A FINITE VOLUME METHOD FOR THE NAVIER-STOKES EQUATIONS WITH FINITE RATE CHEMISTRY

by

THOMAS RICHARD ARTHUR BUSSING

S.M., Massachusetts Institute of Technology, Cambridge
(1982)
B.Eng., McGill University, Montreal
(1980)

Submitted to the Department of Aeronautics and Astronautics in Partial Fulfillment of the Requirements for the Degree of

DOCTOR OF SCIENCE IN

COMPUTATIONAL FLUID DYNAMICS

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

August 1985

Signature of Author ______________________
Department of Aeronautics and Astronautics, 21 August 1985

Certified by ______________________
Professor Earll M. Murman — Thesis Supervisor

Certified by ______________________
Professor Judson R. Baron — Thesis Committee

Certified by ______________________
Professor Manuel Martinez-Sanchez — Thesis Committee

Accepted by ______________________
Professor Harold Y. Wachman — Chairman, Departmental Graduate Committee

A FINITE VOLUME METHOD FOR THE NAVIER-STOKES EQUATIONS WITH FINITE RATE CHEMISTRY

by

THOMAS R. A. BUSSING

Submitted to the Department of Aeronautics and Astronautics
on 27 August 1985 in partial fulfillment of the
requirements of the Degree of Doctor of Science in
Computational Fluid Dynamics

Abstract

The time-dependent Navier-Stokes equations, which include the effects of finite rate chemistry, are numerically integrated foward in time to predict the steady state behavior of model scramjet flame holders. Several efficient acceleration techniques are developed for calculating steady state chemically reacting flows. The techniques include preconditioning the conservation equations, a preconditioned multiple-grid accelerator, and a constant CFL condition. One possible choice for the preconditioner leads to a matrix which is equivalent to the one obtained by treating the species source terms implicitly. These methods can be viewed as ways of rescaling the equations in time such that all chemical and convective phenomena evolve on comparable pseudo time scales. If only the steady state is desired the number of iterations needed to solve reacting problems is approximately the same as for non-reacting problems. Only steady state problems are considered in this thesis. The methods are applied to the 2-D Euler equations with H_2-air chemistry and to the 2-D Navier-Stokes equations with H_2-air chemistry.

Two candidate supersonic flame holders are analyzed to assess their application to scramjet engines. The geometries included a ramp and a rearward facing step. With combustion, several different kinds of flow fields are generated depending upon the level of heat release. For each geometry it is suggested that the different flow fields can be summarized on a plot of fuel equivalence ratio vs inlet Mach number or the fuel equivalence ratio vs the ratio of the maximum temperature to the fuel ignition temperature. All flows considered a premixed H_2-air stream and used the global chemistry model of Rogers and Chinitz.

Thesis Supervisor: E.M. Murman

Title: Professor of Aeronautics and Astronautics

Table of Contents

List of Figures

List of Tables

Nomenclature

A	Channel Cross-sectional	
Aw	Atomic Weight	moles/kg
A	Area	m^2
a	Local Speed of Sound	m/sec
c_p	Specific Heat At Constant Pressure	J/kg·K
c_v	Specific Heat At Constant Volume	J/kg·K
C	Species Concentation	mole/cm^3
E	Total Internal Energy	J·kg/K
F	x Component Flux Vector	
G	y Component Flux Vector	
H	Source Term Vector	
H	Total Enthalpy	J/kg
k	Thermal Conductivity	W/m·K
L	Geometric Length	m
Le	Lewis Number	
M	Mach Number	
P	Pressure	N/m^2
Pr	Prandtl Number	
q	Heat Flux	J/s·m^2
R	Residual	
Re	Reynolds Number	
Ru	Universal Gas Constant	1.987cal/mole·K
S	Scaling Matrix, Cell Surface Area	m^2
Sc	Schmidt Number	
T	Temperature	K
T_{ig}	Fuel Ignition Temperature	K
T_{max}	Maximum Temperature	K
t	Time	s
U	State Quantities	
u	Velocity Component	m/sec
v	Velocity Component	m/sec
w	Reaction Rate	kg/s·m^3
x	Spatial Coordinate	m
y	Spatial Coordinate	m
Y_i	i^{th} Component Species	

	Density Fraction	

Greek

α	Numerical Time Step Constants	
ϕ	Fuel Equivalence Ratio	
μ	Viscosity	$N\cdot/m^2$
ρ	Density	kg/m^3
σ	Artifical Viscosity Coefficient	
τ	Characteristic Time	s
Γ	CFL Number	
Σ	$= T_{max}/T_{ig}$	

Acknowledgements

I would like to thank Professor Murman for his guidance and helpful discussions. I would also like to thank Professor Martinez-Sanchez and Professor Baron for their support. I would like to extend special thanks to Dr. Beach and Mr. Anderson of NASA Langley, for their financial support and the time they spent teaching me the hows and whys of scramjets.

