**2)*(G/GF*B-R4R1)
      PS1=(1.-BETA)/2.*(1.+G*SSL**2) $ PPC(1,13)=P(1,NSHK+1)*PS1
      DPC(1,13)=D(1,NSHK+1)/(1.-(1.-BETA)*(PS1-PG)/(PS1+BETA))
      UPC(1,13)=(SQRT(G)*A*SQRT((1.-BETA)*(PS1-PG)*(PS1-1.)/G/
     *(PS1+BETA))+U(1,NSHK+1)/D(1,NSHK+1))*DPC(1,13)
      EPC(1,13)=DPC(1,13)*(PPC(1,13)/DPC(1,13)/(GF-1.)+UPC(1,13)**2/
     *DPC(1,13)**2/2.)
201   EPS=(RSSAV-R(NSHK))/DELR
      C1=2.*(2.-EPS)/(1.+EPS)
      C2=2.*EPS-3. $ C3=(1.-EPS)*(2.*EPS-1.)/(1.+EPS)
      DPC(1,2)=D(1,NSHK)-DTDX*(C1*UPC((1,13)+C2*U(1,NSHK)+C3*U(1,NSHK-1))
     *-AT*DTDX*(C1*DPC(1,13)+C2*D(1,NSHK)+C3*D(1,NSHK-1))
      UPC(1,2)=U(1,NSHK)-DTDX*(C1*CM(13)+C2*PM(NSHK)+C3*PM(NSHK-1))
     *-AT*DTDX*(C1*UPC(1,13)+C2*U(1,NSHK)+C3*U(1,NSHK-1))
      EPC(1,2)=E(1,NSHK)-DTDX*(C1*CE(13)+C2*PE(NSHK)+C3*PE(NSHK-1))
     *-AT*DTDX*(C1*EPC(1,13)+C2*E(1,NSHK)+C3*E(1,NSHK-1))
      PPC(1,2)=PP(2,GG)
C-----CORRECTOR FOR NSHK-NXSS*(NXSS-1)/2 = 2
 2    M=NSHK-NXSS*(NXSS-1)/2
      D(2,M)=DPC(2,2)=DC(2,M,0) $ U(2,M)=UPC(2,2)=UC(2,M,0)
      E(2,M)=EPC(2,2)=EC(2,M,0) $ P(2,M)=PPC(2,2)=PC(2,GG)
      IF(NAME.EQ.8HSHKCDFLM) GO TO 4
      IF(NAME.EQ.8HSHKCDSHK) GO TO 10
C-----PREDICTOR FOR NCD+2+NXSCD*(NXSCD+1)/2 = 8
      M=NCD+2+NXSCD*(NXSCD+1)/2
      DPC(1,8)=DP(M,0) $ UPC(1,8)=UP(M,0)
      EPC(1,8)=EP(M,0) $ PPC(1,8)=PP(8,GE)
      IF(NXSCD.EQ.-1) GO TO 4
C-----PREDICTOR FOR NCD+3+NXSCD = 9
      M=NCD+3+NXSCD
      DPC(1,9)=DP(M,0) $ UPC(1,9)=UP(M,0)
      EPC(1,9)=EP(M,0) $ PPC(1,9)=PP(9,GE)
 4    M=NCD $ IF(NCD-NSHK-1.NE.0) M=NCD-1
      M3=NCD+2 $IF((NAME.EQ.8HSKCD DET).OR.(NAME.EQ.8HSKCDDT   ))M3=NCD+1
      CALL GLIM(M,M3,6,NCD,SECD)
      IF((NAME.NE.8HRARCD    ).AND.(NAME.NE.8HDETRARCD)) GO TO 5
      D(2,NCD)=DPC(1,6)=DPC(2,6)=D(1,NCD)
      U(2,NCD)=UPC(1,6)=UPC(2,6)=U(1,NCD)
      E(2,NCD)=EPC(1,6)=EPC(2,6)=E(1,NCD)
      P(2,NCD)=PPC(1,6)=PPC(2,6)=P(1,NCD)
      GO TO 6
 5    IF(NAME.NE.8HSHKCDRAR) GO TO 6
      D(2,NCD+1)=DPC(1,7)=DPC(2,7)=D(1,NCD+1)
      U(2,NCD+1)=UPC(1,7)=UPC(2,7)=U(1,NCD+1)
      E(2,NCD+1)=EPC(1,7)=EPC(2,7)=E(1,NCD+1)
```

```
      P(2,NCD+1)=PPC(1,7)=PPC(2,7)=P(1,NCD+1)
 6    RCD=RCDSAV+SECD*DT
      IF(((RCD.GT.R2(NCD+2)).AND.(NXSCD.EQ.0)).OR.((RCD.LT.R2(NCD+2)).
     *AND.(NXSCD.EQ.1))) RCD=R2(NCD+2)
      IF(((RCD.LT.R2(NCD-1)).AND.(NXSCD.EQ.0)).OR.((RCD.GT.R2(NCD-1)).
     *AND.(NXSCD.EQ.-1))) RCD=R2(NCD-1)
C
      IF(NXSCD.NE.-1) GO TO 7
C-----PREDICTOR AND CORRECTOR FOR NCD-1 = 5
      M=NCD-1
      D(2,M)=DPC(2,5)=DPC(1,5)=DPC(1,7)
      U(2,M)=UPC(2,5)=UPC(1,5)=UPC(1,7)
      E(2,M)=EPC(2,5)=EPC(1,5)=EPC(1,7)
      P(2,M)=PPC(2,5)=PPC(1,5)=PPC(1,7)
      IF(NAME.EQ.8HSHKCDFLM) GO TO 10
C-----CORRECTOR FOR NCD+2 = 8
      M=NCD+2
      D(2,M)=DPC(2,8)=DC(8,M,0) $ U(2,M)=UPC(2,8)=UC(8,M,0)
      E(2,M)=EPC(2,8)=EC(8,M,0) $ P(2,M)=PPC(2,8)=PC(8,GE)
      GO TO 10
 7    IF(NXSCD.EQ.0) GO TO 8
C-----PREDICTOR AND CORRECTOR FOR NCD+2 = 10(=6)
      M=NCD+2
      D(2,M)=DPC(2,10)=DPC(1,10)=DPC(1,6)
      U(2,M)=UPC(2,10)=UPC(1,10)=UPC(1,6)
      E(2,M)=EPC(2,10)=EPC(1,10)=EPC(1,6)
      P(2,M)=PPC(2,10)=PPC(1,10)=PPC(1,6)
C-----CORRECTOR FOR NCD+2+NXSCD = 8
 8    IF(NAME.EQ.8HSHKCDFLM) GO TO 10
      M=NCD+2+NXSCD
      EPS=(R2(NCD+2)-RCD)/DELR
      C1=2.*(2.-EPS)/(1.+EPS)
      C2=2.*EPS-3. $ C3=(1.-EPS)*(2.*EPS-1.)/(1.+EPS)
      D(2,M)=DPC(2,6)=(D(1,M)+DPC(1,8)+DTDX*(C1*UPC(1,7)+C2*UPC(1,8)
     *+C3*UPC(1,9))+AT*DTDX*(C1*DPC(1,7)+C2*DPC(1,8)+C3*DPC(1,9)))/2.
      U(2,M)=UPC(2,8)=(U(1,M)+UPC(1,8)+DTDX*(C1*CM(7)+C2*CM(8)+C3*CM(9))
     *+AT*DTDX*(C1*UPC(1,7)+C2*UPC(1,8)+C3*UPC(1,9)))/2.
      E(2,M)=EPC(2,8)=(E(1,M)+EPC(1,8)+DTDX*(C1*CE(7)+C2*CE(8)+C3*CE(9))
     *+AT*DTDX*(C1*EPC(1,7)+C2*EPC(1,8)+C3*EPC(1,9)))/2.
      P(2,M)=PPC(2,8)=PC(8,GE)
C
C
 10   IF((NSC.NE.200).AND.(NSC.NE.310).AND.(NSC.NE.201).AND.(NSC.NE.311)
     *.AND.(NSC.NE.199).AND.(NSC.NE.309)) GO TO 20
C-----PREDICTOR AND CORRECTOR FOR NSHK+1+NXSS = 4
      M=NSHK+1+NXSS
      D(2,M)=DPC(2,4)=DPC(1,4)=DP(M,0) $U(2,M)=UPC(2,4)=UPC(1,4)=UP(M,0)
      E(2,M)=EPC(2,4)=EPC(1,4)=EP(M,0)$P(2,M)=PPC(2,4)=PPC(1,4)=PP(4,GE)
      IF((NSC.EQ.199).OR.(NSC.EQ.309)) GO TO 11
C-----PREDICTOR FOR NSHK+2+NXSS = 5
      M=NSHK+2+NXSS
      EPS=(RCDSAV-R(M))/DELR
      C1=2.*(2.-EPS)/(1.+EPS)
      C2=2.*EPS-3. $ C3=(1.-EPS)*(2.*EPS-1.)/(1.+EPS)
      DPC(1,5)=D(1,M)-DTDX*(C1*U(1,NCD)+C2*U(1,M)+C3*U(1,M-1))
     *-AT*DTDX*(C1*D(1,NCD)+C2*D(1,M)+C3*D(1,M-1))
```

```
      UPC(1,5)=U(1,M)-DTDX*(C1*PM(NCD)+C2*PM(M)+C3*PM(M-1))
     *-AT*DTDX*(C1*U(1,NCD)+C2*U(1,M)+C3*U(1,M-1))
      EPC(1,5)=E(1,M)-DTDX*(C1*PE(NCD)+C2*PE(M)+C3*PE(M-1))
     *-AT*DTDX*(C1*E(1,NCD)+C2*E(1,M)+C3*E(1,M-1))
      PPC(1,5)=PP(5,GE)
 11   IF(NTYPE(NFIRST).EQ.5) GO TO 202
      IF(NXSS.EQ.0) GO TO 12
C-----PREDICTOR FOR NSHK+(NXSS+1)/2 = 3
      M=NSHK+(NXSS+1)/2
      PPC(1,3)=P(1,M)*(2.*GE/(GE+1.)*SSL**2-(GE-1.)/(GE+1.))
      DPC(1,3)=D(1,M)/((GE-1.)/(GE+1.)+2./(GE+1.)/SSL**2)
      UPC(1,3)=DPC(1,3)*(U(1,M)/D(1,M)+SQRT(GE*P(1,M)/D(1,M))*2./(GE+1.)
     **SSL*(1.-1./SSL**2))
      EPC(1,3)=DPC(1,3)*(PPC(1,3)/DPC(1,3)/(GE-1.)+UPC(1,3)**2/DPC(1,3)
     ***2/2.)
C-----PREDICTOR FOR SHK VEL AND POS
 12   MH=4-2*NSHKSGN+NXSS $ ML=2+2*NSHKSGN
      SSLSAV=SQRT((PPC(1,MH)/PPC(1,ML)+(GE-1.)/(GE+1.))*(GE+1.)/2./GE)*
     *SSL/ABS(SSL)
      RS=RSSAV+(SSLSAV*SQRT(GE*PPC(1,ML)/DPC(1,ML))+UPC(1,ML)/DPC(1,ML)+
     *SSE)/2.*DT
      GO TO 203
 202  IF(NXSS.NE.0) GO TO 204
      A=SQRT(PPC(1,4)/DPC(1,4))
      PG=1.+(PG0-1.)/A**2+GF*(1.-1./A**2)*(G/GF*F-R4R1)
      SSLSAV=SQRT((2.*PG-(1.-BETA)+2.*SQRT(PG**2-(1.-BETA)*PG-BETA))/
     *(1.-BETA)/G)
      RS=RSSAV+(SSE+SSLSAV*SQRT(G)*A+UPC(1,4)/DPC(1,4))*DT/2.
      GO TO 203
 204  RS=RSSAV+(2.*SSE)*DT/2.
 203  IF(((RS.GT.R2(NSHK+1+MSC)).AND.(NXSS.EQ.0)).OR.((RS.LT.R2(NSHK+1
     *+MSC)).AND.(NXSS.EQ.1))) RS=R2(NSHK+1+MSC)
      IF(((RS.LT.R2(NSHK)).AND.(NXSS.EQ.0)).OR.((RS.GT.R2(NSHK)).
     *AND.(NXSS.EQ.-1))) RS=R2(NSHK)
C
      IF(((RS.GT.RCD).AND.(NSHKCDX.EQ.0)).OR.((RS.LT.RCD).AND.(NSHKCDX
     *.EQ.1))) RS=RCD
      IF((NSC.NE.200).AND.(NSC.NE.310).AND.(NSC.NE.201).AND.(NSC.NE.311)
     *.AND.(NSC.NE.211).AND.(NSC.NE.101).AND.(NSC.NE.91).AND.(NSC.NE.90)
     *) GO TO 13
C
      IF((NXSS.EQ.0).AND.(NSHKSGN.EQ.0)) GO TO 14
C-----CORRECTOR FOR NSHK+1+NXSS*(NXSS+1)/2 = 4
      M=NSHK+1+NXSS*(NXSS+1)/2
      MM=NXSS*(NXSS+1)/2-(NXSS+1)*(NXSS-1)
      D(2,M)=DPC(2,4)=DC(4,M,MM) $ U(2,M)=UPC(2,4)=UC(4,M,MM)
      E(2,M)=EPC(2,4)=EC(4,M,MM) $ P(2,M)=PPC(2,4)=PC(4,GE)
      GO TO 13
C-----CORRECTOR FOR NSHK+1+NXSS*(NXSS+1)/2 = 4
 14   M=NSHK+1+NXSS*(NXSS+1)/2  $ SL1=(SSL+SSLSAV)/2.
      PS=P(1,NSHK)*(2.*GE/(GE+1.)*SL1**2-(GE-1.)/(GE+1.))
      DS=D(1,NSHK)/((GE-1.)/(GE+1.)+2./(GE+1.)/SL1**2)
      US=DS*(U(1,NSHK)/D(1,NSHK)+SQRT(GE*P(1,NSHK)/D(1,NSHK))*2./(GE+1.)
     **SL1*(1.-1./SL1**2))
      ES=DS*(PS/DS/(GE-1.)+US**2/DS**2/2.)
      EPS=(R2(NSHK+1)-RS)/DELR
```

```
      C1=2.*(2.-EPS)/(1.+EPS)
      C2=2.*EPS-3. $ C3=(1.-EPS)*(2.*EPS-1.)/(1.+EPS)
      D(2,M)=DPC(2,4)=(D(1,M)+DPC(1,4)+DTDX*(C1*US+C2*UPC(1,4)+
     *C3*UPC(1,5))+AT*DTDX*(C1*DS+C2*DPC(1,4)+C3*DPC(1,5)))/2.
      U(2,M)=UPC(2,4)=(U(1,M)+UPC(1,4)+DTDX*(C1*(US**2/DS+PS)+C2*CM(4)+
     *C3*CM(5))+AT*DTDX*(C1*US+C2*UPC(1,4)+C3*UPC(1,5)))/2.
      E(2,M)=EPC(2,4)=(E(1,M)+EPC(1,4)+DTDX*(C1*US/DS*(ES+PS)+C2*CE(4)+
     *C3*CE(5))+AT*DTDX*(C1*ES+C2*EPC(1,4)+C3*EPC(1,5)))/2.
      P(2,M)=PPC(2,4)=PC(4,GE)
 13   IF((NSC.NE.200).AND.(NSC.NE.310).AND.(NSC.NE.201).AND.(NSC.NE.311)
     *) GO TO 15
C-----CORRECTOR FOR NSHK+2+NXSS = 5
      M=NSHK+2+NXSS
      D(2,M)=DPC(2,5)=DC(5,M,0) $ U(2,M)=UPC(2,5)=UC(5,M,0)
      E(2,M)=EPC(2,5)=EC(5,M,0) $ P(2,M)=PPC(2,5)=PC(5,GE)
 15   IF(NTYPE(NFIRST).EQ.5) GO TO 205
      IF(NXSS.EQ.0) GO TO 16
C-----CORRECTOR FOR NSHK+(NXSS+1)/2 = 3
      M=NSHK+(NXSS+1)/2
      SSLSAV=(SSL+SSLSAV)/2.
      DPC(2,3)=D(1,M)/((GE-1.)/(GE+1.)+2./(GE+1.)/SSLSAV**2)
      UPC(2,3)=DPC(2,3)*(U(1,M)/D(1,M)+SQRT(GE*P(1,M)/D(1,M))*2./(
     *GE+1.)*SSLSAV*(1.-1./SSLSAV**2))
      PPC(2,3)=P(1,M)*(2.*GE/(GE+1.)*SSLSAV**2-(GE-1.)/(GE+1.))
      EPC(2,3)=DPC(2,3)*(PPC(2,3)/DPC(2,3)/(GE-1.)+UPC(2,3)**2
     */DPC(2,3)**2/2.)
      IF((NAME.EQ.8HSHKCDSHK).AND.(NSC.EQ.-11)) GO TO 16
      D(2,M)=DPC(2,3) $U(2,M)=UPC(2,3) $E(2,M)=EPC(2,3) $P(2,M)=PPC(2,3)
C-----CORRECTOR FOR SHK VEL
 16   MH=4-2*NSHKSGN+NXSS $ ML=2+2*NSHKSGN
      SSL=SQRT((PPC(2,MH)/PPC(2,ML)+(GE-1.)/(GE+1.))*(GE+1.)/2./GE)*
     *SSL/ABS(SSL)
      SSE=SSL*SQRT(GE*PPC(2,ML)/DPC(2,ML))+UPC(2,ML)/DPC(2,ML)
      GO TO 206
 205  IF(NXSS.EQ.0) GO TO 207
      D(2,NSHK+1)=DPC(1,3)=DPC(2,3)=DPC(1,13)
      U(2,NSHK+1)=UPC(1,3)=UPC(2,3)=UPC(1,13)
      E(2,NSHK+1)=EPC(1,3)=EPC(2,3)=EPC(1,13)
      P(2,NSHK+1)=PPC(1,3)=PPC(2,3)=PPC(1,13)
 207  A=SQRT(PPC(2,4)/DPC(2,4))
      PG=1.+(PG0-1.)/A**2+GF*(1.-1./A**2)*(C/GF*E-R4R1)
      SSL=SQRT((2.*PG-(1.-BETA)+2.*SQRT(PG**2-(1.-BETA)*PG-BETA))/
     *(1.-BETA)/G)
      SSE=SSL*SQRT(G)*A+UPC(2,4)/DPC(2,4)
 206  IF(NSC.NE.-11) GO TO 17
      IF(NAME.EQ.8HSHKCDSHK) GO TO 35
C-----CORRECTOR FOR NCD(=6) = CORRECTOR 3
      D(2,NCD)=DPC(2,3) $ U(2,NCD)=UPC(2,3)
      E(2,NCD)=EPC(2,3) $ P(2,NCD)=PPC(2,3)
C-----CORRECTOR FOR NSHK(=3) = CORRECTOR 7
      D(2,NSHK)=DPC(2,7) $ U(2,NSHK)=UPC(2,7)
      E(2,NSHK)=EPC(2,7) $ P(2,NSHK)=PPC(2,7)
 17   IF(NSC.NE.11) GO TO 35
C-----CORRECTOR FOR NCD(=6) = CORRECTOR 6
      D(2,NCD)=DPC(2,6) $ U(2,NCD)=UPC(2,6)
      E(2,NCD)=EPC(2,6) $ P(2,NCD)=PPC(2,6)
```

```
C-----CORRECTOR FOR NCD+2 = CORRECTOR 3
      D(2,NCD+2)=DPC(2,3) $ U(2,NCD+2)=UPC(2,3)
      E(2,NCD+2)=EPC(2,3) $ P(2,NCD+2)=PPC(2,3)
C-----ADVANCE SHK AND CD CELL POSITION IF NECCESSARY
 35   NSHK=NSHK+NXSS $ IF(NTYPE(NFIRST).EQ.5) NFLM=NSHK
      IF((NXSCD.EQ.0).OR.(NAME.EQ.8HSHKCDSHK)) GO TO 19
      M=NCD+2+3*(NXSCD-1)/2
      PS=P(2,M) $ DS=D(2,M) $ US=U(2,M) $ ES=E(2,M) $ R2S=R2(M)
      IF(NXSCD.EQ.-1) GO TO 18
      P(2,NCD+2)=P(2,NCD+1) $ P(2,NCD+1)=P(2,NCD) $ P(2,NCD)=PS
      U(2,NCD+2)=U(2,NCD+1) $ U(2,NCD+1)=U(2,NCD) $ U(2,NCD)=US
      E(2,NCD+2)=E(2,NCD+1) $ E(2,NCD+1)=E(2,NCD) $ E(2,NCD)=ES
      D(2,NCD+2)=D(2,NCD+1) $ D(2,NCD+1)=D(2,NCD) $ D(2,NCD)=DS
      R2(NCD)=R2S
      GO TO 19
 18   P(2,NCD-1)=P(2,NCD) $ P(2,NCD)=P(2,NCD+1) $ P(2,NCD+1)=PS
      U(2,NCD-1)=U(2,NCD) $ U(2,NCD)=U(2,NCD+1) $ U(2,NCD+1)=US
      E(2,NCD-1)=E(2,NCD) $ E(2,NCD)=E(2,NCD+1) $ E(2,NCD+1)=ES
      D(2,NCD-1)=D(2,NCD) $ D(2,NCD)=D(2,NCD+1) $ D(2,NCD+1)=DS
      R2(NCD+1)=R2S
C
 19   NDISCL(N1)=NCD=NCD+NXSCD $ RD(N1)=R2(NCD)=R2(NCD+1)=RCD
      SF(N1)=SECD $ NDISCL(NFIRST)=NSHK $ SL(NFIRST)=SSL $SE(NFIRST)=SSF
      RD(NFIRST)=RS
C
      IF(NSHKCDX.EQ.0) RETURN
C
C
      IF((NSC.EQ.0).OR.(NSC.EQ.110).OR.(NSC.EQ.109).OR.(NSC.EQ.99).OR.
     *(NSC.EQ.209).OR.(NSC.EQ.11)) GO TO 27
      RETURN
C
 20   IF((NSC.NE.210).AND.(NSC.NE.211)) GO TO 21
C-----PREDICTOR AND CORRECTOR FOR NSHK+2 = 4
      M=NSHK+2
      EPS=(R(M+1)-R(M))/DELR
      C1=2.*(2.-EPS)/(1.+EPS)
      C2=2.*EPS-3. $ C3=(1.-EPS)*(2.*EPS-1.)/(1.+EPS)
      D(2,M)=DPC(2,4)=DPC(1,4)=D(1,M)-DTDX*(C1*U(1,M+1)+C2*U(1,M)+
     *C3*U(1,M-1))-AT*DTDX*(C1*D(1,M+1)+C2*D(1,M)+C3*D(1,M-1))
      U(2,M)=UPC(2,4)=UPC(1,4)=U(1,M)-DTDX*(C1*PM(M+1)+C2*PM(M)+
     *C3*PM(M-1))-AT*DTDX*(C1*U(1,M+1)+C2*U(1,M)+C3*U(1,M-1))
      E(2,M)=EPC(2,4)=EPC(1,4)=E(1,M)-DTDX*(C1*PE(M+1)+C2*PE(M)+
     *C3*PE(M-1))-AT*DTDX*(C1*E(1,M+1)+C2*E(1,M)+C3*E(1,M-1))
      P(2,M)=PPC(2,4)=PPC(1,4)=PP(4,GE)
C
      IF(NSC.EQ.210) GO TO 22
C-----PREDICTOR FOR NSHK+2 = 5(=10)
      DPC(1,5)=DPC(1,10) $ UPC(1,5)=UPC(1,10)
      EPC(1,5)=EPC(1,10) $ PPC(1,5)=PPC(1,10)
 22   GO TO 11
C
 21   IF((NSC.NE.101).AND.(NSC.NE.91).AND.(NSC.NE.100).AND.(NSC.NE.90))
     *GO TO 23
C-----PREDICTOR AND CORRECTOR FOR NSHK+1 = 4
      M=NSHK+1
```

```
      D(2,M)=DPC(2,4)=DPC(1,4)=D(1,M) $ U(2,M)=UPC(2,4)=UPC(1,4)=U(1,M)
      E(2,M)=EPC(2,4)=EPC(1,4)=E(1,M) $ P(2,M)=PPC(2,4)=PPC(1,4)=P(1,M)
C
      IF(NSC.NE.101) GO TO 11
      IF(NAME.EQ.8HSHKCDSHK) GO TO 33
C-----PREDICTOR FOR NCD+2 = 5(=10)
      DPC(1,5)=DPC(1,10) $ UPC(1,5)=UPC(1,10)
      EPC(1,5)=EPC(1,10) $ PPC(1,5)=PPC(1,10)
      GO TO 11
C-----PREDICTOR FOR NSHK+1 = 5
 33   DPC(1,5)=D(2,M) $ UPC(1,5)=U(2,M)
      EPC(1,5)=E(2,M) $ PPC(1,5)=P(2,M)
      GO TO 11
C
 23   IF((NSC.NE.111).AND.(NSC.NE.-9).AND.(NSC.NE.1)) GO TO 24
C-----PREDICTOR AND CORRECTOR FOR NCD+2 = 4(=10)
      IF(NAME.EQ.8HSHKCDSHK) GO TO 36
      M=NCD+2
      D(2,M)=DPC(2,4)=DPC(1,4)=DPC(1,10)
      U(2,M)=UPC(2,4)=UPC(1,4)=UPC(1,10)
      E(2,M)=EPC(2,4)=EPC(1,4)=EPC(1,10)
      P(2,M)=PPC(2,4)=PPC(1,4)=PPC(1,10)
      GO TO 11
 36   DPC(2,4)=DPC(1,4)=DPC(1,6) $ UPC(2,4)=UPC(1,4)=UPC(1,6)
      EPC(2,4)=EPC(1,4)=EPC(1,6) $ PPC(2,4)=PPC(1,4)=PPC(1,6)
      GO TO 11
C
 24   IF((NSC.EQ.-10 ).OR.(NSC.EQ.89)) GO TO 11
      IF((NSC.NE.0).AND.(NSC.NE.110).AND.(NSC.NE.109).AND.(NSC.NE.99).
     *AND.(NSC.NE.209).AND.(NSC.NE.-11).AND.(NSC.NE.11)) GO TO 25
C-----PREDICTOR AND CORRECTOR FOR NSHK+1 = 4(=6)
      DPC(2,4)=DPC(1,4)=DPC(1,6) $ UPC(2,4)=UPC(1,4)=UPC(1,6)
      EPC(2,4)=EPC(1,4)=EPC(1,6) $ PPC(2,4)=PPC(1,4)=PPC(1,6)
      IF(NSHKCDX.EQ.1) GO TO 26
      GO TO 11
C
 25   PRINT 1001,NSC,NSHKSGN,NSHKCDX
 1001 FORMAT(1H ,5/,* NSC=*,3I5,10X,*TROUBLE IN SHK-CD*)
      DELR=-ABS(DELR) $ RETURN
C
 26   IF(((NSC.EQ.-11).OR.(NSC.EQ.99).OR.(NSC.EQ.0)).AND.(NSHKSGN.EQ.0))
     *GO TO 25
C
      TRFLCT=(RCDSAV-RSSAV)/(SSESAV-SECDSAV)
      RS=RCD=RCDSAV+TRFLCT*SECDSAV
      GO TO 11
C
 27   IF(NSC.NE.11) GO TO 28
      IF(RCD.LT.R2(NSHK)) RCD=RS=R2(NCD)=R2(NCD+1)=RD(NFIRST)=RD(N1)=
     *R2(NSHK)
      TRFLCT=(RS-RSSAV)/SSESAV
C
 28   IF(NSC.NE.109) GO TO 29
      IF(RCD.GT.R2(NSHK+2)) GO TO 30
      NDISCL(NFIRST)=NSHK=NSHK-1
C-----CORRECTOR FOR NCD+2 = CORRECTOR FOR NCD+1
```

```
      M=NCD+1
      D(2,M+1)=D(2,M) $ U(2,M+1)=U(2,M)
      E(2,M+1)=E(2,M) $ P(2,M+1)=P(2,M)
      GO TO 29
 30   M=NCD
      PS=P(2,M) $ DS=D(2,M) $ US=U(2,M) $ ES=E(2,M) $ R2S=R2(M+2)
      U(2,NCD)=U(2,NCD+2) $ U(2,NCD+2)=U(2,NCD+1) $ U(2,NCD+1)=US
      E(2,NCD)=E(2,NCD+2) $ E(2,NCD+2)=E(2,NCD+1) $ E(2,NCD+1)=ES
      P(2,NCD)=P(2,NCD+2) $ P(2,NCD+2)=P(2,NCD+1) $ P(2,NCD+1)=PS
      D(2,NCD)=D(2,NCD+2) $ D(2,NCD+2)=D(2,NCD+1) $ D(2,NCD+1)=DS
      NDISCL(N1)=NDISCL(N1)+1
      R2(NCD)=R2S $ RD(NFIRST)=R2(NCD+2)=R2(NCD+1)=RCD
C
C
 29   RD(NFIRST)=RD(N1)
      IF(NTYPE(NFIRST).EQ.2) CALL SHKCDA(NFIRST,N1,TRFLCT)
      IF(NTYPE(NFIRST).EQ.5) CALL DETCDA(NFIRST,N1,TRFLCT)
      RETURN
C
C
C-----SPCL FLW FLD ADJSTMNT FR NSHKCDX=1
 100  DO 110 M=1,N
      IF(M.GT.NCD+1) GO TO 102
      IF(M.LT.NCD) GO TO 101
      GO TO 110
 101  D(2,M)=D(1,M)=D(1,NCD+1) $ U(2,M)=U(1,M)=U(1,NCD+1)
      E(2,M)=E(1,M)=E(1,NCD+1) $ P(2,M)=P(1,M)=P(1,NCD+1)
      GO TO 110
 102  D(1,M-2)=D(1,M) $ U(1,M-2)=U(1,M) $ E(1,M-2)=E(1,M)
      D(2,M-2)=D(2,M) $ U(2,M-2)=U(2,M) $ E(2,M-2)=E(2,M)
      P(1,M-2)=P(1,M) $ P(2,M-2)=P(2,M) $ R(M-2)=R(M) $ R2(M-2)=R2(M)
 110  CONTINUE
      DO 120 M=1,NTDISCT
      IF(NDISCL(M).LT.NCD+1) NDISCL(M)=-1
      IF(NDISCL(M).GE.NCD+1) NDISCL(M)=NDISCL(M)-2
 120  CONTINUE
      NFLM=NFLM-2 $ N=N-2
      IF(N1+1.LE.NTDISCS) NRHS(N1+1)=NRHS(N1+1)-2
      IF(N1+1.LE.NTDISCS) NLHS(N1+1)=NLHS(N1+1)-2
C
C-----ACCT FOR PRTCLE PTH AND NEG AND POS CHRCT TRAJ CELL LOCTNS CHANGES
      IF(NPPTH.LE.0) GO TO 126
      DO 111 I=1,NPPTH
      IF(NCLPPTH(I).GE.NCD+1) NCLPPTH(I)=NCLPPTH(I)-2
 111  CONTINUE
 126  IF(NUMA.LE.0) GO TO 112
      DO 113 I=NUMAFST,NUMA
      IF(NCLUMA(I).GE.NCD+1) NCLUMA(I)=NCLUMA(I)-2
 113  CONTINUE
 112  IF(NUPA.LE.0) GO TO 114
      DO 115 I=1,NUPA
      IF(NCLUPA(I).GE.NCD+1) NCLUPA(I)=NCLUPA(I)-2
 115  CONTINUE
C
 114  NUBC=3HYES
C
```

```
      NCODE=10HSHKCD        $ CALL FRNTFF(NCODE)
C
      RETURN $ END
```

```
      SUBROUTINE SHKCDA(NFIRST,N1,IREFLCT)
      COMMON/PARAM/N,J,AJ,G,GF,DELR,NFLM
      COMMON/TIME/T,DT,DTL,TWRITE,DELT,DTDX,AT
      COMMON/FIRFETC/INDEX,NCYCLE,NN,NNN,NSTORE,NS,NITRCTN
      COMMON/ARRAYS/P(501),U(2,501),R(2,501),D(2,501),E(2,501),R2(501)
      COMMON/DISCS/NTDISC,NDISCNO(51),NTYPE(51),NDISCL(51),SE(51),SL(51)
     *,RD(51),NTDISCT,NTDISCS
C
C-----DET STATES 4 AND 5 DUE TO SHK-CD INTERACTION
C     SET STATE1,0 AS REF STATE FOR POLAR SOLN, AS OVERALL REF STATE
C     STD CONDITS ASSUMED TO HLD ACROSS CD  P=CONST , U=CONST
C     VEL REF WRT SQRT(GF*P1/D1)
C     P21=TO BE READ AS PRESS IN STATE2 REL TO AND NONDIM BY STATE1
C     U410=READ AS PRTCLE VEL IN STTE4 REL TO STTE1 AND NONDIM BY STTE0
C     U41=READ AS PRTCLE VEL IN STATE4 REL TO AND NONDIM BY STATE1
C
C
C-----TWO POSSIBLE INTERACTIONS -  DEPENDING ON THE SPD OF SND RATIO
C
C     A2.GT.A1
C
C     - STATE -    -3---S--5--CD--4--S---1-
C                  -3---S--2--CD--1--
C
C
C     A2.LT.A1
C
C     - STATE -    -3---H+RARE+T--5--CD--4--S---1-
C                            -3---S--2--CD--1--
C
C
      GE=GF $ IF(NDISCL(NFIRST).GT.NFLM) GE=G
C-----SET STATES 1,2 AND 3
      M=NDISCL(NFIRST) $ IS=0 $ IF(NTYPE(NFIRST).NE.2) IS=-1
      P10=P(2,M+2+2*IS) $ D10=D(2,M+2+2*IS) $ U10=U(2,M+2+2*IS)
      P20=P(2,M+1) $ D20=D(2,M+1) $ U20=U(2,M+1)
      P30=P(2,M-2*IS) $ D30=D(2,M-2*IS) $ U30=U(2,M-2*IS)
      U10=U10/D10/SQRT(G) $ U20=U20/D20/SQRT(G) $ U30=U30/D30/SQRT(G)
      A10=SQRT(P10/D10*GE/G) $ A20=SQRT(P20/D20*GE/G)
      A30=SQRT(P30/D30*GE/G)
      SL32=SL(NFIRST-IS) $ SE30=SE(NFIRST-IS)
      SIGN=1.0 $ IF(SL32.LT.0.0) SIGN=-1.0
C
      P11=D11=A11=1.0 $ U11=0.0
      P21=P20/P10 $ D21=D20/D10 $ A21=A20/A10
      U210=U20-U10 $ U211=U210/A10
      SE21=SL32*A21+U211
      P32=P30/P20 $ P31=P32*P21
      U321=(U30-U20)/A10 $ U31=U321+U211
      D32=D30/D20 $ D31=D32*D21
      A32=A30/A20 $ A31=A32*A21
C
C-----SET SUB CONSTS AND ITER COUNTER
      SL53=SL523=SL535=SE51=SEH1=SET1=99999.999
      BETA=(GE-1.)/(GE+1.) $ ALPHA=2./(GE-1.)
      NSIGN=-1 $ DELP541=0.5 $ EPS=1.E-14 $ NITER=0
```

```
C
C-----GUESS P51=P41 AS AVG BET P31 AND P FOR U31 AT A11
      U41=U31
      XI=(U41-U11)**2/A11**2*GE/(1.-BETA)
      P41=(2.+XI+SQRT(XI**2+4.*XI*(1.+BETA)))/2.
      P51=P41=(P41+P31)/2.-0.333333*DELP541
C
C-----PERFORM ITER OF THE PLR SOLN
      PRINT 1003,SL32,A21
 1003 FORMAT(1H1,5X,*SHK-CD INTERACTN - SHK MACH NO., A21=*,2F8.4)
 1    NITER=NITER+1
      IF(NITER.LT.75) GO TO 2
      IF(NITER.EQ.75) DELUSAV=DELU
      IF(NITER.LT.86) GO TO 2
      IF((DELUSAV.LT.2.E-12).AND.(DELU.LT.2.E-12)) GO TO 100
      PRINT 1000,DELU,DELUSAV
 1000 FORMAT(1H ,5/,5X,*NO. OF ITERATIONS HAS REACHED MAX OF 85*,2E20.10
     *)
      IF(DELU.LT.9.E-10) GO TO 100
      DELR=-ABS(DELR) $ RETURN
 2    P51=P41=P41+DELP541
      SL41=SQRT((P41+BETA)/(1.+BETA))*SIGN
      SE41=SL41*A11+U11
      U41=(1.-BETA)*(1.-1./SL41**2)*SL41*A11
      D41=1./(BETA+(1.-BETA)/SL41**2)
      A41=SQRT((1.-BETA)**2*(SL41**2-BETA/(1.+BETA))*(1./SL41**2+
     *BETA/(1.-BETA))*GE)
C
C-----CHECK IF  A2.GT.A1
      IF(A21.LT.1.0) GO TO 10
C
C-----HERE A2.GT.A1 SO SHK-SHK
      P53=P51/P31
      SL53=-SQRT((P53+BETA)/(1.+BETA))*SIGN
      SES1=SL53*A31+U31
      U531=(1.-BETA)*(1.-1./SL53**2)*SL53*A31
      U51=U531+U31
      D53=1./(BETA+(1.-BETA)/SL53**2)
      D51=D53*D31
      A53=SQRT((1.-BETA)**2*(SL53**2-BETA/(1.+BETA))*(1./SL53**2+
     *BETA/(1.-BETA))*GE)
      A51=A53*A31
      GO TO 3
C
C
C-----HERE A2.LT.A1 SO RARE-SHK
C-----ASSUME A LEFT RUNNING RARE
 10   P53=P51/P31
      D53=((2.+ALPHA)/ALPHA/(E*(P53-1.)+1.)**(ALPHA/(2.+ALPHA))
      D51=D53*D31
      A53=D53**(1./ALPHA)
      A51=A53*A31
      U531=-ALPHA*(A53-1.)*A31*SIGN
      U51=U531+U31
      SL533=SL535=-1.*SIGN
      SEH1=SL533*A31+U31 $ SET1=SL535*A51+U51
```

```
C
 3    DELU=ABS(U41-U51)*(ABS(1./U41)+ABS(1./U51))/2.0
C
      PRINT 1001,NITER,DELP541,DELU,A21,P21,D21,U211,U11,
     *P32,P31,D32,D31,A32,A31,U321,U31,SL32,SE21,
     *P41,D41,A41,U41,SL41,SE41,SL533,SL535,SEH1,SET1,
     *P53,P51,D53,D51,A53,A51,U531,U51,SL53,SE51
 1001 FORMAT(1H ,I5,2E12.5,5F12.7,/,10F13.5,/,10F13.5,/,10F13.5,/)
C
      IF(DELU.LE.EPS) GO TO 100
C
      IF((U51*SIGN.LT.U41*SIGN).AND.(NSIGN.LE.0)) DELP541=-ABS(DELP541)
      IF((U51*SIGN.LT.U41*SIGN).AND.(NSIGN.EQ.-1)) NSIGN=0
      IF((U51*SIGN.LT.U41*SIGN).AND.(NSIGN.EQ.1)) DELP541=-ABS(DELP541)/
     *2.
      IF((U41*SIGN.LT.U51*SIGN).AND.(NSIGN.EQ.-1)) DELP541=ABS(DELP541)
      IF((U41*SIGN.LT.U51*SIGN).AND.(NSIGN.GE.0))DELP541=ABS(DELP541)/2.
      IF((U41*SIGN.LT.U51*SIGN).AND.(NSIGN.EQ.0)) NSIGN=1
      GO TO 1
C
C
C-----ADD DISCS TO THE FLOW FIELD
 100  NT=NTDISCT $ U10=U10*SQRT(G)
      NDISCNO(NT+1)=NTDISC+1 $ NDISCNO(NT+2)=NTDISC+2
      NDISCNO(NT+3)=NTDISC+3
      NTYPE(NT+1)=NTYPE(NT+3)=2 $ NTYPE(NT+2)=3
      SL(NT+3)=SL53 $ SL(NT+1)=SL41 $ SL(NT+2)=0.0
      SE(NT+3)=SE51*A10*SQRT(G)+U10 $ SE(NT+1)=SE41*A10*SQRT(G)+U10
      SE(NT+2)=U41*A10*SQRT(G)+U10
      NCD=NDISCL(N1+IS)
      R2(NCD)=R2(NCD+1)=RD(NT+2)=RD(NFIRST)+(U41*A10*SQRT(G)+U10)*
     *(DT-TRFLCT)
      RD(NT+3)=RD(NFIRST)+SE(NT+3)*(DT-TRFLCT)
      RD(NT+1)=RD(NFIRST)+SE(NT+1)*(DT-TRFLCT)
      IF(RD(NT+3+2*IS).LT.R2(NCD-1)) RD(NT+3+2*IS)=R2(NCD-1)
      IF(RD(NT+1-2*IS).GT.R2(NCD+2)) RD(NT+1-2*IS)=R2(NCD+2)
      NDISCL(N1)=NDISCL(NFIRST)=-1 $ NDISCL(NT+3+2*IS)=NCD-1
      NDISCL(NT+1-2*IS)=NCD+1 $ NDISCL(NT+2)=NCD
C-----SET THERMO AND GAS PARAMETERS IN CELLS NCD, NCD+1
      U50=U51*A10*SQRT(G)+U10 $ U40=U41*A10*SQRT(G)+U10
      M=NCD-IS
      D50=D(2,M)=D51*D10 $ U(2,M)=U50*D50 $ P50=P(2,M)=P51*P10
      E(2,M)=D50*(P50/D50/(GE-1.)+U50**2/2.)
      M=NCD+1+IS
      D40=D(2,M)=D41*D10 $ U(2,M)=U40*D40 $ P40=P(2,M)=P41*P10
      E(2,M)=D40*(P40/D40/(GE-1.)+U40**2/2.)
      NTDISCT=NTDISCT+3 $ NTDISC=NTDISC+3
      IF(A21.GT.1.0) GO TO 101
      NTDISCT=NTDISCT-1 $ NTDISC=NTDISC-1
      NTYPE(NT+2)=4-4*IS
      M=NCD-IS $ MM=NCD-1-3*IS
      D(2,M)=D(2,MM) $ U(2,M)=U(2,MM) $ E(2,M)=E(2,MM) $ P(2,M)=P(2,MM)
C
C
 101  NCODE=10HSHKCDA      $ IF(IS.EQ.-1) NCODE=10HCDSHKA
      CALL PRNTFF(NCODE)
```

```
      NITRCTN=3HYES
C
      RETURN $ END
```

```
      SUBROUTINE DETCDA(N1,N2,TRFLCT)
      COMMON/PARAM/N,J,AJ,G,GF,DELR,NFLM
      COMMON/TIME/T,DT,DTL,TWRITE,DELT,DTDX,AT
      COMMON/FIRFETC/INDEX,NCYCLE,NN,NNN,NSTORE,NS,NITRCTN
      COMMON/DISCSKF/RDSAV(51),NLHS(51),NRHS(51),NCROSS(51)
      COMMON/POWER/VCAPF,R4R1,PDPOWER,PPOWER,CND,CCAPF,SFECLD,SFE,NCJ
      COMMON/ARRAYS/R(501),U(2,501),P(2,501),D(2,501),E(2,501),R2(501)
      COMMON/DISCS/NTDISC,NDISCNO(51),NTYPE(51),NDISCL(51),SE(51),SL(51)
     *,RD(51),NTDISCT,NTDISCS
C
C     -STATE--         ---1---4-DETONATION-3---
C
C     -STATE--     ---1-DETONATION--2-CD-3---
C
      NCD=NDISCL(N2) $ MI=NCD+2
      DO 1 M=MI,N
      P(1,M-2)=P(1,M) $D(1,M-2)=D(1,M) $U(1,M-2)=U(1,M) $E(1,M-2)=E(1,M)
      P(2,M-2)=P(2,M) $D(2,M-2)=D(2,M) $U(2,M-2)=U(2,M) $E(2,M-2)=E(2,M)
      R(M-2)=R(M) $ R2(M-2)=R2(M)
 1    CONTINUE
      DO 2 M=N2,NTDISCT
      IF(NDISCL(M).GT.MI) NDISCL(M)=NDISCL(M)-2
 2    CONTINUE
      N=N-2 $ NDISCL(N2)=-1
      IF(N2+1.LE.NTDISCS) NRHS(N2+1)=NRHS(N2+1)-2
      IF(N2+1.LE.NTDISCS) NLHS(N2+1)=NLHS(N2+1)-2
C
      BETA=(GF-1.)/(GF+1.) $ B=(GF-1.)/(G-1.)
      PG0=(1.+BETA)*(VCAPF-BETA)/(1.-BETA)-BETA
      A=SQRT(P(2,NCD)/D(2,NCD))
      PG=1.+(PG0-1.)/A**2+GF*(1.-1./A**2)*(G/GF*B-R4R1)
      SL(N1)=SQRT((2.*PG-(1.-BETA)+2.*SQRT(PG**2-(1.-BETA)*PG-BETA))/
     *(1.-BETA)/G)
      SE(N1)=SL(N1)*SQRT(G)*A+U(2,NCD)/D(2,NCD)
C
      RD(N1)=RD(N1)+SE(N1)*(DT-TRFLCT)
      IF(RD(N1).LT.R2(NCD)) GO TO 3
      IF(RD(N1).GT.R2(NCD+1)) RD(N1)=R2(NCD+1)
      PS1=(1.-BETA)/2.*(1.+G*SL(N1)**2) $ P(2,NCD)=PS1*P(2,NCD)
      US=SQRT(G)*A*SQRT((1.-BETA)*(PS1-PG)*(PS1-1.)/G/
     *(PS1+BETA))+U(2,NCD)/D(2,NCD)
      D(2,NCD)=D(2,NCD)/(1.-(1.-BETA)*(PS1-PG)/(PS1+BETA))
      U(2,NCD)=US*D(2,NCD)
      E(2,NCD)=D(2,NCD)*(P(2,NCD)/D(2,NCD)/(GF-1.)+US**2/2.)
      NFLM=NDISCL(N1)=NDISCL(N1)+1
      A=SQRT(P(2,NFLM+1)/D(2,NFLM+1))
      PG=1.+(PG0-1.)/A**2+GF*(1.-1./A**2)*(G/GF*B-R4R1)
      SL(N1)=SQRT((2.*PG-(1.-BETA)+2.*SQRT(PG**2-(1.-BETA)*PG-BETA))/
     *(1.-BETA)/G)
      SE(N1)=SL(N1)*SQRT(G)*A+U(2,NFLM+1)/D(2,NFLM+1)
C
 3    NCODE=10HDETCD       $ CALL PRNTFF(NCODE)
      NITRCTN=3HYES
C
      RETURN $ END
```

```
      SUBROUTINE SCDFLM(NFIRST,NSECOND,NTHIRD,NAME)
      COMMON/PARAM/N,J,AJ,G,GF,DELR,NFLM
      COMMON/TIME/T,DT,DTL,TWRITE,DELT,DTDX,AT
      COMMON/PREDCOR/DPC(2,13),UPC(2,13),EPC(2,13),PPC(2,13)
      COMMON/ARRAYS/R(501),U(2,501),P(2,501),D(2,501),E(2,501),R2(501)
      COMMON/DISCS/NTDISC,NDISCNO(51),NTYPE(51),NDISCL(51),SE(51),SL(51)
     *,RD(51),NTDISCT,NTDISCS
C
C
      PRINT 1001,NAME
 1001 FORMAT(1H ,1X,A4)
      RS=RD(NFIRST) $ SSL=SL(NFIRST) $ SSE=SE(NFIRST)
      NSHK=NDISCL(NFIRST) $ RCD=RD(NSECOND) $ SECD=SE(NSECOND)
      NCD=NDISCL(NSECOND) $ RF=RD(NTHIRD) $ SFE=SE(NTHIRD)
      SFL=SL(NTHIRD)
      RCDTEST=RCD+SECD*DT
      NXSCD=0
      IF(RCDTEST.GT.R2(NCD+2))NXSCD=1 $ IF(RCDTEST.LT.R2(NCD-1))NXSCD=-1
      SFESAV=SFE
      IF(RCDTEST.LE.RF+SFE*DT) GO TO 1
 6    PRINT 1000,NCD,RCD,SECD,RCDTEST,RF,SFE,DT
 1000 FORMAT(1H ,5/,1X,I5,6E14.5,/,* TROUBLE SHK-CD FLM, CD-FLM MERGE*)
      RETURN
 1    NSCDF=10*(NFLM-NCD-1)+NXSCD
      CALL SHKCD(NFIRST,NSECOND,8HSHKCDFLM)
      IF(RCD.GT.RF+SFE*DT) GO TO 6
      IF(NAME.EQ.4HSCFS) GO TO 2
C-----PREDICTOR AND CORRECTOR FOR NFLM+1 = 10
      M=NFLM+1
      D(2,M)=DPC(2,10)=DPC(1,10)=DP(M,0)
      U(2,M)=UPC(2,10)=UPC(1,10)=UP(M,0)
      E(2,M)=EPC(2,10)=EPC(1,10)=EP(M,0)
      P(2,M)=PPC(2,10)=PPC(1,10)=PP(10,G)
C-----PREDICTOR AND CORRECTOR FOR NFLM = 9
      CALL SFVDPD(G,GF,PPC(1,10),DPC(1,10),SFL,VD,PD,DELR)
      D(2,NFLM)=DPC(2,9)=DPC(1,9)=DPC(1,10)/VD
      P(2,NFLM)=PPC(2,9)=PPC(1,9)=PPC(1,10)*PD
      U(2,NFLM)=UPC(2,9)=UPC(1,9)=DPC(1,9)*(SFL*(1.-VD)+UPC(1,10)/
     *DPC(1,10))
      E(2,M)=EPC(2,9)=EPC(1,9)=DPC(1,9)*(PPC(1,9)/DPC(1,9)/(GF-1.)+
     *UPC(1,9)**2/DPC(1,9)**2/2.)
 2    IF((NSCDF.EQ.10).OR.(NSCDF.EQ.21)) GO TO 5
      IF(NSCDF.EQ.9) GO TO 3
C-----PREDICTOR FOR NFLM-1 = 8
      M=NFLM-1
      DPC(1,8)=DP(M,0) $ UPC(1,8)=UP(M,0)
      EPC(1,8)=EP(M,0) $ PPC(1,8)=PP(8,GF)
C-----CORRECTOR FOR NFLM-1 = 8
      M=NFLM-1
      EPS=(R2(M)-RCD)/DELR
      C1=2.*(2.-EPS)/(1.+EPS)
      C2=2.*EPS-3. $ C3=(1.-EPS)*(2.*EPS-1.)/(1.+EPS)
      D(2,M)=DPC(2,8)=(D(1,M)+DPC(1,8)+DTDX*(C1*UPC(1,7)+C2*UPC(1,8)+
     *C3*UPC(1,9))+AT*DTDX*(C1*DPC(1,7)+C2*DPC(1,8)+C3*DPC(1,9)))/2.
      U(2,M)=UPC(2,8)=(U(1,M)+UPC(1,8)+DTDX*(C1*CM(7)+C2*CM(8)+C3*CM(9))
     *+AT*DTDX*(C1*UPC(1,7)+C2*UPC(1,8)+C3*UPC(1,9)))/2.
```

```
      E(2,M)=EPC(2,8)=(E(1,M)+EPC(1,8)+DTDX*(C1*CE(7)+C2*CE(8)+C3*CE(9))
     *+AT*DTDX*(C1*EPC(1,7)+C2*EPC(1,8)+C3*EPC(1,9)))/2.
      P(2,M)=PPC(2,8)=PC(8,GF)
      GO TO 4
C-----PREDICTOR FOR NFLM-2 = 8(=C5)
 3    DPC(1,8)=DPC(2,5) $ UPC(1,8)=UPC(2,5)
      EPC(1,8)=EPC(2,5) $ PPC(1,8)=PPC(2,5)
C-----CORRECTOR FOR NFLM = 9
 4    D(2,NFLM)=DPC(2,9)=DC(9,NFLM,0) $ U(2,NFLM)=UPC(2,9)=UC(9,NFLM,0)
      E(2,NFLM)=EPC(2,9)=EC(9,NFLM,0) $ P(2,NFLM)=PPC(2,9)=PC(9,GF)
C-----CORRECTOR FOR NFLM+1 = 10
      VD=DPC(1,10)/DPC(1,9) $ CALL FLM43(9,VD,SFL)
      M=NFLM+1
      D(2,M)=DPC(2,10) $ U(2,M)=UPC(2,10)
      E(2,M)=EPC(2,10) $ P(2,M)=PPC(2,10)
 5    RD(NTHIRD)=R2(NFLM+1)=R2(NFLM)=RF=RF+SFESAV*DT
      SF(NTHIRD)=SFL+U(2,NFLM+1)/D(2,NFLM+1) $ SL(NTHIRD)=SFL
C
      RETURN $ END
```

```
      SUBROUTINE SCDFLMS(NFIRST,NSECOND,NTHIRD,N1)
      COMMON/PARAM/N,J,AJ,C,CF,DELR,NFLM
      COMMON/PREDCOR/DPC(2,13),UPC(2,13),EPC(2,13),PPC(2,13)
      COMMON/ARRAYS/R(501),U(2,501),P(2,501),D(2,501),E(2,501),R2(501)
C
C
      CALL FLMSHK(4HSCFS,NTHIRD,N1)
C-----PREDICTOR AND CORRECTOR FOR NFLM = 9(=7)
      D(2,NFLM)=DPC(2,9)=DPC(1,9)=DPC(1,7)
      U(2,NFLM)=UPC(2,9)=UPC(1,9)=UPC(1,7)
      E(2,NFLM)=EPC(2,9)=EPC(1,9)=EPC(1,7)
      P(2,NFLM)=PPC(2,9)=PPC(1,9)=PPC(1,7)
C-----PREDICTOR AND CORRECTOR FOR NFLM+1 = 10(=8)
      D(2,NFLM+1)=DPC(2,10)=DPC(1,10)=DPC(1,8)
      U(2,NFLM+1)=UPC(2,10)=UPC(1,10)=UPC(1,8)
      E(2,NFLM+1)=EPC(2,10)=EPC(1,10)=EPC(1,8)
      P(2,NFLM+1)=PPC(2,10)=PPC(1,10)=PPC(1,8)
      CALL SCDFLM(NFIRST,NSECOND,NTHIRD,4HSCFS)
      CALL FLMSHK(4HSCFK,NTHIRD,N1)
C
      RETURN $ END
```

```
      SUBROUTINE SKCDSHK(NFIRST,NSECOND,N1)
      COMMON/SCS/SECDSCS,RCDSCS,NXCDSCS
      COMMON/DISCSKF/RDSAV(51),NLHS(51),NRHS(51),NCRCSS(51)
      COMMON/DISCS/NTDISC,NDISCND(51),NTYPE(51),NDISCL(51),SE(51),SL(51)
     *,RD(51),NTDISCT,NTDISCS
C
      PRINT 1000
 1000 FORMAT(1H ,1X,*SHK-CD-SHK CALLED BY DISC*)
C
      NCD=NDISCL(NSECOND) $ SECD=SE(NSECOND) $ RCD=RD(NSECOND)
      NAME=6HSCDGT1 $ IF(NCD-NDISCL(NFIRST).EQ.1) NAME=6HSCDEQ1
      CALL CDSHK(NSECOND,N1,NAME)
C
      RCDSCS=RD(NSECOND) $ RD(NSECOND)=RCD
      NCDSCS=NDISCL(NSECOND) $ NDISCL(NSECOND)=NCD
      SECDSCS=SE(NSECOND) $ SE(NSECOND)=SECD $ NXCDSCS=NCRCSS(NSECOND)
C
      CALL SHKCD(NFIRST,NSECOND,8HSHKCDSHK)
C
      RETURN $ END
```

```
      SUBROUTINE DSORSD(II,NI,NJ,NAME)
      COMMON/PARAM/N,J,AJ,G,GF,DELR,NFLM
      COMMON/TIME/T,DT,DTL,TWRITE,DELT,DTCX,AT
      COMMON/DISCSKP/RDSAV(51),NLHS(51),NRHS(51),NCROSS(51)
      COMMON/PREDCOR/DPC(2,13),UPC(2,13),EPC(2,13),PPC(2,13)
      COMMON/POWER/VCAPF,R4R1,PDPOWER,PPOWER,CND,CCAPF,SFEDLD,SFE,NCJ
      COMMON/ARRAYS/R(501),U(2,501),P(2,501),D(2,501),E(2,501),R2(501)
      COMMON/DISCS/NTDISC,NDISCND(51),NTYPE(51),NDISCL(51),SE(51),SL(51)
     *,RD(51),NTDISCT,NTDISCS
C
      D2M(SSL,GE,M)=D(1,M)/((GE-1.)/(GE+1.)+2./(GE+1.)/SSL**2)
      U2M(SSL,GE,M)=D(2,M)*(U(1,M)/D(1,M)+SQRT(GE*P(1,M)/D(1,M))*(2./
     *(GE+1.)*SSL*(1.-1./SSL**2)))
      P2M(SSL,GE,M)=P(1,M)*(2.*GE/(GE+1.)*SSL**2-(GE-1.)/(GE+1.))
      E2M(GE,M)=D(2,M)*(P(2,M)/D(2,M)/(GE-1.)+U(2,M)**2/D(2,M)**2/2.)
C
      SSLS(PAHD,PBHD,GE)=SQRT((PBHD/PAHD+(GE-1.)/(GE+1.))*(GE+1.)/2./GE)
      SSES(SSL,GE,K,L)=SSL*SQRT(GE*PPC(K,L)/DPC(K,L))+UPC(K,L)/DPC(K,L)
C
      PG1(A)=1.+(PG0-1.)/A**2+GF*(1.-1./A**2)*(G/GF*B-R4R1)
      BETA=(GF-1.)/(GF+1.) $ B=(GF-1.)/(G-1.)
      PG0=(1.+BETA)*(VCAPF-BETA)/(1.-BETA)-BETA
C
C
      G1=G $ IF(NDISCL(NI)+1.LE.NFLM) G1=GF
      G2=G $ IF(NDISCL(II)+1.LE.NFLM) G2=GF
      G3=G $ IF(NDISCL(II).LE.NFLM) G3=GF
      NCRSII=NCROSS(II) $ MII=NDISCL(II)+1
      NCRSNI=NCROSS(NI) $ MNI=NDISCL(NI)+1
      NSC=100*(NDISCL(NI)-NDISCL(II))+10*NCRSII+NCRSNI
      PRINT 2000,NAME,NCRSII,NCRSNI,MII,MNI,II,NI,NSC
 2000 FORMAT(1H ,1X,A7,1X,7I5)
      IF((MII.EQ.MNI).AND.(NCRSII.EQ.1).AND.(NCRSNI.EQ.0)) GO TO 72
      GO TO 71
 72   NCRSII=NCROSS(II)=0 $ NRHS(II)=NRHS(II)-1 $ GO TO 1
 71   IF(NRHS(II).NE.MNI) GO TO 1
C
C-----SHKNI(NCRS=0) PRDTR FR NSHK-1,0,1,2=1,2,3,4 - CRRCTR FR NSHK,1=2,3
C-----SHKNI(NCRS=1) PRDTR FR NSHK-1,0,1,2,3 AND CRRCTR FR NSHK,1,2=2,3,4
      IF(NAME.EQ.7HDTSKSKO) GO TO 2
      IF((NAME.EQ.7HSPECIAL).OR.(NAME.EQ.7HSPECIAL)) GO TO 2
      IF(NTYPE(NI).EQ.2) CALL SHK(RD(NI),SE(NI),SL(NI),NDISCL(NI))
      IF(NTYPE(NI).EQ.5) CALL DET(RD(NI),SE(NI),SL(NI),NDISCL(NI),NCRSNI
     *,7H DSORSD,NI)
      GO TO 2
C-----SHKNI PREDICTOR FOR NSHK+2+NCRSNI = 4
 1    IF((NAME.EQ.7HDTSKSKO).OR.(NAME.EQ.7HSPECIAL)) GO TO 2
      GE=G1
      M=NDISCL(NI)+2+NCRSNI
      DPC(1,4)=DP(M,0) $ UPC(1,4)=UP(M,0)
      EPC(1,4)=EP(M,0) $ PPC(1,4)=PP(4,GE)
C-----SHKNI PREDICTOR FOR NSHK+1+NCRSNI = 3
      M=NDISCL(NI)+1+NCRSNI
      DPC(1,3)=DP(M,0) $ UPC(1,3)=UP(M,0)
      EPC(1,3)=EP(M,0) $ PPC(1,3)=PP(3,GE)
C-----SHKNI CORRECTOR FOR NSHK+1+NCRSNI = 3
```

```
      D(2,M)=DPC(2,3)=DC(3,M,1) $ U(2,M)=UPC(2,3)=UC(3,M,1)
      E(2,M)=EPC(2,3)=EC(3,M,1) $ P(2,M)=PPC(2,3)=PC(3,GE)
C------SHKII PREDICTOR FOR NSHK-1 = 6
 2    IF(NAME.EQ.7H DTSKSK) GO TO 7
      M=NDISCL(II)-1
      DPC(1,6)=DP(M,0) $ UPC(1,6)=UP(M,0)
      EPC(1,6)=EP(M,0) $ PPC(1,6)=PP(6,G3)
C------SHKII PREDICTOR FOR NSHK = 7
      M=NDISCL(II)
      IF(NDISCL(II).EQ.NDISCL(NI)) GO TO 5
      PS=P2M(SL(II),G2,MII) $ DS=D2M(SL(II),G2,MII)
      US=U2M(SL(II),G2,MII)/D(2,MII)*DS
      IF(NTYPE(II).EQ.2) GO TO 6
      A=SQRT(P(1,MII)/D(1,MII)) $ PG=PG1(A)
      PS1=(1.-BETA)/2.*(1.+     G*SL(II)**2)    $ PDT=PS=PS1*P(1,MII)
      DDT=DS=D(1,MII)/(1.-(1.-BETA)*(PS1-PG)/(PS1+BETA))
      UDT=US=DS*(SQRT(G )*A*SQRT((1.-BETA)    *(PS1-PG)*(PS1-1.)/G
     * /(PS1+BETA))+U(1,MII)/D(1,MII))
      EDT=DS*(PS/DS/(GF-1.)+US**2/DS**2/2.)
      GO TO 6
 5    IF((NAME.EQ.7HDTSKSKO).AND.(NDISCL(NJ)+1.EQ.MNI)) GO TO 96
      IF((NAME.EQ.7HSPECIAL).AND.(NDISCL(NJ)+1.EQ.MNI)) GO TO 102
      PS=P2M(SL(NI),G1,MNI) $ DS=D2M(SL(NI),G1,MNI)
      US=U2M(SL(NI),G1,MNI)/D(2,MNI)*DS
      IF(NTYPE(NI).EQ.2) GO TO 97
      A=SQRT(P(1,MNI)/D(1,MNI)) $ PG=PG1(A)
      PS1=(1.-BETA)/2.*(1.+     G*SL(NI)**2)    $ PDT=PS=PS1*P(1,MNI)
      DDT=DS=D(1,MNI)/(1.-(1.-BETA)*(PS1-PG)/(PS1+BETA))
      UDT=US=DS*(SQRT(G )*A*SQRT((1.-BETA)    *(PS1-PG)*(PS1-1.)/G
     * /(PS1+BETA))+U(1,MNI)/D(1,MNI))
      EDT=DS*(PS/DS/(GF-1.)+US**2/DS**2/2.)
      GO TO 97
 102  PS=PPC(1,1) $ DS=DPC(1,1) $ US=UPC(1,1)
      GO TO 97
 96   PS=P2M(SL(NJ),G ,MNI) $ DS=D2M(SL(NJ),G ,MNI)
      US=U2M(SL(NJ),G ,MNI)/D(2,MNI)*DS
      IF(NTYPE(NJ).EQ.2) GO TO 82
      A=SQRT(P(1,MNI)/D(1,MNI)) $ PG=PG1(A)
      PS1=(1.-BETA)/2.*(1.+     G*SL(NJ)**2)    $ PDT=PS=PS1*P(1,MNI)
      DDT=DS=D(1,MNI)/(1.-(1.-BETA)*(PS1-PG)/(PS1+BETA))
      UDT=US=DS*(SQRT(G )*A*SQRT((1.-BETA)    *(PS1-PG)*(PS1-1.)/G
     * /(PS1+BETA))+U(1,MNI)/D(1,MNI))
      EDT=DS*(PS/DS/(GF-1.)+US**2/DS**2/2.)
 82   IF(NTYPE(NI).EQ.2) GO TO 83
      A=SQRT(PS/DS) $ PG=PG1(A)
      PS1=(1.-BETA)/2.*(1.+     G*SL(NI)**2)    $ PDT=PS=PS1*PS
      US=(SQRT(G )*A*SQRT((1.-BETA)    *(PS1-PG)*(PS1-1.)/G
     * /(PS1+BETA))+US/DS)
      DDT=DS=DS/(1.-(1.-BETA)*(PS1-PG)/(PS1+BETA))
      UDT=US=US*DS
      EDT=DS*(PS/DS/(GF-1.)+US**2/DS**2/2.)
      DS1=DS $ PS1=PS $ US1=US
      GO TO 97
 83   GE=G $ IF(NTYPE(NJ).EQ.5) GE=GF
      US=US/DS+SQRT(GE*PS/DS)*2./(GE+1.)*SL(NI)*(1.-1./SL(NI)**2)
      DS1=DS=D2M(SL(NI),GE,MNI)/D(1,MNI)*DS
```

```
      US1=US=US*DS $ PS1=PS=P2M(SL(NI),GE,MNI)/P(1,MNI)*PS
 97   IF(NTYPE(II).EQ.5) GO TO 63
      GE=G3
      US=US/DS+SQRT(GE*PS/DS)*2./(GE+1.)*SL(II)*(1.-1./SL(II)**2)
      DS=D2M(SL(II),GE,MNI)/D(1,MNI)*DS
      US=US*DS
      PS=P2M(SL(II),GE,MNI)/P(1,MNI)*PS
      GO TO 6
 63   A=SQRT(PS/DS) $ PC=PCI(A)
      PS1=(1.-BETA)/2.*(1.+     G*SL(II)**2)     $ PDT=PS=PS1*PS
      US=(SQRT(G )*A*SQRT((1.-BETA)    *(PS1-PC)*(PS1-1.)/G
     * /(PS1+BETA))+US/DS)
      DDT=DS=DS/(1.-(1.-BETA)*(PS1-PC)/(PS1+BETA))
      UDT=US=US*DS
      EDT=DS*(PS/DS/(GE-1.)+US**2/DS**2/2.)
 6    ES=(PS/DS/(G3-1.)+US**2/DS**2/2.)*DS
      EPS=(RC(II)-R(M))/DELR
      C1=2.*(2.-EPS)/(1.+EPS)
      C2=2.*EPS-3. $ C3=(1.-EPS)*(2.*EPS-1.)/(1.+EPS)
      DPC(1,7)=D(1,M)-DTDX*(C1*US+C2*U(1,M)+C3*U(1,M-1))
      UPC(1,7)=U(1,M)-DTDX*(C1*(US*US/DS+PS*RC(II)**J)+C2*PM(M)+
     *C3*PM(M-1))
      EPC(1,7)=E(1,M)-DTDX*(C1*US/DS*(ES+PS*RC(II)**J)+C2*PE(M)+C3*
     *PE(M-1))
      PPC(1,7)=PP(7,G3)
C-----SHKII CORRECTOR FOR NSHK = 7
      D(2,M)=DPC(2,7)=DC(7,M,0) $ U(2,M)=UPC(2,7)=UC(7,M,0)
      E(2,M)=EPC(2,7)=EC(7,M,0) $ P(2,M)=PPC(2,7)=PC(7,G3)
 7    IF(NCROSS(II).NE.1) GO TO 3
C-----SHKII PREDICTOR FOR NSHK+1 = 8
      GE=G3
      DPC(1,8)=D2M(SL(II),GE,MII)
      UPC(1,8)=U2M(SL(II),GE,MII)/D(2,MII)*DPC(1,8)
      P(2,MII)=PPC(1,8)=P2M(SL(II),GE,MII)
      EPC(1,8)=DPC(1,8)*(PPC(1,8)/DPC(1,8)/(GE-1.)+UPC(1,8)**2/DPC(1,8)
     ***2/2.)
      IF(NTYPE(II).EQ.2) GO TO 3
      P(2,MII)=PPC(1,8)=PDT $ DPC(1,8)=DDT $ EPC(1,8)=EDT $ UPC(1,8)=UDT
 3    IF((NCROSS(NI).NE.1).OR.(NRHS(II).EQ.MNI)) GO TO 4
      IF((NAME.EQ.7HCTSKSKO).OR.(NAME.EQ.7HSPECIAL)) GO TO 4
C-----SHKNI PREDICTOR FOR NSHK+1 = 2
      P(2,MNI)=PPC(1,2)=P2M(SL(NI),G1,MNI)
C
C
 4    IF(NRHS(II).EQ.MNI) GO TO 10
      IF(NRHS(II).EQ.MNI+1) GO TO 20
      IF(NRHS(II).EQ.MNI+2) GO TO 30
      IF(NRHS(II).EQ.MNI+3) GO TO 24
C
 10   IF(NAME.NE.7HCTSKSKO) GO TO 80
      GO TO 22
C-----SHKNI PREDICTOR FOR NSHK = 2(=1)
 81   DPC(1,2)=DPC(1,1) $ UPC(1,2)=UPC(1,1)
      EPC(1,2)=EPC(1,1) $ PPC(1,2)=PPC(1,1)
C-----SHKNI PREDICTOR FOR NSHK-1 = 1
      M=NDISCL(NI)-1
```

```
      DPC(1,1)=DP(M,0) $ UPC(1,1)=UP(M,0)
      EPC(1,1)=EP(M,0) $ PPC(1,1)=PP(1,G2)
C-----SHKNI CORRECTOR FOR NSHK = 2
      M=NDISCL(NI)
      D(2,M)=DPC(2,2)=DC(2,M,0) $ U(2,M)=UPC(2,2)=UC(2,M,0)
      E(2,M)=EPC(2,2)=EC(2,M,0) $ P(2,M)=PPC(2,2)=PC(2,G2)
C-----SHKII CORRECTOR FOR NSHK+1+NCRSII = 1
 80   D(2,MII+NCRSII)=DPC(2,1)=DC(1,MII+NCRSII,1)
      U(2,MII+NCRSII)=UPC(2,1)=UC(1,MII+NCRSII,1)
      E(2,MII+NCRSII)=EPC(2,1)=EC(1,MII+NCRSII,1)
      P(2,MII+NCRSII)=PPC(2,1)=PC(1,G2)
 12   IF(NTYPE(II).EQ.5) GO TO 52
      SSL=SSLS(PPC(1,1),PPC(1,7+NCRSII),G2) $ SSE=SSES(SSL,G2,1,1)
      IF(NCRSII.NE.1) GO TO 11
C-----SHKII CORRECTOR FOR NSHK+1 = 8
      SSL=(SSL+SL(II))/2.0 $ NDISCL(II)=NDISCL(II)+NCRSII
      IF(NTYPE(II).EQ.5) NFLM=NDISCL(II)
      GE=G2
      D(2,MII)=DPC(2,8)=D2M(SSL,GE,MII)
      U(2,MII)=UPC(2,8)=U2M(SSL,GE,MII)
      P(2,MII)=PPC(2,8)=P2M(SSL,GE,MII) $ E(2,MII)=EPC(2,8)=E2M(GE,MII)
      GO TO 11
 52   IF(NCRSII.EQ.0) GO TO 53
      D(2,MII)=DPC(2,8)=DDT $ U(2,MII)=UPC(2,8)=UDT
      E(2,MII)=EPC(2,8)=EDT $ P(2,MII)=PPC(2,8)=PDT
      SSE=SE(II) $ NDISCL(II)=NDISCL(II)+NCRSII
      IF(NTYPE(II).EQ.5) NFLM=NDISCL(II)
      GO TO 11
 53   A=SQRT(PPC(1,1)/DPC(1,1)) $ PG=PG1(A)
      SSE=A*SQRT(G)*SQRT((2.*PG-(1.-BETA)+2.*SQRT(PG**2-(1.-BETA)*PG
     *-BETA))/(1.-BETA)/G)+UPC(1,1)/DPC(1,1)
 11   RD(II)=RD(II)+(SSE+SE(II))/2.*DT
      IF(NTYPE(II).EQ.5) GO TO 54
      SL(II)=SSLS(PPC(2,1),PPC(2,7+NCRSII),G2)
      SE(II)=SSES(SL(II),G2,2,1)
      GO TO 55
 54   A=SQRT(PPC(2,1)/DPC(2,1)) $ PG=PG1(A)
      SL(II)=SQRT((2.*PG-(1.-BETA)+2.*SQRT(PG**2-(1.-BETA)*PG-BETA))/
     *(1.-BETA)/G)
      SE(II)=SL(II)*SQRT(G)*A+UPC(2,1)/DPC(2,1)
 55   IF((RD(II).GT.R2(MII)).AND.(NCRSII.EQ.0)) RD(II)=R2(MII)-DELR*
     *1.E-10
      IF((RD(II).LT.R2(MII)).AND.(NCRSII.EQ.1)) RD(II)=R2(MII)+DELR*
     *1.E-10
      IF((RD(II).LT.R2(MII-1)).AND.(NCRSII.EQ.0)) RD(II)=R2(MII-1)
     *+DELR*1.E-10
 9    IF((RD(II).GT.RD(NI)).AND.(NAME.EQ.7H DISCDS)) GO TO 100
      IF((RD(II).GT.RD(NI)).AND.(NAME.EQ.7H DISCSD)) GO TO 100
      RETURN
C
 20   IF((NCRSII.EQ.0).AND.(NCRSNI.EQ.0)) GO TO 21
      IF((NCRSII.EQ.1).AND.(NCRSNI.EQ.0)) GO TO 22
C-----HERE NCRSII=1 AND NCRSNI=1 OR NCRSII=0 AND NCRSNI=1
C-----SHKNI COMPLETE PREDICTOR FOR NSHK+1 = 2
      IF(NAME.EQ.7HSPECIAL) GO TO 27
      GE=G1
```

```
      D(2,MNI)=DPC(1,2)=D2M(SL(NI),GE,MNI)
      U(2,MNI)=UPC(1,2)=U2M(SL(NI),GE,MNI) $ EPC(1,2)=E2M(GE,MNI)
      IF(NTYPE(NI).EQ.2) GO TO 27
      A=SQRT(P(1,MNI)/D(1,MNI)) $ PG=PG1(A)
      PS1=(1.-BETA)/2.*(1.+     G*SL(NI)**2)    $ PDT=PS=PS1*P(1,MNI)
      DDT=DS=D(1,MNI)/(1.-(1.-BETA)*(PS1-PG)/(PS1+BETA))
      UDT=US=DS*(SQRT(G )*A*SQRT((1.-BETA)    *(PS1-PG)*(PS1-1.)/G
     * /(PS1+BETA))+U(1,MNI)/D(1,MNI))
      EDT=DS*(PS/DS/(GF-1.)+US**2/DS**2/2.)
      D(2,MNI)=DPC(1,2)=DDT $ U(2,MNI)=UPC(1,2)=UDT
      E(2,MNI)=EPC(1,2)=EDT $ P(2,MNI)=PPC(1,2)=PDT
 27   IF((NCRSII.EQ.0).AND.(NCRSNI.EQ.1)) GO TO 21
C-----SHKNI PREDICTOR FOR NSHK = 1
 22   IF((NAME.EQ.7HSPECIAL).AND.(MNI-MII.EQ.2).AND.(NCRSII.EQ.1))
     *GO TO 76
      IF((NAME.EQ.7HDTSKSKO).AND.(NDISCL(NJ)+1.EQ.MNI)) GO TO 77
      GO TO 78
 77   PS=P2M(SL(NJ),G ,MNI) $ DS=D2M(SL(NJ),G ,MNI)
      US=U2M(SL(NJ),G ,MNI)/D(2,MNI)*DS
      IF(NTYPE(NJ).EQ.2) GO TO 73
      A=SQRT(P(1,MNI)/D(1,MNI)) $ PG=PG1(A)
      PS1=(1.-BETA)/2.*(1.+     G*SL(NJ)**2)    $ PDT=PS=PS1*P(1,MNI)
      DDT=DS=D(1,MNI)/(1.-(1.-BETA)*(PS1-PG)/(PS1+BETA))
      UDT=US=DS*(SQRT(G )*A*SQRT((1.-BETA)    *(PS1-PG)*(PS1-1.)/G
     * /(PS1+BETA))+U(1,MNI)/D(1,MNI))
      EDT=DS*(PS/DS/(GF-1.)+US**2/DS**2/2.)
 73   IF(NTYPE(NI).EQ.2) GO TO 74
      A=SQRT(PS/DS) $ PG=PG1(A)
      PS1=(1.-BETA)/2.*(1.+     G*SL(NI)**2)    $ PDT=PS=PS1*PS
      US=(SQRT(G )*A*SQRT((1.-BETA)    *(PS1-PG)*(PS1-1.)/G
     * /(PS1+BETA))+US/DS)
      DDT=DS=DS/(1.-(1.-BETA)*(PS1-PG)/(PS1+BETA))
      UDT=US=US*DS
      EDT=DS*(PS/DS/(GF-1.)+US**2/DS**2/2.)
      GO TO 79
 74   GE=G $ IF(NTYPE(NJ).EQ.5) GE=GF
      US=US/DS+SQRT(GE*PS/DS)*2./(GE+1.)*SL(NI)*(1.-1./SL(NI)**2)
      DS=D2M(SL(NI),GE,MNI)/D(1,MNI)*DS
      US=US*DS
      PS=P2M(SL(NI),GE,MNI)/P(1,MNI)*PS
      GO TO 79
 78   IF(NTYPE(NI).EQ.5) GO TO 60
      GE=G1
      PS=P2M(SL(NI),GE,MNI) $ DS=D2M(SL(NI),GE,MNI)
      US=U2M(SL(NI),GE,MNI)/D(2,MNI)*DS
      GO TO 79
 60   A=SQRT(P(1,MNI)/D(1,MNI)) $ PG=PG1(A)
      PS1=(1.-BETA)/2.*(1.+     G*SL(NI)**2)    $ PDT=PS=PS1*P(1,MNI)
      DDT=DS=D(1,MNI)/(1.-(1.-BETA)*(PS1-PG)/(PS1+BETA))
      UDT=US=DS*(SQRT(G )*A*SQRT((1.-BETA)    *(PS1-PG)*(PS1-1.)/G
     * /(PS1+BETA))+U(1,MNI)/D(1,MNI))
      EDT=DS*(PS/DS/(GF-1.)+US**2/DS**2/2.)
 79   ES=(PS/DS/(G2-1.)+US**2/DS**2/2.)*DS
      M=NDISCL(NI)
      EPS=(RD(NI)-R(M))/DELR
      C1=2.*(2.-EPS)/(1.+EPS)
```

```
      C2=2.*EPS-3. $ C3=(1.-EPS)*(2.*EPS-1.)/(1.+EPS)
      D(2,M)=DPC(1,1)=DPC(2,1)=D(1,M)-DTDX*(C1*US+C2*U(1,M)+C3*U(1,M-1))
      U(2,M)=UPC(1,1)=UPC(2,1)=U(1,M)-DTDX*(C1*(US*US/DS+PS*RD(II)**J)+
     *C2*PM(M)+C3*PM(M-1))
      E(2,M)=EPC(1,1)=EPC(2,1)=E(1,M)-DTDX*(C1*US/DS*(ES+PS*RD(II)**J)+
     *C2*PE(M)+C3*PE(M-1))
      P(2,M)=PPC(1,1)=PPC(2,1)=PP(1,G2)
      IF((NAME.EQ.7HDTSKSKO).AND.(NRHS(II).EQ.MNI)) GO TO 81
 76   IF((NCRSII.EQ.1).AND.(NCRSNI.EQ.0)) GO TO 24
C-----SHKNI CORRECTOR FOR NSHK = 1
 23   M=NDISCL(NI)
      D(2,M)=DPC(2,1)=DC(1,M,1) $ U(2,M)=UPC(2,1)=UC(1,M,1)
      E(2,M)=EPC(2,1)=EC(1,M,1) $ P(2,M)=PPC(2,1)=PC(1,G2)
      GO TO 24
C-----SHKII PREDICTOR AND CORRECTOR FOR NSHK = 1
 21   D(2,MII)=DPC(1,1)=DPC(2,1)=D(1,MII)
      U(2,MII)=UPC(1,1)=UPC(2,1)=U(1,MII)
      E(2,MII)=EPC(1,1)=EPC(2,1)=E(1,MII)
      P(2,MII)=PPC(1,1)=PPC(2,1)=P(1,MII)
      IF((NCRSII.EQ.0).AND.(NCRSNI.EQ.0)) GO TO 24
      GO TO 23
C-----CALC SHKNI(HERE) AND SHKII(AT 12) SPDS AND POSITION
 24   IF(NAME.EQ.7HDTSKSKO) GO TO 12
      IF(NAME.EQ.7HSPECIAL) GO TO 26
      IF(NTYPE(NI).EQ.5) GO TO 56
      MH=1+NCRSNI
      SSL=SSLS(PPC(1,3),PPC(1,MH),G1) $ SSE=SSES(SSL,G1,1,3)
      IF(NCRSNI.NE.1) GO TO 25
C-----SHKNI CORRECTOR FOR NSHK+II = 2
      SSL=(SSL+SL(NI))/2. $ NDISCL(NI)=NDISCL(NI)+NCRSNI
      IF(NTYPE(NI).EQ.5) NFLM=NDISCL(NI)
      GE=G1
      D(2,MNI)=DPC(2,2)=D2M(SSL,GE,MNI)
      U(2,MNI)=UPC(2,2)=U2M(SSL,GE,MNI)
      P(2,MNI)=PPC(2,2)=P2M(SSL,GE,MNI) $ E(2,MNI)=EPC(2,2)=E2M(GE,MNI)
      GO TO 25
 56   IF(NCRSNI.EQ.0) GO TO 57
      D(2,MNI)=DPC(2,2)=DDT $ U(2,MNI)=UPC(2,2)=UDT
      E(2,MNI)=EPC(2,2)=EDT $ P(2,MNI)=PPC(2,2)=PDT
      SSE=SE(NI) $ NDISCL(NI)=NDISCL(NI)+NCRSNI
      IF(NTYPE(NI).EQ.5) NFLM=NDISCL(NI)
      GO TO 25
 57   A=SQRT(PPC(1,3)/DPC(1,3)) $ PD=PG1(A)
      SSF=A*SQRT(G)*SQRT((2.*PG-(1.-BETA)+2.*SQRT(PG**2-(1.-BETA)*PG
     *-BETA))/(1.-BETA)/G)+UPC(1,3)/DPC(1,3)
 25   RD(NI)=RD(NI)+(SSF+SE(NI))/2.*DT
      IF(NTYPE(NI).EQ.5) GO TO 58
      SL(NI)=SSLS(PPC(2,3),PPC(2,MH),G1) $ SE(NI)=SSES(SL(NI),G1,2,3)
      GO TO 59
 58   A=SQRT(PPC(2,3)/DPC(2,3)) $ PD=PG1(A)
      SL(NI)=SQRT((2.*PG-(1.-BETA)+2.*SQRT(PG**2-(1.-BETA)*PG-BETA))/
     *(1.-BETA)/G)
      SE(NI)=SL(NI)*SQRT(G)*A+UPC(2,3)/DPC(2,3)
 59   IF((RD(NI).GT.R2(MNI)).AND.(NCRSNI.EQ.0)) RD(NI)=R2(MNI)-DELR*
     *1.E-10
      IF((RD(NI).LT.R2(MNI)).AND.(NCRSNI.EQ.1)) RD(NI)=R2(MNI)+DELR*
```

```
     *1.E-10
      IF((RD(NI).LT.R2(MNI-1)).AND.(NCRSNI.EQ.0)) RD(NI)=R2(MNI-1)
     *+DELR*1.E-10
 26   IF(NRHS(II).EQ.MNI+3) GO TO 40
      GO TO 12
C
C
 30   IF((NCRSII.EQ.0).AND.(NCRSNI.EQ.0)) GO TO 31
      IF((NCRSII.EQ.1).AND.(NCRSNI.EQ.0)) GO TO 32
C-----HERE NCRSII=1 AND NCRSNI=1 OR NCRSII=0 AND NCRSNI=1
C-----PRDCTR AND CRRCTR FOR FCTTOUS NDE =1
      IF(NAME.EQ.7HSPECIAL) GO TO 24
      IF(NTYPE(NI).EQ.5) GO TO 67
      GE=G1
      D(2,MNI)=DPC(1,1)=D2M(SL(NI),GE,MNI)
      UPC(1,1)=U2M(SL(NI),GE,MNI) $ PPC(1,1)=PPC(1,2)
      SSL=(SSLS(PPC(1,3),PPC(1,2),GE)+SL(NI))/2.
      DPC(2,1)=D2M(SSL,GE,MNI) $ PPC(2,1)=P2M(SSL,GE,MNI)
      UPC(2,1)=U2M(SSL,GE,MNI)/D(2,MNI)*DPC(2,1)
      GO TO 24
 67   A=SQRT(P(1,MNI)/D(1,MNI)) $ PG=PG1(A)
      PS1=(1.-BETA)/2.*(1.+     G*SL(NI)**2)     $ PDT=PS=PS1*P(1,MNI)
      DDT=DS=D(1,MNI)/(1.-(1.-BETA)*(PS1-PG)/(PS1+BETA))
      UDT=US=DS*(SQRT(G )*A*SQRT((1.-BETA)    *(PS1-PG)*(PS1-1.)/G
     * /(PS1+BETA))+U(1,MNI)/D(1,MNI))
      EDT=DS*(PS/DS/(GE-1.)+US**2/DS**2/2.)
      DPC(2,1)=DPC(1,1)=DDT $ UPC(2,1)=UPC(1,1)=UDT
      EPC(2,1)=EPC(1,1)=EDT $ PPC(2,1)=PPC(1,1)=PDT
      GO TO 24
C-----SHKII PRDCTR AND CRRCTR FOR FCTTOUS NDE =1
 31   GE=G1
      DPC(2,1)=DPC(1,1)=D2M(SL(NI),GE,MNI)
      PPC(2,1)=PPC(1,1)=P2M(SL(NI),GE,MNI)
      UPC(2,1)=UPC(1,1)=U2M(SL(NI),GE,MNI)/D(2,MNI)*DPC(1,1)
      IF(NTYPE(NI).EQ.2) GO TO 66
      IF(NAME.NE.7H DTSKSK) GO TO 84
      A=SQRT(P(1,MNI)/D(1,MNI)) $ PG=PG1(A)
      PS1=(1.-BETA)/2.*(1.+     G*SL(NI)**2)     $ PDT=PS=PS1*P(1,MNI)
      DDT=DS=D(1,MNI)/(1.-(1.-BETA)*(PS1-PG)/(PS1+BETA))
      UDT=US=DS*(SQRT(G )*A*SQRT((1.-BETA)    *(PS1-PG)*(PS1-1.)/G
     * /(PS1+BETA))+U(1,MNI)/D(1,MNI))
      EDT=DS*(PS/DS/(GE-1.)+US**2/DS**2/2.)
 84   DPC(2,1)=DPC(1,1)=DDT $ UPC(2,1)=UPC(1,1)=UDT
      EPC(2,1)=EPC(1,1)=EDT $ PPC(2,1)=PPC(1,1)=PDT
 66   IF((NAME.EQ.7HDTSKSKO).AND.(NDISCL(NJ)+1.EQ.MNI)) GO TO 101
      GO TO 24
 101  DPC(2,1)=DPC(1,1)=DS1 $ UPC(2,1)=UPC(1,1)=US1
      PPC(2,1)=PPC(1,1)=PS1
      GO TO 24
C-----SHKII PREDICTOR AND CORRECTOR FOR NSHKII = 1
 32   DPC(2,1)=DPC(1,1)=D(1,MII) $ UPC(2,1)=UPC(1,1)=U(1,MII)
      PPC(2,1)=PPC(1,1)=P(1,MII)
      GO TO 24
C
C-----SHKII PREDICTOR FOR NSHKII = 0
 40   IF(NTYPE(II).EQ.5) GO TO 68
```

```
      IF(NAME.EQ.7HSPECIAL) GO TO 41
      DPC(1,1)=DDT $ UPC(1,1)=UDT $ PPC(1,1)=PDT $ EPC(1,1)=EDT
 41   GE=G3
      PPC(1,8)=P2M(SL(II),GE,MII)/P(1,MII)*PPC(1,1)
      DPC(1,8)=D2M(SL(II),GE,MII)/D(1,MII)*DPC(1,1)
      UPC(1,8)=DPC(1,8)*(UPC(1,1)/DPC(1,1)+SQRT(GE*PPC(1,1)/DPC(1,1))*
     *(2./(GE+1.)*SL(II)*(1.-1./SL(II)**2)))
      EPC(1,8)=DPC(1,8)*(PPC(1,8)/DPC(1,8)/(GE-1.)+UPC(1,8)**2/DPC(1,8)
     ***2/2.)
      SSL=SSLS(PPC(1,1),PPC(1,8),GE) $ SSE=SSES(SSL,GE,1,1)
      SSL=(SSL+SL(II))/2.
      D(2,MII)=DPC(2,8)=D2M(SSL,GE,MII)/D(1,MII)*DPC(1,1)
      P(2,MII)=PPC(2,8)=P2M(SSL,GE,MII)/P(1,MII)*PPC(1,1)
      U(2,MII)=UPC(2,8)=DPC(2,8)*(UPC(1,1)/DPC(1,1)+SQRT(GE*PPC(1,1)/
     *DPC(1,1))*(2./(GE+1.)*SSL*(1.-1./SSL**2)))
      E(2,MII)=EPC(2,8)=E2M(GE,MII)
      GO TO 69
 68   D(2,MII)=DPC(2,8)=DPC(1,8)=DDT $ U(2,MII)=UPC(2,8)=UPC(1,8)=UDT
      E(2,MII)=EPC(2,8)=EPC(1,8)=EDT $ P(2,MII)=PPC(2,8)=PPC(1,8)=PDT
      SSE=SE(II)
 69   DPC(2,1)=DPC(2,2) $ UPC(2,1)=UPC(2,2)
      EPC(2,1)=EPC(2,2) $ PPC(2,1)=PPC(2,2)
      NDISCL(II)=NDISCL(II)+NCRSII
      IF(NTYPE(II).EQ.5) NFLM=NDISCL(II)
      GO TO 11
C
C
 100  TARFLCT=(RD(II)-RD(NI))/(SE(II)-SE(NI))
      RD(NI)=RD(II)=RD(NI)-SE(NI)*TARFLCT
      IF(NAME.EQ.7H DISCSD) CALL SHKDETA(II,NI,TARFLCT)
      IF(NAME.EQ.7H DISCDS) CALL DETSHKA(II,NI,TARFLCT)
C
      RETURN $ END
```

```
      SUBROUTINE SHKDETA(II,NI,TARFLCT)
      COMMON/PARAM/N,J,AJ,G,GF,DELR,NFLM
C
      DELR=-ABS(DELR)
      PRINT 1000
 1000 FORMAT(1H ,2X,*SHKDETA STOP*)
C
      RETURN $ END
```

```
      SUBROUTINE DETSHKA(II,NI,TARFLCT)
      COMMON/PARAM/N,J,AJ,G,GF,DELR,NFLM
      COMMON/TRCT/PLTR,DLTR,ULTR,ELTR,OVERDNM,NOVERDN
      COMMON/FIRFETC/INDEX,NCYCLE,NN,NNN,NSTORE,NS,NITRCTN
      COMMON/POWER/VCAPF,R4R1,PDPOWER,PPOWER,CND,CCAPF,SFECLD,SFE,NCJ
      COMMON/ARRAYS/R(501),U(2,501),P(2,501),D(2,501),E(2,501),R2(501)
      COMMON/DISCS/NTDISC,NDISCNO(51),NTYPE(51),NDISCL(51),SE(51),SL(51)
     *,RD(51),NTDISCT,NTDISCS
C
C
C-----OVERDNM=OVERDRIVEN DETONATION MACH NO.
C-----NOVERDN=NO. OF TIME STEPS DET HAS BEEN OVERDRIVEN
C
      PG1(A)=1.+(PG0-1.)/A**2+GF*(1.-1./A**2)*(G/GF*B-R4R1)
      BETA=(GF-1.)/(GF+1.) $ B=(GF-1.)/(G-1.)
      PG0=(1.+BETA)*(VCAPF-BETA)/(1.-BETA)-BETA
C
C-----ASSUME OVERDRIVEN DET INSTTSLY DECYS TO A DFLAG PRESS EQUAL TO THE
C      AVG OF (PREV CJ + DFLAG PRESS CORS TO A RL PASSNG THRU PREV CJ)/2
      NDISCL(NI)=-1 $ NSHK=NDISCL(II)
      A=SQRT(P(2,NSHK+1)/D(2,NSHK+1)) $ PG=PG1(A)
      AM2=(P(2,NSHK)/P(2,NSHK+1)-1.)/(1.-D(2,NSHK+1)/D(2,NSHK))/G
      PDFLAG=(1.-BETA)/2.*(1.+G*AM2)
      PDFLAG=(PDFLAG+SQRT(PDFLAG**2-G*(1.-BETA)*AM2*PG+BETA))*P(2,NSHK+1
     *)
      PAVG=(PDFLAG+P(2,NSHK))/2.
      PRATIO=PAVG/P(2,NSHK+1)
      SL(II)=OVERDNM=SQRT((PRATIO+BETA)*(PRATIO-1.)/(1.-BETA)/(PRATIO
     *-PG)/G)
      SE(II)=SL(II)*A*SQRT(G)+U(2,NSHK+1)/D(2,NSHK+1)
      NOVERDN=0
C
      RD(II)=RD(II)+SE(II)*TARFLCT
      IF(RD(II).LT.R2(NSHK+1)) GO TO 1
      IF(RD(II).GT.R2(NSHK+2)) RD(II)=R2(NSHK+2)
      P(2,NSHK+1)=PAVG
      U(2,NSHK+1)=SQRT(G)*A*SQRT((1.-BETA)*(PRATIO-PG)*(PRATIO-1.)/
     *G/(PRATIO+BETA))+U(2,NSHK+1)/D(2,NSHK+1)
      D(2,NSHK+1)=D(2,NSHK+1)/(1.-(1.-BETA)*(PRATIO-PG)/(PRATIO+BETA))
      U(2,NSHK+1)=U(2,NSHK+1)*D(2,NSHK+1)
      E(2,NSHK+1)=D(2,NSHK+1)*(P(2,NSHK+1)/D(2,NSHK+1)/(GF-1.)+
     *U(2,NSHK+1)**2/D(2,NSHK+1)**2/2.)
      NFLM=NDISCL(II)=NDISCL(II)+1
C
 1    NCODE=10HDETSHKA     $ CALL PRNTFF(NCODE)
      NITRCTN=3HYES
C
      RETURN $ END
```