Most importantly, I would like to thank my wife Julie Anne for her constant support. Without her this thesis would not have been possible.

Chapter 1
INTRODUCTION

Since the early 1950's the superiority of supersonic combustion ramjets (scramjets) over other more conventional types of engines (subsonic-combustion ramjets, rockets, turbojets, etc.) for hypersonic flight within the atmosphere has been recognized. At this time, no United States nor foreign-developed scramjet has been flight tested. However, it is now well established that the scramjet has unique performance advantages over all other air breathing engine cycles at speeds of Mach 5 and above [24]. In fact above Mach 6 the scramjet is the only engine cycle which offers adequate thrust and specific impulse for efficient propulsion. Rocket propulsion is limited by its much lower propulsive efficiency (as it must carry its own oxidizer) and is limited to research and nonreusable vehicles.

A plot [1], figure 1-1, of Mach number vs. specific impulse can provide useful information about the type of propulsion system which should be considered for a particular mission. For example both aerodynamic heating and turbojet engine performance limit today's operational aircraft to speeds of about Mach 3. From Mach 0 up to Mach 3 turbojets, fanjets and propjets produce more thrust per unit mass of fuel burned than other engines due to inherently higher thermal efficiency. Above Mach 3 the turbine-inlet temperature reaches its maximum allowable level for structural integrity. To operate at higher Mach numbers, the combustor must be operated very lean or special turbine cooling must be employed. In either case, the cycle efficiency drops rapidly with increase in speed. In addition at Mach numbers greater than 3 the ram-air effect can provide the needed compression thus making the turbojet compressor a liability. The ramjet thus becomes more efficient than the turbojet.

Within the Mach 3.5 - 5 range the ramjet proves more efficient in terms of specific impulse (Isp) than the scramjet. The ramjet has two major loss mechanisms: a total pressure loss through the inlet normal shock and a total pressure loss associated with combustion. A scramjet on the other hand suffers primarily only from the latter of these two loss mechanisms. For flight within the Mach 3.5 - 5 range the losses in the ramjet are less then those in the scramjet. However, the normal shock losses in

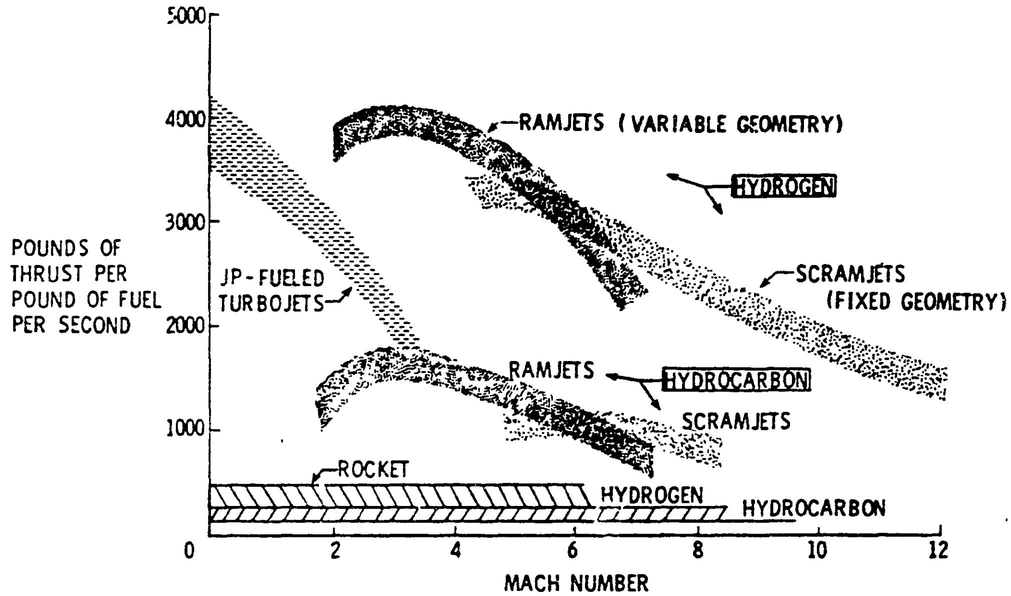


Figure 1-1: Mach Number vs ISP Plot

the ramjet increases with flight speed, and at Mach 5 - 6 the two cycles have comparable losses. Above Mach 6 the scramjet, in addition to the loss considerations, clearly surpasses the ramjet for two additional reasons. First, the low static temperature in the combustor leads to a lower level of dissociation of the reaction products, which in the case of ramjets can represent a significant amount of energy unavailable for thrust. Secondly, the lower static temperature and pressure in the combustor produces less heat transfer and lower structural loads which leads to an engine weight reduction. In conclusion, normal shock total pressure losses, reaction product dissociation, and engine heat load problems, significantly degrade ramjet performance for Mach numbers above 6.