```
      SUBROUTINE SCTRDTR(N1,N2,N3,NTR)
      COMMON/DISCS/NTDISC,NDISCNC(51),NTYPE(51),NDISCL(51),SF(51),SL(51)
     *,RD(51),NTDISCT,NTDISCS
C
      NTR=-1
      PRINT 1000
 1000 FORMAT(1H ,5/,5X,*TERMINATING TAIL OF RARE (ASSOC WITH CJ-DET) IN
     *SUBROUTINE SCTRDTR*)
C
      CALL SHKCD(N1,N2,8HSKCD DET)
      NTYPE(N2)=3
      PRINT 1001
 1001 FORMAT(1H ,/,5X,*CD TYPE SET TO 3 IN SCTRDTR*)
C
      RETURN $ END
```

```
      SUBROUTINE SCDTDET(N1,NXS,N2,NXCD,N3,NXTR,N4,NXD,NAME)
      COMMON/PARAM/N,J,AJ,G,GF,DELR,NFLM
      COMMON/TIME/T,DT,DTL,TWRITE,DELT,DTCX,AT
      COMMON/PREDCOR/DPC(2,13),UPC(2,13),EPC(2,13),PPC(2,13)
      COMMON/ARRAYS/R(501),U(2,501),P(2,501),D(2,501),E(2,501),R2(501)
      COMMON/DISCS/NTDISC,NDISCNC(51),NTYPE(51),NDISCL(51),SE(51),SL(51)
     *,RD(51),NTDISCT,NTDISCS
C
C
      NTR=NDISCL(N3) $ NCD=NDISCL(N2)
      NSC=100*(NTR-NCD-1)+10*NXCD+NXTR
      PRINT 1000,NAME,NSC,(I,NDISCL(I),NTYPE(I),I=N1,N4),NXS,NXCD,NXTR,
     *NXD
 1000 FORMAT(1H ,1X,A4,17I5)
      IF(NXTR.NE.-1) GO TO 1
      PRINT 1001
 1001 FORMAT(1H ,5/,5X,*TROUBLE IN SCDTDET, NXTR=-1*,2/)
      DELR=-ABS(DELR) $ RETURN
C
 1    IF(NAME.EQ.4H2675) CALL SHKCD(N1,N2,8HSKCD DET)
      IF(NAME.EQ.4H6750) CALL CD(RD(N2),SE(N2),NDISCL(N2),5HCDTRD)
      NAMES=8HSCTD YES $ IF(NSC.LT.200) NAMES=8HSCTD  NO
      CALL TRDET(N3,NXTR,N4,NXD,NAMES)
      IF((NSC.EQ.0).OR.(NSC.EQ.-10).OR.(NSC.EQ.-9).OR.(NSC.EQ.1).OR.
     *(NSC.EQ.110).OR.(NSC.EQ.111)) RETURN
      IF(NSC.NE.11) GO TO 5
C-----CORRECTOR FOR NDISCL(N2)-1
      M=NDISCL(N2)-1
      D(2,M)=D(2,M+1) $U(2,M)=U(2,M+1) $E(2,M)=E(2,M+1) $P(2,M)=P(2,M+1)
      RETURN
 5    IF((NSC.NE.210).AND.(NSC.NE.211)) GO TO 4
C-----CORRECTOR FOR NCD+3 = (P12)
      M=NCD+3
      D(2,M)=DPC(1,12) $ U(2,M)=UPC(1,12)
      E(2,M)=EPC(1,12) $ P(2,M)=PPC(1,12)
      RETURN
C-----PREDICTOR FOR NDISCL(N2)+1+(NXCD-1)*NXCD/2 = 11
 4    M=NDISCL(N2)+1+(NXCD-1)*NXCD/2
      DPC(1,11)=D(2,M) $ UPC(1,11)=U(2,M)
      EPC(1,11)=E(2,M) $ PPC(1,11)=P(2,M)
      IF((NSC.EQ.200).OR.(NSC.EQ.201).OR.(NSC.EQ.310).OR.(NSC.EQ.311))
     *GO TO 2
C-----PREDICTOR AND CORRECTOR FOR NCD+2 = 12
      M=NCD+2
      D(2,M)=DPC(2,12)=DPC(1,12)=D(1,M)
      U(2,M)=UPC(2,12)=UPC(1,12)=U(1,M)
      E(2,M)=EPC(2,12)=EPC(1,12)=E(1,M)
      P(2,M)=PPC(2,12)=PPC(1,12)=P(1,M)
      IF(NSC.EQ.100) RETURN
      IF(NSC.EQ.101) GO TO 3
C-----CORRECTOR FOR NDISCL(N2)+2 = 12
      D(2,M)=DC(12,M,0) $ U(2,M)=UC(12,M,0)
      E(2,M)=EC(12,M,0) $ P(2,M)=PC(12,CF)
      RETURN
C-----PREDICTOR FOR NCD+3 = 13
 3    M=NCD+3
```

```
      DPC(2,13)=D(2,M) $ UPC(2,13)=U(2,M)
      EPC(2,13)=E(2,M) $ PPC(2,13)=P(2,M)
C-----CORRECTOR FOR NCD+2+NXCD = 12
 2    M=NCD+2+NXCD
      EPS=(R2(M)-RD(N2))/DELR
      C1=2.*(2.-EPS)/(1.+EPS)
      C2=2.*EPS-3. $ C3=(1.-EPS)*(2.*EPS-1.)/(1.+EPS)
      D(2,M)=(D(1,M)+DPC(1,12)+DTDX*(C1*UPC(1,11)+C2*UPC(1,12)+C3*UPC(1,
     *13)))/2.
      U(2,M)=(U(1,M)+UPC(1,12)+DTDX*(C1*CM(11)+C2*CM(12)+C3*CM(13)))/2.
      E(2,M)=(E(1,M)+EPC(1,12)+DTDX*(C1*CE(11)+C2*CE(12)+C3*CE(13)))/2.
      DPC(2,12)=D(2,M) $ UPC(2,12)=U(2,M)
      EPC(2,12)=E(2,M) $ P(2,M)=PC(12,GF)
C
      RETURN $ END
```

```
      SUBROUTINE CDFLM(NAME,NFIRST,N1)
      COMMON/PARAM/N,J,AJ,G,GF,DELR,NFLM
      COMMON/TIME/T,DT,DTL,TWRITE,DELT,DTDX,AT
      COMMON/PREDCOR/DPC(2,13),UPC(2,13),EPC(2,13),PPC(2,13)
      COMMON/ARRAYS/R(501),U(2,501),P(2,501),D(2,501),E(2,501),R2(501)
      COMMON/DISCS/NTDISC,NDISCNC(51),NTYPE(51),NDISCL(51),SE(51),SL(51)
     *,RD(51),NTDISCT,NTDISCS
C
C
      RCD=RD(NFIRST) $ NCD=NDISCL(NFIRST) $ SECD=SE(NFIRST)
      RF=RD(N1) $ SFE=SE(N1) $ SFL=SL(N1)
C
      RCDSAV=RCD $ RCD=RCD+SECD*DT
      NCROSS=0
      IF(RCD.GT.R2(NCD+2)) NCROSS=1 $ IF(RCD.LT.R2(NCD-1)) NCROSS=-1
      NFLMCDX=10*(NFLM-NCD)+NCROSS
      NSC=100*(NDISCL(N1)-NDISCL(NFIRST)-1)+10*NCROSS
      PRINT 1001,NAME,NFLMCDX,NCROSS,NSC
 1001 FORMAT(1H ,1X,A4,3I5)
      IF(NFLMCDX.NE.21) GO TO 9
      PRINT 1000,RCDSAV,RCD,SECD,DT,R(NFLM),R2(NFLM),SFE,SFL,NCD,NFLM
 1000 FORMAT(1H ,8E14.7,2I5,/,* SUBROUTINE CD-FLM*)
      RETURN
C
 9    IF(NAME.EQ.4H CDS) GO TO 2
C-----PREDICTOR AND CORRECTOR FOR NFLM+1 = 8
      M=NFLM+1
      D(2,M)=DPC(2,8)=DPC(1,8)=DP(M,0) $U(2,M)=UPC(2,8)=UPC(1,8)=UP(M,0)
      E(2,M)=EPC(2,8)=EPC(1,8)=EP(M,0) $P(2,M)=PPC(2,8)=PPC(1,8)=PP(8,G)
C-----PREDICTOR AND CORRECTOR FOR NFLM = 7
      CALL SFVDPD(G,GF,PPC(1,8),DPC(1,8),SFL,VD,PD,DELR)
      D(2,NFLM)=DPC(2,7)=DPC(1,7)=DPC(1,8)/VD
      P(2,NFLM)=PPC(2,7)=PPC(1,7)=PPC(1,8)*PD
      U(2,NFLM)=UPC(2,7)=UPC(1,7)=DPC(1,7)*(SFL*(1.-VD)+UPC(1,8)/
     *DPC(1,8))
      E(2,NFLM)=EPC(2,7)=EPC(1,7)=DPC(1,7)*(PPC(1,7)/DPC(1,7)/(GF-1.)
     *+UPC(1,7)**2/DPC(1,7)**2/2.)
      RD(N1)=RF=R2(NFLM)=R2(NFLM+1)=RF+(SFL+UPC(1,8)/DPC(1,8))*DT
 2    IF(NFLMCDX.EQ.19) GO TO 3
C-----PREDICTOR FROR NCD-2 = 1
      M=NCD-2
      DPC(1,1)=DP(M,0) $ UPC(1,1)=UP(M,0)
      EPC(1,1)=EP(M,0) $ PPC(1,1)=PP(1,GF)
C-----PREDICTOR FOR NCD-1 = 2
      M=NCD-1
      EPS=(R(NCD)-R(M))/DELR
      C1=2.*(2.-EPS)/(1.+EPS)
      C2=2.*EPS-3. $ C3=(1.-EPS)*(2.*EPS-1.)/(1.+EPS)
      DPC(1,2)=D(1,M)-DTDX*(C1*U(1,NCD)+C2*U(1,M)+C3*U(1,M-1))
     *-AT*DTDX*(C1*D(1,NCD)+C2*D(1,M)+C3*D(1,M-1))
      UPC(1,2)=U(1,M)-DTDX*(C1*FM(NCD)+C2*FM(M)+C3*FM(M-1))
     *-AT*DTDX*(C1*U(1,NCD)+C2*U(1,M)+C3*U(1,M-1))
      EPC(1,2)=E(1,M)-DTDX*(C1*PE(NCD)+C2*PE(M)+C3*PE(M-1))
     *-AT*DTDX*(C1*E(1,NCD)+C2*E(1,M)+C3*E(1,M-1))
      PPC(1,2)=PP(2,GF)
C-----CORRECTOR FOR NCD-1 = 2
```

```
      D(2,M)=DPC(2,2)=DC(2,M,0) $ U(2,M)=UPC(2,2)=UC(2,M,0)
      E(2,M)=EPC(2,2)=EC(2,M,0) $ P(2,M)=PPC(2,2)=PC(2,GF)
C-----PREDICTOR AND CORRECTOR FOR NCD,NCD+1 = 3,4
 3    CALL GLIM(NCD-1,NCD+2,3,NCD,SECD)
      RCD=RCDSAV+SECD*DT
      IF(((RCD.GT.R2(NCD+2)).AND.(NCROSS.EQ.0)).OR.((RCD.LT.R2(NCD+2)).
     *AND.(NCROSS.EQ.1))) RCD=R2(NCD+2)
      IF(((RCD.LT.R2(NCD-1)).AND.(NCROSS.EQ.0)).OR.((RCD.GT.R2(NCD-1)).
     *AND.(NCROSS.EQ.-1))) RCD=R2(NCD-1)
      IF(NCROSS) 4,5,6
C-----PREDICTOR AND CORRECTOR FOR NCD-1 = 2,6(=4)
 4    M=NCD-1
      D(2,M)=DPC(2,2)=DPC(1,2)=DPC(2,6)=DPC(1,6)=DPC(1,4)
      U(2,M)=UPC(2,2)=UPC(1,2)=UPC(2,6)=UPC(1,6)=UPC(1,4)
      E(2,M)=EPC(2,2)=EPC(1,2)=EPC(2,6)=EPC(1,6)=EPC(1,4)
      P(2,M)=PPC(2,2)=PPC(1,2)=PPC(2,6)=PPC(1,6)=PPC(1,4)
      GO TO 8
C-----PREDICTOR AND CORRECTOR FOR NCD+2 = 5(=3)
 6    M=NCD+2
      D(2,M)=DPC(2,5)=DPC(1,5)=DPC(1,3)
      U(2,M)=UPC(2,5)=UPC(1,5)=UPC(1,3)
      E(2,M)=EPC(2,5)=EPC(1,5)=EPC(1,3)
      P(2,M)=PPC(2,5)=PPC(1,5)=PPC(1,3)
 5    IF((NFLMCDX.EQ.20).OR.(NFLMCDX.EQ.31)) GO TO 10
C-----PREDICTOR FOR NFLM-1 = 6
      M=NFLM-1
      DPC(1,6)=DP(M,0) $ UPC(1,6)=UP(M,0)
      EPC(1,6)=EP(M,0) $ PPC(1,6)=PP(6,GF)
C-----CORRECTOR FOR NFLM-1 = 6
      EPS=(R2(NCD+2)-RCD)/DELR
      C1=2.*(2.-EPS)/(1.+EPS)
      C2=2.*EPS-3. $ C3=(1.-EPS)*(2.*EPS-1.)/(1.+EPS)
      D(2,M)=DPC(2,6)=(D(1,M)+DPC(1,6)+DTDX*(C1*UPC(1,4)+C2*UPC(1,6)+
     *C3*UPC(1,7))-AT*DTDX*(C1*DPC(1,4)+C2*DPC(1,6)+C3*DPC(1,7)))/2.
      U(2,M)=UPC(2,6)=(U(1,M)+UPC(1,6)+DTDX*(C1*CM(4)+C2*CM(6)+C3*CM(7))
     *-AT*DTDX*(C1*UPC(1,4)+C2*UPC(1,6)+C3*UPC(1,7)))/2.
      E(2,M)=EPC(2,6)=(E(1,M)+EPC(1,6)+DTDX*(C1*CE(4)+C2*CE(6)+C3*CE(7))
     *-AT*DTDX*(C1*EPC(1,4)+C2*EPC(1,6)+C3*EPC(1,7)))/2.
      P(2,M)=PPC(2,6)=PC(6,GF)
C-----CORRECTOR FOR NFLM = 7
 8    D(2,NFLM)=DPC(2,7)=DC(7,NFLM,0) $ U(2,NFLM)=UPC(2,7)=UC(7,NFLM,0)
      E(2,NFLM)=EPC(2,7)=EC(7,NFLM,0) $ P(2,NFLM)=PPC(2,7)=PC(7,GF)
C-----CORRECTOR FOR NFLM+1 = 8
 7    VD=DPC(1,8)/DPC(1,7)
      CALL FLM43(7,VD,SFL)
      M=NFLM+1
      D(2,M)=DPC(2,8) $ U(2,M)=UPC(2,8)
      E(2,M)=EPC(2,8) $ P(2,M)=PPC(2,8)
      SL(N1)=SFL $ SE(N1)=SFL+U(2,M)/D(2,M)
C-----ADVANCE CD CELL POSITION NUMBER IF NECCESSARY
 10   IF(NCROSS.EQ.0) GO TO 16
      M=NCD+2+3*(NCROSS-1)/2
      PS=P(2,M) $ DS=D(2,M) $ US=U(2,M) $ ES=E(2,M) $ RS=R2(M)
      IF(NCROSS.EQ.-1) GO TO 15
      P(2,NCD+2)=P(2,NCD+1) $ P(2,NCD+1)=P(2,NCD) $ P(2,NCD)=PS
      U(2,NCD+2)=U(2,NCD+1) $ U(2,NCD+1)=U(2,NCD) $ U(2,NCD)=US
```

```
      E(2,NCD+2)=E(2,NCD+1) $ E(2,NCD+1)=E(2,NCD) $ E(2,NCD)=ES
      D(2,NCD+2)=D(2,NCD+1) $ D(2,NCD+1)=D(2,NCD) $ D(2,NCD)=DS
      R2(NCD)=RS
      GO TO 16
 15   P(2,NCD-1)=P(2,NCD) $ P(2,NCD)=P(2,NCD+1) $ P(2,NCD+1)=PS
      U(2,NCD-1)=U(2,NCD) $ U(2,NCD)=U(2,NCD+1) $ U(2,NCD+1)=US
      D(2,NCD-1)=D(2,NCD) $ D(2,NCD)=D(2,NCD+1) $ D(2,NCD+1)=DS
      E(2,NCD-1)=E(2,NCD) $ E(2,NCD)=E(2,NCD+1) $ E(2,NCD+1)=ES
      R2(NCD+1)=RS
C
 16   NDISCL(NFIRST)=NCD=NCD+NCROSS $ RC(NFIRST)=R2(NCD)=R2(NCD+1)=RCD
      SE(NFIRST)=SECD
C
      RETURN $ END
```

```
      SUBROUTINE CDFLWSK(NFIRST,NSECOND,N1)
C
      CALL FLMSHK(4HCFS1,NSECOND,N1)
      CALL CDFLW(4H CFS,NFIRST,NSECOND)
      CALL FLMSHK(4HCFS2,NSECOND,N1)
C
      RETURN $ END
```

```
      SUBROUTINE CDSHK(NFIRST,N1,NAME)
      COMMON/UBDCND/NUBC
      COMMON/PARAM/N,J,AJ,G,GF,DELR,NFLM
      COMMON/TIME/T,DT,DTL,TWRITE,DELT,DTDX,AT
      COMMON/DISCSKF/RDSAV(51),NLHS(51),NRHS(51),NCROSS(51)
      COMMON/PREDCOR/DPC(2,13),UPC(2,13),EPC(2,13),PPC(2,13)
      COMMON/ARRAYS/R(501),U(2,501),F(2,501),D(2,501),E(2,501),R2(501)
      COMMON/DISCS/NTDISC,NDISCNC(51),NTYPE(51),NDISCL(51),SE(51),SL(51)
     *,RD(51),NTDISCT,NTDISCS
      COMMON/PPCHR/NPPTH,NUMA,NUMAFST,NUFA,RPPTH(24),RUMA(150),RUPA(150)
     *,NCLPPTH(24),NCLUMA(150),NCLUFA(150),PTHNEXT,RUMANXT,TUFANXT,
     *DELPPTH,DELUMA,DELUFA
C
C
      RCD=RD(NFIRST) $ SECD=SE(NFIRST) $ NCD=NDISCL(NFIRST)
      RS=RD(N1) $ SSL=SL(N1) $ SSE=SE(N1) $ NSHK=NDISCL(N1)
C
      GE=G $ IF(NSHK.LE.NFLM) GE=GF
      NSHKSGN=0 $ IF(SSL.GT.0.0) NSHKSGN=1
      SECDSAV=SECD $ SSESAV=SSE
      RSSAV=RS $ RS=RS+SSE*DT $ RCDSAV=RCD $ RCD=RCD+SECD*DT
      NXSS=NXSCD=0 $ MCS=0 $ IF(NSHK.EQ.NCD+1) MCS=-2
      IF(RS.GT.R2(NSHK+1)) NXSS=1 $ IF(RS.LT.R2(NSHK+MCS)) NXSS=-1
      IF(RCD.GT.R2(NCD+2)) NXSCD=1 $ IF(RCD.LT.R2(NCD-1)) NXSCD=-1
      NCDSHKX=0 $ IF(RS.LT.RCD) NCDSHKX=1
      NSC=100*(NSHK-1-NCD)+10*NXSCD+NXSS
      PRINT 1001,NAME,NXSS,NXSCD,NCDSHKX,NSC
 1001 FORMAT(1H ,1X,A6,4I5)
      IF((NSC.EQ.-1).AND.(NCDSHKX.EQ.0)) NXSS=0
      IF((NSC.EQ.-1).AND.(NCDSHKX.EQ.0)) NSC=0
      IF((SL(N1).LT.0.0).AND.(N1.EQ.3).AND.(ABS(SL(1)).LT.1.001)) GO TO
     * 100
      IF((SL(N1).LT.0.0).AND.(N1.EQ.4).AND.(ABS(SL(1)).LT.1.001)) GO TO
     * 100
C
C
      IF((NAME.EQ.6HSCDEQ1).OR.(NAME.EQ.6HSCDCT1).OR.(NXSCD.EQ.-1))
     * GO TO 1
C------PREDICTOR FOR NCD-2 = 4
      M=NCD-2
      DPC(1,4)=DP(M,0) $ UPC(1,4)=UP(M,0)
      EPC(1,4)=EP(M,0) $ PPC(1,4)=PP(4,GE)
C------PREDICTOR FOR NCD-1 = 5
      M=NCD-1
      EPS=(RCDSAV-R(M))/DELR
      C1=2.*(2.-EPS)/(1.+EPS)
      C2=2.*EPS-3. $ C3=(1.-EPS)*(2.*EPS-1.)/(1.+EPS)
      DPC(1,5)=D(1,M)-DTDX*(C1*U(1,NCD)+C2*U(1,M)+C3*U(1,M-1))
     *-AT*DTDX*(C1*D(1,NCD)+C2*D(1,M)+C3*D(1,M-1))
      UPC(1,5)=U(1,M)-DTDX*(C1*FM(NCD)+C2*FM(M)+C3*FM(M-1))
     *-AT*DTDX*(C1*U(1,NCD)+C2*U(1,M)+C3*U(1,M-1))
      EPC(1,5)=E(1,M)-DTDX*(C1*PE(NCD)+C2*PE(M)+C3*PE(M-1))
     *-AT*DTDX*(C1*E(1,NCD)+C2*E(1,M)+C3*E(1,M-1))
      PPC(1,5)=PP(5,GE)
C------CORRECTOR FOR NCD-1 = 5
      D(2,M)=DPC(2,5)=DC(5,M,0) $ U(2,M)=UPC(2,5)=UC(5,M,0)
```

```
      E(2,M)=EPC(2,5)=EC(5,M,0) $ P(2,M)=PPC(2,5)=PC(5,GE)
C-----PREDICTOR AND CORRECTOR FOR NCD,NCD+1 = 6,7
 1    M=NCD-1 $ IF(NAME.EQ.6HSCDEC1) M=NCD
      MM=NCD+1 $ IF(NSHK-NCD-1.GE.1) MM=NCD+2
      CALL GLIM(M,MM,6,NCD,SECD)
      IF(NAME.NE.6H RARCD) GO TO 27
      D(2,NCD)=DPC(1,6)=DPC(2,6)=D(1,NCD)
      U(2,NCD)=UPC(1,6)=UPC(2,6)=U(1,NCD)
      E(2,NCD)=EPC(1,6)=EPC(2,6)=E(1,NCD)
      P(2,NCD)=PPC(1,6)=PPC(2,6)=P(1,NCD)
      GO TO 28
 27   IF(NAME.NE.6HCDRSHK) GO TO 28
      D(2,NCD+1)=DPC(1,7)=DPC(2,7)=D(1,NCD+1)
      U(2,NCD+1)=UPC(1,7)=UPC(2,7)=U(1,NCD+1)
      E(2,NCD+1)=EPC(1,7)=EPC(2,7)=E(1,NCD+1)
      P(2,NCD+1)=PPC(1,7)=PPC(2,7)=P(1,NCD+1)
 28   RCD=RCDSAV+SECD*DT
      IF(((RCD.GT.R2(NCD+2)).AND.(NXSCD.EQ.0)).OR.((RCD.LT.R2(NCD+2)).
     *AND.(NXSCD.EQ.1))) RCD=R2(NCD+2)
      IF(((RCD.LT.R2(NCD-1)).AND.(NXSCD.EQ.0)).OR.((RCD.GT.R2(NCD-1)).
     *AND.(NXSCD.EQ.-1))) RCD=R2(NCD-1)
C
      IF(NXSCD.EQ.0) GO TO 2
C-----PDTR,CRTR FR NCD+(3*NXSCD+1)/2=6+(3*NXSCD+1)/2=P(6-(NXSCD-1)/2)
      M=NCD+(3*NXSCD+1)/2 $ MM=6+(3*NXSCD+1)/2 $ MMM=6-(NXSCD-1)/2
      D(2,M)=DPC(1,MM)=DPC(2,MM)=DPC(1,MMM)
      U(2,M)=UPC(1,MM)=UPC(2,MM)=UPC(1,MMM)
      E(2,M)=EPC(1,MM)=EPC(2,MM)=EPC(1,MMM)
      P(2,M)=PPC(1,MM)=PPC(2,MM)=PPC(1,MMM)
 2    IF(NXSS.EQ.0) GO TO 3
C-----PREDICTOR FOR NSHK+(NXSS+1)/2 = 11
      M=NSHK+(NXSS+1)/2
      PPC(1,11)=P(1,M)*(2.*GE/(GE+1.)*SSL**2-(GE-1.)/(GE+1.))
      DPC(1,11)=D(1,M)/((GE-1.)/(GE+1.)+2./(GE+1.)/SSL**2)
      IF(NXSS*NSHKSGN.EQ.-1) GO TO 40
      UPC(1,11)=DPC(1,11)*(U(1,M)/D(1,M)+SQRT(GE*P(1,M)/D(1,M))*2./
     *(GE+1.)*SSL*(1.-1./SSL**2))
      GO TO 41
 40   PPC(1,11)=P(1,M)**2/PPC(1,11) $ DPC(1,11)=D(1,M)**2/DPC(1,11)
      UPC(1,11)=DPC(1,11)*(U(1,M)/D(1,M)-SQRT(GE*P(1,M+1)/D(1,M+1))*2./
     *(GE+1.)*SSL*(1.-1./SSL**2))
 41   EPC(1,11)=DPC(1,11)*(PPC(1,11)/DPC(1,11)/(GE-1.)+UPC(1,11)**2/
     *DPC(1,11)**2/2.)
C-----PREDICTOR FOR NSHK+1+(NXSS+1)*NXSS/2 = 12
 3    M=NSHK+1+(NXSS+1)*NXSS/2
      DPC(1,12)=DP(M,0) $ UPC(1,12)=UP(M,0)
      EPC(1,12)=EP(M,0) $ PPC(1,12)=PP(12,GE)
      IF(NXSS.EQ.-1) GO TO 4
C-----PREDICTOR FOR NSHK+2+NXSS = 13
      M=NSHK+2+NXSS
      DPC(1,13)=DP(M,0) $ UPC(1,13)=UP(M,0)
      EPC(1,13)=EP(M,0) $ PPC(1,13)=PP(13,GE)
      IF((NXSS.EQ.0).AND.(NSHKSGN.EQ.0)) GO TO 5
C-----CORRECTOR FOR NSHK+1+NXSS*(NXSS+1)/2 = 12
 4    M=NSHK+1+NXSS*(NXSS+1)/2
      MM=NXSS*(NXSS+1)/2-(NXSS+1)*(NXSS-1)
```

```
      D(2,M)=DPC(2,12)=DC(12,M,MM) $ U(2,M)=UPC(2,12)=UC(12,M,MM)
      E(2,M)=EPC(2,12)=EC(12,M,MM) $ P(2,M)=PPC(2,12)=PC(12,GE)
C
C
C
 5    IF((NSC.NE.311).AND.(NSC.NE.310).AND.(NSC.NE.211).AND.(NSC.NE.210)
     *.AND.(NSC.NE.200).AND.(NSC.NE.309).AND.(NSC.NE.189).AND.(NSC.NE.
     *199).AND.(NSC.NE.201)) GO TO 10
      IF((NSHKSGN.EQ.0).OR.(NSHKSGN*NXSS.EQ.-1)) GO TO 6
C-----PREDICTOR AND CORRECTOR FOR NSHK = 10
      DS=D(1,NSHK+1)/((GE-1.)/(GE+1.)+2./(GE+1.)/SSL**2)
      PS=P(1,NSHK+1)*(2.*GE/(GE+1.)*SSL**2-(GE-1.)/(GE+1.))
      US=DS*(U(1,NSHK+1)/D(1,NSHK+1)+SQRT(GE*P(1,NSHK+1)/
     *D(1,NSHK+1))*2./(GE+1.)*SSL*(1.-1./SSL**2))
      ES=DS*(PS/DS/(GE-1.)+US**2/DS**2/2.)
      EPS=(RSSAV-R(NSHK))/DELR
      C1=2.*(2.-EPS)/(1.+EPS)
      C2=2.*EPS-3. $ C3=(1.-EPS)*(2.*EPS-1.)/(1.+EPS)
      D(2,NSHK)=DPC(2,10)=DPC(1,10)=D(1,NSHK)-DTDX*(C1*US+C2*U(1,NSHK)+
     *C3*U(1,NSHK-1))-AT*DTDX*(C1*DS+C2*D(1,NSHK)+C3*D(1,NSHK-1))
      U(2,NSHK)=UPC(2,10)=UPC(1,10)=U(1,NSHK)-DTDX*(C1*(US**2/DS+PS)+C2
     **PM(NSHK)+C3*PM(NSHK-1))-AT*DTDX*(C1*US+C2*U(1,NSHK)+C3*U(1,NSHK-1
     *))
      E(2,NSHK)=EPC(2,10)=EPC(1,10)=E(1,NSHK)-DTDX*(C1*US/DS*(ES+PS)+C2
     **PE(NSHK)+C3*PE(NSHK-1))-AT*DTDX*(C1*ES+C2*E(1,NSHK)+C3*E(1,NSHK-1
     *))
      P(2,NSHK)=PPC(2,10)=PPC(1,10)=PP(10,GE)
      GO TO 7
C-----PREDICTOR AND CORRECTOR FOR NSHK+NXSS = 10
 6    M=NSHK+NXSS $ MM=NXSS*NXSS-1
      D(2,M)=DPC(2,10)=DPC(1,10)=DP(M,MM)
      U(2,M)=UPC(2,10)=UPC(1,10)=UP(M,MM)
      E(2,M)=EPC(2,10)=EPC(1,10)=EP(M,MM)
      P(2,M)=PPC(2,10)=PPC(1,10)=PP(10,GE)
C
 7    IF((NSC.NE.310).AND.(NSC.NE.311).AND.(NSC.NE.200).AND.(NSC.NE.201)
     *) GO TO 8
C-----PREDICTOR FOR NSHK-1 = 9
      M=NSHK-1
      DPC(1,9)=DP(M,0) $ UPC(1,9)=UP(M,0)
      EPC(1,9)=EP(M,0) $ PPC(1,9)=PP(9,GE)
C-----CORRECTOR FOR NSHK = 10
      M=NSHK
      D(2,NSHK)=DPC(2,10)=DC(10,M,0) $ U(2,NSHK)=UPC(2,10)=UC(10,M,0)
      E(2,NSHK)=EPC(2,10)=EC(10,M,0) $ P(2,NSHK)=PPC(2,10)=PC(10,GE)
      GO TO 8
 10   IF((NSC.NE.101).AND.(NSC.NE.91).AND.(NSC.NE.100).AND.(NSC.NE.90))
     *GO TO 15
C-----PREDICTOR AND CORRECTOR FOR NSHK = 10(=(1,NSHK))
      D(2,NSHK)=DPC(2,10)=DPC(1,10)=D(1,NSHK)
      U(2,NSHK)=UPC(2,10)=UPC(1,10)=U(1,NSHK)
      E(2,NSHK)=EPC(2,10)=EPC(1,10)=E(1,NSHK)
      P(2,NSHK)=PPC(2,10)=PPC(1,10)=P(1,NSHK)
 8    IF((NSC.NE.189).AND.(NSC.NE.91).AND.(NSC.NE.90)) GO TO 11
C-----PREDICTOR(9) = PREDICTOR(7)
      DPC(1,9)=DPC(1,7) $ UPC(1,9)=UPC(1,7)
```

```
      EPC(1,9)=EPC(1,7) $ PPC(1,9)=PPC(1,7)
C-----CORRECTOR FOR NSHK = 10
      D(2,NSHK)=DPC(2,10)=DC(10,NSHK,0)
      U(2,NSHK)=UPC(2,10)=UC(10,NSHK,0)
      E(2,NSHK)=EPC(2,10)=EC(10,NSHK,0)
      P(2,NSHK)=PPC(2,10)=PC(10,GE)
      GO TO 15
 11   IF((NSC.NE.311).AND.(NSC.NE.310).AND.(NSC.NE.211).AND.(NSC.NE.200)
     *.AND.(NSC.NE.101).AND.(NSC.NE.201))GO TO 15
C-----CORRECTOR FOR NSHK-1+MD = 9+MD
      MD=0 $ IF((NSC.EQ.211).OR.(NSC.EQ.101)) MD=1
      M=NSHK-1+MD $ L=9+MD
      EPS=(R2(M)-RCD)/DELR
      C1=2.*(2.-EPS)/(1.+EPS)
      C2=2.*EPS-3. $ C3=(1.-EPS)*(2.*EPS-1.)/(1.+EPS)
      D(2,M)=DPC(2,L)=(D(1,M)+DPC(1,L)+DTDX*(C1*UPC(1,7)+C2*UPC(1,L)
     *+C3*UPC(1,L+1))+AT*DTDX*(C1*DPC(1,7)+C2*DPC(1,L)+C3*DPC(1,L+1)))/2
     *.
      U(2,M)=UPC(2,L)=(U(1,M)+UPC(1,L)+DTDX*(C1*CM(7)+C2*CM(L)+C3*CM(L+1
     *))+AT*DTDX*(C1*UPC(1,7)+C2*UPC(1,L)+C3*UPC(1,L+1)))/2.
      E(2,M)=EPC(2,L)=(E(1,M)+EPC(1,L)+DTDX*(C1*CE(7)+C2*CE(L)+C3*CE(L+1
     *))+AT*DTDX*(C1*EPC(1,7)+C2*EPC(1,L)+C3*EPC(1,L+1)))/2.
      P(2,M)=PPC(2,L)=PC(L,GE)
C-----PREDICTOR FOR SHK VEL AND POS
C
C
 15   MH=10+NXSS**2+(NXSS**2-1)*(NSHKSGN-1)*2
      IF(((NSC.EQ.110).OR.(NSC.EQ.-10).OR.(NSC.EQ.0)).AND.(NSHKSGN.EQ.1)
     *) MH=7
      IF(((NSC.EQ.189).OR.(NSC.EQ.199)).AND.(NXSS*NSHKSGN.EQ.-1)) MH=10
      IF(((NSC.EQ.89).OR.(NSC.EQ.99).OR.(NSC.EQ.-11)).AND.(NXSS*NSHKSGN
     *.EQ.-1)) MH=7
      ML=11+NXSS-(NXSS**2-1)*(-1+2*NSHKSGN)
      IF(((NSC.EQ.209).OR.(NSC.EQ.109).OR.(NSC.EQ.89).OR.(NSC.EQ.-11).OR
     *.(NSC.EQ.99)).OR.(((NSC.EQ.-10).OR.(NSC.EQ.110).OR.(NSC.EQ.0)).AND
     *.(NSHKSGN.EQ.0))) ML=7
      IF(((NSC.EQ.189).OR.(NSC.EQ.89).OR.(NSC.EQ.199).OR.(NSC.EQ.99).OR.
     *(NSC.EQ.-11)).AND.(NXSS*NSHKSGN.EQ.-1)) ML=11
      SSLSAV=SQRT((PPC(1,MH)/PPC(1,ML)+(GE-1.)/(GE+1.))*(GE+1.)/2./GE)*
     *SSL/ABS(SSL)
      RS=RSSAV+(SSLSAV*SQRT(GE*PPC(1,ML)/DPC(1,ML))+UPC(1,ML)/DPC(1,ML)+
     *SSE)/2.*DT
      IF(((RS.GT.R2(NSHK+1)).AND.(NXSS.EQ.0)).OR.((RS.LT.R2(NSHK+1)).
     *AND.(NXSS.EQ.1))) RS=R2(NSHK+1)
      IF(((RS.LT.R2(NSHK+MCS)).AND.(NXSS.EQ.0)).OR.((RS.GT.R2(NSHK+MCS))
     *.AND.(NXSS.EQ.-1))) RS=R2(NSHK+MCS)
C
      IF(((RS.LT.RCD).AND.(NCDSHKX.EQ.0)).OR.((RS.GT.RCD).AND.(NCDSHKX
     *.EQ.1))) RS=RCD
C
C
      IF((NXSS*NXSS.EQ.1).OR.((NXSS.EQ.0).AND.(NSHKSGN.EQ.1))) GO TO 29
C-----CORRECTOR FOR NSHK+1 = 12
      M=NSHK+1 $ SL1=(SSL+SSLSAV)/2.
      PS=P(1,NSHK)*(2.*GE/(GE+1.)*SL1**2-(GE-1.)/(GE+1.))
      DS=D(1,NSHK)/((GE-1.)/(GE+1.)+2./(GE+1.)/SL1**2)
```

```
      US=DS*(U(1,NSHK)/D(1,NSHK)+SQRT(GE*P(1,NSHK)/D(1,NSHK))*2./(GE+1.)
     **SL1*(1.-1./SL1**2))
      ES=DS*(PS/DS/(GE-1.)+US**2/DS**2/2.)
      EPS=(R2(NSHK+1)-RS)/DELR
      C1=2.*(2.-EPS)/(1.+EPS)
      C2=2.*EPS-3. $ C3=(1.-EPS)*(2.*EPS-1.)/(1.+EPS)
      D(2,M)=DPC(2,12)=(D(1,M)+DPC(1,12)+DTDX*(C1*US+C2*UPC(1,12)+
     *C3*UPC(1,13))+AT*DTDX*(C1*DS+C2*DPC(1,12)+C3*DPC(1,13)))/2.
      U(2,M)=UPC(2,12)=(U(1,M)+UPC(1,12)+DTDX*(C1*(US**2/DS+PS)+C2*
     *CM(12)+C3*CM(13))+AT*DTDX*(C1*US+C2*UPC(1,12)+C3*UPC(1,13)))/2.
      E(2,M)=EPC(2,12)=(E(1,M)+EPC(1,12)+DTDX*(C1*US/DS*(ES+PS)+C2
     **CE(12)+C3*CE(13))+AT*DTDX*(C1*ES+C2*EPC(1,12)+C3*EPC(1,13)))/2.
      P(2,M)=PPC(2,12)=PC(12,GE)
 29   IF(NXSS.EQ.0) GO TO 16
C-----CORRECTOR FOR NSHK+(NXSS+1)/2 = 11
      M=NSHK+(NXSS+1)/2
      SSLSAV=(SSL+SSLSAV)/2.
      D(2,M)=DPC(2,11)=D(1,M)/((GE-1.)/(GE+1.)+2./(GE+1.)/SSLSAV**2)
      P(2,M)=PPC(2,11)=P(1,M)*(2.*GE/(GE+1.)*SSLSAV**2-(GE-1.)/(GE+1.))
      IF(NXSS*NSHKSGN.EQ.-1) GO TO 42
      U(2,M)=UPC(2,11)=D(2,M)*(U(1,M)/D(1,M)+SQRT(GE*P(1,M)/D(1,M))*2./(
     *GE+1.)*SSLSAV*(1.-1./SSLSAV**2))
      GO TO 43
 42   P(2,M)=PPC(2,11)=P(1,M)**2/PPC(2,11)
      D(2,M)=DPC(2,11)=D(1,M)**2/DPC(2,11)
      U(2,M)=UPC(2,11)=D(2,M)*(U(1,M)/D(1,M)-SQRT(GE*P(1,M+1)/D(1,M+1))
     **2./(GE+1)*SSLSAV*(1.-1./SSLSAV**2))
 43   E(2,M)=EPC(2,11)=DPC(2,11)*(PPC(2,11)/DPC(2,11)/(GE-1.)+UPC(2,11)
     ***2/DPC(2,11)**2/2.)
C-----CORRECTOR FOR SHK VEL
 16   SSL=SQRT((PPC(2,MH)/PPC(2,ML)+(GE-1.)/(GE+1.))*(GE+1.)/2./GE)*
     *SSL/ABS(SSL)
      SSE=SSL*SQRT(GE*PPC(2,ML)/DPC(2,ML))+UPC(2,ML)/DPC(2,ML)
      IF(NSC.NE.-11) GO TO 25
C-----CORRECTOR FOR NCD+1(=7) = CORRECTOR 7
      D(2,NCD+1)=DPC(2,7) $ U(2,NCD+1)=UPC(2,7)
      E(2,NCD+1)=EPC(2,7) $ P(2,NCD+1)=PPC(2,7)
C-----CORRECTOR FOR NCD-1(=5) = 11
      D(2,NCD-1)=DPC(2,11) $ U(2,NCD-1)=UPC(2,11)
      E(2,NCD-1)=EPC(2,11) $ P(2,NCD-1)=PPC(2,11)
      GO TO 17
 25   IF(NSC.NE.11) GO TO 17
C-----CORRECTOR NSHK+1(=6) = CORRECTOR 6
      D(2,NSHK+1)=DPC(2,6) $ U(2,NSHK+1)=UPC(2,6)
      E(2,NSHK+1)=EPC(2,6) $ P(2,NSHK+1)=PPC(2,6)
C
C-----ADVANCE SHK AND CD CELL POSITION IF NECCESSARY
 17   NSHK=NSHK+NXSS
      IF(NSC.EQ.109) GO TO 20
      IF(NXSCD.EQ.0) GO TO 19
      M=NCD+2+3*(NXSCD-1)/2
      PS=P(2,M) $ DS=D(2,M) $ US=U(2,M) $ ES=E(2,M) $ R2S=R2(M)
      IF(NXSCD.EQ.-1) GO TO 18
      P(2,NCD+2)=P(2,NCD+1) $ P(2,NCD+1)=P(2,NCD) $ P(2,NCD)=PS
      U(2,NCD+2)=U(2,NCD+1) $ U(2,NCD+1)=U(2,NCD) $ U(2,NCD)=US
      E(2,NCD+2)=E(2,NCD+1) $ E(2,NCD+1)=E(2,NCD) $ E(2,NCD)=ES
```

```
      D(2,NCD+2)=D(2,NCD+1) $ D(2,NCD+1)=D(2,NCD) $ D(2,NCD)=DS
      R2(NCD)=R2S
      GO TO 19
 18   P(2,NCD-1)=P(2,NCD) $ P(2,NCD)=P(2,NCD+1) $ P(2,NCD+1)=PS
      U(2,NCD-1)=U(2,NCD) $ U(2,NCD)=U(2,NCD+1) $ U(2,NCD+1)=US
      E(2,NCD-1)=E(2,NCD) $ E(2,NCD)=E(2,NCD+1) $ E(2,NCD+1)=ES
      D(2,NCD-1)=D(2,NCD) $ D(2,NCD)=D(2,NCD+1) $ D(2,NCD+1)=DS
      R2(NCD+1)=R2S
C
 19   NDISCL(NFIRST)=NCD=NCD+NXSCD $ RD(NFIRST)=R2(NCD)=R2(NCD+1)=RCD
      SE(NFIRST)=SECD $ NDISCL(N1)=NSHK $ SL(N1)=SSL $ SE(N1)=SSE
      RD(N1)=RS
C
 26   IF(NCDSHKX.EQ.1) GO TO 20
      RETURN
C
C
 20   TRFLCT=(RCDSAV-RSSAV)/(SSESAV-SECDSAV)
      RD(N1)=RD(NFIRST)=RS=RCD=RCDSAV+TRFLCT*SECDSAV
      IF((NSC.EQ.110).OR.(NSC.EQ.99).OR.(NSC.EQ.0).OR.((NSC.EQ.-11).AND
     *.(RCD.LT.R2(NSHK+1)))) GO TO 30
      IF((NSC.NE.109).OR.(RCD.GT.R2(NSHK+1))) GO TO 21
      NDISCL(NFIRST)=NCD $ R2(NCD)=R2(NCD+1)=RCD
 23   NDISCL(N1)=NSHK $ SE(N1)=SSE $ SL(N1)=SSL
      GO TO 30
 21   IF(NSC.NE.109) GO TO 22
      D(2,NCD+2)=D(2,NCD+1) $ U(2,NCD+2)=U(2,NCD+1)
      E(2,NCD+2)=E(2,NCD+1) $ P(2,NCD+2)=P(2,NCD+1)
      R2(NCD)=R2(NCD+2) $ RD(NFIRST)=R2(NCD+1)=R2(NCD+2)=RCD
      NDISCL(NFIRST)=NCD=NCD+1 $ NSHK=NSHK-1
      GO TO 23
 22   IF((NSC.EQ.-11).AND.(RCD.GT.R2(NSHK+1))) GO TO 24
      PRINT 1000,NSC,NSHK,NCD,NXSS,NXSCD,NCDSHKX
 1000 FORMAT(1H ,1X,*TROUBLE IN CD-SHK*,6I5)
      DELR=-ABS(DELR) $ RETURN
 24   D(2,NCD+2)=D(2,NCD+1) $ U(2,NCD+2)=U(2,NCD+1)
      E(2,NCD+2)=E(2,NCD+1) $ P(2,NCD+2)=P(2,NCD+1)
      D(2,NCD+1)=D(2,NCD) $ U(2,NCD+1)=U(2,NCD)
      E(2,NCD+1)=E(2,NCD) $ P(2,NCD+1)=P(2,NCD)
      R2(NCD+2)=R2(NCD) $ R2(NCD+1)=R2(NCD+2)=RCD
      NDISCL(NFIRST)=NDISCL(NFIRST)+1 $ NDISCL(N1)=NDISCL(N1)+1
C
 30   CALL SHKCDA(NFIRST,N1,TRFLCT)
      RETURN
C
C-----SPCL FLW FLD ADJSTMNT FR NCDSHKX=1
 100  DO 110 M=1,N
      IF(M.GT.NCD+1) GO TO 101
      IF(M.LT.NCD) GO TO 102
      GO TO 110
 102  D(1,M)=D(1,NSHK) $ U(1,M)=U(1,NSHK) $ E(1,M)=E(1,NSHK)
      D(2,M)=D(1,NSHK) $ U(2,M)=U(1,NSHK) $ E(2,M)=E(1,NSHK)
      P(1,M)=P(1,NSHK) $ P(2,M)=P(1,NSHK)
      IF((N1.EQ.4).AND.(M.GE.NDISCL(2))) R2(M)=R2(M+2)
      GO TO 110
 101  MM=M-2 $ IF(N1.EQ.4) MM=M-4
```

```
      D(1,MM)=D(1,M) $ U(1,MM)=U(1,M) $ F(1,MM)=F(1,M) $ E(1,MM)=E(1,M)
      D(2,MM)=D(2,M) $ U(2,MM)=U(2,M) $ F(2,MM)=F(2,M) $ F(2,MM)=E(2,M)
      R(MM)=R(M) $ R2(MM)=R2(M)
 110  CONTINUE
      MM=2 $ IF(N1.EQ.4) MM=4
      DO 120 M=1,NTDISCT
      IF(NDISCL(M).LT.NSHK) NDISCL(M)=-1
      IF(NDISCL(M).GE.NSHK) NDISCL(M)=NDISCL(M)-MM
 120  CONTINUE
      NFLM=NFLM-MM $ N=N-MM
      IF(N1+1.LE.NTDISCS) NRHS(N1+1)=NRHS(N1+1)-MM
      IF(N1+1.LE.NTDISCS) NLHS(N1+1)=NLHS(N1+1)-MM
C
C-----ACCT FOR PRTCLE PTH AND NEG AND POS CHRCT TRAJ CELL LOCTNS CHANGES
      IF(NPPTH.LE.0) GO TO 126
      DO 111 I=1,NPPTH
      IF(NCLPPTH(I).GE.NSHK) NCLPPTH(I)=NCLPPTH(I)-MM
 111  CONTINUE
 126  IF(NUMA.LE.0) GO TO 112
      DO 113 I=NUMAFST,NUMA
 113  CONTINUE
 112  IF(NUPA.LE.0) GO TO 114
      DO 115 I=1,NUPA
      IF(NCLUPA(I).GE.NSHK) NCLUPA(I)=NCLUPA(I)-MM
 115  CONTINUE
C
 114  NUPC=3HYES
      CALL SHK(RD(N1),SE(N1),SL(N1),NDISCL(N1))
C
      NCODE=10HCDSHK       $ CALL PRNTFF(NCODE)
C
      RETURN $ END
```

```
      SUBROUTINE CDTRDTR(N1,N2,NTR)
      COMMON/DISCS/NTDISC,NDISCNO(51),NTYPE(51),NDISCL(51),SE(51),SL(51)
     *,RD(51),NTDISCT,NTDISCS
C
      NTR=-1
      PRINT 1000
 1000 FORMAT(1H ,5/,5X,*TERMINATING TAIL OF RARE (ASSOC WITH CJ-DET) IN
     *SUBROUTINE CDTRDTR*)
C
      CALL CD(RD(N1),SE(N1),NDISCL(N1),SHOCTRC)
      NTYPE(N2)=3
      PRINT 1001
 1001 FORMAT(1H ,/,5X,*CD TYPE SET TO 3 IN CDTRDTR*)
C
      RETURN $ END
```

```
      SUBROUTINE TRARE(NTR)
C
      NTR=-1
      PRINT 1000
 1000 FORMAT(1H ,5/,5X,*TERMINATING TAIL OF RARE (ASSOC WITH CJ-DET) IN
     *SUBROUTINE TRARE*)
C
      RETURN $ END
```

```
      SUBROUTINE TRDET(N1,NXTR,N2,NXD,NAME)
      COMMON/PARAM/N,J,AJ,G,GF,DELR,NFLM
      COMMON/TIME/T,DT,DTL,TWRITE,DELT,DTDX,AT
      COMMON/TRDT/PLTR,DLTR,ULTR,ELTR,OVERDAM,NOVERDN
      COMMON/PREDCOR/DPC(2,13),UPC(2,13),EPC(2,13),PPC(2,13)
      COMMON/ARRAYS/R(501),U(2,501),P(2,501),D(2,501),E(2,501),R2(501)
      COMMON/POWER/VCAPF,R4R1,POPOWER,PPOWER,CND,CCAPF,SFEOLD,SFE,NCJ
      COMMON/DISCS/NTDISC,NDISCNO(51),NTYPE(51),NDISCL(51),SE(51),SL(51)
     *,RD(51),NTDISCT,NTDISCS
C
C
      NSC=100*(NDISCL(N2)-NDISCL(N1))+10*NXTR+NXD
      PRINT 1000,NAME,(I,NDISCL(I),NTYPE(I),RD(I),I=N1,N2),NXTR,NXD,NSC
 1000 FORMAT(1H ,1X,A8,2(3I5,F14.7),3I5)
      NTR=NDISCL(N1)
      IF(NAME.EQ.8HSCTD  NC) GO TO 3
C-----PROCEDURE FOR INCORPORATING THE RARE ASSOC WITH CJ DET
      IF(NXTR.EQ.-1) GO TO 1
C-----PREDICTOR FOR NTR-1 = 12
      M=NTR-1
      DPC(1,12)=DP(M,0) $ UPC(1,12)=UP(M,0)
      EPC(1,12)=EP(M,0) $ PPC(1,12)=PP(12,GF)
C-----PREDICTOR FOR NTR = 13
      NSHK=NTR $ PS=PLTR $ DS=DLTR $ US=ULTR $ ES=ELTR
      EPS=(RD(N1)-R(NTR))/DELR
      C1=2.*(2.-EPS)/(1.+EPS)
      C2=2.*EPS-3. $ C3=(1.-EPS)*(2.*EPS-1.)/(1.+EPS)
      DPC(1,13)=D(1,NSHK)-DTDX*(C1*US+C2*U(1,NSHK)+C3*U(1,NSHK-1))
      UPC(1,13)=U(1,NSHK)-DTDX*(C1*(US*US/DS+PS*RD(N1)**J)+C2*PM(NSHK)+
     *C3*PM(NSHK-1))
      EPC(1,13)=E(1,NSHK)-DTDX*(C1*US/DS*(ES+PS*RD(N1)**J)+C2*PE(NSHK)+
     *C3*PE(NSHK-1))
      PPC(1,13)=PP(13,GF)
C-----CORRECTOR FOR NTR = 13
      D(2,NSHK)=DPC(2,13)=DC(13,NSHK,0)
      U(2,NSHK)=UPC(2,13)=UC(13,NSHK,0)
      E(2,NSHK)=EPC(2,13)=EC(13,NSHK,0)
      P(2,NSHK)=PPC(2,13)=PC(13,GF)
 3    IF(NXTR.NE.1) GO TO 1
      D(2,NTR+1)=DLTR $ U(2,NTR+1)=ULTR
      E(2,NTR+1)=ELTR $ P(2,NTR+1)=PLTR
 1    RD(N1)=RD(N1)+SE(N1)*DT $ NDISCL(N1)=NTR=NDISCL(N1)+NXTR
C
      CALL DET(RD(N2),SE(N2),SL(N2),NDISCL(N2),NXD,7HTRDET   )
C
C-----PREDICT S CONDITIONS
      NSHK=NDISCL(N2)
      BETA=(GF-1.)/(GF+1.) $ B=(GF-1.)/(G-1.)
      PG0=(1.+BETA)*(VCAPF-BETA)/(1.-BETA)-BETA
      A=SQRT(P(1,NSHK+1)/D(1,NSHK+1))
      PG=1.+(PG0-1.)/A**2+GF*(1.-1./A**2)*(G/GF*B-R4R1)
      PS1=(1.-BETA)/2.*(1.+G*SL(N2)**2) $ PS=PS1*P(1,NSHK+1)
      DS=D(1,NSHK+1)/(1.-(1.-BETA)*(PS1-PG)/(PS1+BETA))
      US=SQRT(G)*A*SQRT((1.-BETA)*(PS1-PG)*(PS1-1.)/G/
     *(PS1+BETA))+U(1,NSHK+1)/D(1,NSHK+1)
C
```

```
C-----INPUT RAREFACTION
      ALPHA=2./(GF-1.) $ NTRP1=NTR+1
      DO 2 I=NTRP1,NSHK
      U(2,I)=US-(US-ULTR/DLTR)*(RD(N2)-R2(I))/(RD(N2)-RD(N1))
      UX=(U(2,I)-US)/SQRT(GF*PS/DS)
      AX=UX/ALPHA+1. $ DX=AX**ALPHA $ D(2,I)=DX*DS
      PX=ALPHA*GF/(2.+ALPHA)*(DX**((2.+ALPHA)/ALPHA)-1.)+1.
      P(2,I)=PX*PS
      E(2,I)=D(2,I)*(P(2,I)/D(2,I)/(GF-1.)+U(2,I)**2/2.)
 2    U(2,I)=U(2,I)*D(2,I)
C
      RETURN $ END
```

```
      SUBROUTINE TSTEP
      COMMON/PARAM/N,J,AJ,G,GF,DELR,NFLM
      COMMON/TIME/T,DT,DTL,TWRITE,DELT,DTDX,AT
      COMMON/ARRAYS/R(501),U(2,501),P(2,501),D(2,501),E(2,501),R2(501)
C
C
      DTL=DT
C-----STABILITY CRITERIA
      GE=GF $ DTMIN=1.E+300
      DO 5 M=1,N
      IF(M.GT.NFLM) GE=G
      C=SQRT(GE*P(2,M)/D(2,M))
      IF(U(2,M).LE.0.0) GO TO 1
      US=U(2,M)/D(2,M)+AT $ USPC=US+C $ USMC=US-C
      IF(ABS(US).LT.ABS(USPC))US=USPC $ IF(ABS(US).LT.ABS(USMC))US=USMC
      GO TO 2
 1    US=U(2,M)/D(2,M)+AT-C
 2    DT=DELR/ABS(US)*0.7
      IF(DT.LT.DTMIN) DTMIN=DT
 5    CONTINUE
      DT=DTMIN
C
      T = T+DT
C
C-----REINITIAL PROPERTIES
      DO 10 M=1,N
      U(1,M)=U(2,M) $ D(1,M)=D(2,M) $ P(1,M)=P(2,M) $ E(1,M)=E(2,M)
 10   R(M)=R2(M)
C
      RETURN $ END
```

```
      SUBROUTINE CHARCIR(NCYCLE)
      COMMON/PARAM/N,J,AJ,G,GF,DELR,NFLM
      COMMON/TIME/T,DT,DTL,TWRITE,DELT,DTDX,AT
      COMMON/ARRAYS/R(501),U(2,501),P(2,501),D(2,501),E(2,501),R2(501)
      COMMON/PPCHR/NPPTH,NUMA,NUMAFST,NUPA,RPPTH(24),RUMA(150),RUPA(150)
     *,NCLPPTH(24),NCLUMA(150),NCLUPA(150),PTHNEXT,RUMANXT,TUPANXT,
     *DELPPTH,DELUMA,DELUPA
C
      UAVG(I1,R1)=U(1,I1-1)/D(1,I1-1)+(U(1,I1)/D(1,I1)-U(1,I1-1)/
     *D(1,I1-1))*(R1-R(I1-1))/(R(I1)-R(I1-1))
      PAVG(I1,R1)=P(1,I1-1)+(P(1,I1)-P(1,I1-1))*(R1-R(I1-1))/(R(I1)-
     *R(I1-1))
      DAVG(I1,R1)=D(1,I1-1)+(D(1,I1)-D(1,I1-1))*(R1-R(I1-1))/(R(I1)-
     *R(I1-1))
C
C     NPPTH=NO. OF INITIAL PARTICLE PATHS
C     PTHNEXT=POSITION OF THE NEXT PARTICLE PATH
C     DELPPTH=DST BET SCCSSVE PRTCL PTHS(MST BE > 0. TO HVE ADDTLN PTHS)
C     NUMAFST=NO. OF THE FRST NEG CHARACT TRJ WTH PSTN > 0.0
C     NUMA=NO. OF INITIAL NEGATIVE CHARACTERISTIC TRAJECTORIES
C     RUMANXT=POSITION OF THE NEXT NEGATIVE CHARACTERISTIC TRAJECTORY
C     DELUMA=DST BET SCCSSVE NG CHR TRJS(MST BE > 0. TO HVE ADDTLN TRJS)
C     NUPA=NO. OF INITIAL POSITIVE CHARACTERISTIC TRAJECTORIES
C     TUPANXT=POSITION OF THE NEXT POSITIVE CHARACTERISTIC TRAJECTORY
C     DELUPA=DST BET SCCSSVE PS CHR TRJS(MST BE > 0. TO HVE ADDTLN TRJS)
C     RPPTH,RUMA,RUPA=PRTCLE, NEGTVE CHRACT, POSTVE CHARCT PSTN
C     NCLPPTH,NCLUMA,NCLUPA=CL NO OF PRTCL PTH,NG CHRCT TRJ,PS CHRCT TRJ
C
C
      IF(NCYCLE.GT.0) GO TO 14
C
C-----DET INITIAL PARTICLE PATH CELL POSITIONS
      IF(NPPTH.EQ.0) GO TO 5
      N1=1
      DO 1 I1=1,NPPTH
      DO 2 I2=N1,N
      IF(R2(I2).GT.RPPTH(I1)) GO TO 1
 2    CONTINUE
 1    NCLPPTH(I1)=N1=I2
C
C
C-----DET NEG INITIAL NEG AND POS CHARACT TRAJ CELL POSITIONS
 5    IF(NUMA.EQ.0) GO TO 8
      N1=1
      DO 6 I1=1,NUMA
      DO 7 I2=N1,N
      IF(R2(I2).GT.RUMA(I1)) GO TO 6
 7    CONTINUE
 6    NCLUMA(I1)=N1=I2
C
 8    IF(NUPA.EQ.0) RETURN
      N1=1
      DO 9 I1=1,NUPA
      DO 10 I2=N1,N
      IF(R2(I2).GT.RUPA(I1)) GO TO 9
 10   CONTINUE
```

```
 9    NCLUPA(I1)=N1=I2
      RETURN
C
C
C
C-----CALC PARTCLE PATH AND FINAL CELL LOCATION
 14   IF(NPPTH.EQ.0) GO TO 15
      DO 11 I=1,NPPTH
      II=NCLPPTH(I) $ R1=RPPTH(I)
      RPPTH(I)=UAVG(II,R1)*DT+RPPTH(I)
      IF(RPPTH(I).LT.0.0) RPPTH(I)=0.0
 11   CONTINUE
      DO 12 I1=1,NPPTH
      N1=NCLPPTH(I1)-3 $ IF(N1.LT.2) N1=2
      DO 13 I2=N1,N
      IF(R2(I2).GT.RPPTH(I1)) GO TO 12
 13   CONTINUE
 12   NCLPPTH(I1)=I2
 15   IF(DELPPTH.LE.0.0) GO TO 20
      IF(R2(N).LT.PTHNEXT) GO TO 20
      NPPTH=NPPTH+1 $ RPPTH(NPPTH)=PTHNEXT
      I=N-5
      DO 16 K=I,N
      IF(R2(I).GT.PTHNEXT) GO TO 17
 16   CONTINUE
 17   NCLPPTH(NPPTH)=K
      PTHNEXT=PTHNEXT+DELPPTH
C
C-----CALC NEG AND POS CHARACT TRAJ AND FINAL CELL LOCATIONS
 20   IF(NUMA.EQ.0) GO TO 25
      DO 21 I=NUMAFST,NUMA
      II=NCLUMA(I) $ R1=RUMA(I)
      GE=G $ IF(II.LE.NFLM) GE=GF
 21   RUMA(I)=(UAVG(II,R1)-SQRT(GE*PAVG(II,R1)/DAVG(II,R1)))*DT+RUMA(I)
      DO 22 I1=NUMAFST,NUMA
      IF(RUMA(I1).LT.-DELR) NUMAFST=NUMAFST+1
      N1=NCLUPA(I1)-3 $ IF(N1.LT.2) N1=2
      DO 23 I2=N1,N
      IF(R2(I2).GT.RUMA(I1)) GO TO 22
 23   CONTINUE
 22   NCLUMA(I1)=I2
 25   IF(DELUMA.LE.0.0) GO TO 30
      IF(R2(N).LT.RUMANXT) GO TO 30
      NUMA=NUMA+1 $ RUMA(NUMA)=RUMANXT
      I1=N-5
      DO 26 I=I1,N
      IF(R2(I).GT.RUMANXT) GO TO 27
 26   CONTINUE
 27   NCLUMA(NUMA)=I
      RUMANXT=RUMANXT+DELUMA
C
 30   IF(NUPA.EQ.0) GO TO 35
      DO 31 I=1,NUPA
      II=NCLUPA(I) $ R1=RUPA(I)
      GE=G $ IF(II.LE.NFLM) GE=GF
 31   RUPA(I)=(UAVG(II,R1)+SQRT(GE*PAVG(II,R1)/DAVG(II,R1)))*DT+RUPA(I)
```

```
      DO 32 I1=1,NUPA
      N1=NCLUMA(I1)-3 $ IF(N1.LT.2) N1=2
      DO 33 I2=N1,N
      IF(R2(I2).GT.RUPA(I1)) GO TO 32
   33 CONTINUE
   32 NCLUPA(I1)=I2
   35 IF(DELUPA.LE.0.0) RETURN
      IF(T.LT.TUPANXT) RETURN
      NUPA=NUPA+1 $ RUPA(NUPA)=0.0 $ NCLUPA(NUPA)=2
      TUPANXT=TUPANXT+DELUPA
C
      RETURN $ END
```

```
      SUBROUTINE PRNTFF(NCODE)
      COMMON/PARAM/N,J,AJ,G,GF,DELR,NFLM
      COMMON/TIME/T,DT,DTL,TWRITE,DELT,DTDX,AT
      COMMON/FIRFETC/INDEX,NCYCLE,NN,NNN,NSTORE,NS,NITRCTN
      COMMON/ARRAYS/R(501),U(2,501),P(2,501),D(2,501),E(2,501),R2(501)
      COMMON/POWER/VCAPF,R4R1,POPOWER,PPOWER,CNC,CCAPF,SFEOLD,SFE,NCJ
      COMMON/DISCS/NTDISC,NDISCNO(51),NTYPE(51),NDISCL(51),SE(51),SL(51)
     *,RD(51),NTDISCT,NTDISCS
      COMMON/PPCHR/NPPTH,NUMA,NUMAFST,NUPA,RPPTH(24),RUMA(150),RUPA(150)
     *,NCLPPTH(24),NCLUMA(150),NCLUPA(150),PTHNEXT,RUMANXT,TUPANXT,
     *DELPPTH,DELUMA,DELUPA
C
C
      IARF=3H NO
      IF((NCODE.EQ.10HINITIAL   ).OR.(NCODE.EQ.10HINITIALACC).OR.
     *(NCODE.EQ.10HRESTART   ).OR.(NCODE.EQ.10HRIDIF     )) IARF=3HYES
      IF(IARF.EQ.3HYES) GO TO 1
C
      DO 2 M=1,N
      U(2,M)=U(2,M)/D(2,M)
 2    E(2,M)=E(2,M)/D(2,M)
C
 1    PRINT 2000,T,DT,INDEX,NCYCLE,NTDISCT,NCODE,(I,NDISCNO(I),NTYPE(I),
     *NDISCL(I),RD(I),SE(I),SL(I),I=1,NTDISCT)
 2000 FORMAT(1H1,* T*,E12.5,3X,*DT*,E12.5,3X,*INDEX*,I6,3X,*NCYCLE*,I6,
     *7X,*NTDISCT*,I3,5X,*NCODE *,A10,
     *2/,* I=*,I2,3X,*DISC NO.=*,I3,3X,*NTYPE=*,I1,5X,
     **CELL=*,I4,3X,*POS=*,E12.5,3X,*EULERIAN VEL=*,E12.5,3X,*LAGRANGIAN
     * VEL=*,E12.5,50(/,I5,12X,I3,9X,I1,10X,I4,7X,E12.5,16X,E12.5,
     *18X,E12.5))
      PRINT 2001,(M,R2(M),P(2,M),D(2,M),U(2,M),E(2,M),M=1,N,NNN)
 2001 FORMAT(1H ,/,6X,*CELL*,3X,*POSITION*,5X,*PRESSURE*,6X,*DENSITY*,
     *5X,*VELOCITY*,6X,*ENERGY*,/,1X,100(I8,5(1X,E12.5),/,1X))
      IF(IARF.EQ.3H NO) GO TO 4
      NC1=9HNEGATIVE  $ NC2=7HCHARACT $ NC3=9H TRAJECTS
      NPC1=9HPOSITIVE  $ NC4=8HTRAJ R=  $ NPC4=8HTRAJ T=
      NP1=9HPARTICLE  $ NP2=7HPATHS   $ NP3=9H          $ NP4=8HPATH R=
      PRINT 2003
 2003 FORMAT(1H1)
      IF(NPPTH.NE.0) PRINT 2002,NP1,NP2,NP3,NP4,PTHNEXT,(I,RPPTH(I),
     *NCLPPTH(I),I=1,NPPTH)
 2002 FORMAT(1H ,2/,10X,A9,A7,A9,10X,*NEXT *,A8,F8.4,24(/,5X,6(I3,F8.4,
     *I5,5X))
      IF(NUMA.NE.0) PRINT 2002,NC1,NC2,NC3,NC4,RUMANXT,(I,RUMA(I),
     *NCLUMA(I),I=NUMAFST,NUMA)
      IF(NUPA.NE.0) PRINT 2002,NPC1,NC2,NC3,NPC4,TUPANXT,(I,RUPA(I),
     *NCLUPA(I),I=1,NUPA)
      RETURN
C
 4    DO 3 M=1,N
      U(2,M)=U(2,M)*D(2,M)
 3    E(2,M)=E(2,M)*D(2,M)
C
      RETURN $ END
```

```
      FUNCTION DP(M,MM)
      COMMON/TIME/T,DT,DTL,TWRITE,DELT,DTDX,AT
      COMMON/ARRAYS/R(501),U(2,501),P(2,501),D(2,501),E(2,501),R2(501)
C
      DP=D(1,M)-DTDX*(U(1,M+MM+1)-U(1,M+MM))-AT*DTDX*
     *(D(1,M+MM+1)-D(1,M+MM))
C
      RETURN $ END
```

```
      FUNCTION UP(M,MM)
      COMMON/TIME/T,DT,DTL,TWRITE,DELT,DTDX,AT
      COMMON/ARRAYS/R(501),U(2,501),P(2,501),D(2,501),E(2,501),R2(501)
C
      UP=U(1,M)-DTDX*(U(1,M+1+MM)**2/D(1,M+1+MM)+P(1,M+1+MM)-U(1,M+MM)**
     *2/D(1,M+MM)-P(1,M+MM))-AT*DTDX*(U(1,M+MM+1)-U(1,M+MM))
C
      RETURN $ END
```

```
      FUNCTION EP(M,MM)
      COMMON/TIME/T,DT,DTL,TWRITE,DELT,DTDX,AT
      COMMON/ARRAYS/R(501),U(2,501),P(2,501),D(2,501),E(2,501),R2(501)
C
      EP=E(1,M)-DTDX*(U(1,M+1+MM)/D(1,M+1+MM)*(E(1,M+1+MM)+P(1,M+MM+1))-
     *U(1,M+MM)/D(1,M+MM)*(E(1,M+MM)+P(1,M+MM)))-AT*DTDX*(E(1,M+MM+1)
     *-E(1,M+MM))
C
      RETURN $ END
```

```
      FUNCTION PP(L,GE)
      COMMON/PREDCOR/DPC(2,13),UPC(2,13),EPC(2,13),PPC(2,13)
C
      PP=(EPC(1,L)/DPC(1,L)-UPC(1,L)**2/DPC(1,L)**2/2.0)*DPC(1,L)
     **(GE-1.0)
C
      RETURN $ END
```

```
      FUNCTION DC(L,M,MM)
      COMMON/TIME/T,DT,DTL,TWRITE,DELT,DTDX,AT
      COMMON/PREDCOR/DPC(2,13),UPC(2,13),EPC(2,13),PPC(2,13)
      COMMON/ARRAYS/R(501),U(2,501),P(2,501),D(2,501),E(2,501),R2(501)
C
      DC=(D(1,M)+DPC(1,L)-DTDX*(UPC(1,L+MM)-UPC(1,L-1+MM))
     *-AT*DTDX*(DPC(1,L+MM)-DPC(1,L-1+MM)))/2.
C
      RETURN $ END
```

```
      FUNCTION UC(L,M,MM)
      COMMON/TIME/T,DT,DTL,TWRITE,DELT,DTDX,AT
      COMMON/PREDCOR/DPC(2,13),UPC(2,13),EPC(2,13),PPC(2,13)
      COMMON/ARRAYS/R(501),U(2,501),P(2,501),D(2,501),E(2,501),R2(501)
C
      UC=(U(1,M)+UPC(1,L)-DTDX*(UPC(1,L+MM)**2/DPC(1,L+MM)+
     *PPC(1,L+MM)-UPC(1,L+MM-1)**2/DPC(1,L-1+MM)-PPC(1,L-1+MM))
     *-AT*DTDX*(UPC(1,L+MM)-UPC(1,L-1+MM)))/2.
C
      RETURN $ END
```

```
      FUNCTION EC(L,M,MM)
      COMMON/TIME/T,DT,DTL,TWRITE,DELT,DTDX,AT
      COMMON/PREDCOR/DPC(2,13),UPC(2,13),EPC(2,13),PPC(2,13)
      COMMON/ARRAYS/R(501),U(2,501),P(2,501),D(2,501),E(2,501),R2(501)
C
      EC=(E(1,M)+EPC(1,L)-DTDX*(UPC(1,L+MM)/DPC(1,L+MM)*(EPC(1,L+MM)+
     *PPC(1,L+MM))-UPC(1,L+MM-1)/DPC(1,L+MM-1)*(EPC(1,L-1+MM)+
     *PPC(1,L-1+MM)))-AT*DTDX*(EPC(1,L+MM)-EPC(1,L+MM-1)))/2.
C
      RETURN $ END
```

```
      FUNCTION PC(L,GE)
      COMMON/PREDCOR/DPC(2,13),UPC(2,13),EPC(2,13),PPC(2,13)
C
      PC=(EPC(2,L)/DPC(2,L)-UPC(2,L)**2/DPC(2,L)**2/2.0)*DPC(2,L)
     ***(GE-1.0)
C
      RETURN $ END
```

```
      FUNCTION PM(M)
      COMMON/ARRAYS/R(501),U(2,501),P(2,501),D(2,501),E(2,501),R2(501)
C
      PM=U(1,M)**2/D(1,M)+P(1,M)
C
      RETURN $ END




      FUNCTION PE(M)
      COMMON/ARRAYS/R(501),U(2,501),P(2,501),D(2,501),E(2,501),R2(501)
C
      PE=U(1,M)/D(1,M)*(E(1,M)+P(1,M))
C
      RETURN $ END




      FUNCTION CM(M)
      COMMON/PREDCOR/DPC(2,13),UPC(2,13),EPC(2,13),PPC(2,13)
C
      CM=UPC(1,M)**2/DPC(1,M)+PPC(1,M)
C
      RETURN $ END




      FUNCTION CE(M)
      COMMON/PREDCOR/DPC(2,13),UPC(2,13),EPC(2,13),PPC(2,13)
C
      CE=UPC(1,M)/DPC(1,M)*(EPC(1,M)+PPC(1,M))
C
      RETURN $ END
```