Another interesting feature of a fixed geometry scramjet is its high efficiency at off-design conditions compared to a fixed geometry ramjet. For example, below the design Mach number, where it can operate in the subsonic/supersonic combustion or dual combustion mode, the fixed-geometry scramjet can have a higher airflow than a fixed-geometry ramjet, and can thereby develop higher thrust. This is possible because the scramjet has a larger inlet throat. Above design speed the ramjet experiences strong internal shock losses which greatly reduce its performance. Thus a given scramjet is capable of efficient operation over a larger range of Mach numbers than a given ramjet.

Ferri [18] in his 1964 scramjet survey paper, shows where these results with realistic thrust estimates fit onto an altitude vs. Mach number plot , figure 1-2.

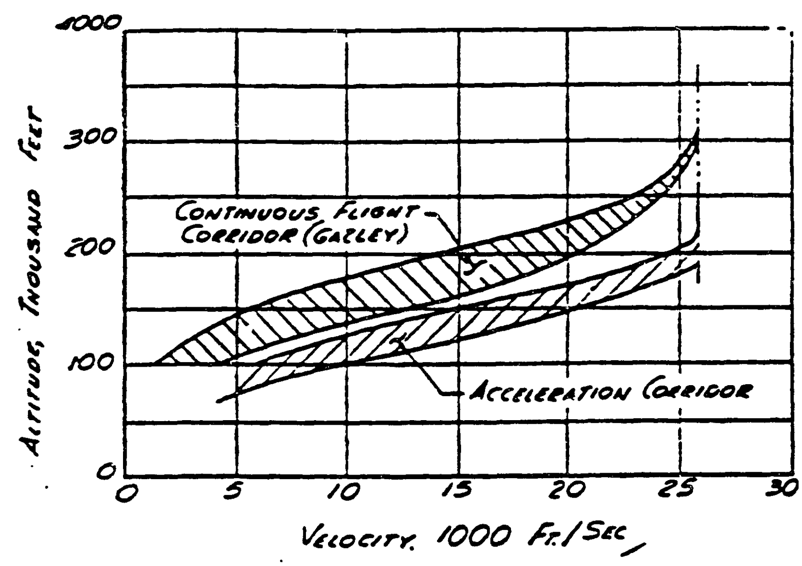


Figure 1-2: Altitude vs Mach Number Plot

Note it is not practical to operate a scramjet at a Mach number of 20 because the inlet contraction ratio would have to be greater than 1000 to 1. One finds that the inlets for such a vehicle become unwieldy. In addition the boundary layer entering the combustor would fill most of the inlet throat. Both effects have made axisymmetric inlets impractical for Mach numbers greater than 10. However, considerable gains can be made to counter these problems by going to the airframe integrated concept [1].

From this discussion, the scramjet has many promising applications in the Mach range 4 and above. Some examples are given below [24]:

Civilian

- Hypersonic, hydrogen-fueled, air-breathing transport for very long range.
- Aircraft-type launch vehicle first stages for future space-shuttle systems.
- Single-stage-to-orbit vehicles.

Military

- Advanced reconnaissance aircraft.
- Acceleration/interceptor aircraft.
- Strategic cruise aircraft.
- Strategic cruise missiles.
- Highly maneuverable interceptor missiles.

For many of the above applications the scramjet would have to be accelerated to scramjet take-over speed by a rocket or aircraft launch vehicle. Other concepts require a companion engine, or hybrid engine to achieve scramjet take-over speed.

The results presented so far show the unequivocal advantages to be had with scramjet propulsion for Mach 4 - 6 and above. However, many fundamental and technical questions need be answered before the scramjet can become a reality. In the following section of this chapter some of the difficulties currently experienced in modeling scramjet components will be discussed.

1.1. Component Modeling

Figure 1-3 shows a schematic of a typical scramjet engine. It consists of four major components: an inlet, a fuel injector/flame holder, a combustor, and a nozzle. In addition some scramjet concepts (airframe integrated [1]) also employ vehicle fore-body inlet precompression and vehicle aft-body nozzle post-expansion. In order to describe what happens to a fluid particle moving through each of these components the following physical disciplines are needed: fluid dynamics and chemistry. Fortunately, through most of the engine, the relative time and length scales of the physics are