```
    0    0    0    0
  850  500   50   50     1 9999 500.0
  110    0 1.3      1.2       0.01      7.0       1.0       2.3       0.5
1.2300322                0.17490552               0.0
1.2373913043             1.1777401                0.18927817381
1.0                 1.0                 0.0                           0.1772
 99.0               0.086206896
    2
    1    1   19 0.028953623             .1772
    2    2  102 1.1                      1.0
    3 1.5      0.5             6 1.1      0.1          11 0.1      0.1
0.3333    0.6666    0.9999
0.5       0.6       0.7       0.8       0.9       1.0
0.00001   0.1       0.2       0.3       0.4       0.5       0.6       0.7
0.8       0.9       1.0
******************************END OF DATA*******************************
```

REFERENCES

1. Berthelot, M., "Sur la Vitesse de Propagation des Phenomenes Explosifs dans les Gaz", C.R. Acad. Sci., Paris, 93, 18-22, Seance du 4 Juillet, 1881.
2. Laderman, A.J. and Oppenheim, A.K., "Initial Flame Acceleration in an Explosive Gas", Proc. Roy. Soc. (London), A268, 153-180, 1962.
3. Laderman, A.J., Urtiew, P.A. and Oppenheim, A.K., "On the Generation of a Shock Wave by a Flame in an Explosive Gas", Ninth Symposium (International) on Combustion, Academic Press Inc., New York, 265-274, 1963.
4. Urtview, P.A., Laderman, A.J. and Oppenheim, A.K., "Dynamics of the Generation of Pressure Waves by Accelerating Flames", Tenth Symposium (International) on Combustion, The Combustion Institute, Pittsburgh, PA., 797-804, 1965.
5. Oppenheim, A.K., Kamel, M.M. and Varvatsoulis, C., "Experimental Study of the Dynamics of a Spherical Flame", Seventh JANNAF Combustion Meeting, CPIA Publication 204, 1, 1971.
6. Oppenheim, A.K. and Soloukhin, R.I., "Experiments in Gasdynamics of Explosions", Annual Review of Fluid Mechanics, 5, 31-58, 1973.
7. Schmidt, E., Steinicke, H. and Neubert, U., "Flame and Schlieren Photographs of the Combustion of Gas-Air Mixtures in Tubes", VDI-Forsch Ungsheft 431, Ed. B., Deutscher Ingenieur-Verlag, Dusseldorf, 17, 31, 1951.
8. Schmidt, E. Steinicke, H. and Neubert, U., "Flame and Schlieren Photographs of Combustion Waves in Tubes", Fourth Symposium (International) on Combustion, Williams and Wilkins, Baltimore, MD., 658-666, 1953.

9. Urtiew, A.J. and Oppenheim, A.K., "Experimental Observations of the Transition to Detonation in an Explosive Gas", Proc. Roy. Soc. (London), A295, 13-28, 1966.

10. Urtiew, A.J. and Oppenheim, A.K., "Detonation Ignition Induced by Shock Merging", Eleventh Symposium (International) on Combustion, The Combustion Institute, Pittsburgh, PA., 665-670, 1967.

11. Urtiew, A.J. and Oppenheim, A.K., "Transverse Flame-Shock Interactions in an Explosive Gas", Proc. Roy. Soc. (London), A304, 379-385, 1968.

12. Strehlow, R.A., "Unconfined Vapor Cloud Explosions - An Overview", Fourteenth Symposium (International) on Combustion, The Combustion Institute, Pittsburgh, PA., 1189-1200, 1973.

13. Fay, J.A., "Unusual Fire Hazard of LNG Tanker Spills", Combustion Science and Technology, 7, 47-49, 1973.

14. Fay, J.A. and Lewis, D. H., Jr., "The Inflammability and Dispersion of LNG Vapor Clouds", Fourth International Symposium on Transportation of Hazardous Cargoes by Sea and Inland Waterways", 489-498, 1975.

15. Haverdings, W., Pasman, H. J. and Wiekema, B. J., "Blast by Non-Ideal Aerial Explosions", Acta Astranautica, 3, 1-2, 107-116, 1976.

16. Oppenheim, A.K., Kurylo, J., Cohen, L.M. and Kamel, M.M., "Blast Waves Generated by Exploding Clouds", Eleventh International Symposium on Shock Tubes and Waves", University of Washington, July 1977 (in press).

17. Dörge, K.J., Pangritz, D. and Wagner, H.Gg., "Experiments on Velocity Augmentation of Spherical Flames by Grids", Acta Astronautica, 3, 11-12, 1067-1076, 1976.

18. Oppenheim, A.K., Kuhl, A.L. and Kamel, M.M., "On Self-Similar Waves Headed by the Chapman-Jouguet Detonation", J. Fluid Mechanics, 55, 2, 257-270, 1972.

19. Oppenheim, A.K., Kuhl, A.L., Lundstrom, E.A. and Kamel, M.M., "A Parametric Study of Self-Similar Blast Waves", J. Fluid Mechanics, 52, 4, 657-682, 1972.

20. Kuhl, A.L., Kamel, M.M. and Oppenheim, A.K., "Pressure Waves Generated by Steady Flames", Fourteenth Symposium (International) on Combustion, The Combustion Institute, Pittsburg, PA., 1201-1215, 1973.

21. Shapiro, A.H. "The Dynamics and Thermodynamics of Compressible Fluid Flow", Vol. 1, The Ronald Press Co., New York, xiii+647 pp., 1953.

22. Rudinger, G., "Wave Diagrams for Non-Steady Flow in Ducts", D. Van Nostrand Co., Inc., New York, xi+278 pp., 1955.

23. Owczarek, J.A., "Fundamentals of Gas Dynamics", International Textbook Co., Scranton, PA., xxii+648 pp., 1964.

24. Cohen. L.M., Short, J.M. and Oppenheim, A.K., "A Computational Technique for the Evaluation of Dynamic Effects of Exothermic Reactions", Combustion and Flame, 24, 3, 319-334, 1975.

25. Tanaka, K., and Hikita, T., "A numerical Study of the Charge Diameter Effect on Unsteady Detonation", Acta Astronautica, 3, 11-12, 1005-1013, 1976.

26. Fishburn, B.D., "Some Aspects of Blast from Fuel-Air Explosives", Acta Astronautica, 3, 11-12, 1049-1065, 1976.

27. von Neumann, J. and Richtmyer, R.D., "A Method for the Numerical Calculation of Hydrodynamic Shocks", J. Apply. Phys., 21, 3, 232-237, 1950.

28. Wilkins, M.L., "Calculation of Elstic-Plastic Flow", Lawrence Livermore Laboratory Report #UCRL - 7322, Rev. 1, Livermore, CA., iv+101 pp., 1969.

29. Lax, P.D., "Weak Solutions of Nonlinear Hyperbolic Equations and their Numerical Computation", Comm. Pure Appl. Math., 7, 159-193, 1954.

30. Richtmyer, R.D. and Morton, K.W., "Difference Methods for Initial-Value Problems", Second Edition, Interscience Publishers, New York, xiv+405 pp., 1967.

31. Moretti, G., "Experiments in Multi-Dimensional Floating Shock-Fitting", Polytechnic Institute of Brooklyn, PIBAL Rept. 73-18, ii+38 pp., Aug. 1973.

32. Moretti, G., "Thoughts and Afterthoughts About Shock Computations", Polytechnic Institute of Brooklyn, PIBAL Rept. 73-37, iv+81 pp., Dec. 1972.

33. Moretti, G. and Abbett, M., "A Time-Dependent Computational Method for Blunt Body Flows", AIAA Journal, 4, 12, 2136-2141, 1966.

34. Oppenheim, A.K. and Stern, R.A., "On the Development of Gaseous Detonation - Analysis of Wave Phenomena", Seventh Symposium (International) on Combustion, Butterworths Scientific Publications, London, 837-850, 1959.

35. Chu, B.T., "On the Generation of Pressure Waves at a Plane Flame Front", Fourth Symposium (International) on Combustion", Williams and Wilkins, Baltimore, MD., 603-612, 1953.

36. Markstein, G.H. and Schwartz, D., "Interaction Between Pressure Waves and Flame Fronts", Jet Propulsion, 25, 173-174, 1955.

37. Markstein, G.H., "A Shock-Tube Study of Falme Front-Pressure Wave Interaction", Sixth Symposium (International) on Combustion, Reinhold Publishing Corp., New York, 387-398, 1957.

38. Salamandra, G.D. and Sevastyanova, I.K., "Formation of Weak Shock Waves Ahead of a Flame Front and Their Intensification During Passage Through the Flame", Combustion and Flame, 7, 169-174, 1963.

39. Kamel, M.M. and Oppenheim, A.K., "Photographic Laboratory Studies of Explosion", L'Aerotechnica Missili e Spazio, 2, Rivista dell'Associazione Italiana di Aeronautica e Astronautica, Tamburini Editore, Milano, 122-134, 1973.

40. Laderman, A.J., Urtview, P.A. and Oppenheim, A.K., "Gasdynamic Effects of Shock-Flame Interactions in an Explosive Gas", AIAA Journal, 3, 5, 876-883, 1965.

41. Gilbert, M., "The Influence of Pressure on Flame Speed", Sixth Symposium (International) on Combustion, Reinhold Publishing Corp., New York, 74-83, 1957.

42. Goldenberg, S.A. and Pelevin, V.S., "Influence of Pressure on Rate of Flame Propagation in Turbulent Flow", Seventh Symposium (International) on Combustion, Butterworths Scientific Publications, 590-594, 1959.

43. Bradley, D. and Hundy, G.F., "Burning Velocity of Methane-Air Mixtures Using Hot-Wire Anemometers in Closed-Vessel Explosions", Thirteenth Symposium (International) on Combustion, The Combustion Institute, Pittsburgh, PA, 575-583, 1971.

44. Andrews, G.E. and Bradley, D., "The Burning Velocity of Methane-Air Mixtures", Combustion and Flame, 19, 2, 275-288, 1972.

45. Andrews, G.E. and Bradley, D. "Determination of Burning Velocity by Double Ignition in a Closed Vessel", Combustion and Flame, 20, 1, 77-89, 1973.

46. Thompson, P.A., "Compressible-Fluid Dynamics", McGraw-Hill Book Co., New York, xvii+665 pp., 1972.

47. Dwyer, H., Allen, R., Ward, M., Karnopp, D. and Margolis, D., "Shock Capturing Finite Difference Methods for Unsteady Gas Transfer". AIAA Paper #74-521, 1974.

48. Ames, W.F., "Numerical Methods for Partial Differential Equations", Barnes and Noble, Inc., x+291 pp., 1969.

49. MacCormack, R.W., "The Effect of Viscosity in Hypervelocity Impact Cratering", AIAA Paper #69-354, 1969.

50. Tyler, L., "Heuristic Analysis of Convective Finite Difference Techniques", Proc. of the Second International Conference on Numerical Methods in Fluid Dynamics, Lecture Notes in Physics, 8, Springer-Verlag, New York, 314-319, 1970.

51. Warming, R.F., Kulter, P. and Lomax, H., "Second- and Third-Order Non-Centered Difference Schemes for Nonlinear Hyperbolic Equations", AIAA Journal, 11, 2, 189-196, 1973.

52. Salas, M.D., "Shock Fitting Method for Complicated Two-Dimensional Supersonic Flows", AIAA Journal, 14, 5, 583-588, 1976.

53. Quarles, D., PhD. Thesis, New York University, 1964.

54. Salas, M.D., Private Communication, 1976.

55. D'Souza, N., Molder, S. and Moretti, G., "Numerical Method for Hypersonic Internal Flow Over Blunt Leading Edges and Two Blunt Bodies", AIAA Journal, 10, 5, 617-622, 1972.

56. Chorin, A.J., Private Communication, 1976.

57. Courant, R. and Friedrichs, K.O., "Supersonic Flow and Shock Waves", Interscience Publishers, Inc., New York, xvi+464 pp., 1948.