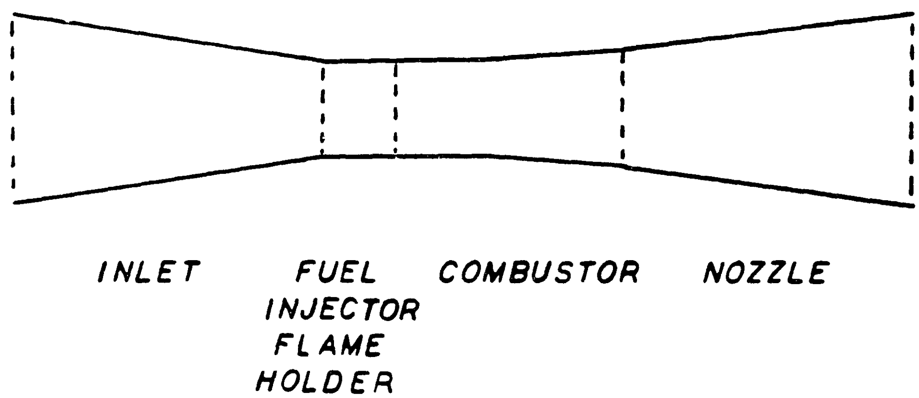


Figure 1-3: Scramjet Engine Component Elements

sufficiently different that part or all of the equations modeling the process can be decoupled. However if some of the physical time and length scales are similar (possible coupling of physical processes), then current theories and methods tend to break down. If the equations are coupled then they all need to be solved simultaneously. If the equations are uncoupled then they can be broken up and each part solved separately which can often lead to substantial work savings. With the idea that time and length scales are useful in assessing the important physics, a critical assessment of the potential modeling difficulties of a scramjet engine can be made.

Inlet Region

From a physics view point the inlet is simple. First, the fluid dynamic length scales of viscous diffusion and convection are such (Re = 10^7) as to make the flow inviscid except near the walls. In addition, the ratio of a typical fluid dynamic time scale to the chemical time scale for the dissociation of air is much less than 1. That is to say that the Damkohler number, $D = \tau_{fl}/\tau_{chem}$, is much less than 1 implying frozen chemistry. Thus to solve for the inlet flow field the Euler equations can be used away from the wall, and the boundary layer equations near the wall, both with chemistry frozen. There are however at least two important difficulties. First, the flow field is highly three dimensional with many interacting oblique shocks necessitating the use of fully three dimensional modeling techniques. The second difficulty arises in trying to accurately model oblique shock-boundary-layer interaction with Euler-boundary layer equation methods. This occurs because an oblique shock interacts with a boundary layer at an angle and can introduce normal pressure gradients which are inconsistent with the boundary layer assumptions. One way around this difficulty is to use the Navier-Stokes equations locally instead of the boundary layer equations. Finally, turbulence will play a role in determining the boundary layer thickness, boundary layer separation, and thus the effective inlet flow area reduction. Turbulence modeling will be discussed in the next section. Kumar has solved the Euler equations [26] and the Navier-Stokes equations [27] for realistic 3-D scramjet inlets and obtained good results. In conclusion, the computation of the inlet flow field and compression processes is well understood .

Fuel Injection/Flame Holder Region

Unlike the inlet, the fuel injector/flame holder region is extremely complicated and the greatest unknown in current scramjet research. Here, in addition to complex geometries, we have complicated fluid dynamics (fuel injection, flame holding, recirculation) chemistry (ignition) and turbulence. Essentially all physical processes have time and length scales of the same order. Thus the full turbulent reacting Navier-Stokes equations are necessary. The strong coupling between finite rate chemistry and fluid dynamics in the scramjet fuel injector/flame holder region was dramatically illustrated in an experiment outlined by Beach [1]. Figure 1-4 shows the effect of moving the fuel injector point, x/h, on the position of ignition. The arrow in the figure shows the location of the step. In case 1, x/h=-1.67, ignition occurs at the point of fuel injection but leads to strong combustor/inlet interaction[1]. In case 2, x/h=1, the base region apparently became too fuel rich and ignition was delayed. And finally, with x/h = 3, ignition occurred at the desired location. Therefore understanding this process is critical to scramjet engine development. Clearly this equation system can not be completely solved with current methods, yet many conclusions can be derived from a simplified analysis. For example, a control volume

---

[1]Combustor/inlet interaction occurs when the combustion process generates disturbances which destabilize the inlet shock system