58. Chorin, A.J., "Random Choice Solution of Hyperbolic System", J. Computational Physics, 22, 4, 517-533, 1976.

59. Oppenheim, A.K., "Introduction to Gasdynamics of Explosions", Springer-Verlag, Wien, New York, v+220 pp., 1972.

60. Dahlquist, G., Björck, A. and Anderson, N., "Numerical Methods", Prentice-Hall, Inc., Englewood, New Jersey, xviii+573 pp., 1974.

FIGURE CAPTIONS

Fig. 1. Streak schlieren photograph and pressure transducer records of flame-shock interactions. Oscilloscope leads the streak record by 3.38 msec (sweep rate = 0.5 msec/cm; vertical deflection = 0.5 psi/cm).

Fig. 2. Finite wave analysis of flame-shock interactions depicted in Fig. 1.

Fig. 3. Comparison of the experimental and analytical pressure profiles at positions PG1 and PG2, indicated in Figs. 1 and 2.

Fig. 4. Pressure-space profiles of the flow fields corresponding to a flame burning speed of 25 m/sec. $j = 0, 1,$ and 2 for plane-, line-, and point-symmetrical flow fields respectively; $\gamma_u = 1.3$; $\gamma_b = 1.2$; $\nu_F = 7$; $M = 1$.

Fig. 5. Density-space profiles of the flow fields associated with Fig. 4.

Fig. 6. Particle velocity-space profiles of the flow fields associated with Fig. 4.

Fig. 7. Shock-shock collision in the hodograph and time-space planes.

Fig. 8. Shock-shock merging in the hodograph and time-space planes.

Fig. 9. Shock-contact discontinuity interaction corresponding to a speed of sound ratio (a_2/a_1) greater than one.

Fig. 10. Shock-contact discontinuity interaction corresponding to a speed of sound ratio (a_2/a_1) less than one.

Fig. 11. Shock reflection off of a plane of symmetry in the hodograph and time-space planes.

Fig. 12. Steady wave solution corresponding to a finite increment in the burning speed.

Fig. 13. (A) Shock-deflagration merging resulting in a burning speed less than the Chapman-Jouguet value. (B) Shock-deflagration merging resulting in a Chapman-Jouguet deflagration. (C) Solution breakdown for burning speeds less than the Chapman-Jouguet value.

Fig. 14. (A) Deflagration-contact discontinuity interaction resulting in a burning speed less than the Chapman-Jouguet value. (B) Deflagration-contact discontinuity interaction resulting in a Chapman-Jouguet deflagration. (C) Solution breakdown for burning speeds less than the Chapman-Jouguet value. (D) Chapman-Jouguet deflagration-contact discontinuity interaction resulting in a Chapman-Jouguet detonation.

Fig. 15. Non-steady analysis of detonation-contact discontinuity interaction.

Fig. 16. Non-steady analysis of detonation-shock merging.

Fig. 17. Comparison of reflected shock pressure-space profiles: MacCormack (·); Cloud Code (+); Exact (-). Incident gasdynamic state: $P_1 = 2.430$; $V_1 = 0.514$; $U_1 = -0.833$; $M_n = 1.505$. Reflected gasdynamic state: $P_2 = 5.413$; $V_2 = 0.282$; $U_2 = 0.0$; $M_n = 1.444$.

Fig. 18. (A) Computational mesh surrounding a right running shock front. (B) Computational mesh surrounding a left running shock front.

Fig. 19. Difference schemes associated with the five possible relative motions of a shock wave and a translating grid.

Fig. 20. Density-space profiles of a contact discontinuity after 40 cycles of MacCormack scheme. Conditions: $\rho_2 = 2.026$; $\rho_1 = 2.133$; $P = 1.493$; $U = -0.456$.

Fig. 21. (A) Riemann's steady wave solution for the two possible pressure ratios corresponding to the independent states 1 and 2. Trajectories; solid line - shock; dashed line - contact discontinuity; chain dotted and chain double dotted lines-head and tail of rarefaction wave. (B) Computational plane surrounding a contact discontinuity.

Fig. 22. Difference schemes associated with the four possible relative motions of a contact discontinuity and a translating grid.

Fig. 23. Difference scheme associated with deflagration.

Fig. 24. Chapman-Jouguet detonation on the pressure-specific volume plane.

Fig. 25. Difference schemes associated with the two possible relative motions of a detonation and a stationary grid.

Fig. 26. Pressure, density, and particle velocity-space profiles of a Chapman-Jouguet detonation wave in a closed end tube. Solid line - self-similar analysis, solid and dashed lines - numerical calculations. Conditions: $\gamma_F = 7$; $\gamma_u = 1.3$; $\gamma_b = 1.2$; $\mathcal{M} = 1$; $M_D = 5.12$.

Fig. 27. Difference schemes associated with the relative motions of a shock-shock system within a translating grid.

Fig. 28. Difference schemes associated with the relative motions of a shock-contact distontinuity system within a translating grid.

Fig. 29. Difference schemes associated with the relative motions of a shock-deflagration system within a translating grid.

Fig. 30. Difference schemes associated with the relative motions of a shock-detonation system within a stationary grid.

Fig. 31. Difference schemes associated with the relative motions of a contact discontinuity-shock system within a translating grid.