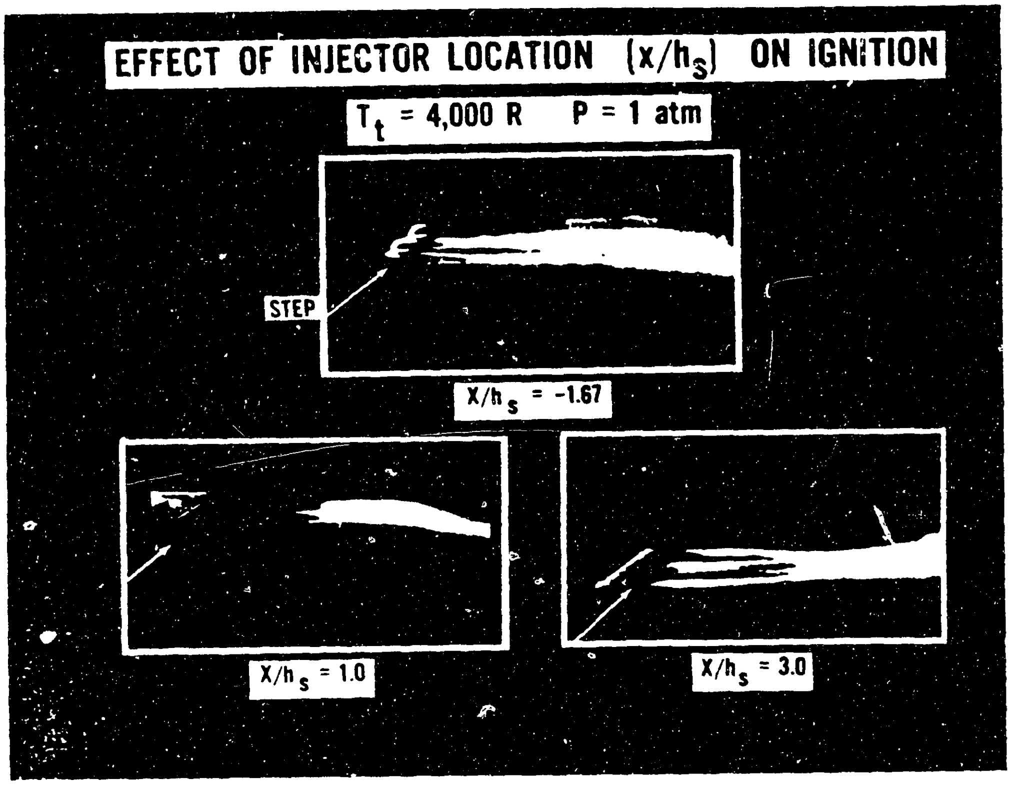


Figure 1-4: Illustration of the Scramjet Flame Holding - From Beach [1]

analysis [43] of the region can provide useful approximate field data to serve as boundary conditions for the combustor analysis. More detailed numerical modeling using current models can provide some useful insight into the character of the region. The work of Drummond [13, 14] and Weidner [57] illustrates the state of the art of computational methods used for this problem. They consider two-dimensional transverse-slot fuel injectors using both nonreacting and reacting chemistry. The chemistry model used was of the complete reaction type which assumes that all available reactants are consumed instantaneously. If the basic approach used by these authors is extended to finite rate chemistry models then several numerical difficulties are encountered. In particular the method becomes computationally very inefficient as the Damkohler number becomes large. In the following chapters it will be shown how this problem can be overcome and that solutions can be obtained efficiently. Drummond and Weidner also indicated that current turbulence models are inadaquate to model scramjet realistic flows. In addition their work indicated that three-dimensional effects can be important and should not be neglected. Thus there are three major problem areas which must be overcome if this part of the scramjet is to be completely understood. The first of these will be addressed in this thesis.

Combustion Region

In a typical scramjet combustor the time scales for cross-flow diffusion of species and species convection are much longer than the time scale for chemical reactions. This implies that the combustor is diffusion controlled and that the chemistry is in equilibrium. Typical diffusion times are of the order of 10^{-3} seconds whereas a typical chemical reaction time is 10^{-8} seconds, implying a Damkohler number of 10^{5}. The flow through the combustor is usually supersonic but can be both subsonic and supersonic in the dual mode concept [1]. If the flow were purely supersonic only the parabolic Reynolds-Averaged turbulent Navier-Stokes and equilibrium chemistry equations would need be solved to fully describe the flow field. In the dual mode concept the parabolic Navier-Stokes would be replaced by the fully elliptic Navier-Stokes equations. Current theory is adaquate to model the combustor to engineering accuracy [14, 48].

Nozzle

The flow through the nozzle is primarily supersonic and the chemistry is frozen ($D \ll 1$). However since the combustor exit flow field contains cross flow species gradients, species diffusion can still occur within the nozzle. Thus the nozzle flow field can be described by the turbulent parabolic Navier-Stokes equations with frozen chemistry. Again as was the case with the combustor and inlet, current theory is sufficient to model the flow processes to engineering accuracy.

1.2. Thesis Objective

The objective of this thesis is to solve a problem important to scramjet research and make a useful contribution to the field of computational fluid dynamics. To this end it was decided to develop a numerical method for flows involving finite rate chemistry. The motivation for this study is the need to better understand the fuel injector/flame holder region within a scramjet, as noted in the previous section. The equations to be solved would include the 1-D inviscid finite rate equations, the 2-D Euler equations with finite rate chemistry and the 2-D Navier-Stokes equations with finite rate chemistry. Due to computer resource limitations the study was limited to two space dimensions.

The research is broken up into two parts.

Part 1

Develop an efficient numerical solution method for flows involving finite rate chemistry. The primary attribute of this method is its ability to achieve steady state quickly irrespective of the Damkohler number.

Part 2

Analyze two representative scramjet flame holders and assess their characteristics. The flame holders chosen were a 2-D inviscid oblique shock flame holder and a 2-D viscous rearward facing step flame holder.