Fig. 32. Difference schemes associated with the relative motions of a contact discontinuity-deflagration system within a translating grid.

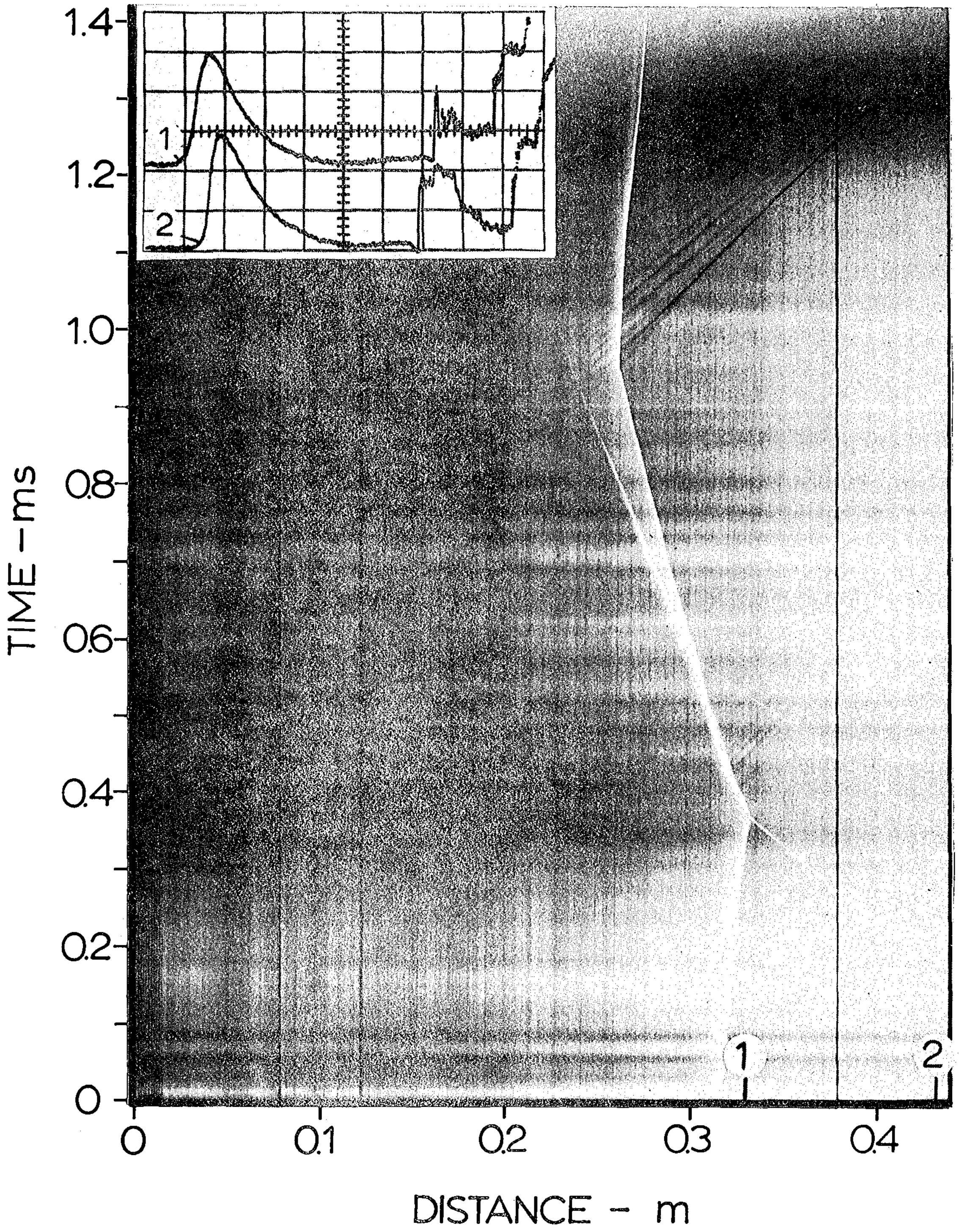
Fig. 33. Difference schemes associated with the interface between a contact discontinuity and a rarefaction within a stationary grid.

Fig. 34. Difference schemes associated with the relative motions of a deflagration-shock system within translating grid.

Fig. 35. Difference schemes associated with the relative motions of a deflagration-contact discontinuity system within a translating grid.

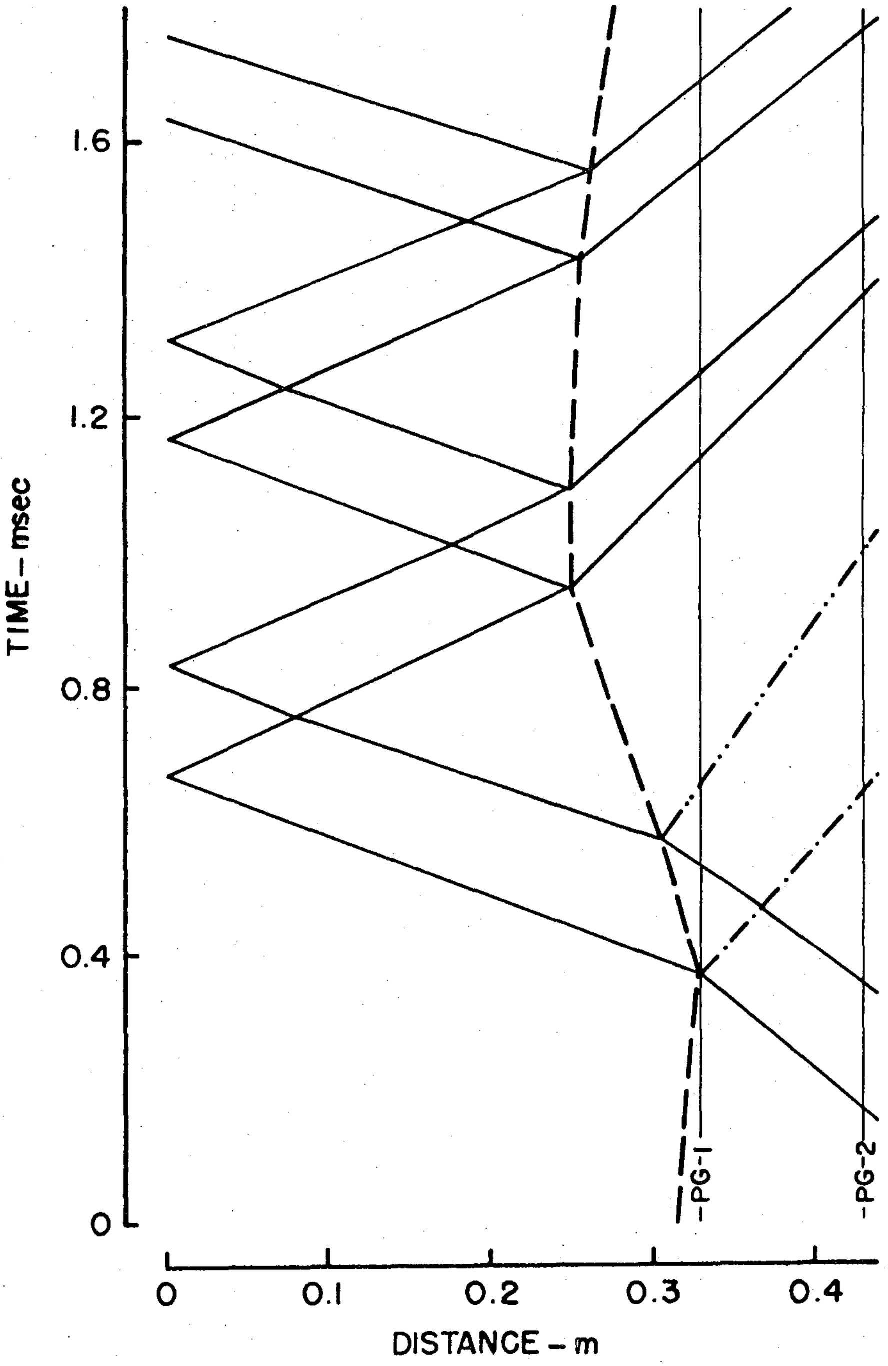
Fig. 36. Difference schemes associated with the relative motions of a detonation-shock system within a stationary grid.

Fig. 37. Difference schemes associated with the relative motions of a detonation-contact discontinuity system within a stationary grid.



XBB 782-1328

Fig. 1



XBL 781-6787

Fig. 2

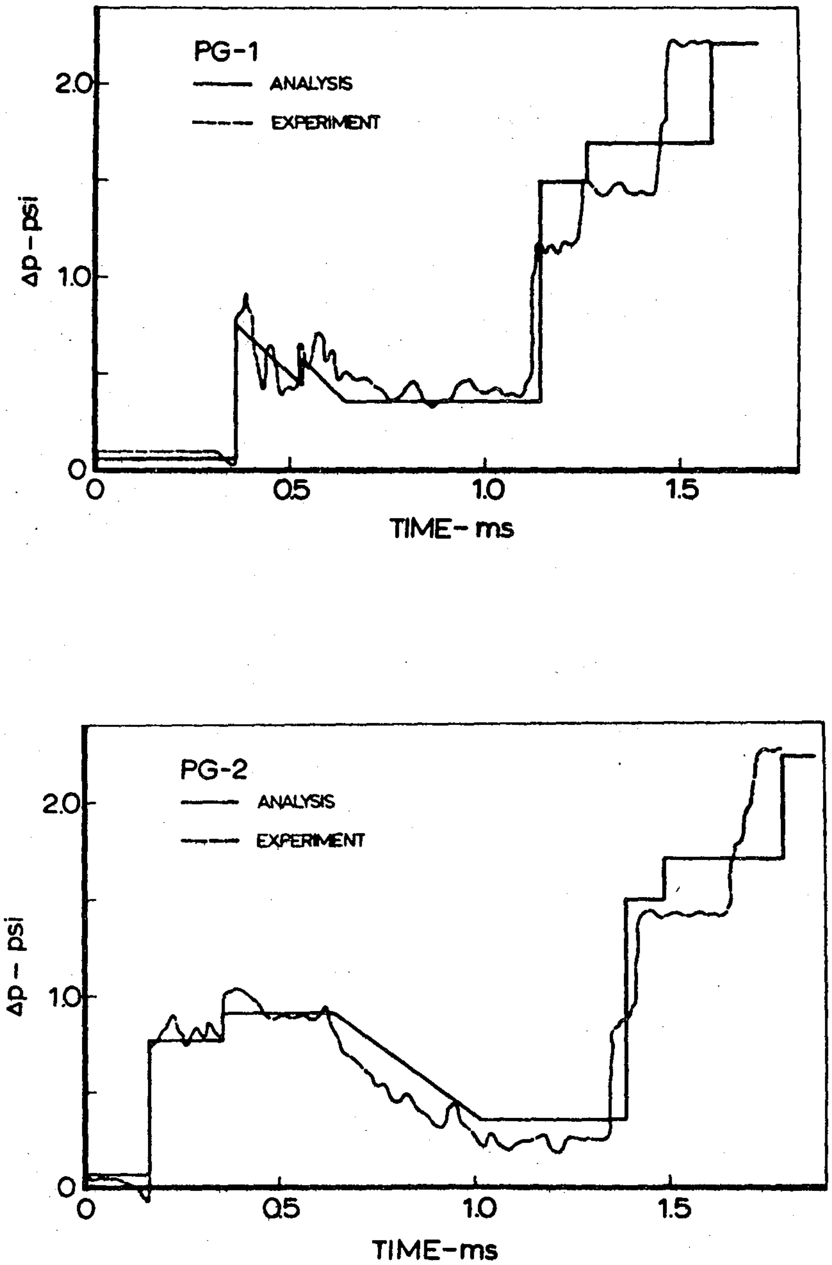
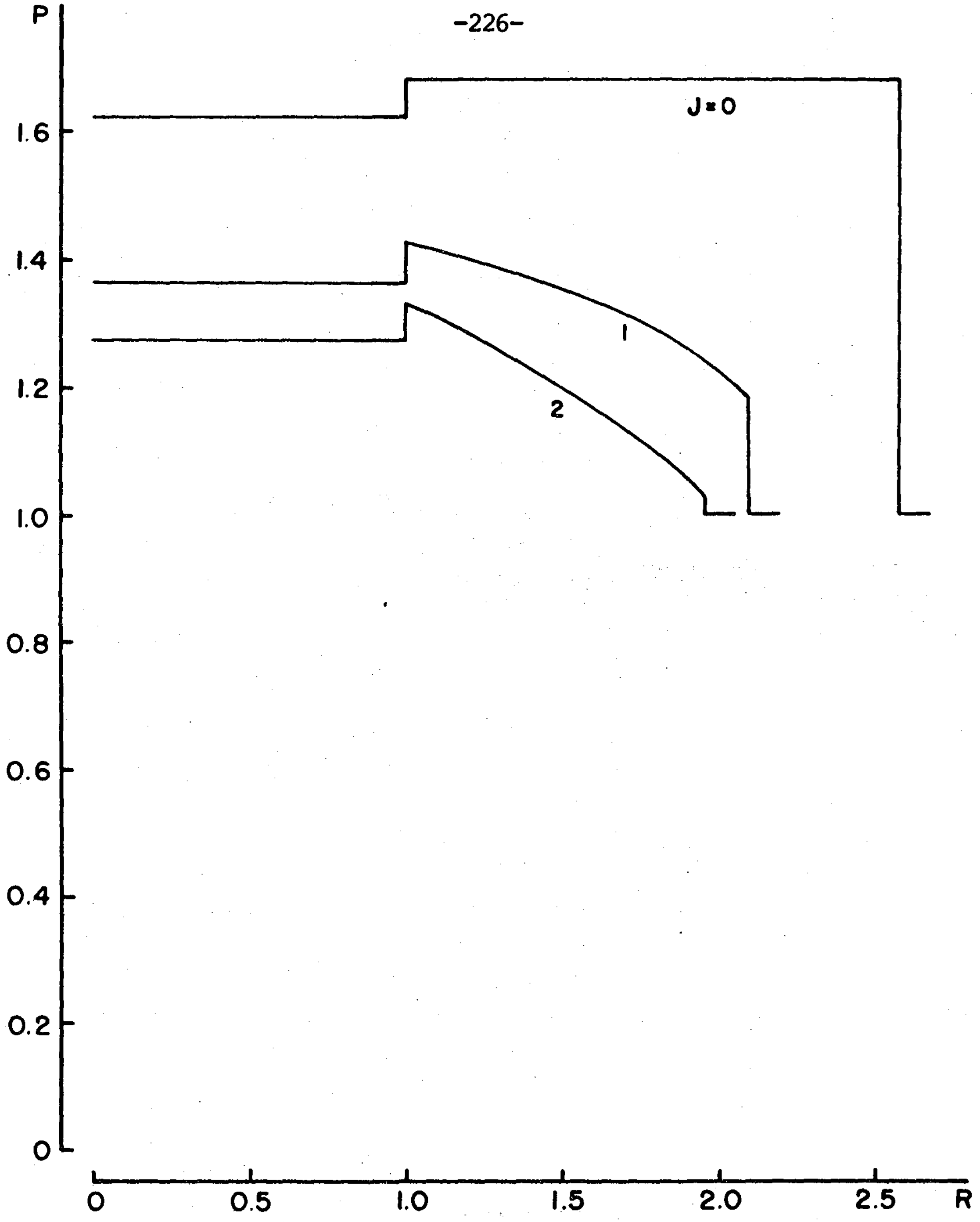
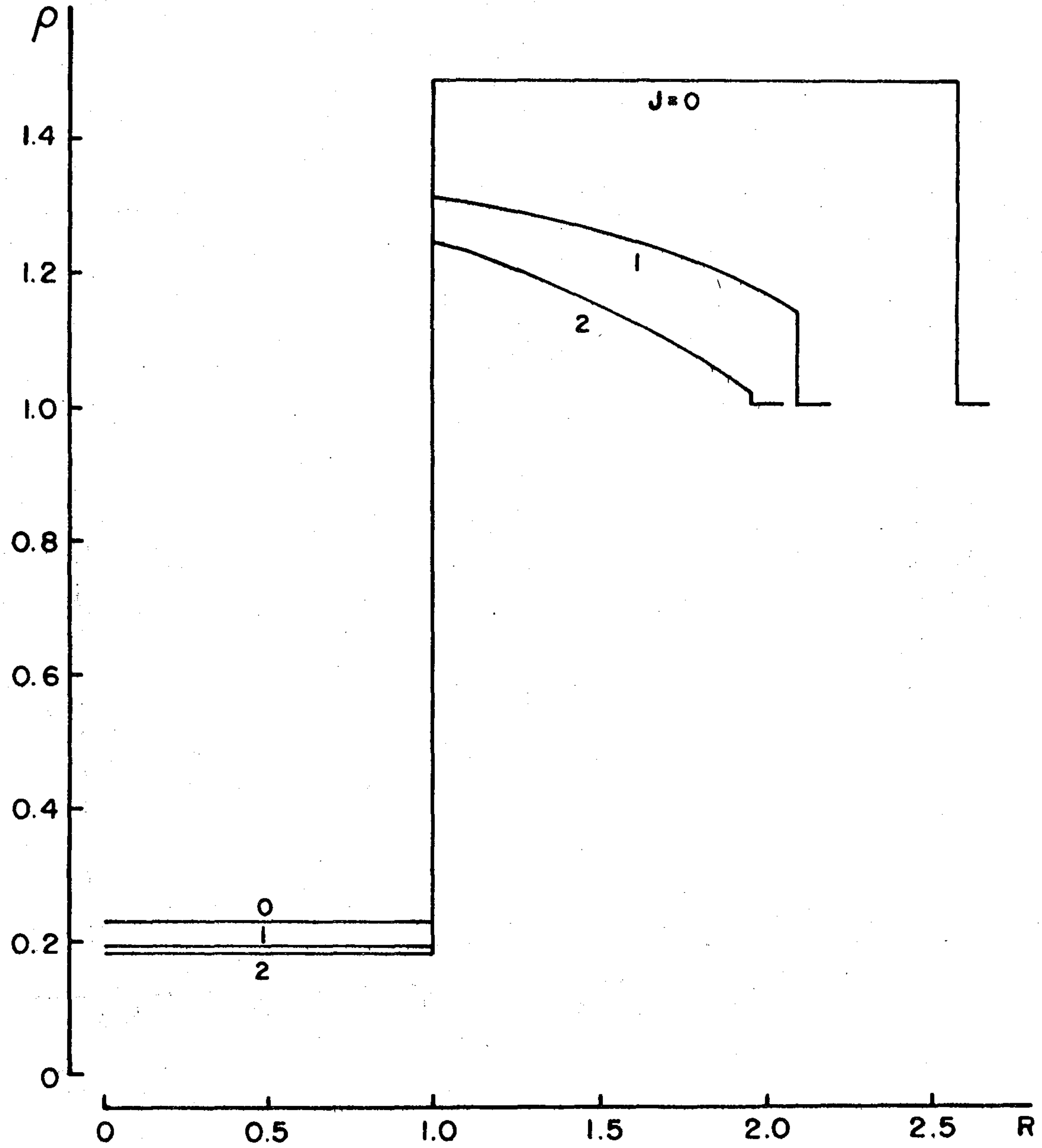

PG-1
ANALYSIS
EXPERIMENT
Δp - psi
2.0
1.0
0
0
0.5
1.0
1.5
TIME- ms
PG-2
ANALYSIS
EXPERIMENT
Δp - psi
2.0
1.0
0
0
0.5
1.0
1.5
TIME-ms

Fig. 3



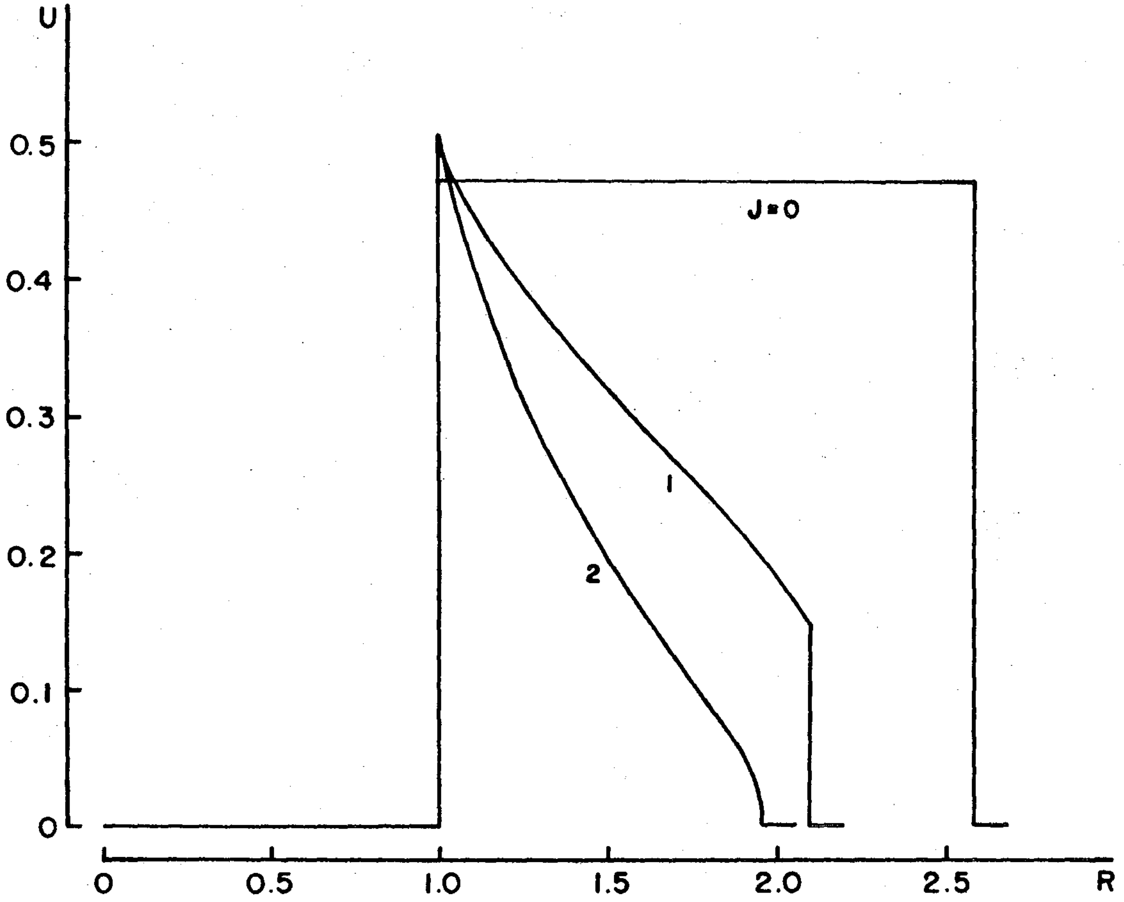
XBL 781-6788

Fig. 4



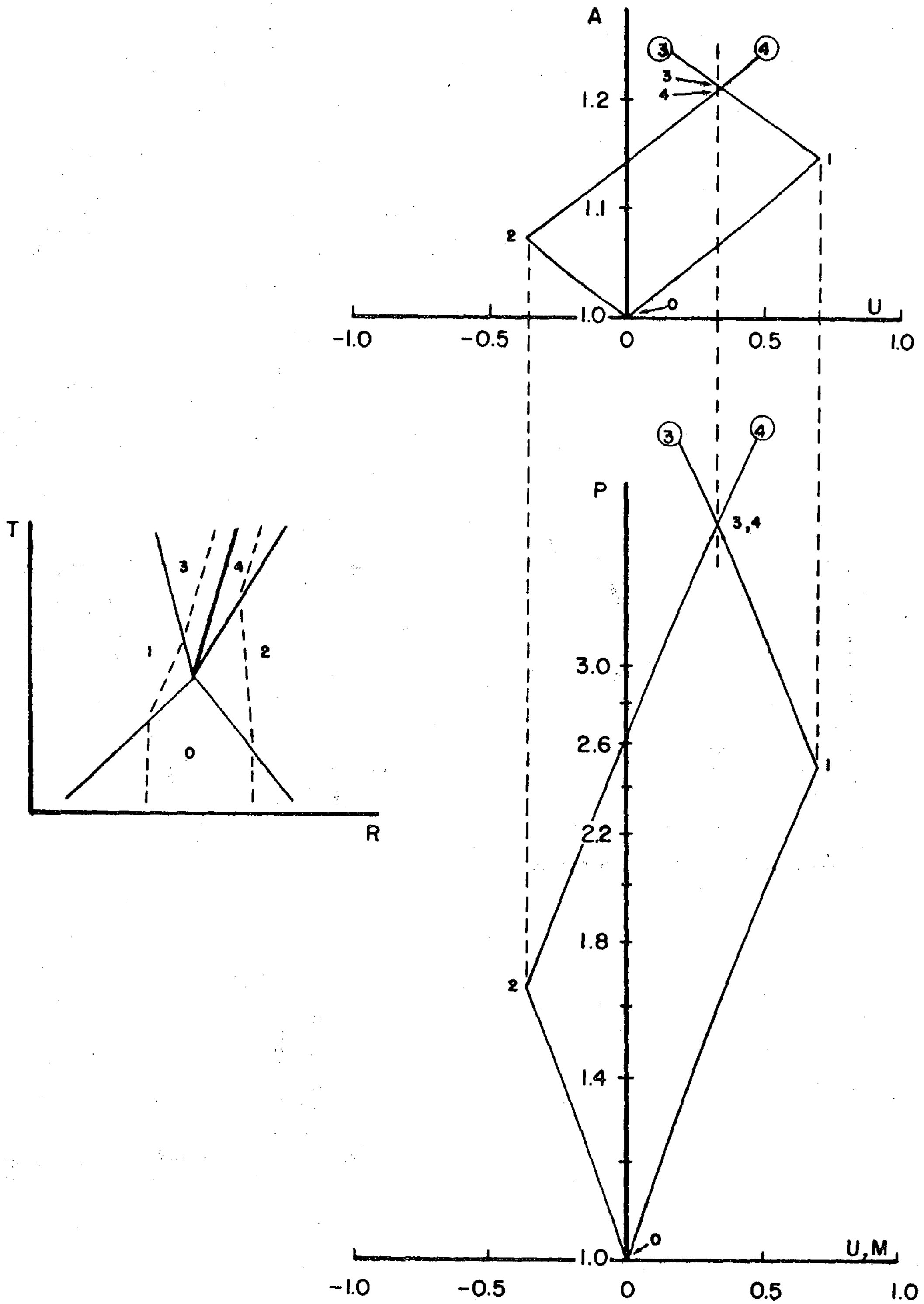
XBL 781-6789

Fig. 5



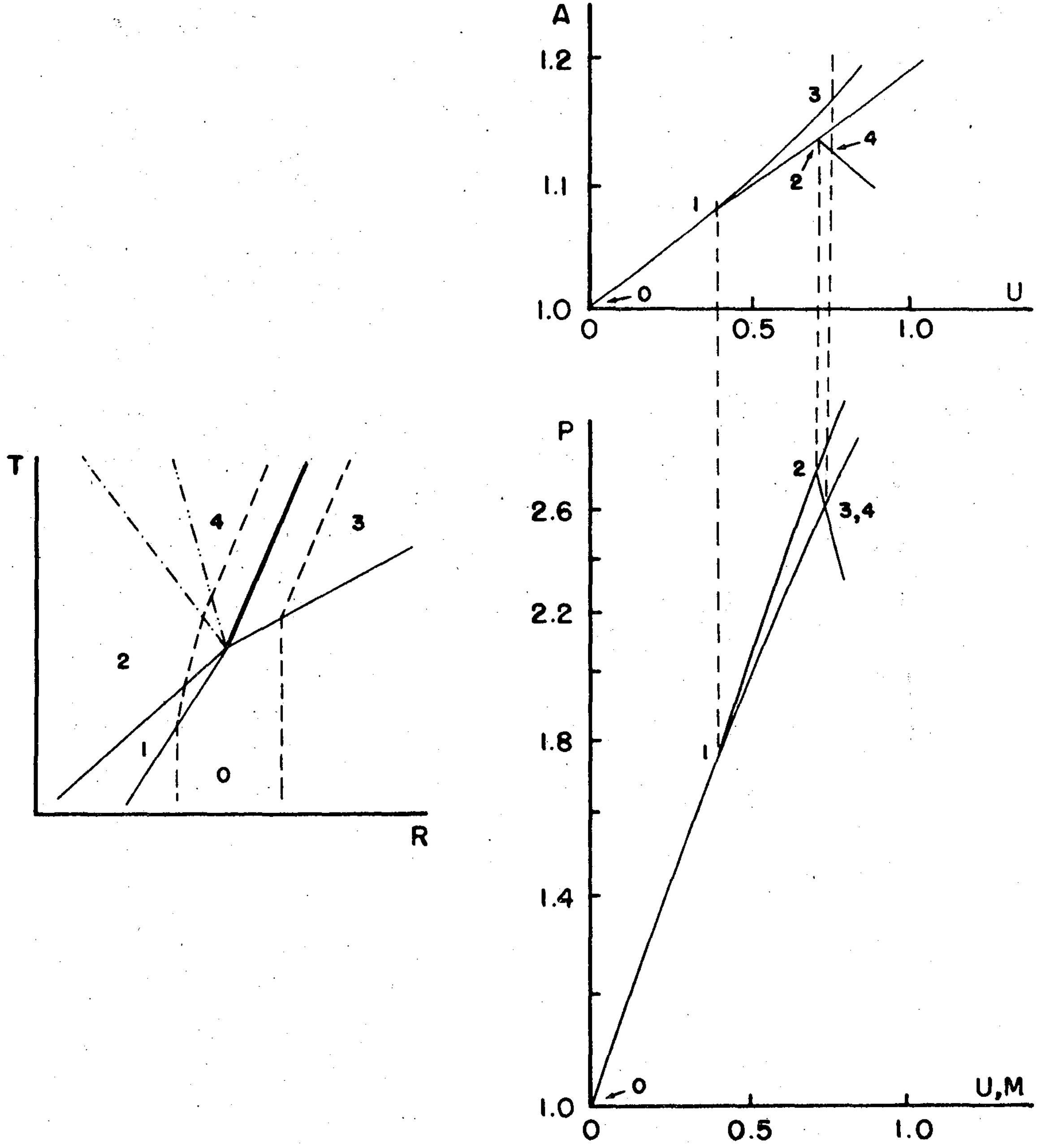
XBL 781-6790

Fig. 6



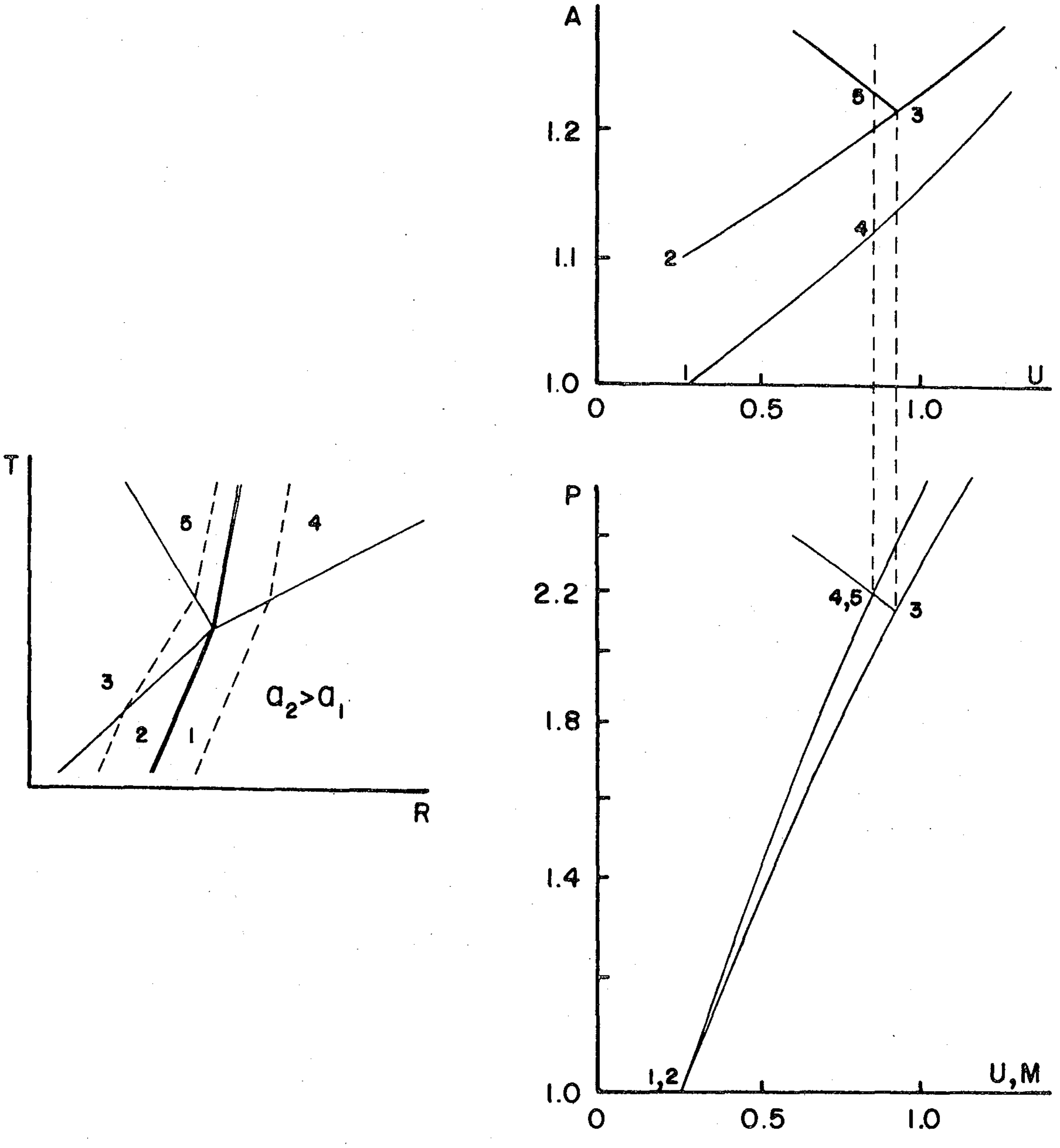
XBL 781-6791

Fig. 7



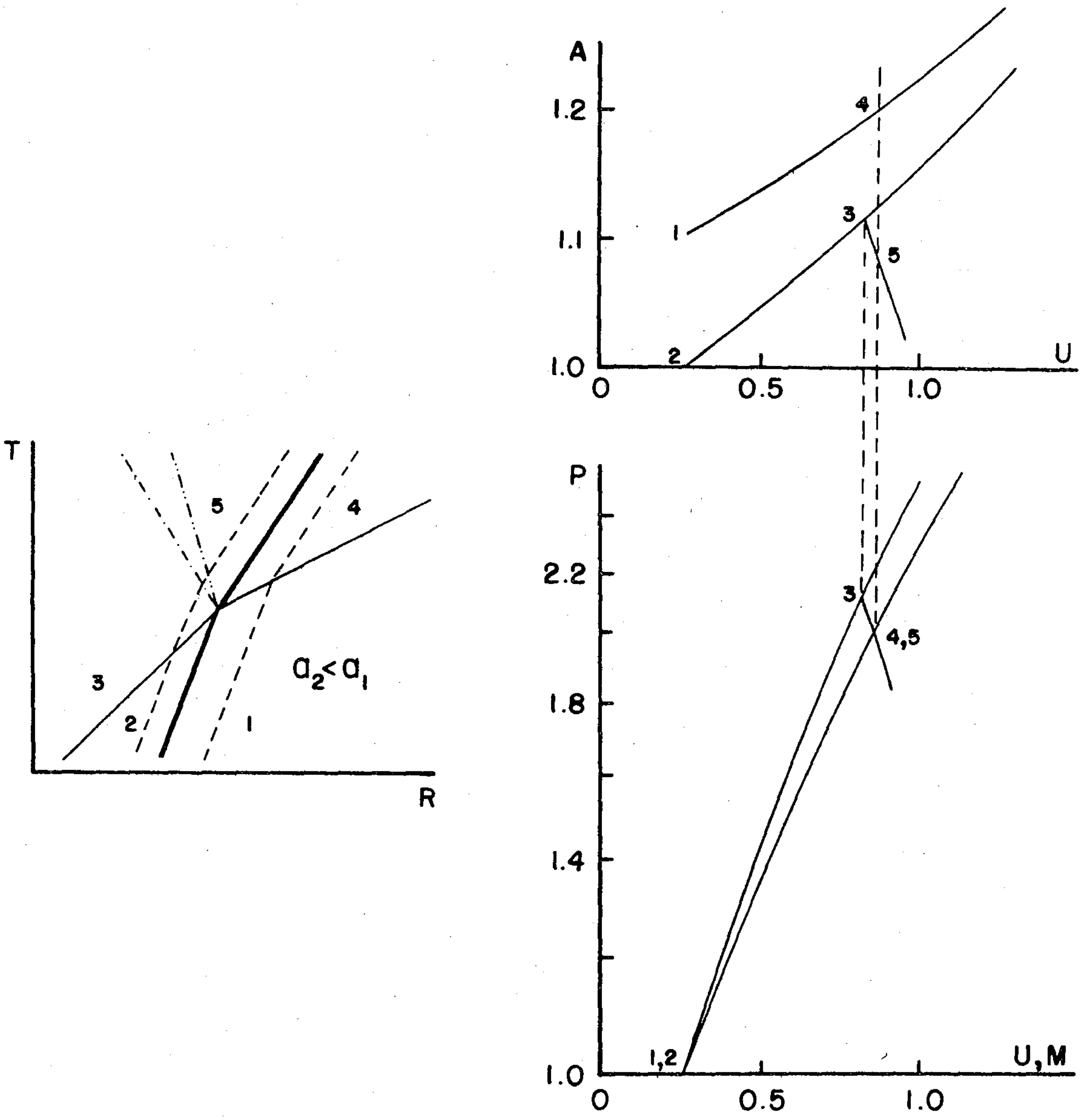
XBL 781-6792

Fig. 8



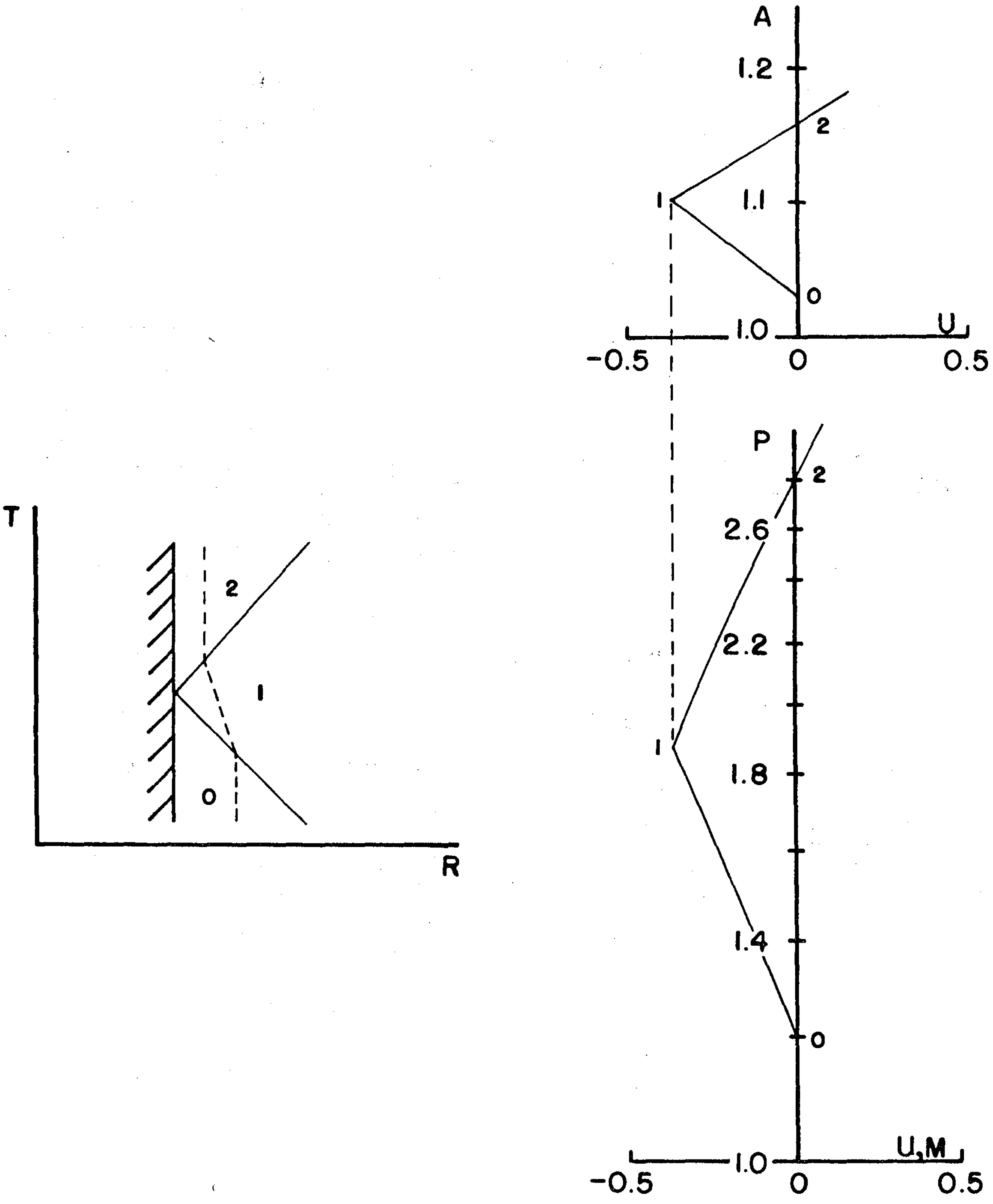
XBL 781-6793

Fig. 9



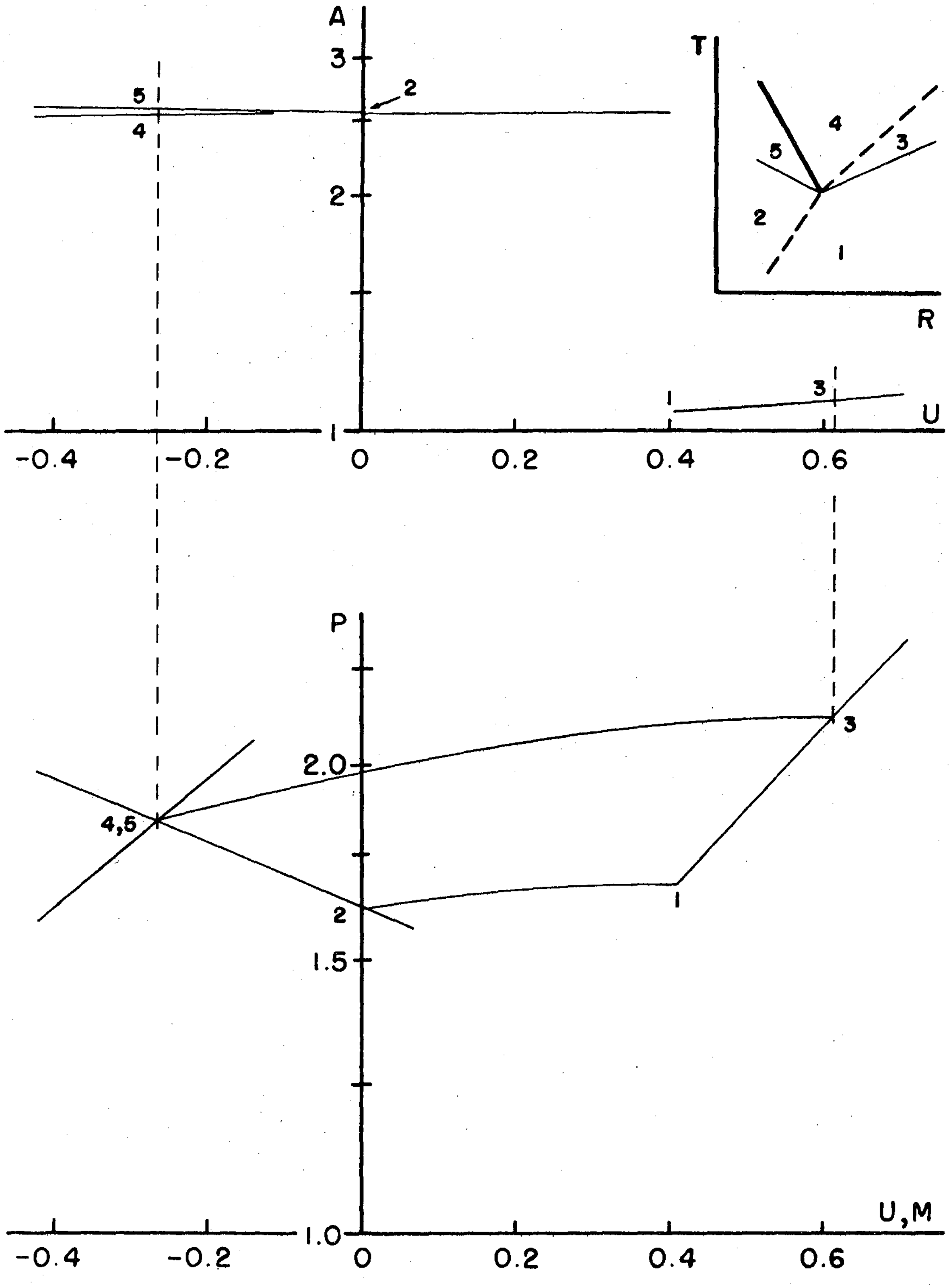
XBL 781-6794

Fig. 10



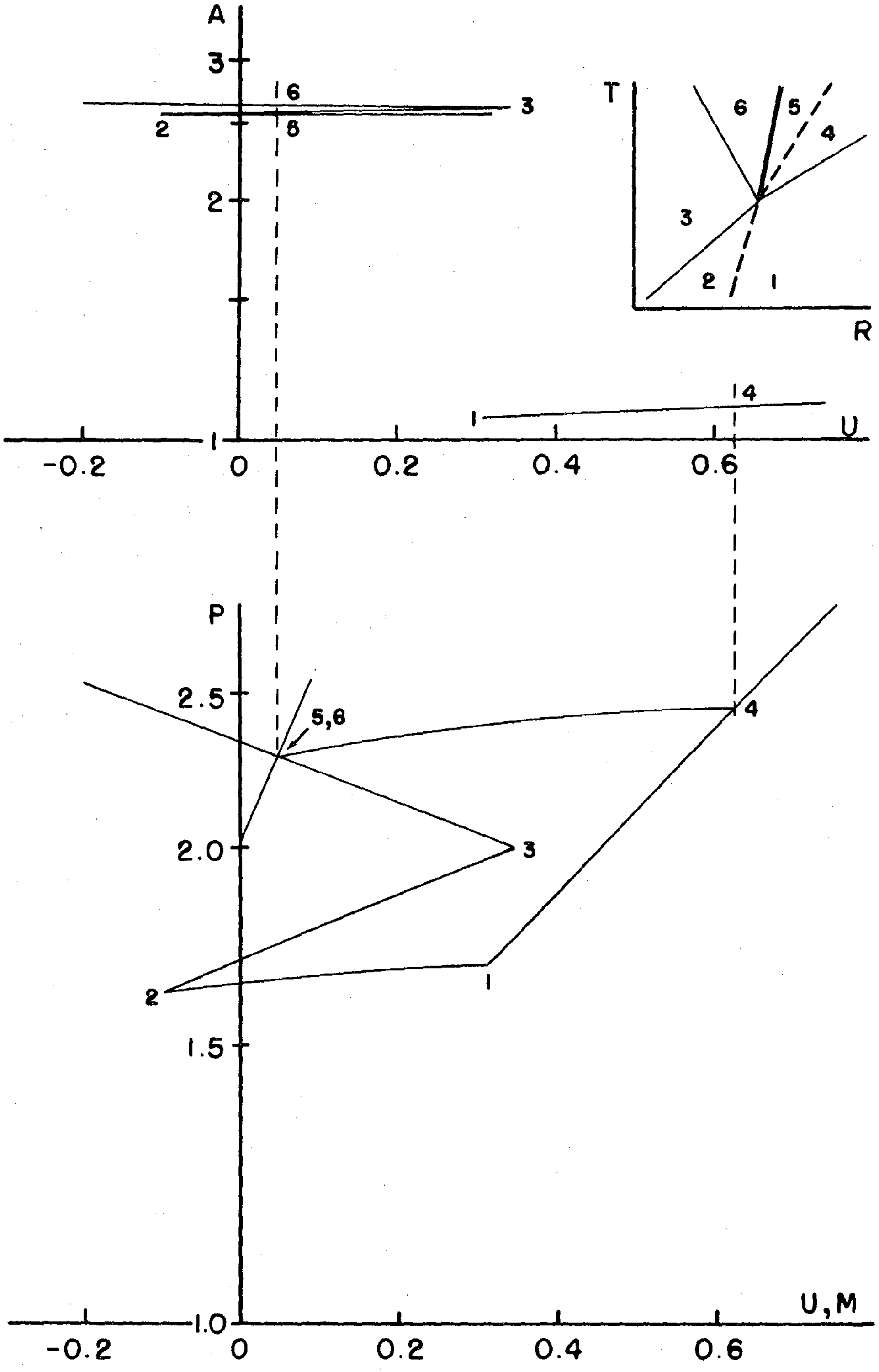
XBL 781-6795

Fig. 11



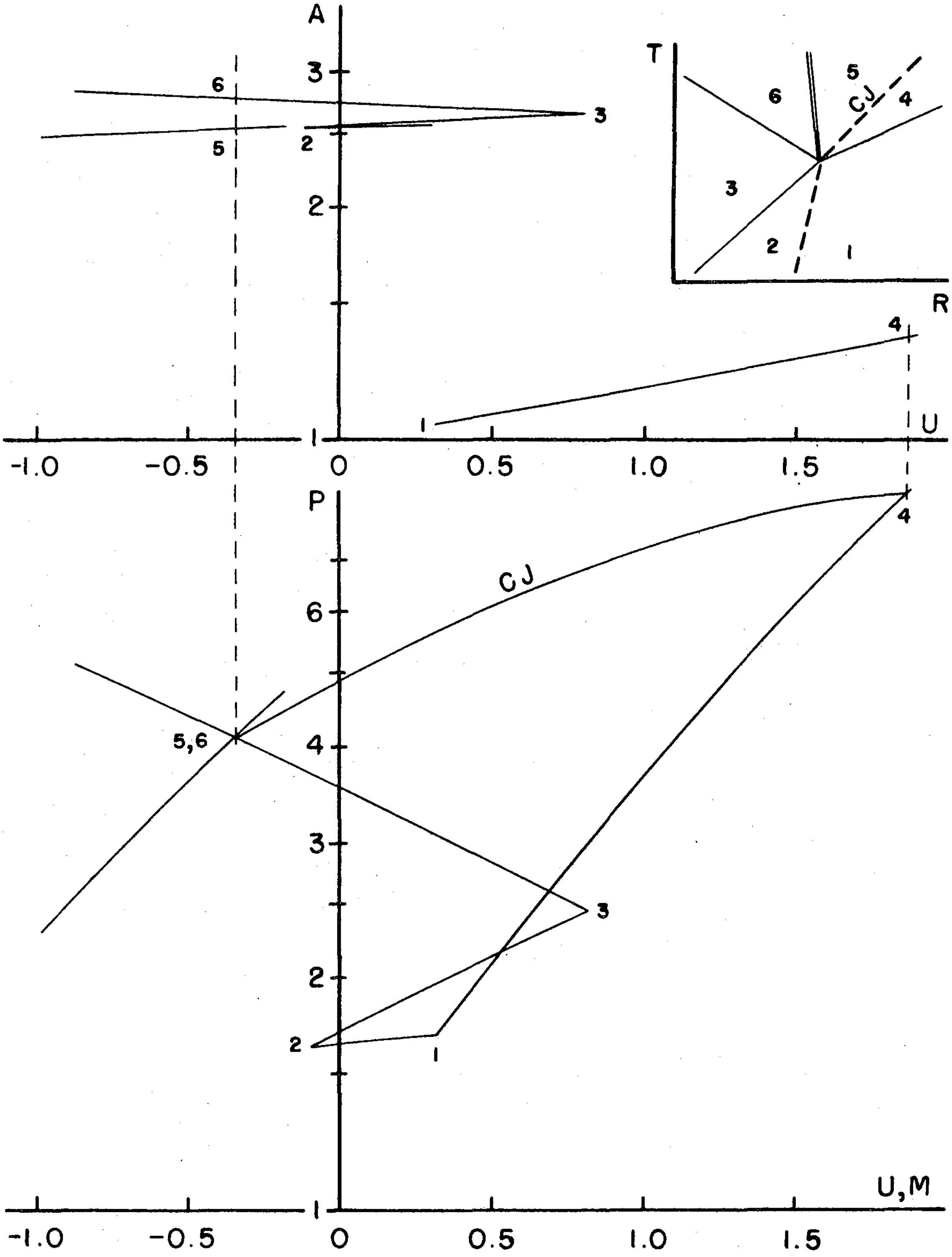
XBL 781-6796

Fig. 12



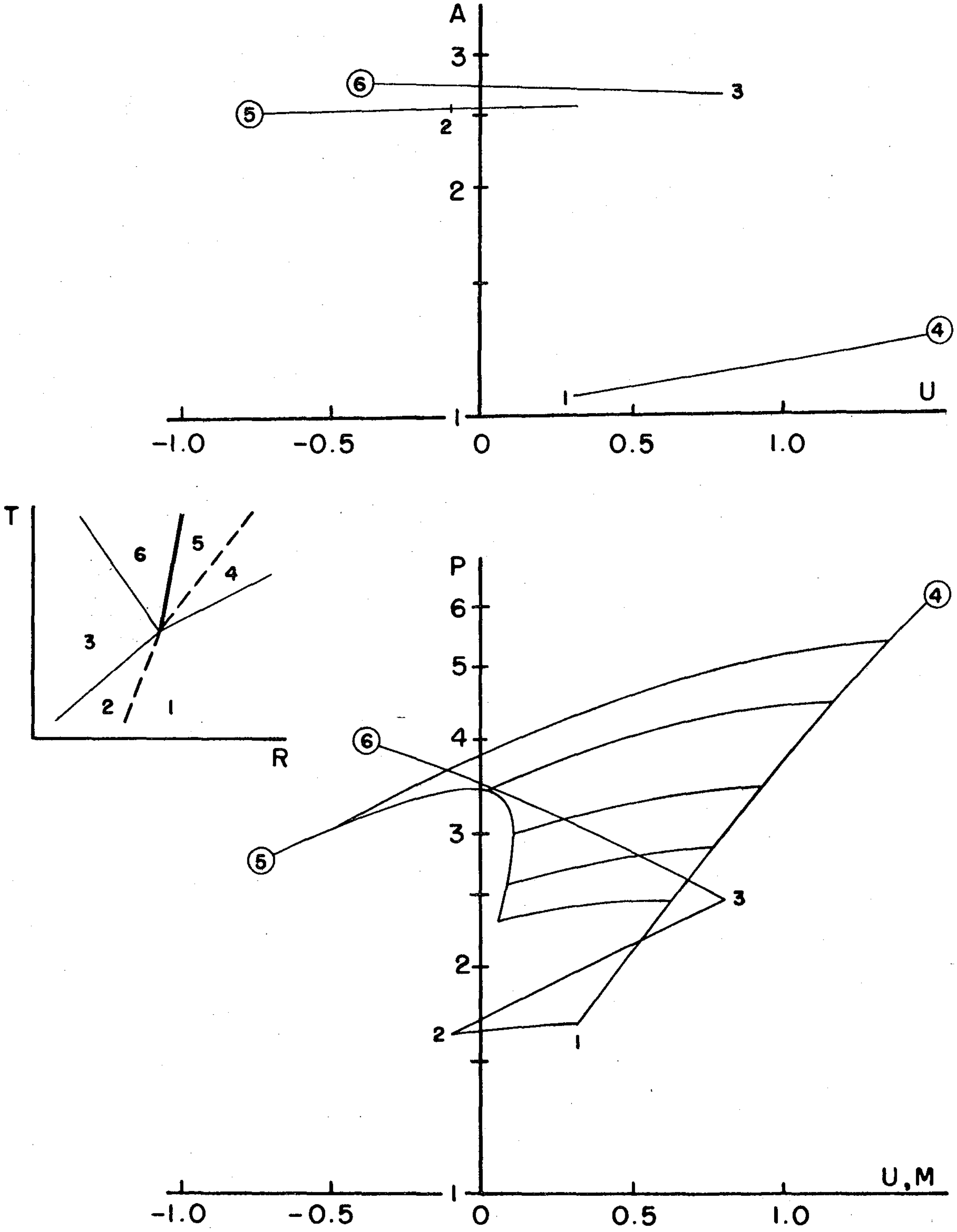
XBL 781-6797

Fig. 13A



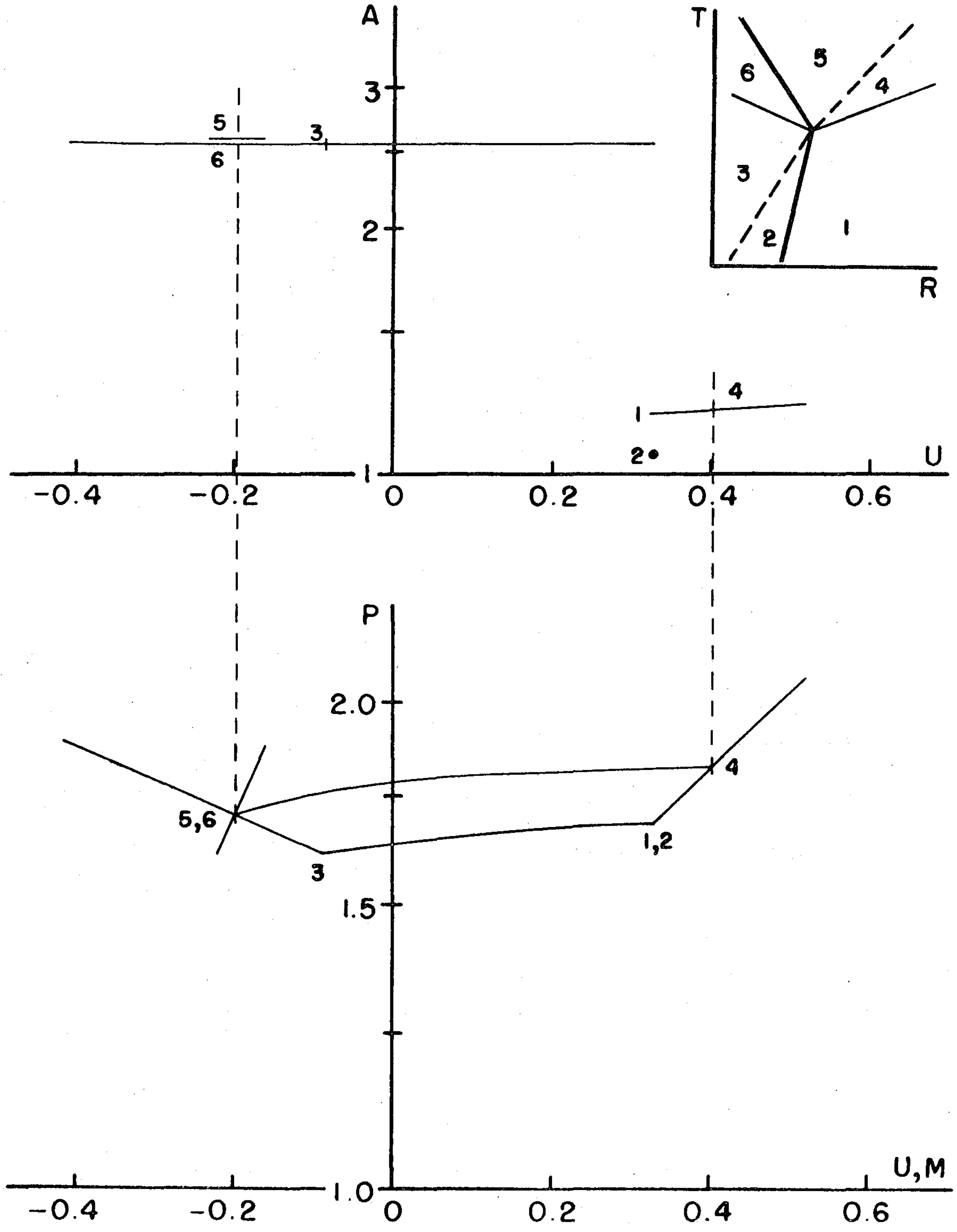
XBL 781-6799

Fig. 13B



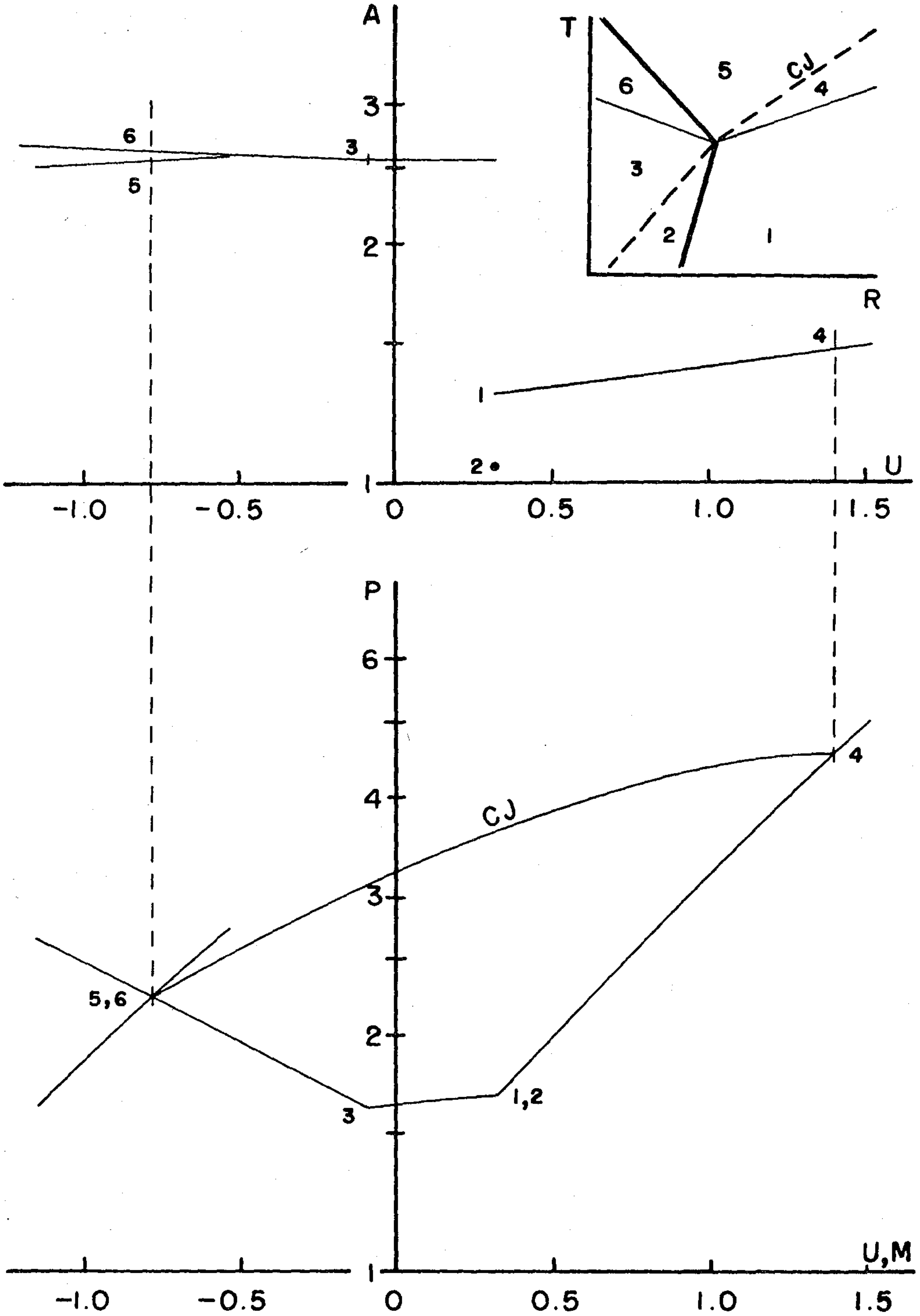
XBL 781-6798

Fig. 13C



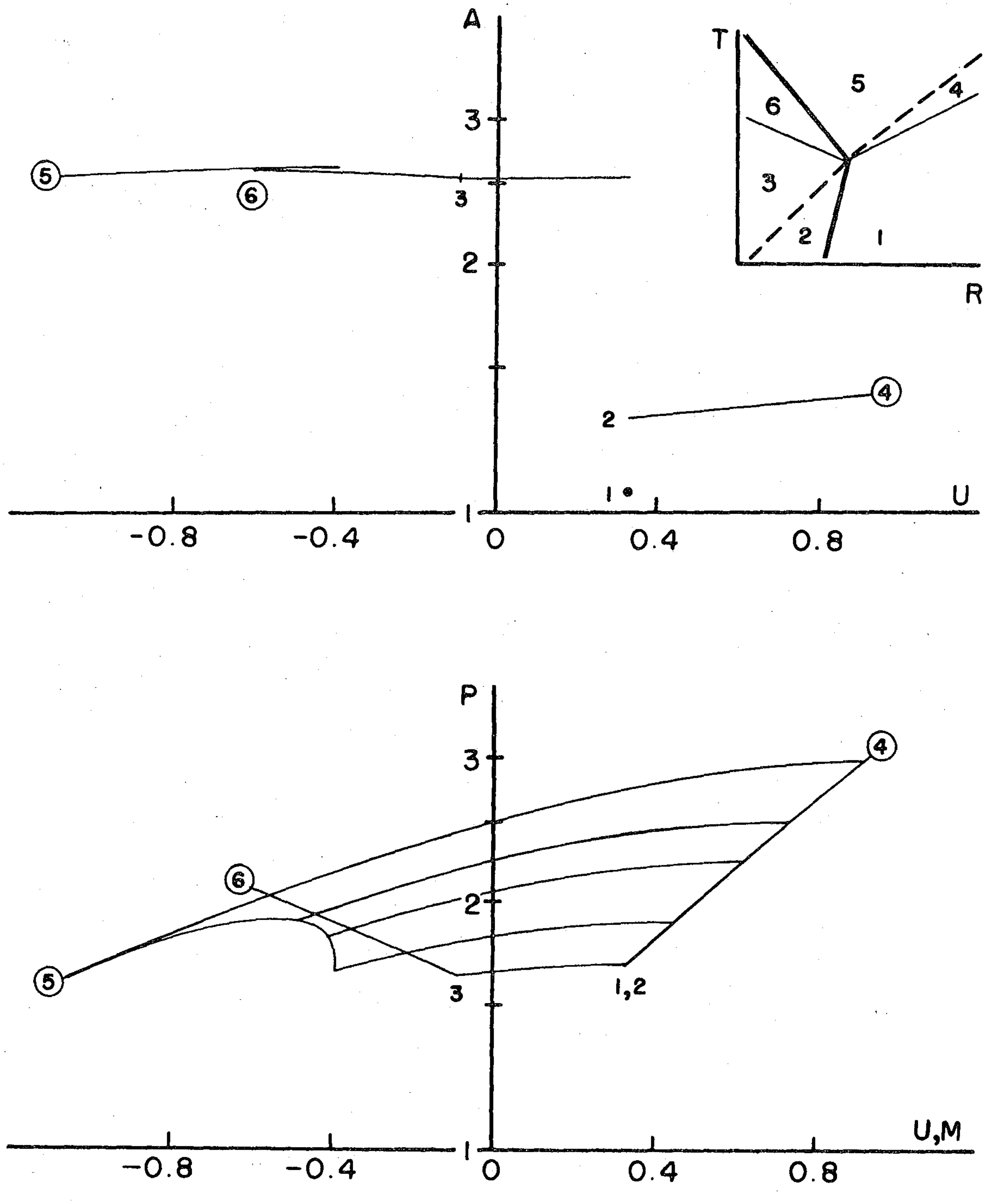
XBL 781-6800

Fig. 14A



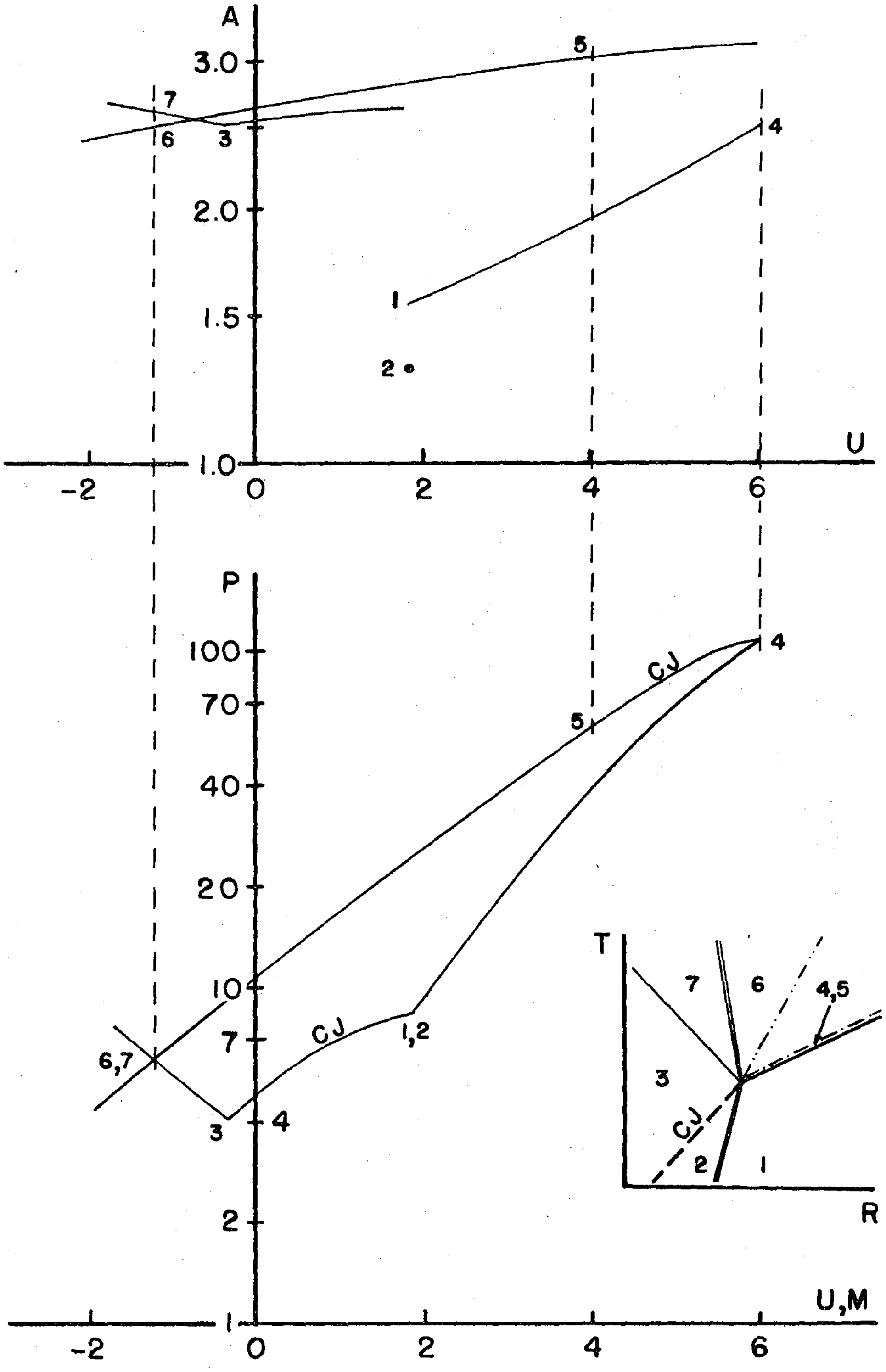
XBL 781-6801

Fig. 14B



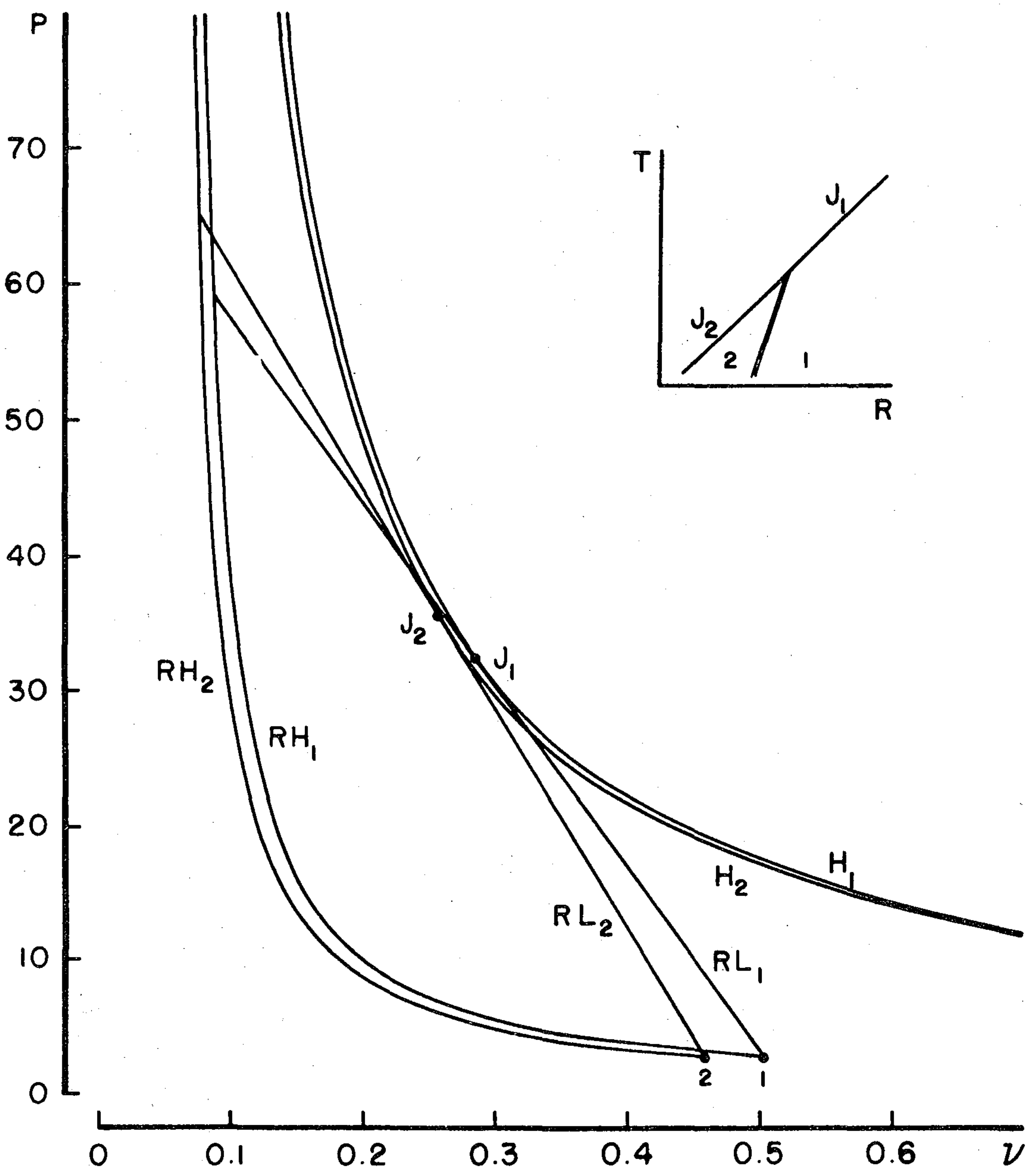
XBL 781-6802

Fig. 14C



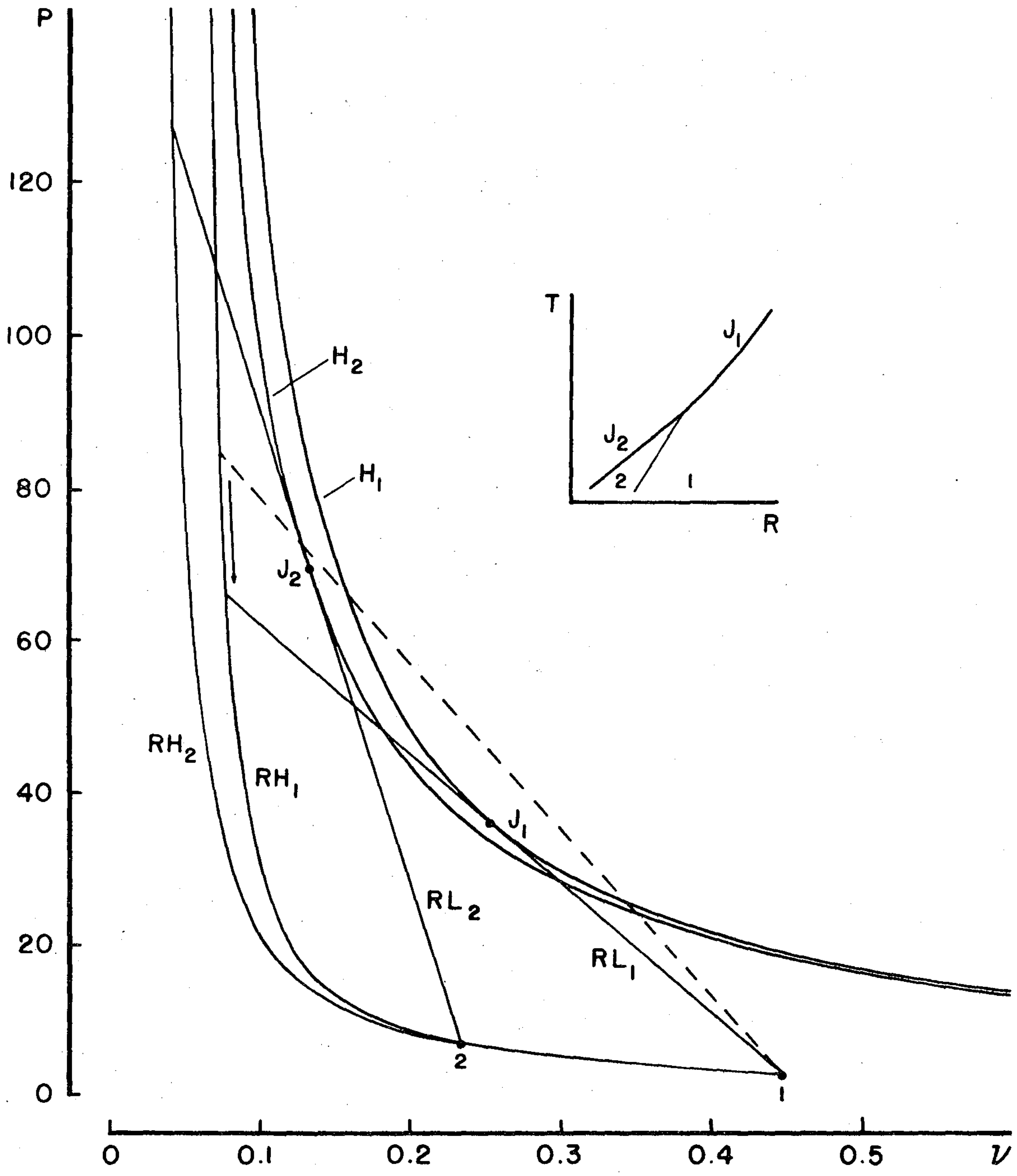
XBL 781-6803

Fig. 14D



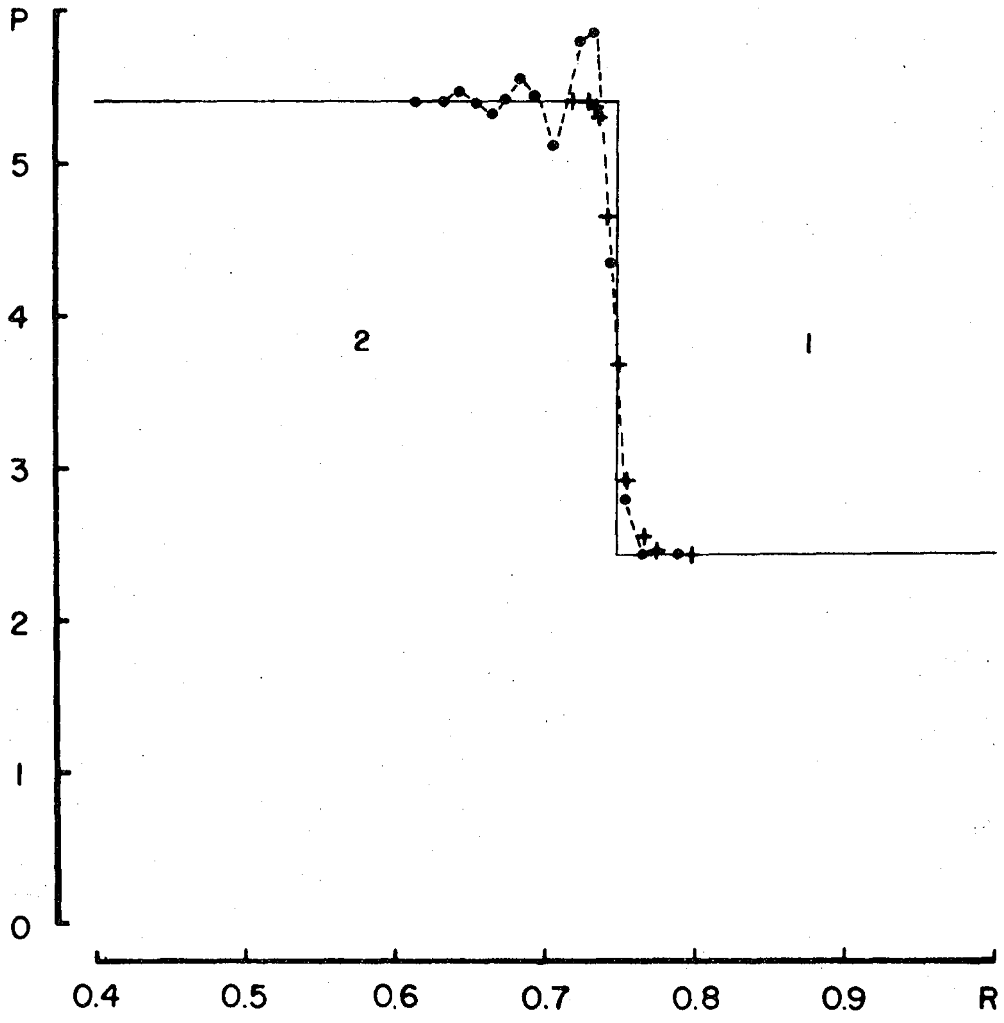
XBL 781-6804

Fig. 15



XBL 781-6805

Fig. 16



XBL 781-6806

Fig. 17

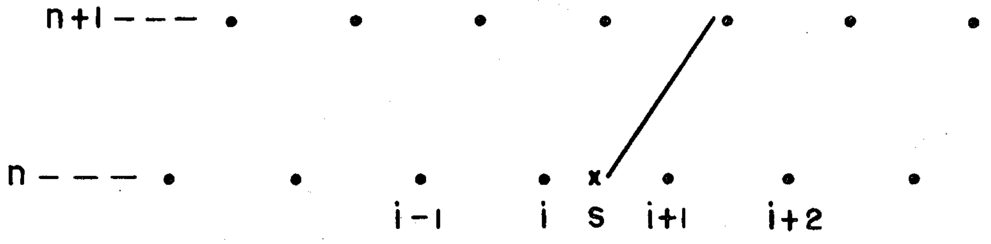
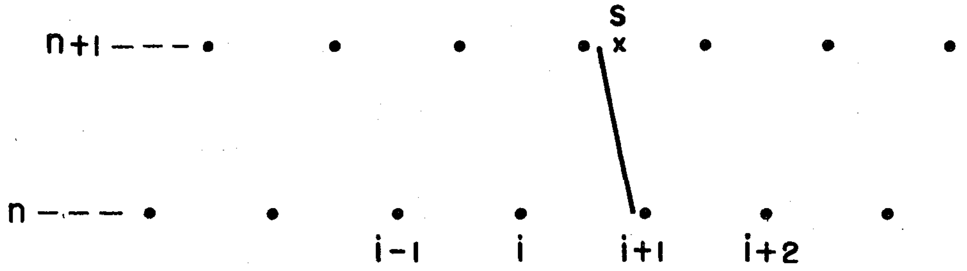
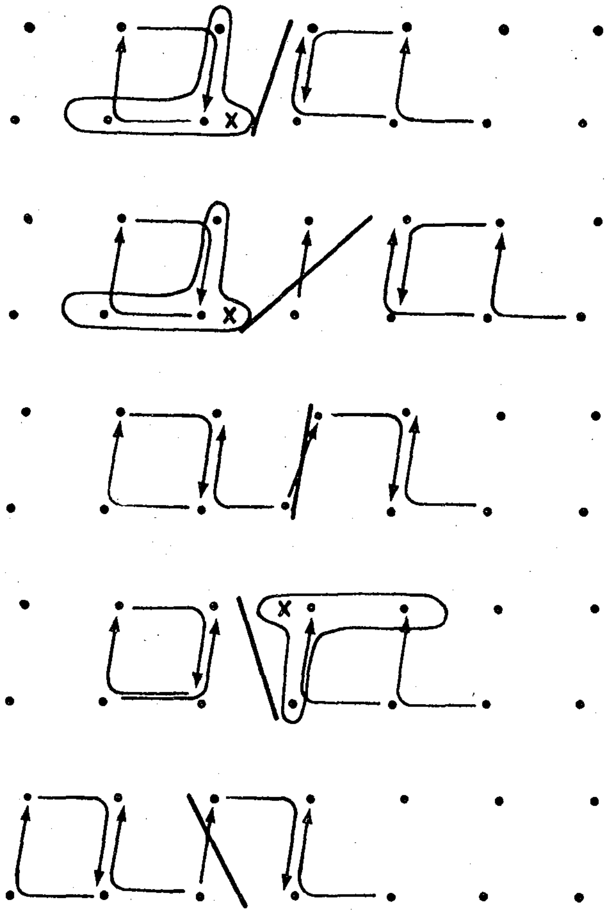


Fig. 18A



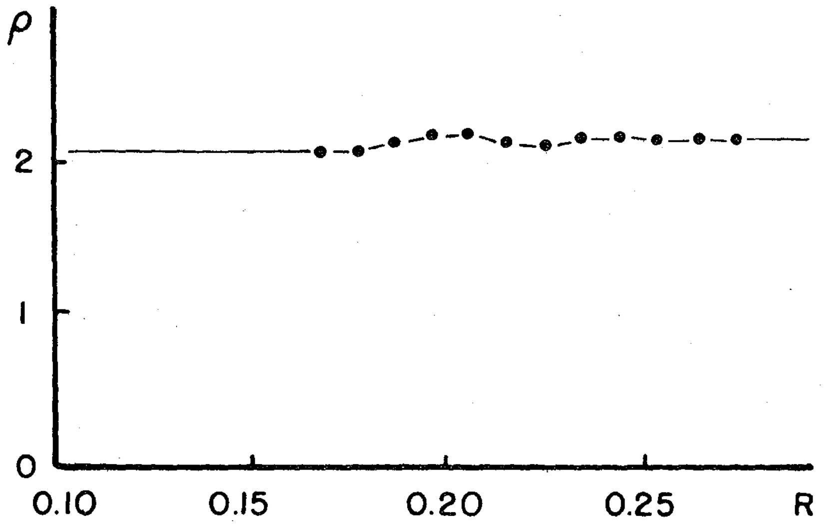
XBL 781-6807

Fig. 18B



XBL 781-6830

Fig. 19



XBL 781-6808

Fig. 20

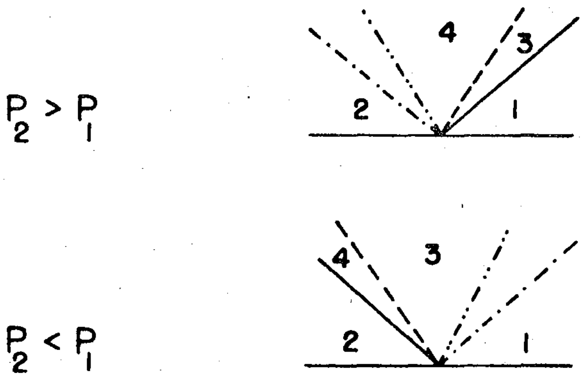
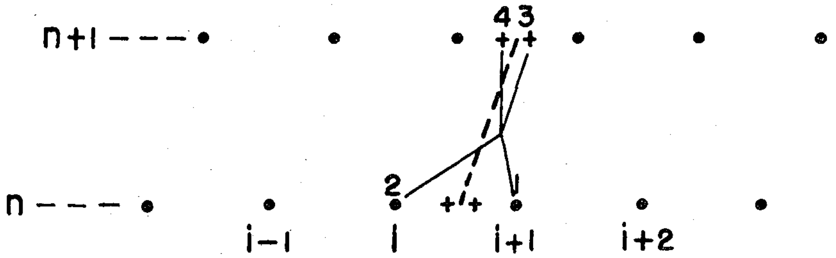
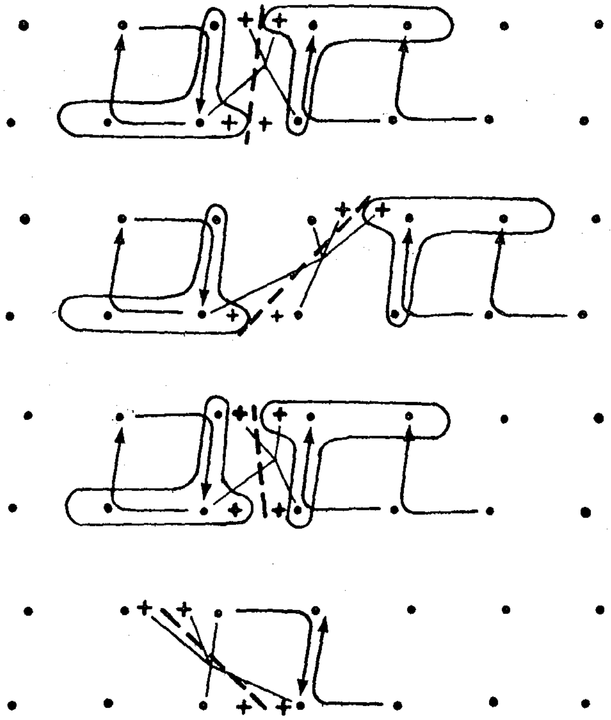


Fig. 21A



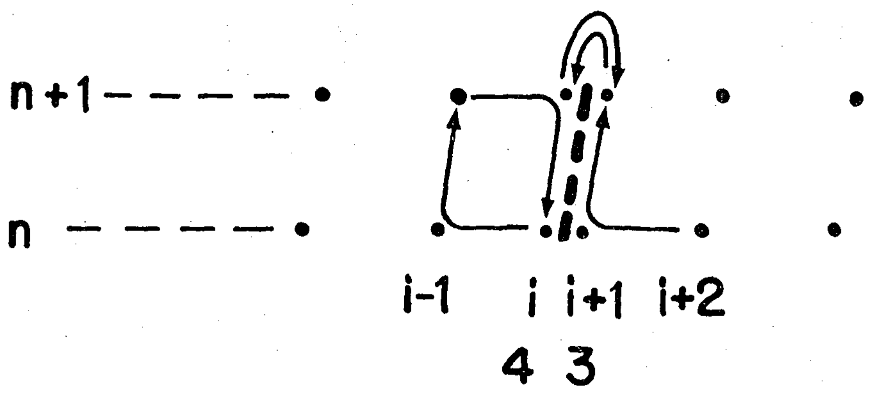
XBL 781-6809

Fig. 21B



XBL 781-6827

Fig. 22



XBL 781-6831

Fig. 23

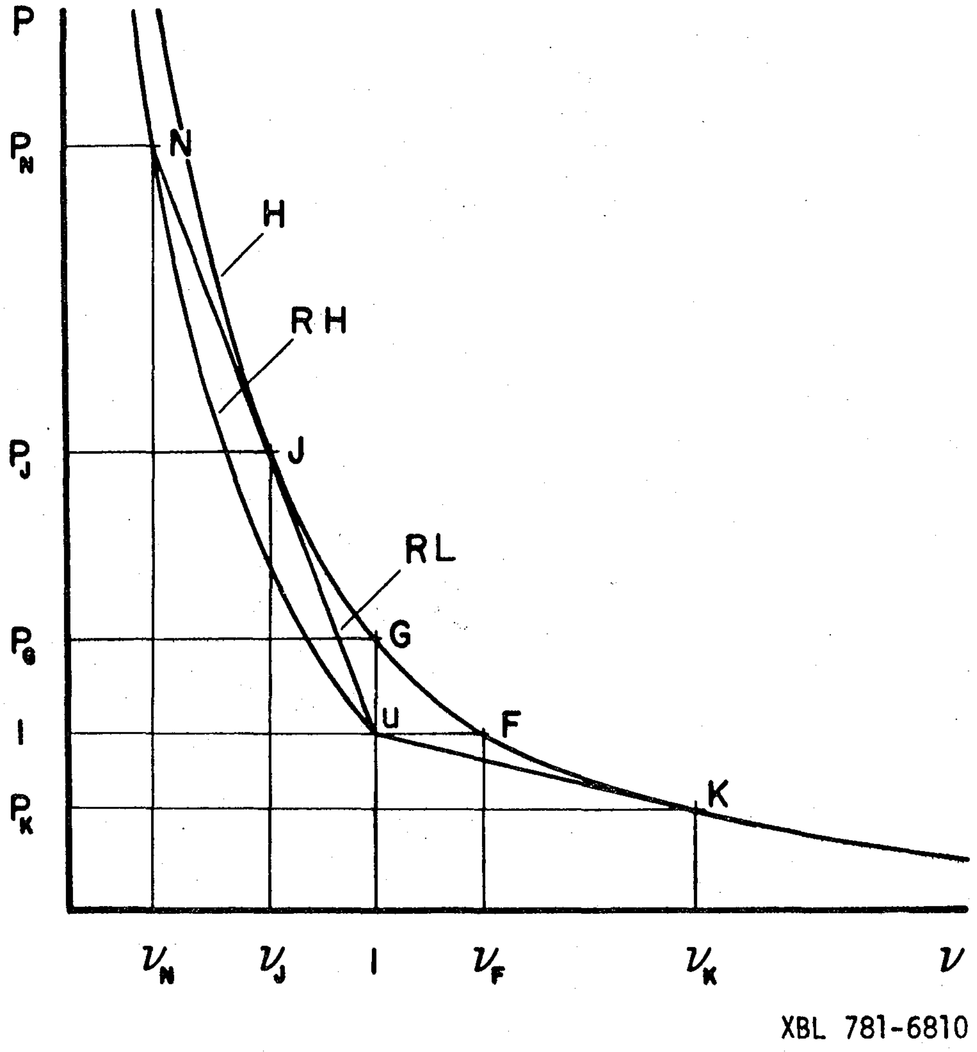
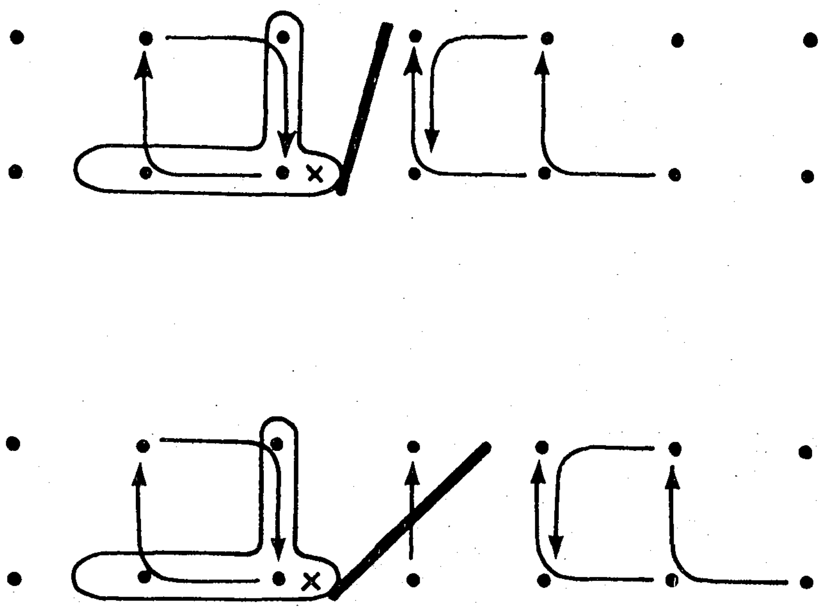
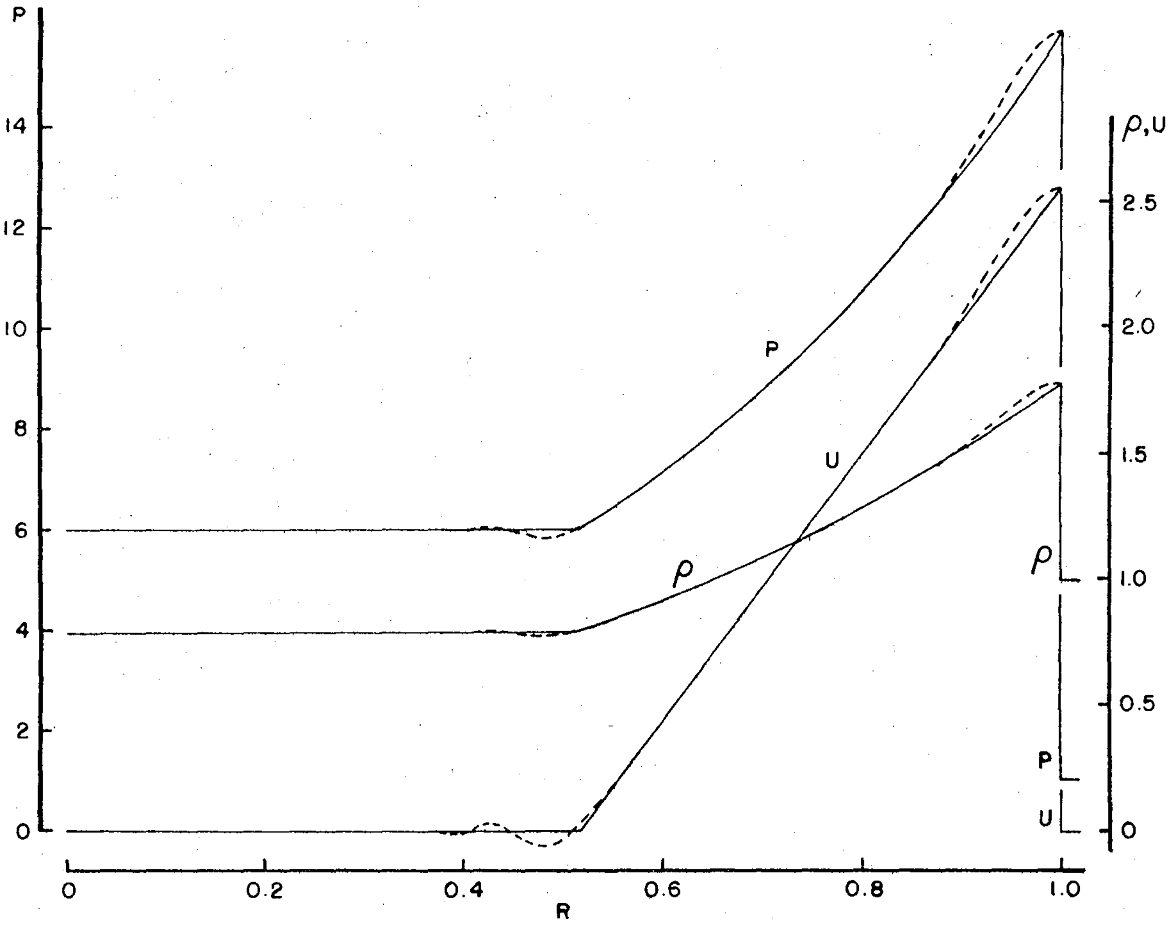


Fig. 24



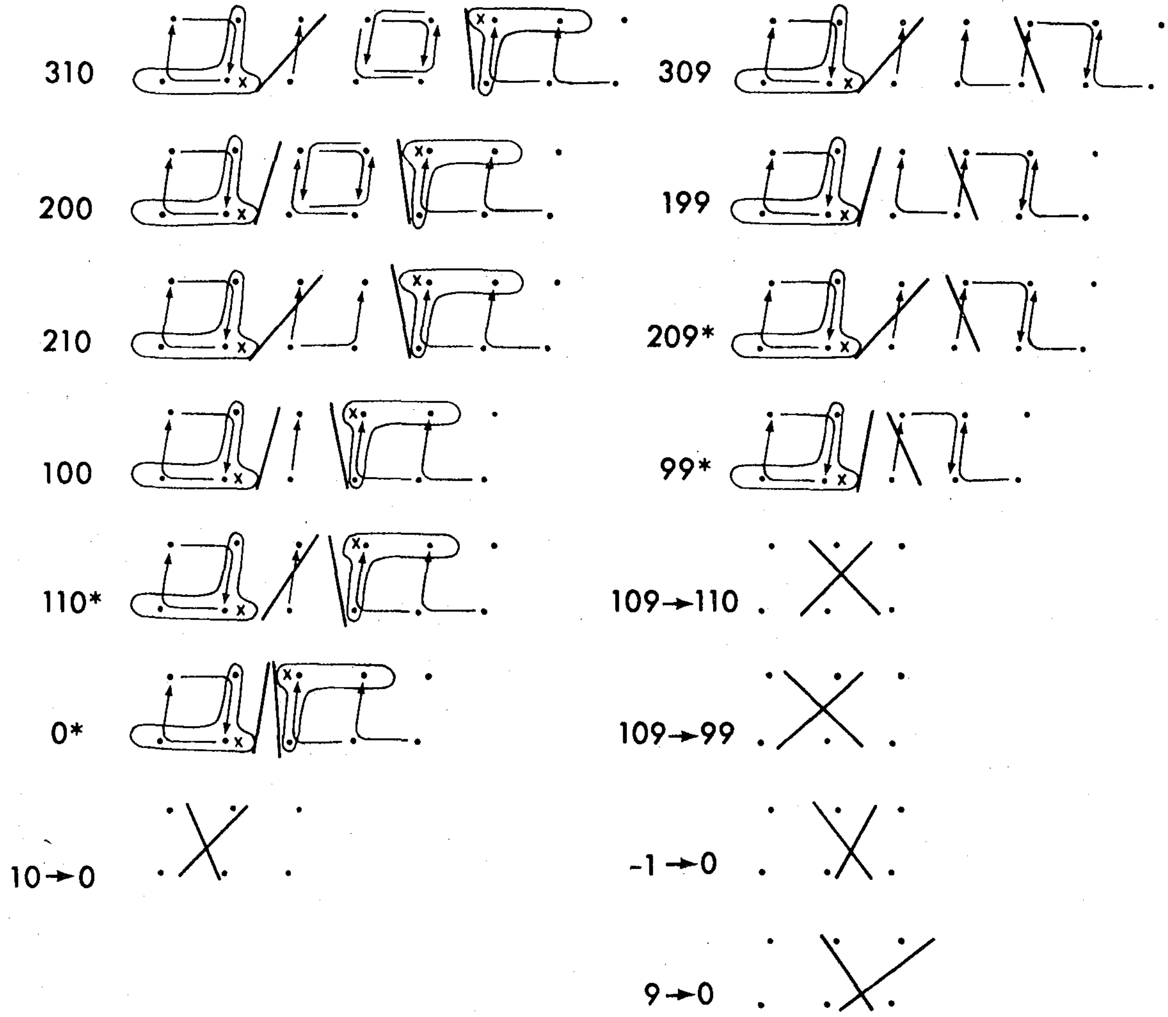
XBL 781-6832

Fig. 25



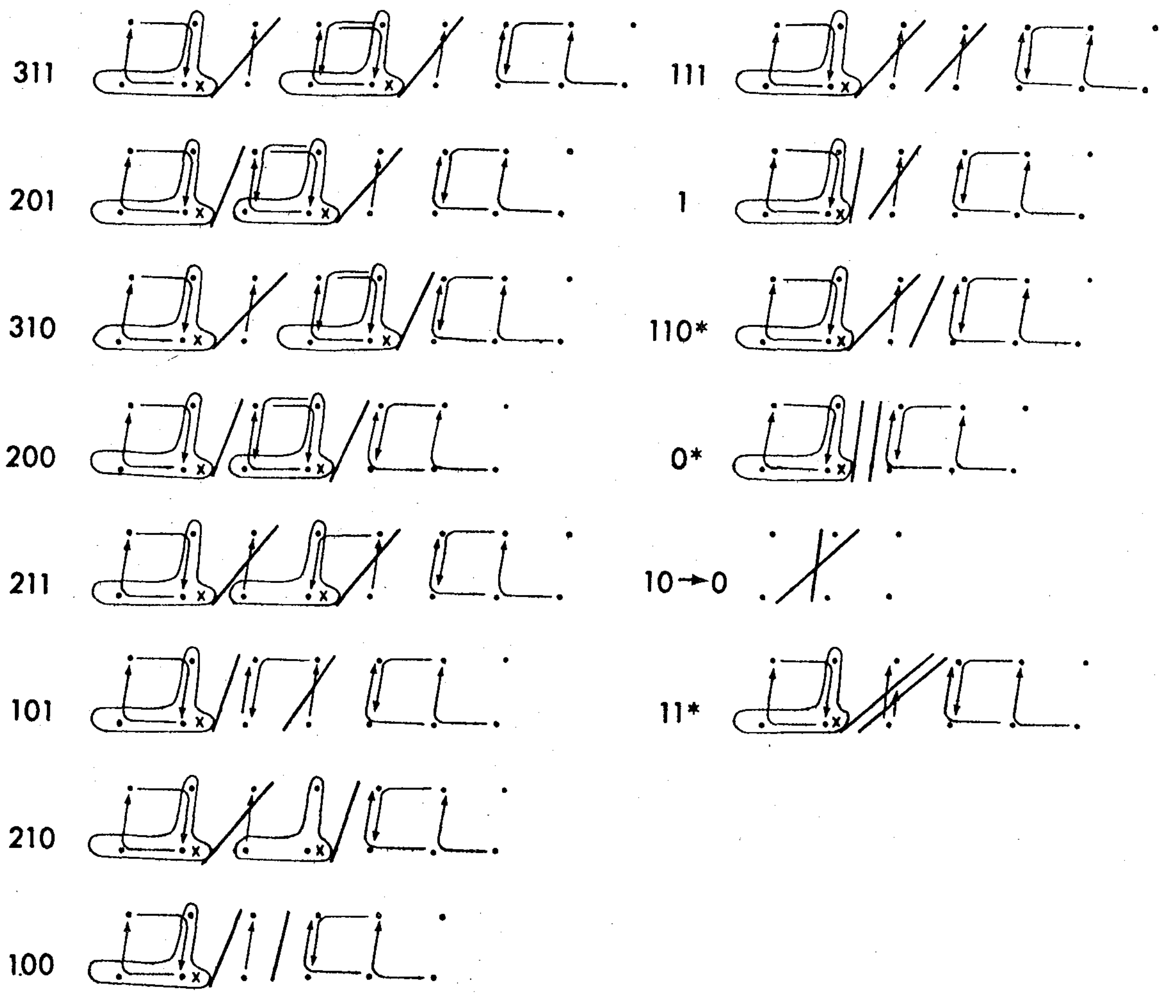
XBL 781-6811

Fig. 26



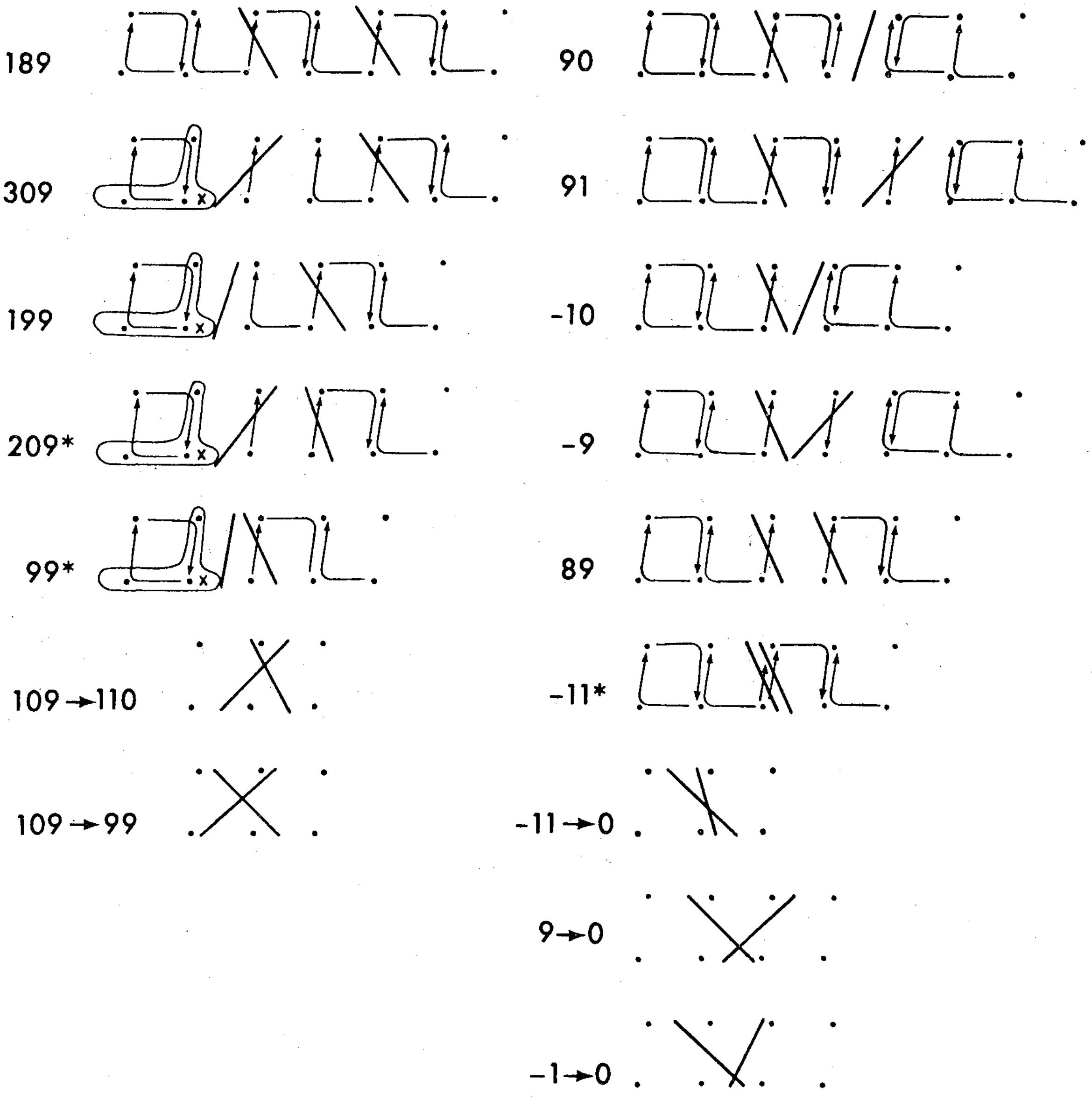
XBL 781-6837

Fig. 27



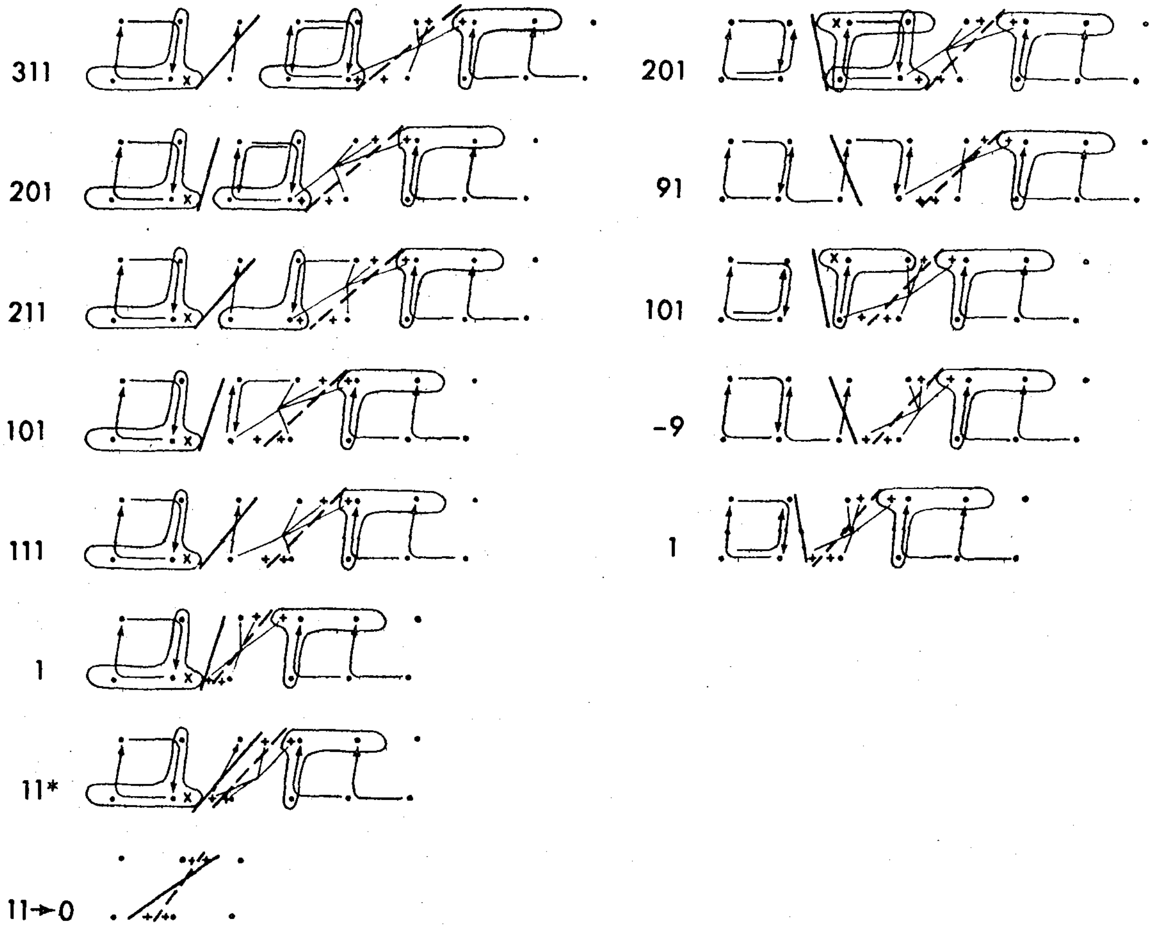
XBL 781-6842

Fig. 27 Cont.



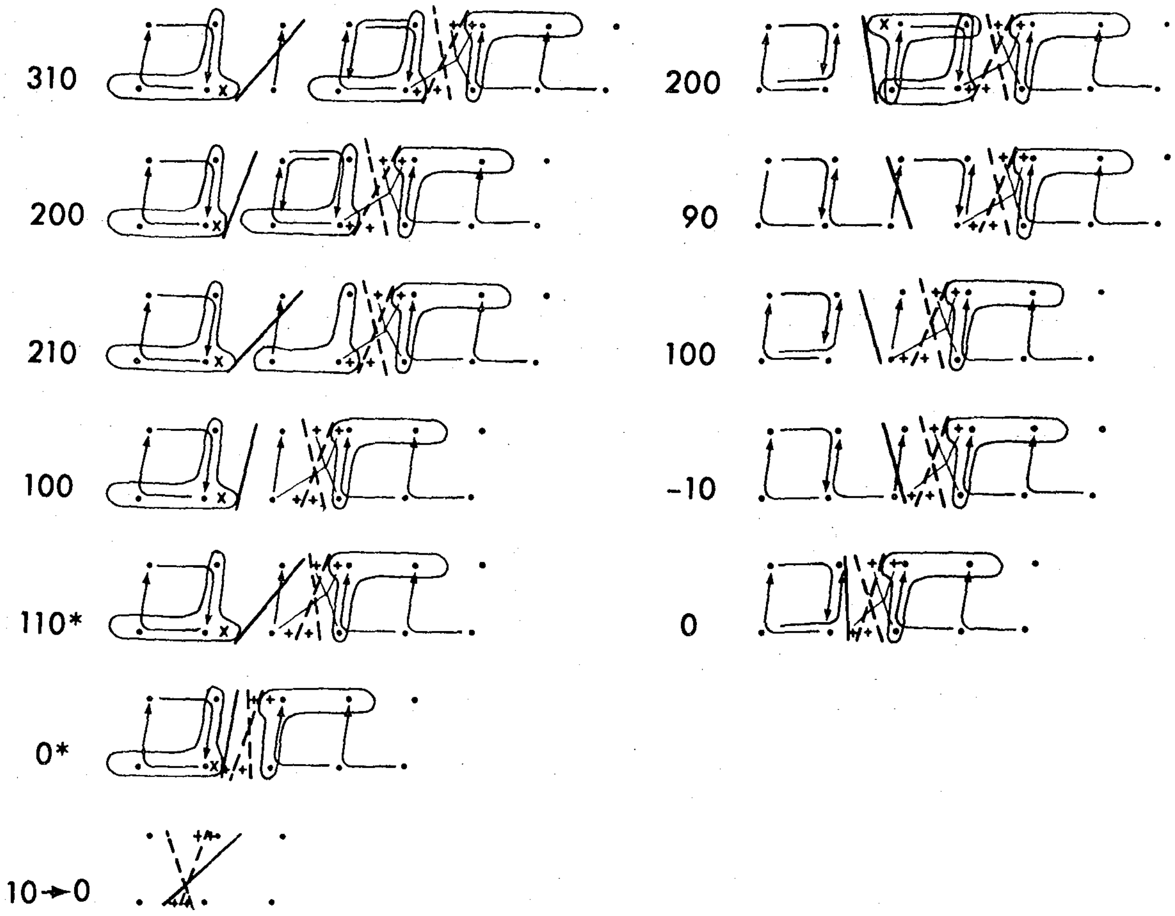
XBL 781-6838

Fig. 27 Cont.



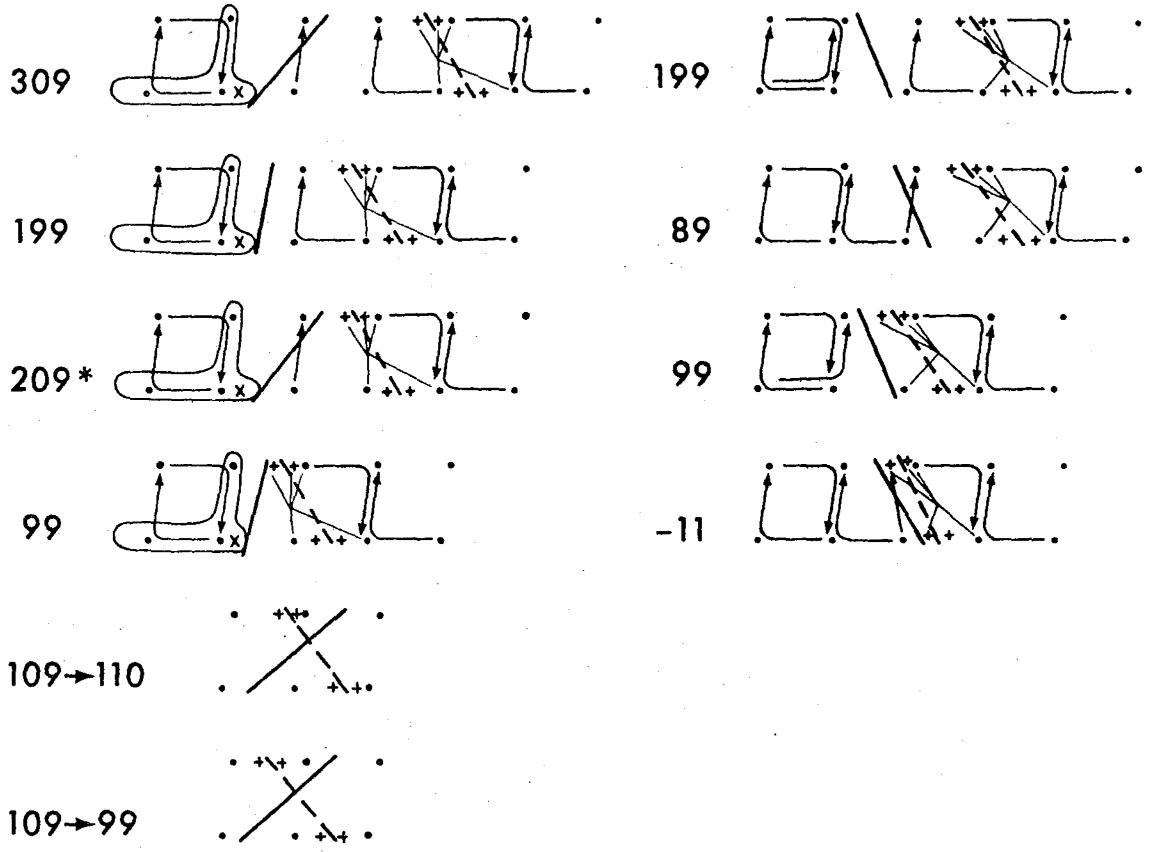
XBL 781-6841

Fig. 28



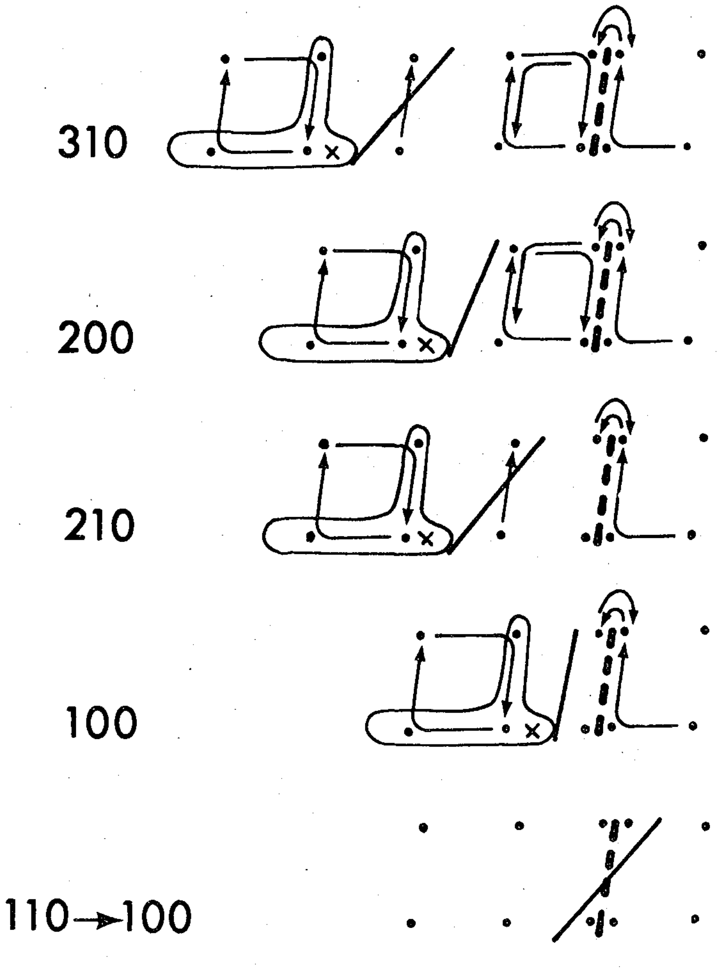
XBL 781-6840

Fig. 28 Cont.



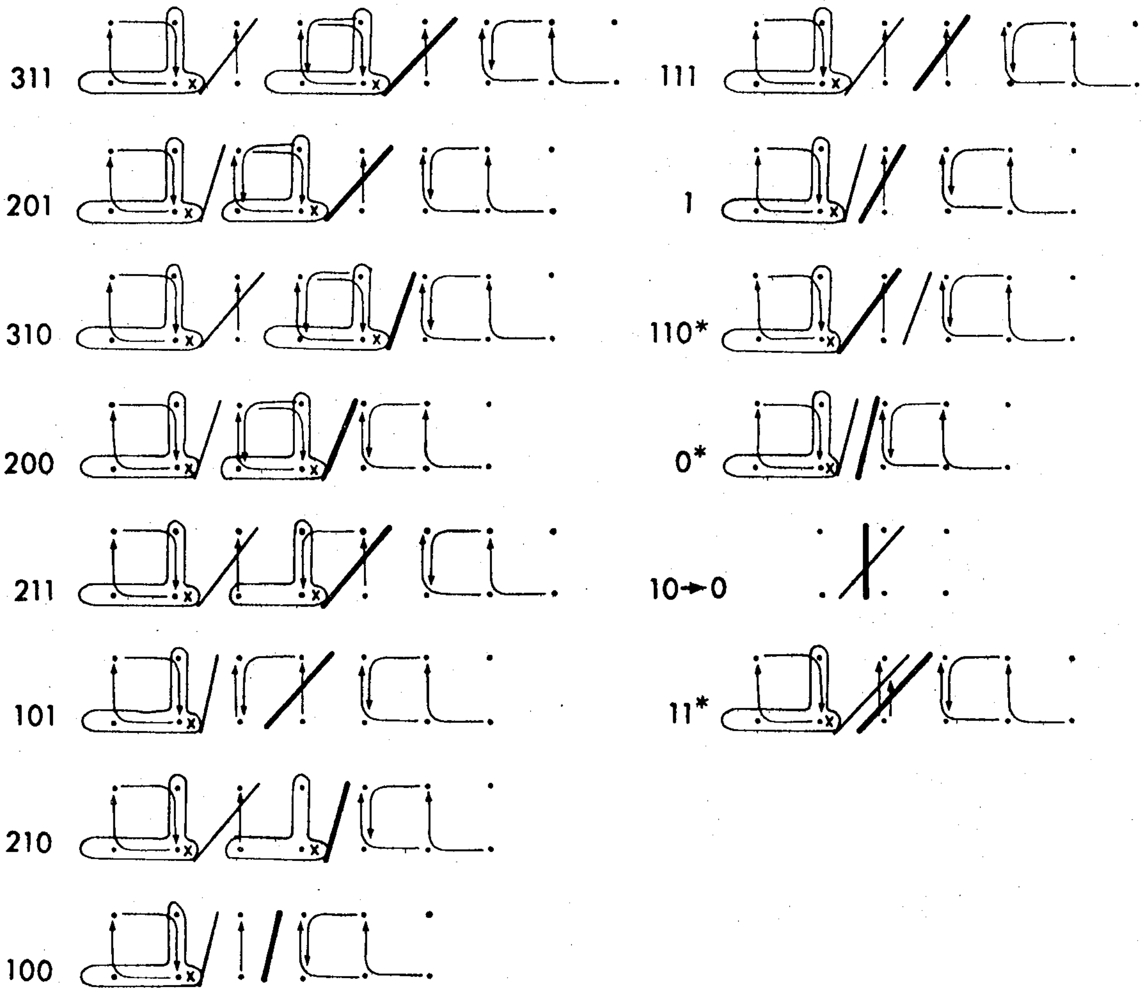
XBL 781-6839

Fig. 28 Cont.



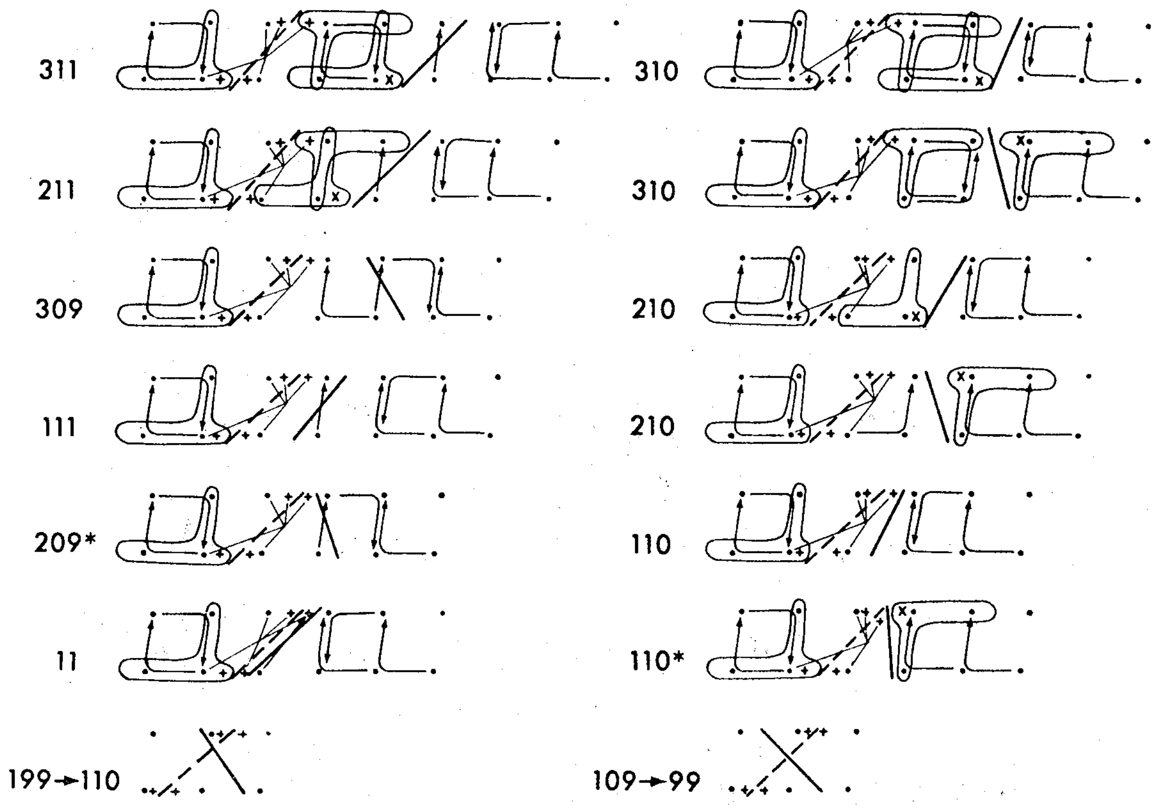
XBL 781-6826

Fig. 29



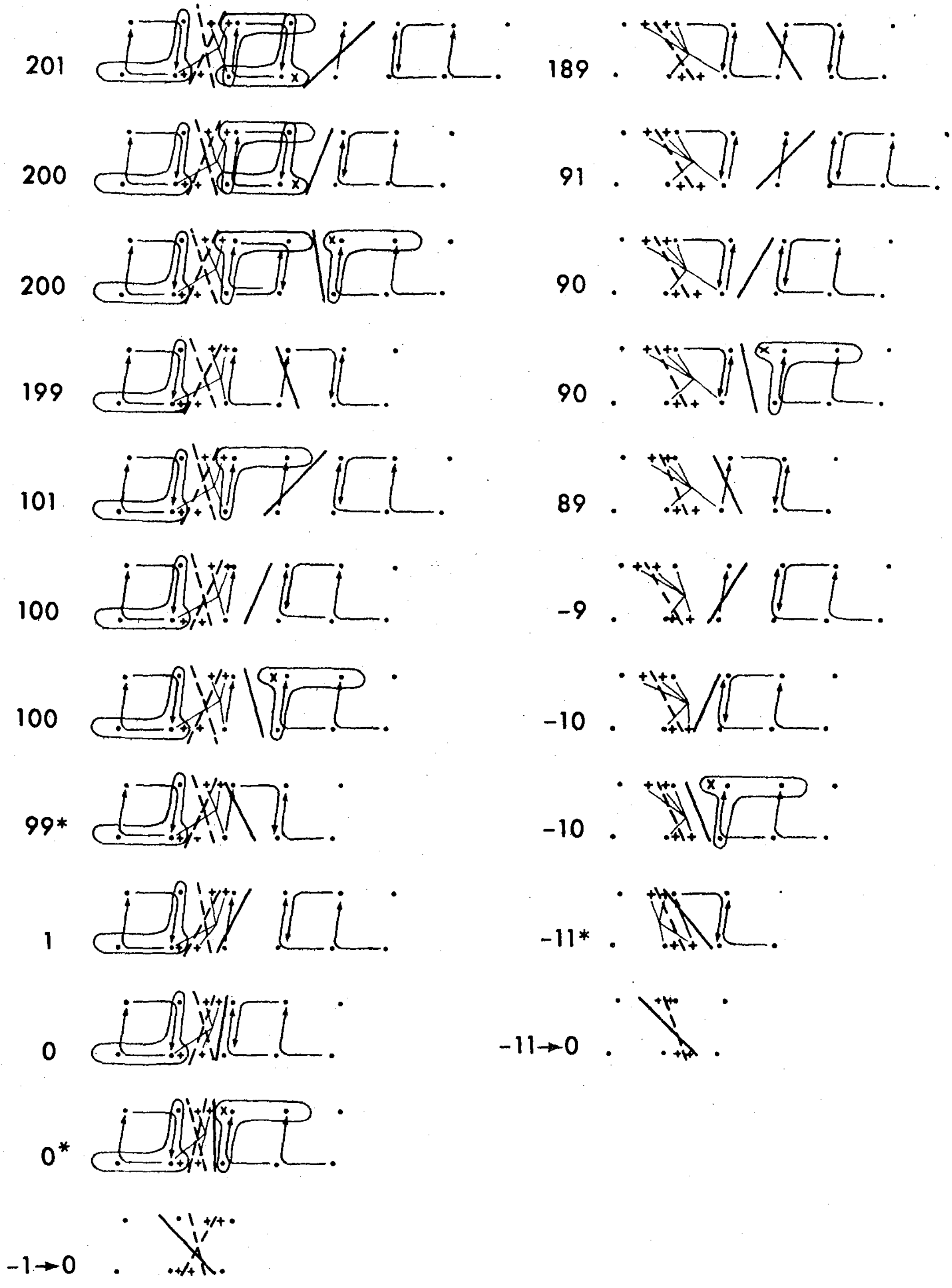
XBL 781-6824

Fig. 30



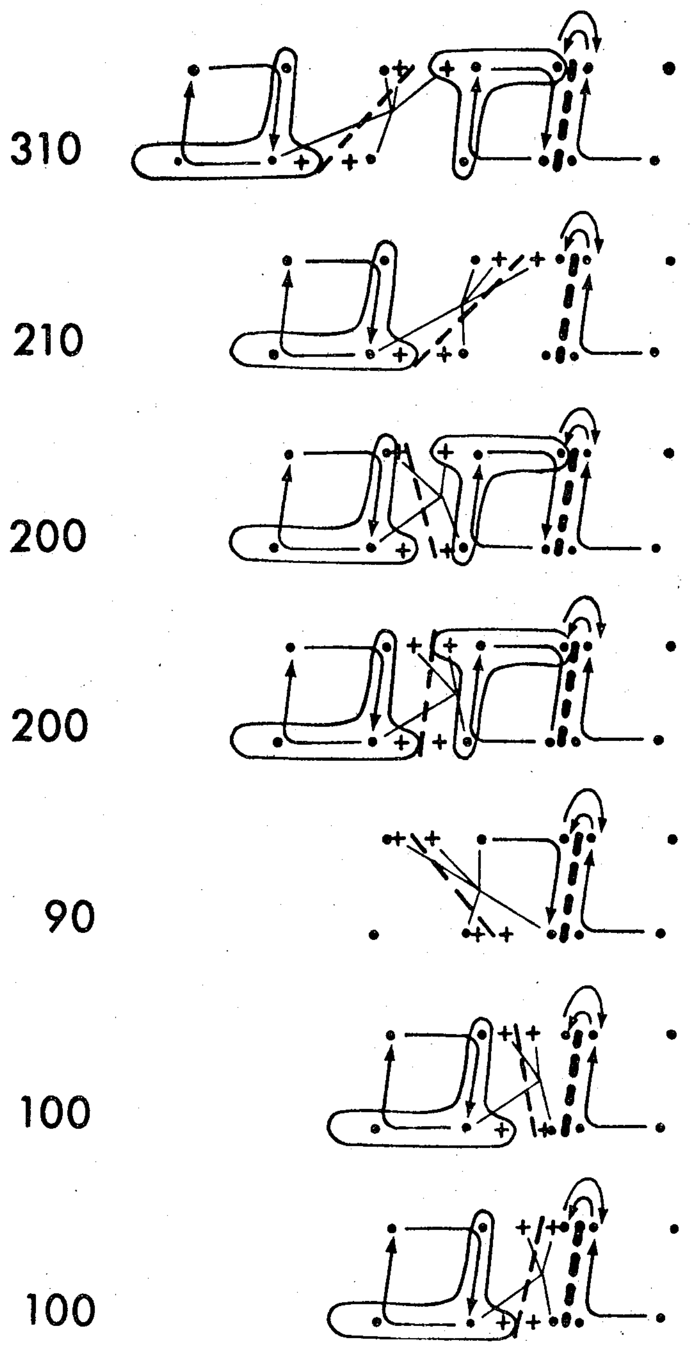
XBL 781-6836

Fig. 31



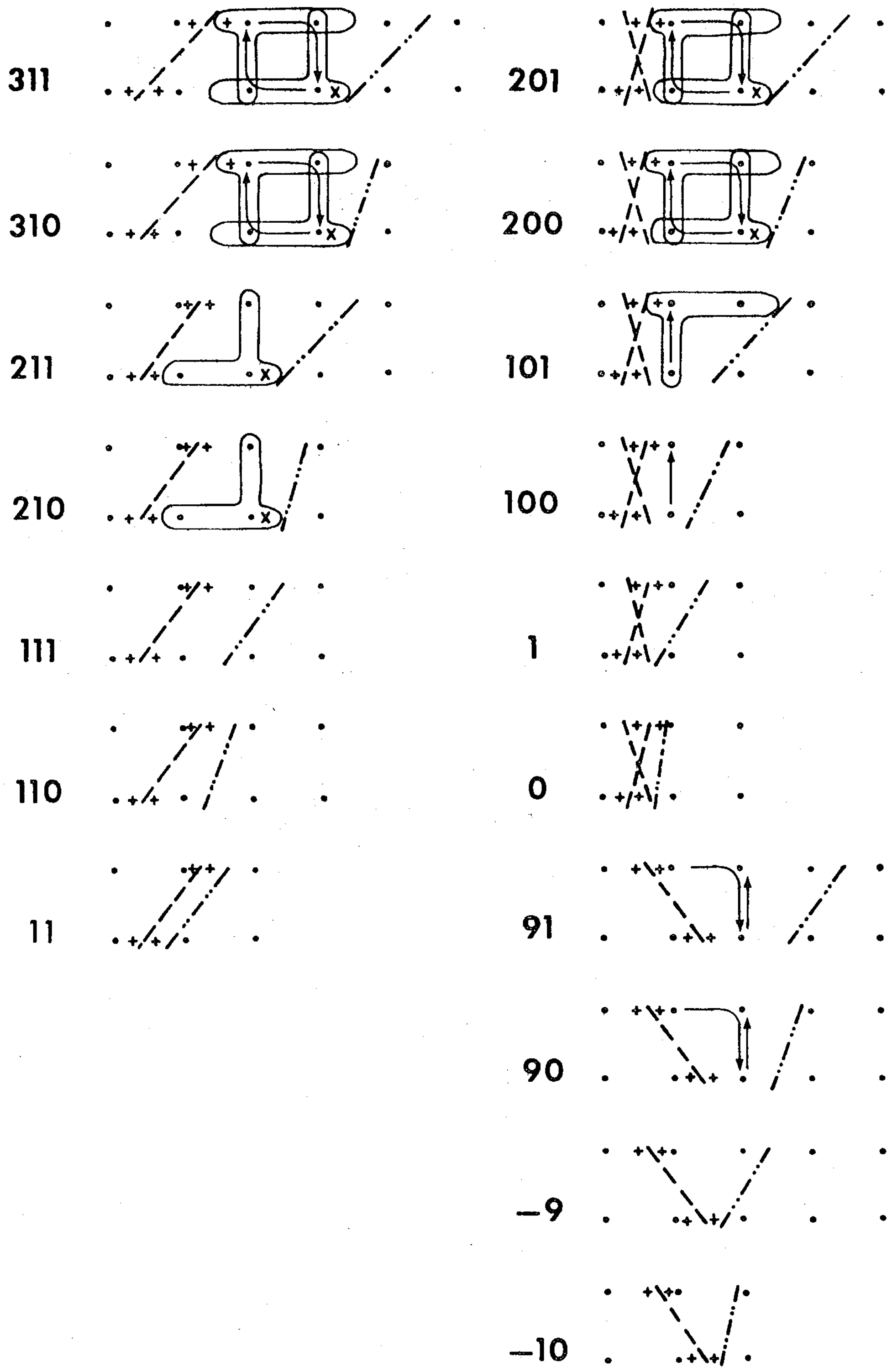
XBL 781-6843

Fig. 31 Cont.



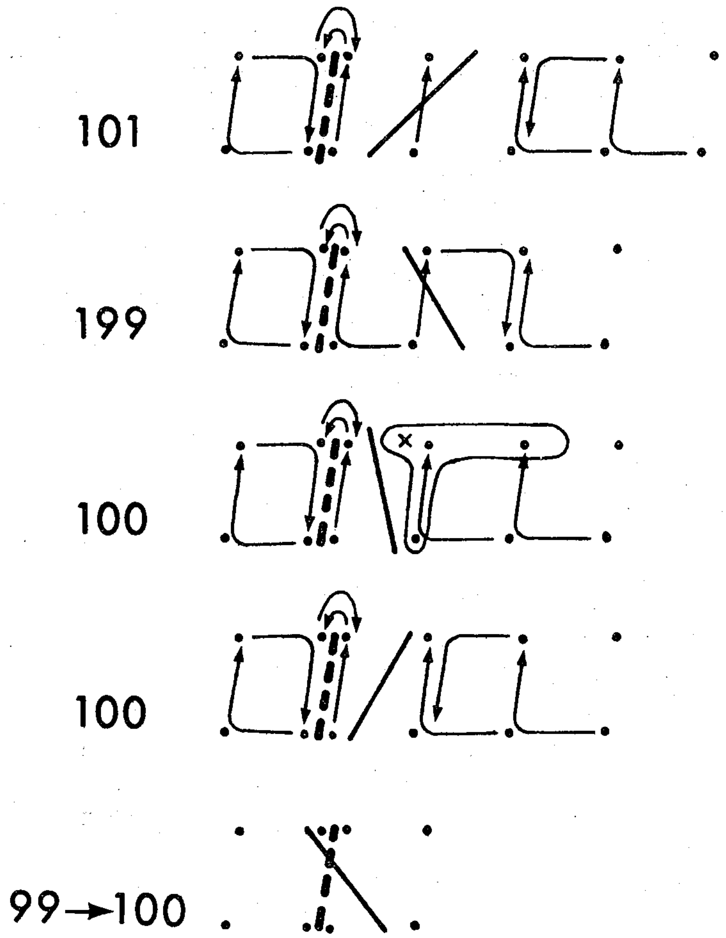
XBL 781-6822

Fig. 32



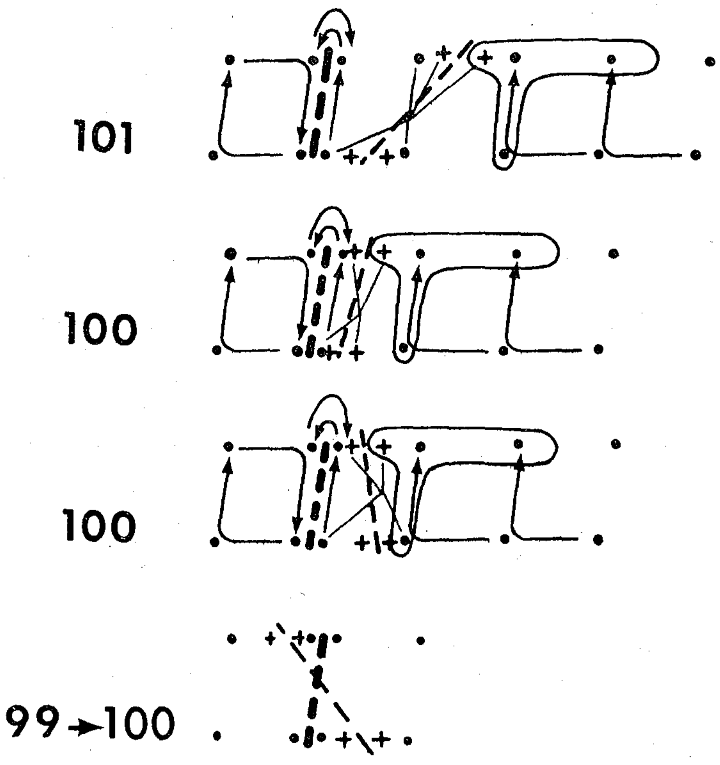
XBL 781-6829

Fig. 33



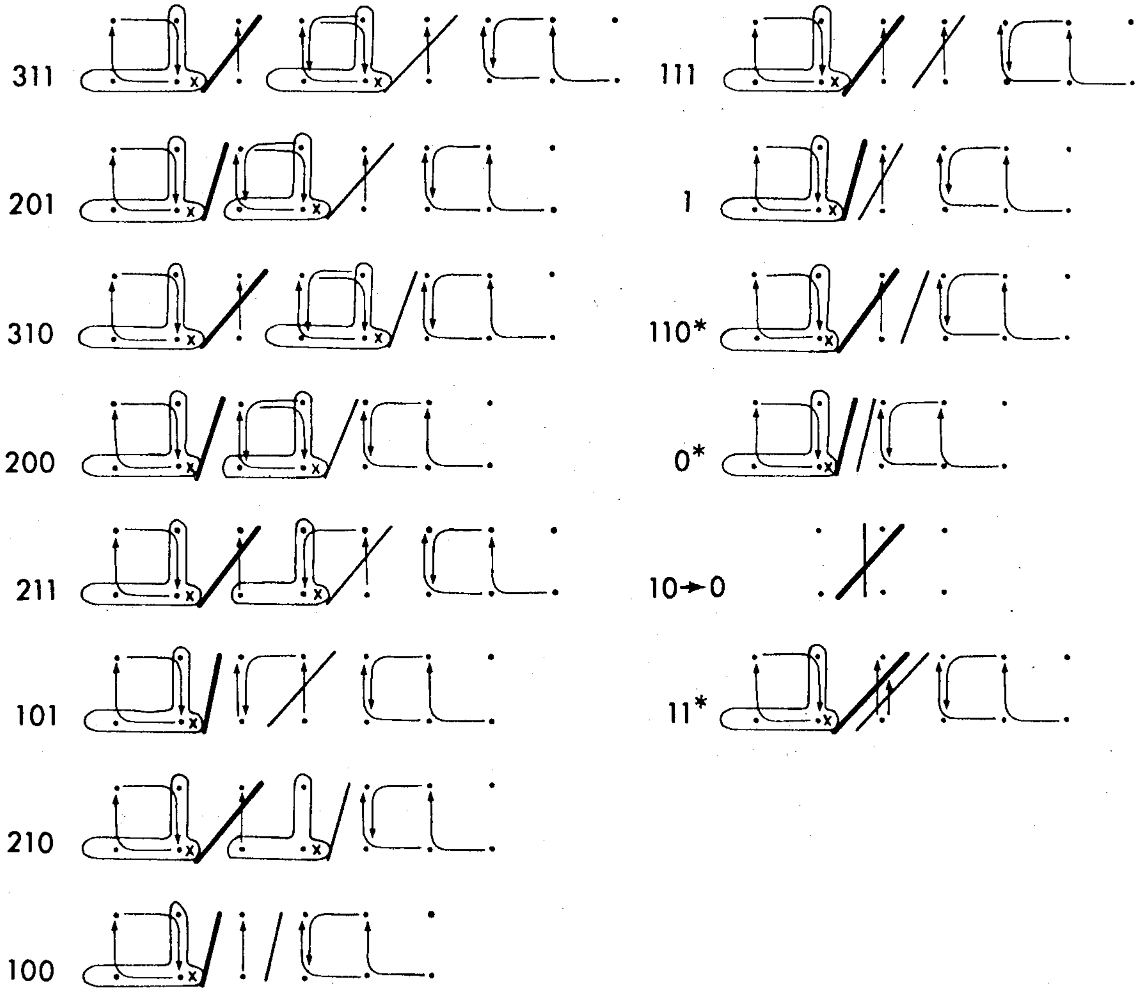
XBL 781-6828

Fig. 34



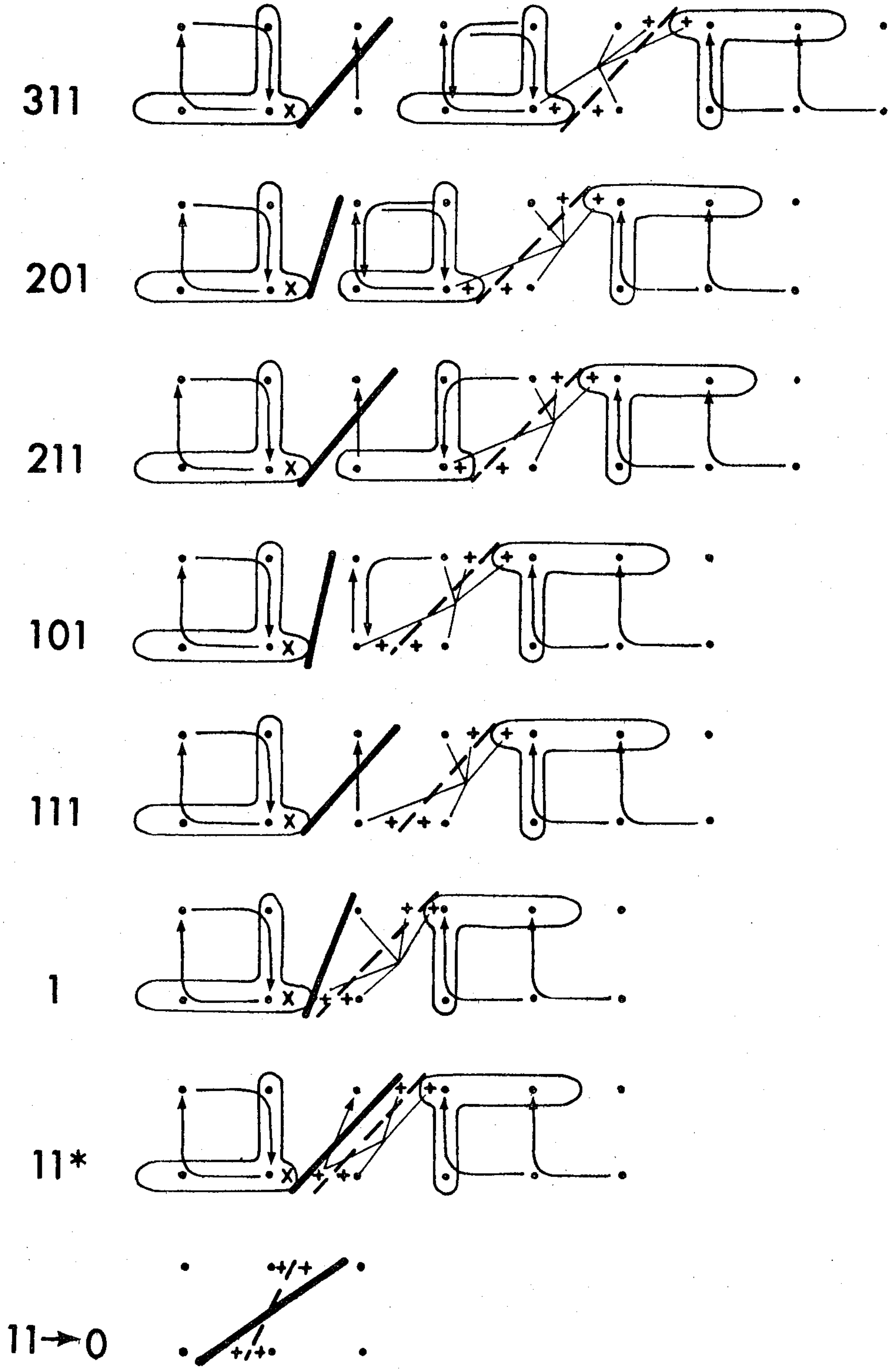
XBL 781-6835

Fig. 35



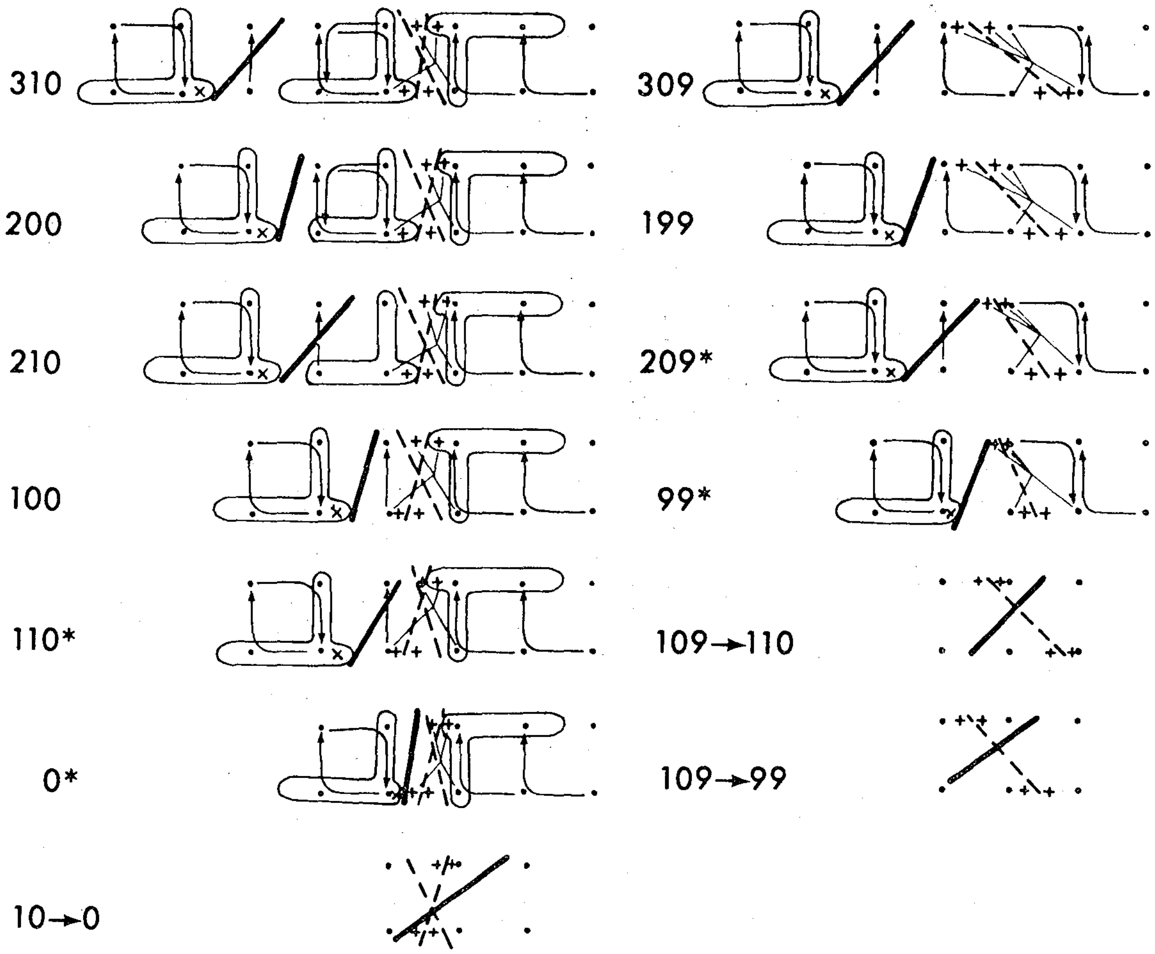
XBL 781-6823

Fig. 36



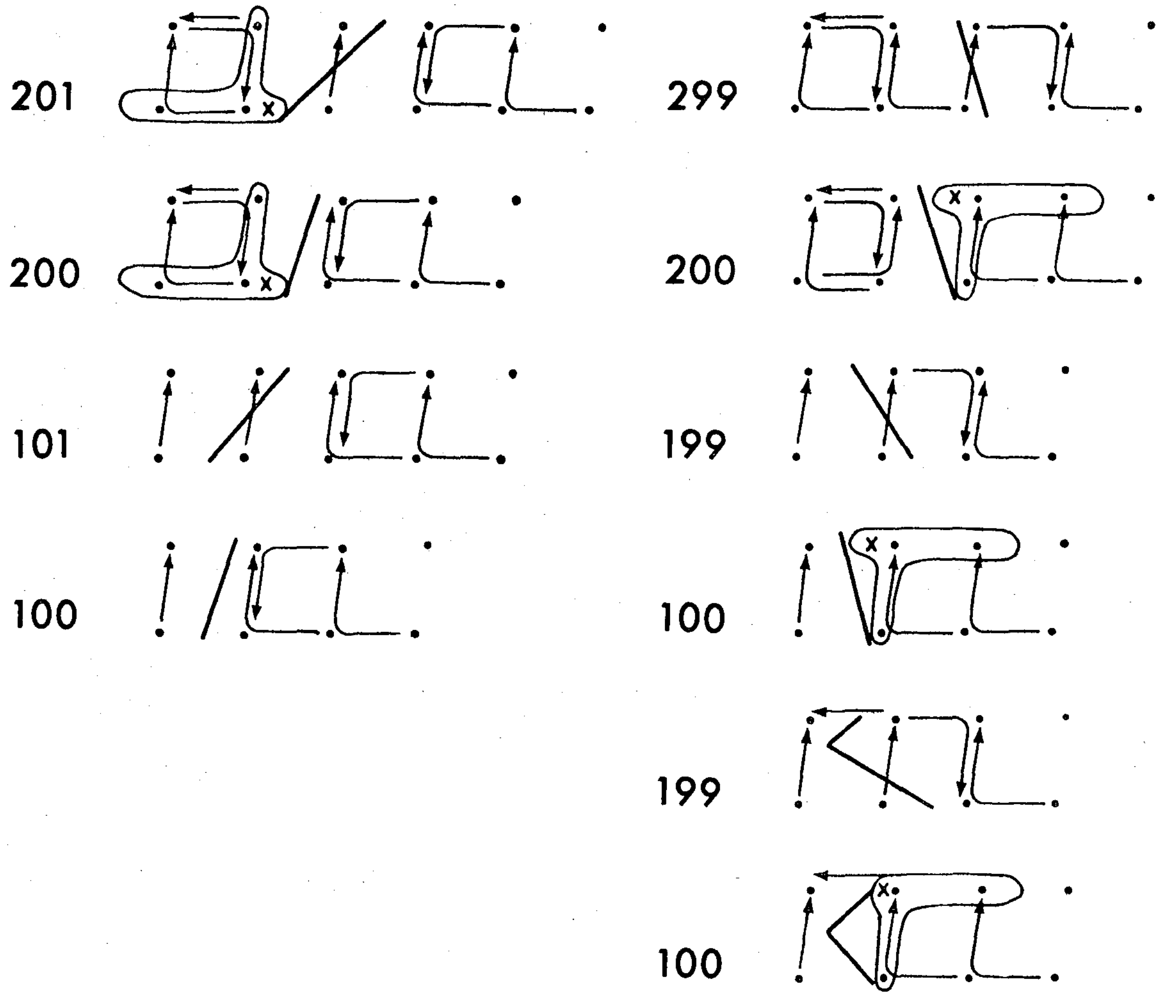
XBL 781-6834

Fig. 37



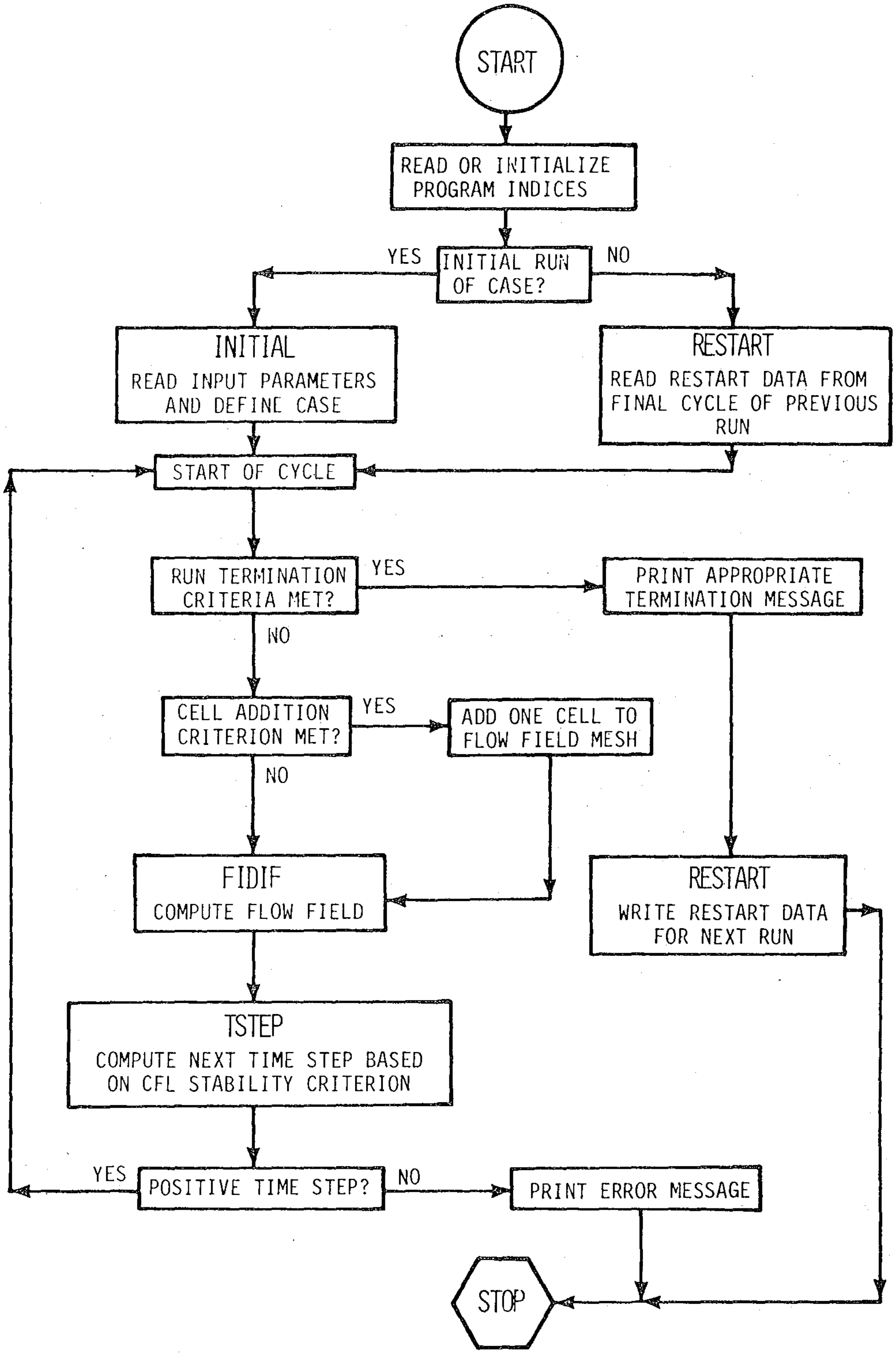
XBL 781-6833

Fig. 37 Cont.



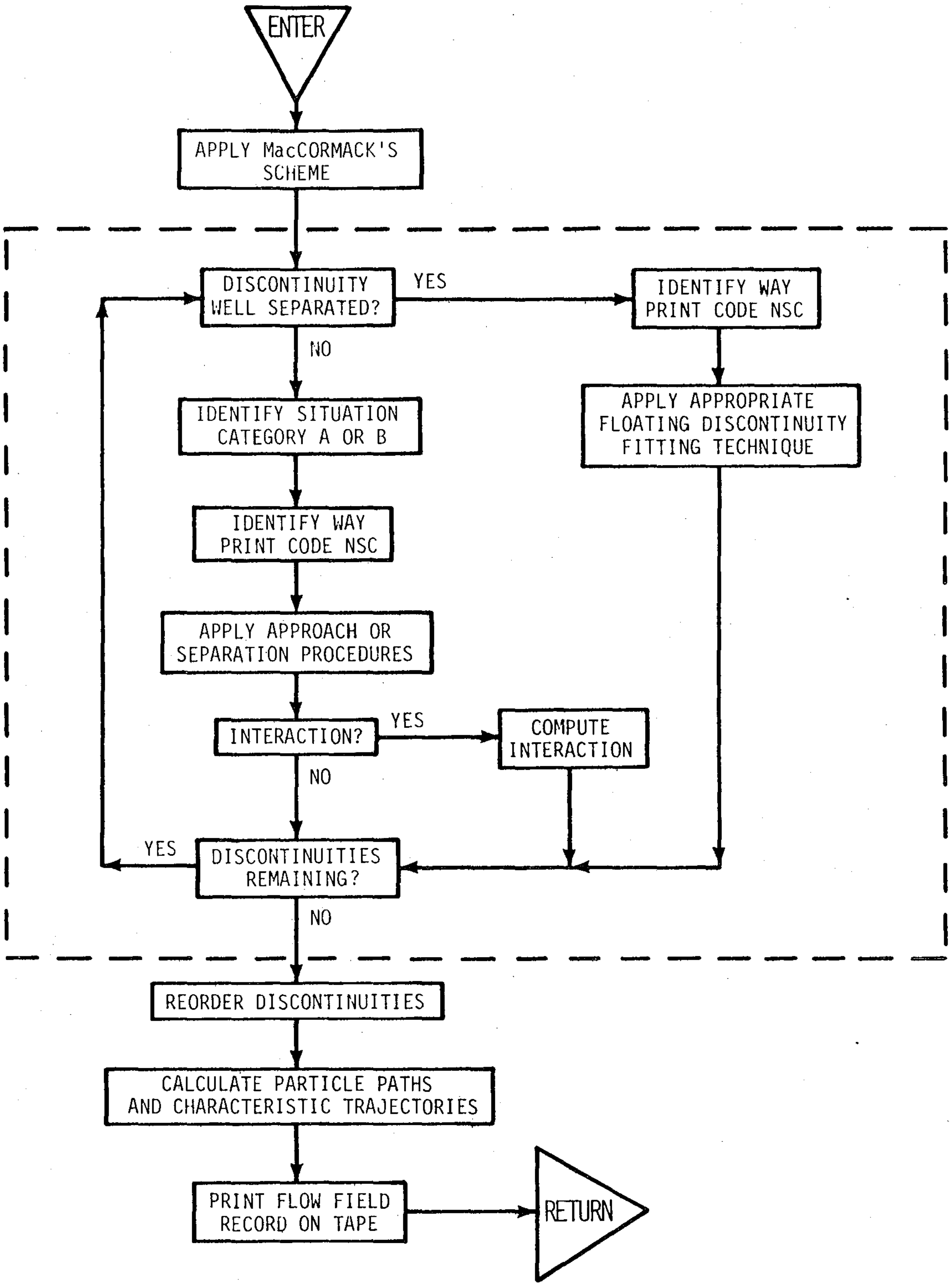
XBL 781-6825

Fig. 38



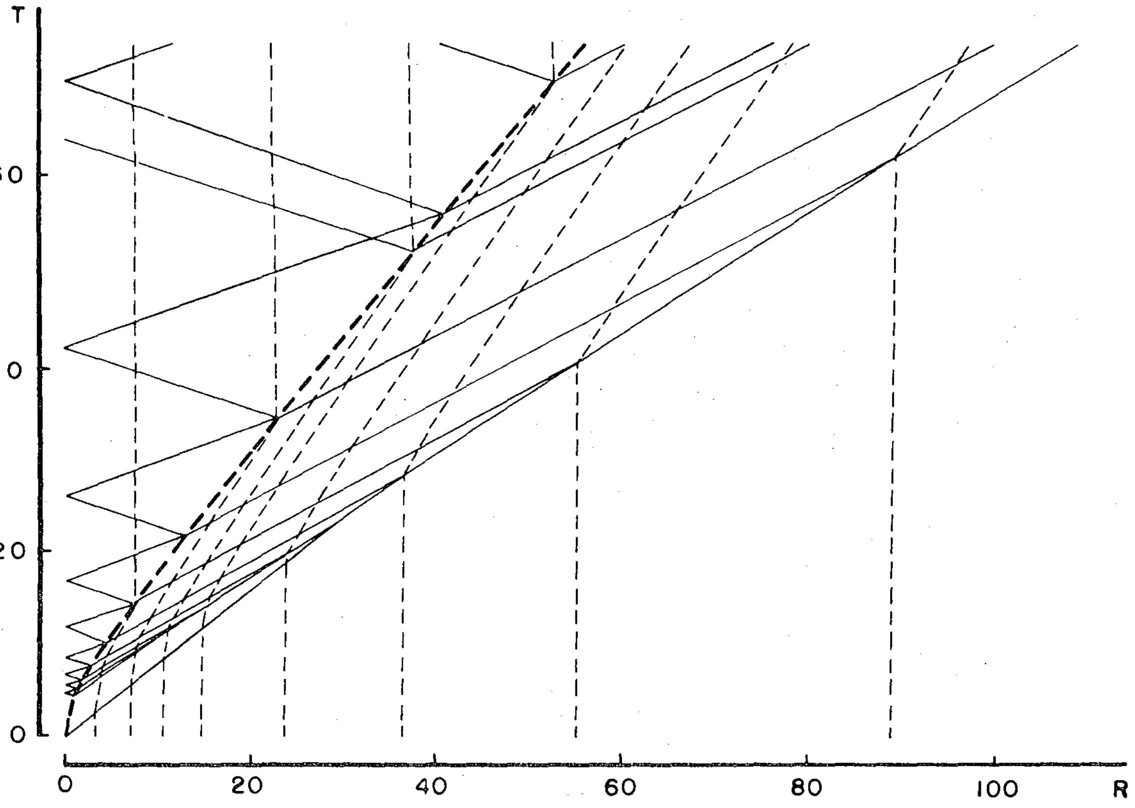
XBL 781-6812

Fig. 39



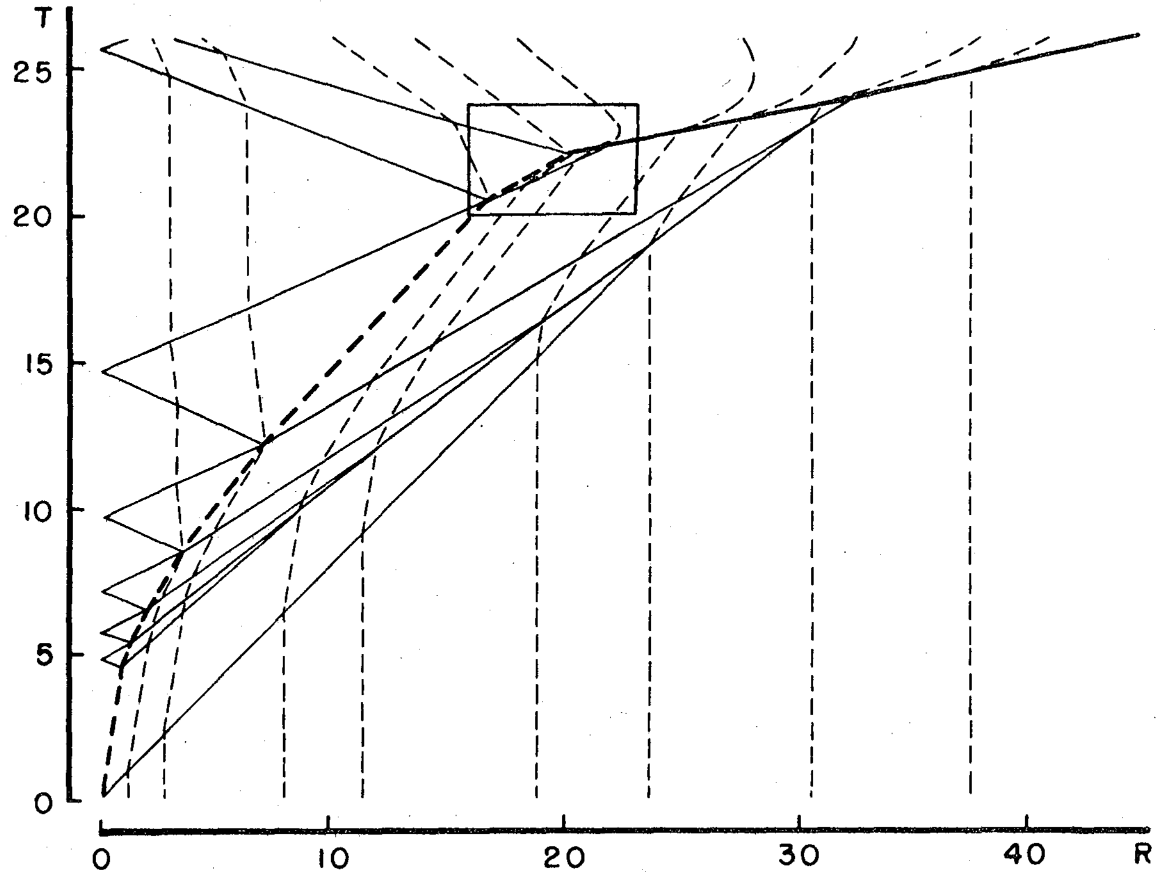
XBL 781-6813

Fig. 40



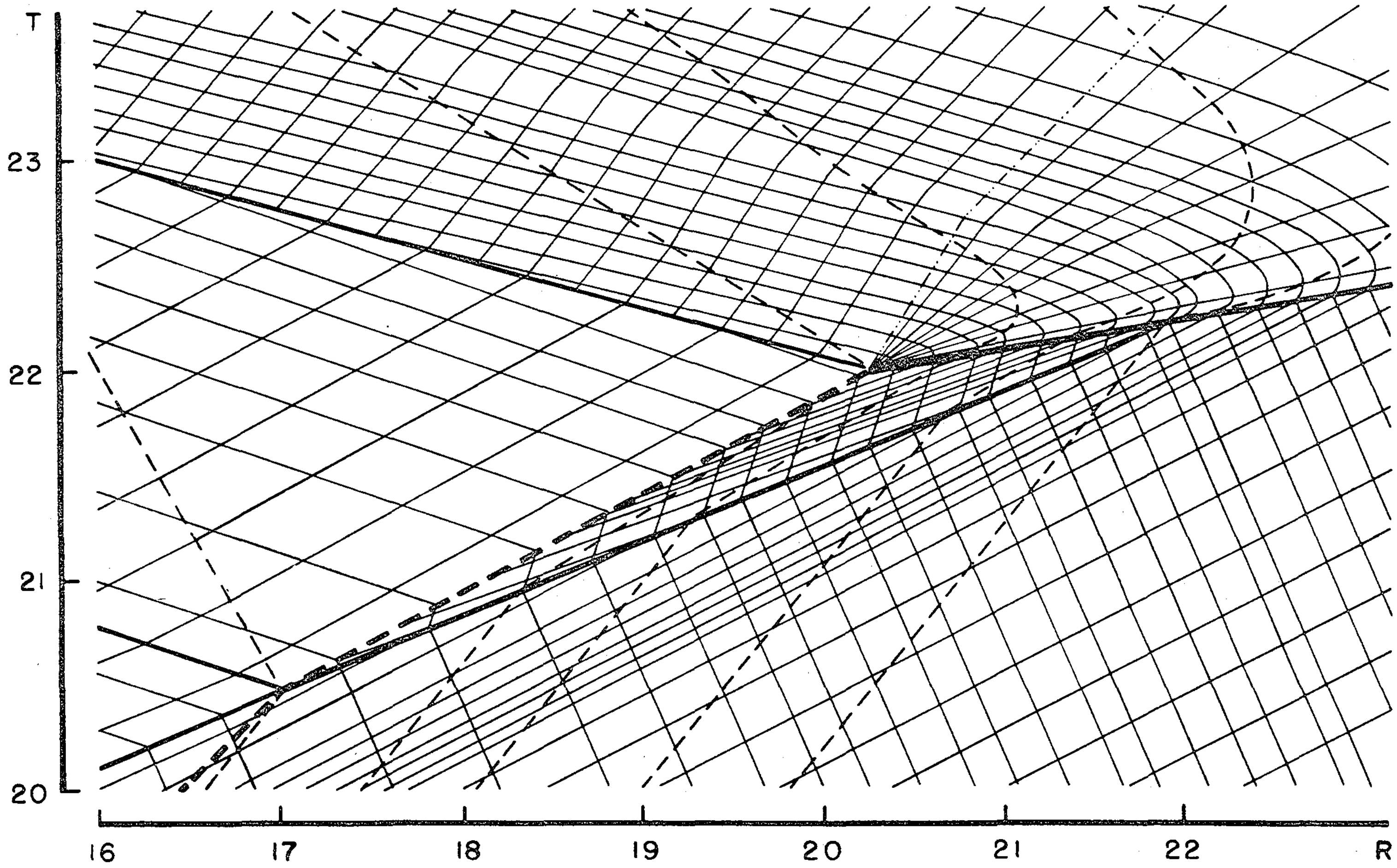
XBL 781-6814

Fig. 41



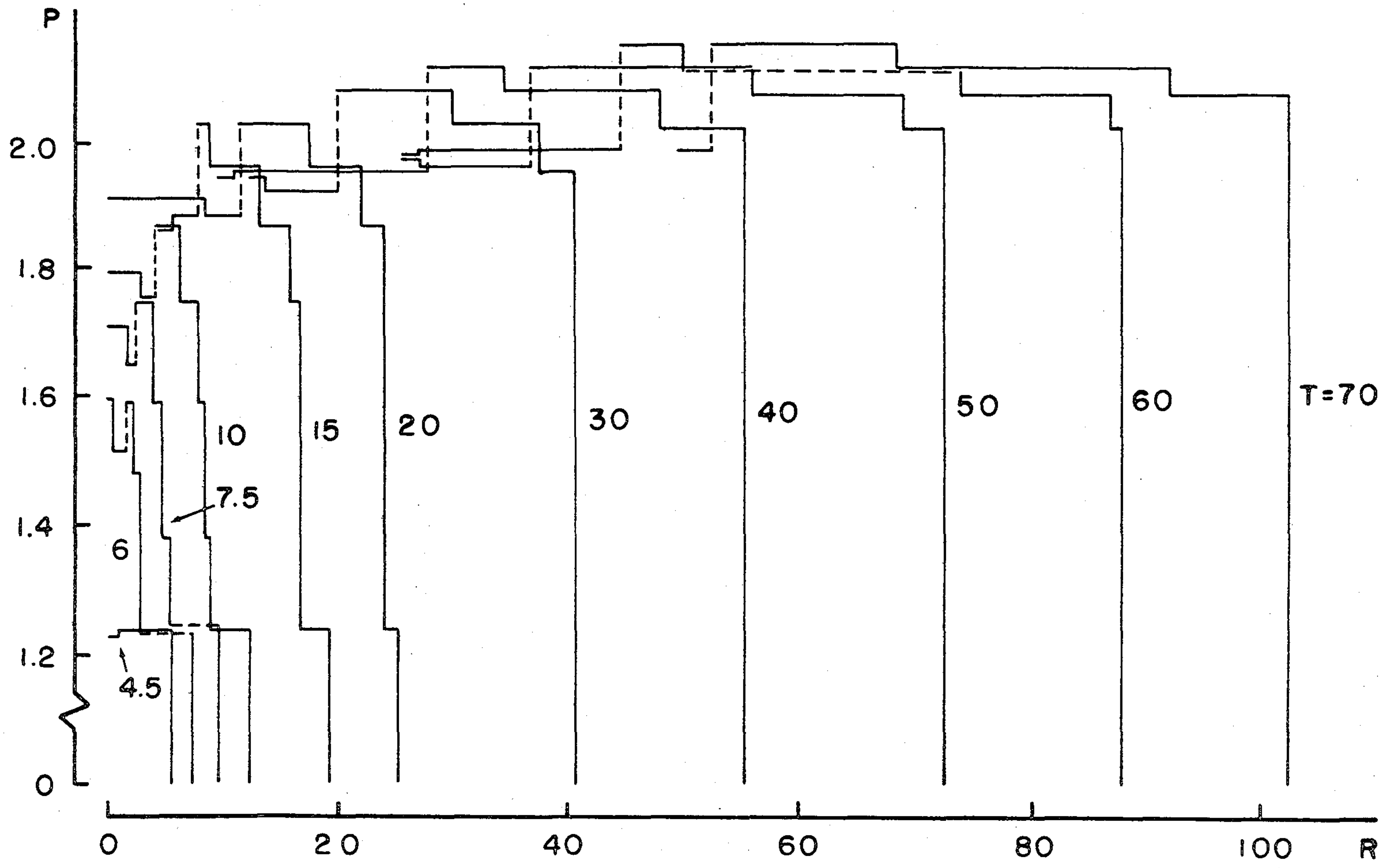
XBL 781-6815

Fig. 42



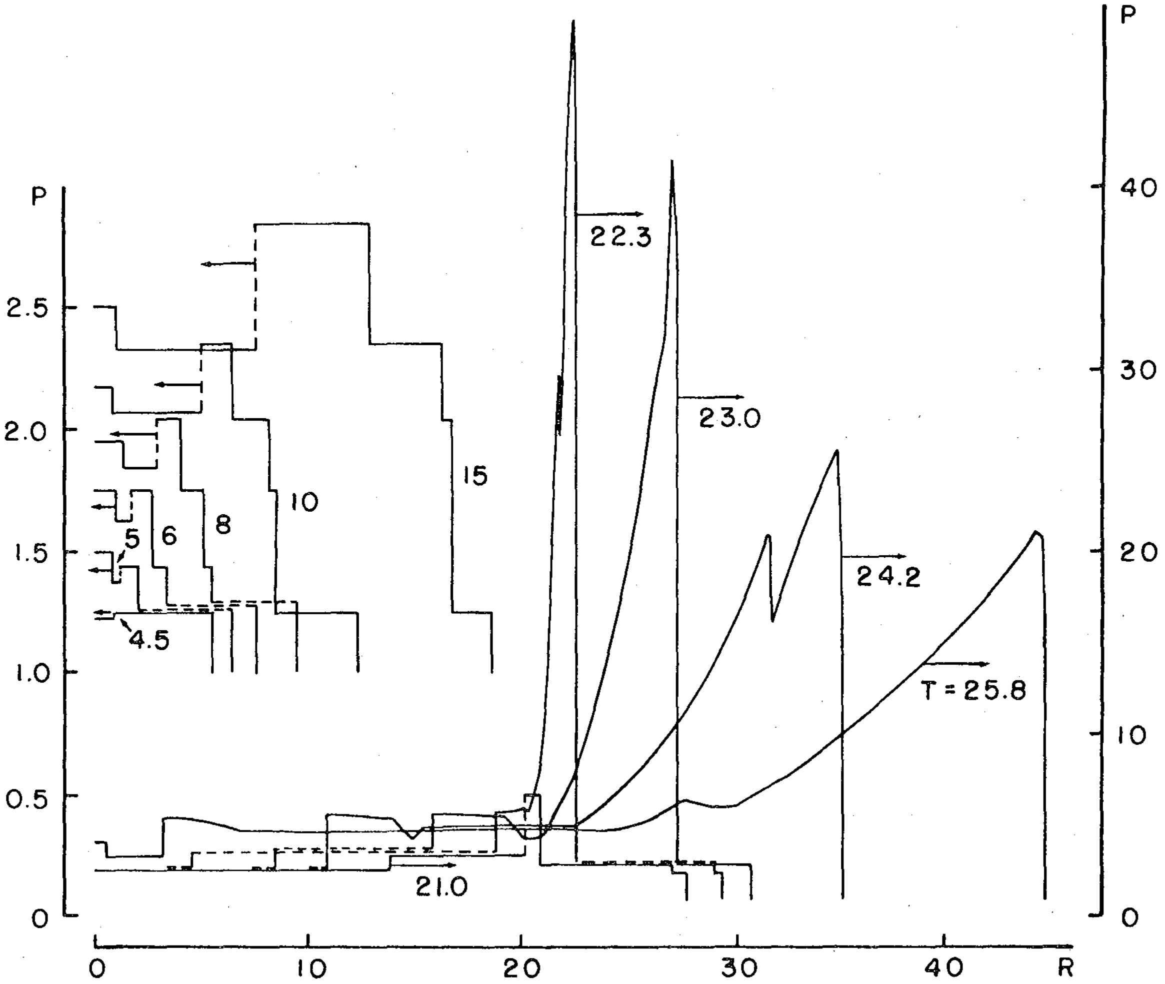
XBL 781-6816

Fig. 43



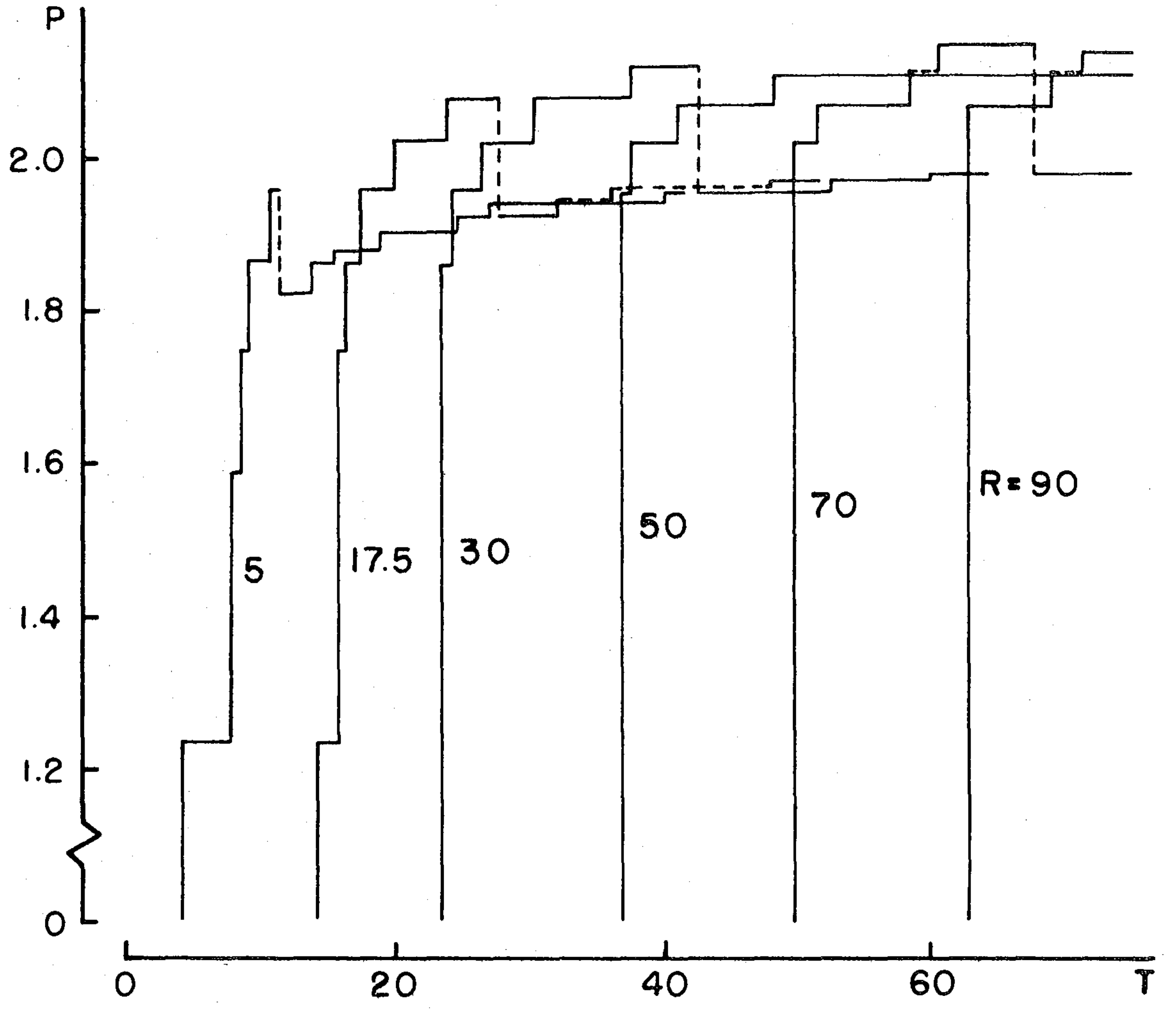
XBL 781-6817

Fig. 44



XBL 781-6818

Fig. 45



XBL 781-6819

Fig. 46

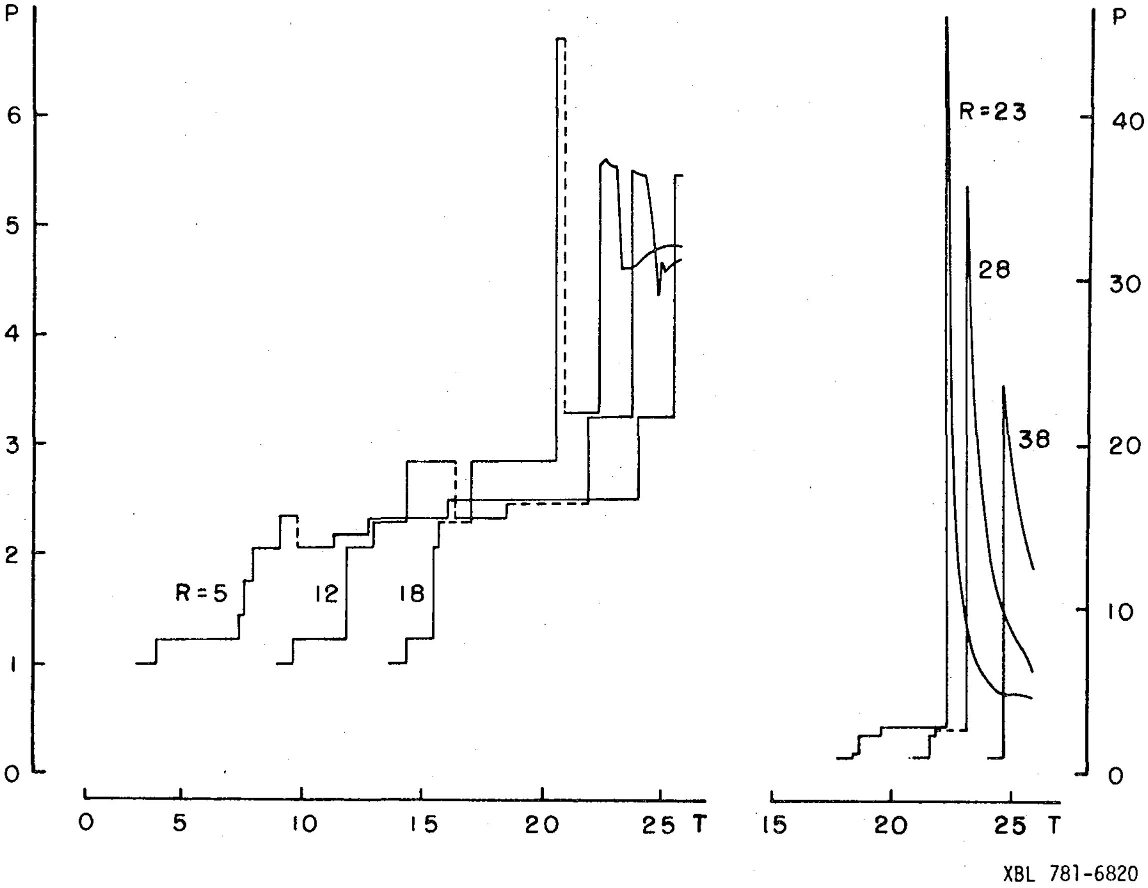
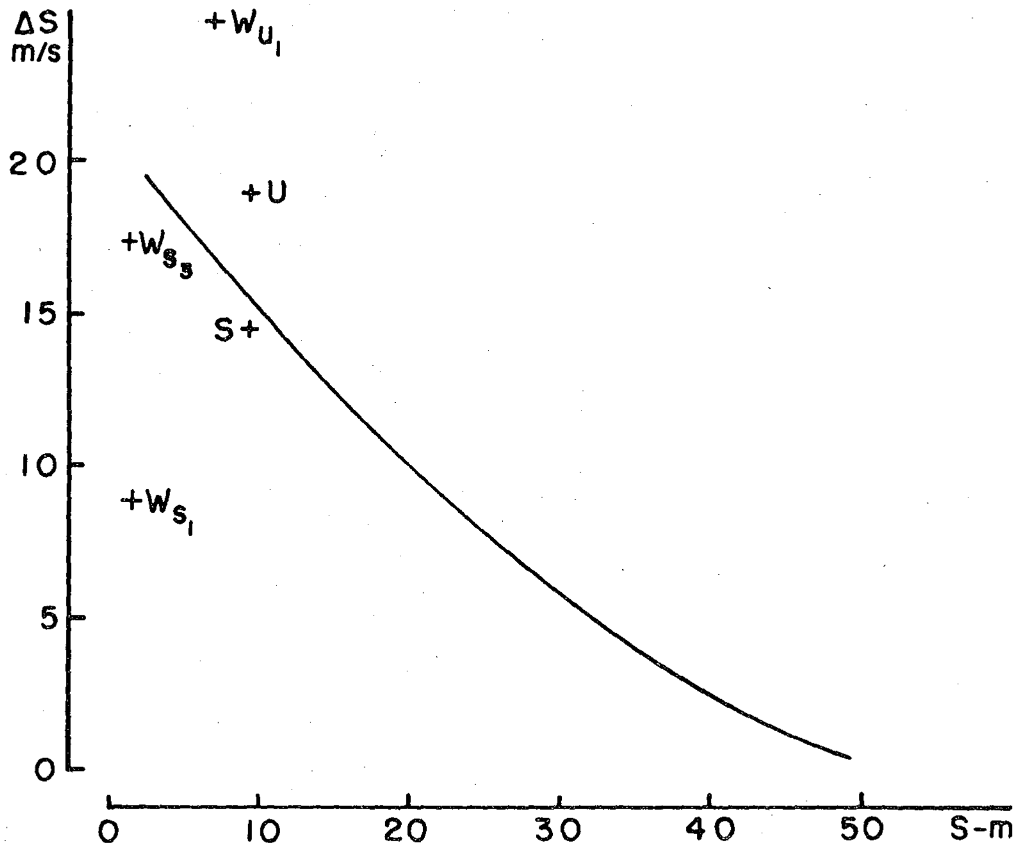

P
0
1
2
3
4
5
6
R = 5
12
18
0
5
10
15
20
25
T
P
0
10
20
30
40
R = 23
28
38
15
20
25
T
XBL 781-6820

Fig. 47



XBL 781-6821

Fig. 48