1.3. Technical Approach

The unsteady quasi-1-D inviscid equations and a simple dissociation model were chosen to develop and illustrate the numerical method for flows involving finite rate chemistry. The dissociation model was simple enough that the convergence behavior of the system of equations to steady state could be studied in detail. This study suggested a way of modifying the time scales, a type of equation preconditioning, which can greatly accelerate the solution to steady state. Additional steady state acceleration techniques are also considered in this thesis. The method is applied to a quasi-1-D duct flow with a realistic H_2 - air chemistry model and the properties of the solution method are analyzed.

The 2-D examples considered in this thesis were chosen because they represent two different types of scramjet flame holders. The first type of flame holder is characterized by an oblique shock which triggers a reaction zone, figure 1-5. For this class of problems, the Reynolds number is sufficiently high, $Re=10^7$, such that only the 2-D Euler equations with finite rate chemistry need be solved. The second class of flame holders is characterized by viscous recirculating flow behind rearward facing steps, figure 1-6. In this case the 2-D Navier-Stokes equations with finite rate chemistry need to be used. Here the flame is anchored through transfer of heat and chemical radicals from the hot zone behind the step to the cold reactants flowing over the step. The examples considered here are restricted to laminar premixed flows but the numerical methods developed apply equally well to nonpremixed flows. The study is restricted to premixed flows so that a tractable problem could be solved.

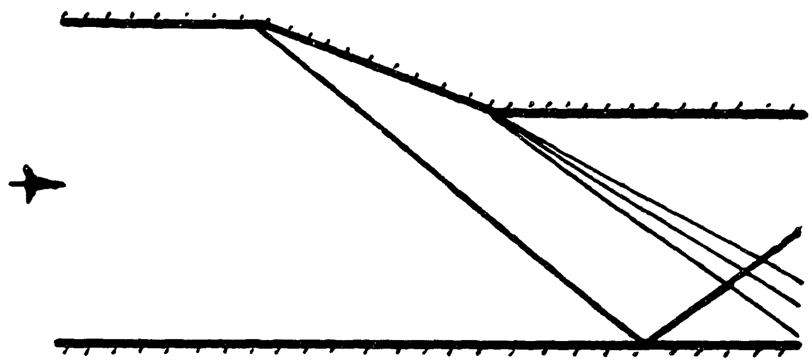


Figure 1-5: Ramp Test Geometry

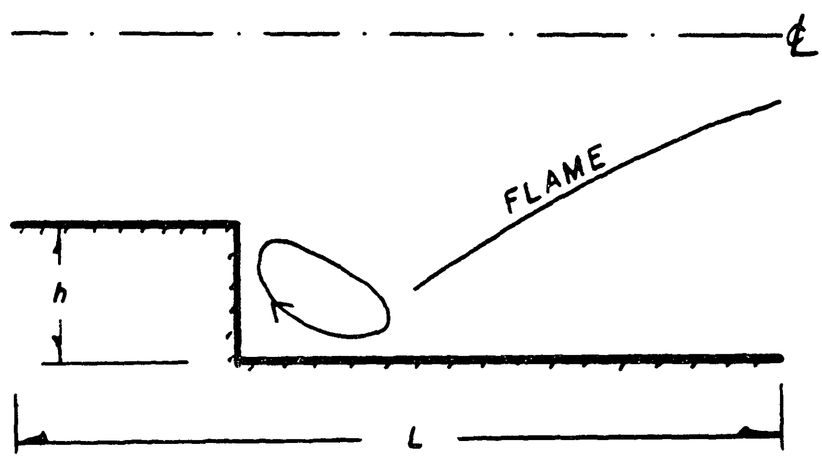


Figure 1-6: Rearward Facing Step Test Geometry

Chapter 2

EQUATIONS TO BE SOLVED

The purpose of this chapter is to outline the conservation equations and the chemistry models used in this study. The chapter will consider only the partial differential and algebraic equations necessary to describe chemically reacting flows. The numerical integration of these equations will be described in a later chapter.

The equations considered in this study include the quasi-1-D Euler equations with finite rate chemistry, the 2-D Euler equations with finite rate chemistry, and the 2-D Navier-Stokes equations with finite rate chemistry.

2.1. Transport Equations

The equations to be solved are the conservation of mass, momentum, energy and species. These equations represent the physics governing a flowing chemically reacting fluid. In general they can be expressed as follows,

Conservation of Mass,

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho V) = 0 \tag{2.1}$$

Conservation of Momentum,

$$\frac{\partial (\rho V)}{\partial t} + \nabla \cdot (\rho V V) + (\nabla \cdot \mathbf{P}) = 0 \tag{2.2}$$

Conservation of Energy,

$$\frac{\partial (\rho E)}{\partial t} + \nabla \cdot (\rho V E) + \mathbf{P} \cdot (\nabla V) + \nabla \cdot q = 0 \tag{2.3}$$

Conservation of Species,

$$\frac{\partial(\rho Y_k)}{\partial t} + \nabla\cdot(\rho V Y_k) + \nabla\cdot(\rho Y_k V_k) = w_k \tag{2.4}$$

Total Internal Energy

$$E = \int C_v dT + .5(V^2) + \Sigma H f_k Y_k \tag{2.5}$$

Equation of State

$$p = \rho \overset{\circ}{R} T \Sigma_{k=1}^{N} Y_k / A w_k \tag{2.6}$$

where Hf_k is heat of formation for species k, V_k is the diffusion velocity of species k and Aw_k is the atomic weight of species k. To close this system of equations relations for **P**, q and $\rho_k V_k$ are needed. w_k, the species production term will be defined later.

The term **P**, which appears in the momentum and energy equations, represent the pressure and viscous shear stress forces. These terms account for the diffusion of momentum and energy. For a compressible, Newtonian fluid (Williams [58]) **P** is defined as,

$$\mathbf{P} = [p + (2/3\mu - \kappa')(\nabla\cdot V)]\mathbf{U}' - \mu[(\nabla V) + (\nabla V)^T] \tag{2.7}$$

where κ' is the bulk viscosity, $\mathbf{U}'$ is the unit tensor, the two dots (:) imply the tensors are to be contracted twice, and the T denotes the transpose of the tensor.

The heat flux vector, q, is given by the following equation [58],

$$q = -\kappa \nabla T + \rho \Sigma h_k Y_k V_k \tag{2.8}$$

where k is the heat diffusion coefficient and

$$h_k = Hf_k + \int_{T_0}^{T} c_{p_k} dT \quad . \tag{2.9}$$

The Dufour contribution to the flux has to be neglected as it is small compared to the other two heat flux terms.

Finally the species diffusion term, $\rho Y_k V_k$, can also be expressed in terms of known quantities using Fick's law as,

$$V_k = -D\nabla \ln Y_k \tag{2.10}$$

$$V_k = -D\frac{\nabla Y_k}{Y_k} \tag{2.11}$$

Thus $\rho Y_k V_k$ can be written as,

$$\rho Y_k V_k = -\frac{\mu}{Sc}\nabla Y_k \tag{2.12}$$

where S_c is the Schmidt number($Sc=\mu/\rho D$). D is assumed to be the same for all species. Now if the Lewis number ($Le=Sc/Pr$), is equal to one then equation (2.12) may be written as,

$$\rho Y_k V_k = \frac{\mu}{Pr}\nabla Y_k \tag{2.13}$$

where the Prandtl number is $Pr=\mu c_p/k$. Note Sc, Le and Pr are based on local quantities here. Similarly the heat flux vector can be written as,

$$q = -k\nabla T - \Sigma\frac{\mu}{Pr}h_k\nabla Y_k \quad . \tag{2.14}$$

Note there are N-1 species transport equations[2].

The laminar viscosity is calculated from Sutherland's law

$$\mu_{laminar} = \frac{1.458 \times 10^{-6}\ T^{3/2}}{(T + 100.33)}\ kg/m{-}s \tag{2.15}$$

where T is measured in degrees Kelvin. The specific heats, c_p and c_v, are functions of the local species concentration.

These equations can be non-dimensionalized following procedures given in reference [41]. The non-dimensional variables used are as follows;

$$u' = \frac{u}{u_\infty} \tag{2.16}$$

$$v' = \frac{v}{u_\infty} \tag{2.17}$$

$$T' = \frac{T}{T_\infty} \tag{2.18}$$

$$x' = \frac{x}{L_\infty} \tag{2.19}$$

$$y' = \frac{y}{L_\infty} \tag{2.20}$$

$$E' = \frac{E}{(u^2)_\infty} \tag{2.21}$$

---

[2]Actually only N-1 species transport equations are needed since the global contunity equation can be used to derive information about the last specie

$$h' = \frac{h}{(u^2)_\infty} \tag{2.22}$$

$$p' = \frac{p}{(\rho u^2)_\infty} \tag{2.23}$$

$$w' = \frac{L_\infty}{(\rho u)_\infty} w \tag{2.24}$$

$$c_p' = \frac{c_p}{c_{p\infty}} \tag{2.25}$$

$$c_v' = \frac{c_v}{c_{v\infty}} \tag{2.26}$$

$$k' = \frac{k}{k_\infty} \tag{2.27}$$

$$\mu' = \frac{\mu}{\mu_\infty} \tag{2.28}$$

As a result of the non-dimensionalization, the following three parameters will fall out;

(2.29)

$$Re = \left(\frac{\rho L u}{\mu}\right)_\infty \tag{2.30}$$

$$Pr = \left(\frac{\mu C_p}{k}\right)_\infty \tag{2.31}$$

$$M = (u/(\gamma RT)^{1/2})_\infty \tag{2.32}$$

The prime denotes the non-dimensional quantities and ∞ denotes the free steam reference quantities. For the remainder of this thesis the primes will be dropped and all the variables will be non-dimensional unless otherwise stated.

With these non-dimensional variables equations (2.1), (2.2), (2.3) and (2.4) can be rewritten as,

Conservation of Mass,

$$\frac{\partial \rho}{\partial t} + \nabla\cdot(\rho V) = 0 \tag{2.33}$$

Conservation of Momentum,

$$\frac{\partial(\rho V)}{\partial t} + \nabla\cdot(\rho VV) + (\nabla\cdot \mathbf{P}) = 0 \tag{2.34}$$

Conservation of Energy,

$$\frac{\partial(\rho E)}{\partial t} + \nabla\cdot(\rho VE) + \mathbf{P} \cdot (\nabla V) + \frac{1}{RePr} \nabla\cdot q = 0 \tag{2.35}$$

Conservation of Species,

$$\frac{\partial(\rho Y_k)}{\partial t} + \nabla\cdot(\rho VY_k) + \frac{1}{RePr} \nabla\cdot(\frac{k}{c_p}\nabla Y_k) = w_k \tag{2.36}$$

where **P** and q become

$$\mathbf{P} = [p + (2/(3Re) - 1/Re_{bulk})(\nabla\cdot V)]\mathbf{U}' - 1/Re[(\nabla V) + (\nabla V)^T] \tag{2.37}$$

$$q = -\frac{1}{M^2(\gamma - 1)}k\nabla T - \frac{k}{c_p}\Sigma h_k \nabla Y_k \quad . \tag{2.38}$$

2.1.1. Quasi 1-D Euler Equations With Finite Rate Chemistry

Equation (2.33) through (2.36) were simplified to the quasi-1-D inviscid equations to provide a simpler model for assessing the numerical integration characteristics of the coupled system. Figure 2-1 shows a typical quasi 1-D geometry.

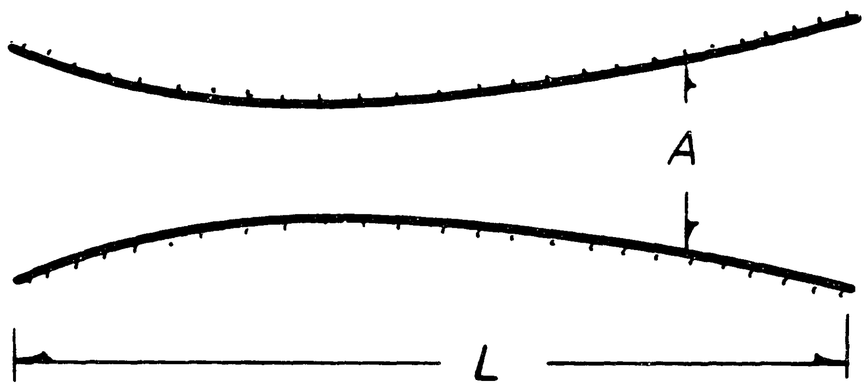


Figure 2-1: Quasi 1 - D Geometry

Written in conservation form these equations are given by,

$$\frac{\partial U}{\partial t} + \frac{\partial F}{\partial x} + H = 0 \tag{2.39}$$

where U, F and H are,

$$U = \begin{bmatrix} \rho A \\ \rho u A \\ \rho E A \\ \rho Y_k A \end{bmatrix} \tag{2.40}$$

$$F = \begin{bmatrix} \rho u A \\ \rho u^2 A + pA \\ \rho H u A \\ \rho Y_k u A \end{bmatrix}$$

$$H = \begin{bmatrix} 0 \\ -p\ dA/dx \\ 0 \\ -w_k \end{bmatrix} .$$

2.1.2. 2-D Euler Equations With Finite Rate Chemistry

The 2-D Euler equations with finite rate chemistry are given as follows,

$$\frac{\partial U}{\partial t} + \frac{\partial F}{\partial x} + \frac{\partial G}{\partial y} + H = 0 \tag{2.41}$$

where U, F, G and H are,

(2.42)

$$U = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho E \\ \rho Y_k \end{bmatrix}$$

$$F = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho Hu \\ \rho Y_k u \end{bmatrix}$$

$$G = \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho Hv \\ \rho Y_k v \end{bmatrix}$$

$$H = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ -w_k \end{bmatrix} .$$

2.1.3. 2-D Navier Stokes Equations With Finite Rate Chemistry

In the first section of this chapter we outlined the general compressible, viscous equations where the viscous stress terms were written in terms of P. In this section these terms will be expanded. The 2-D Navier-Stokes with finite rate chemistry may be written as follows,

$$\frac{\partial U}{\partial t} + \frac{\partial F}{\partial x} + \frac{\partial G}{\partial y} + H = 0 \tag{2.43}$$

where U, F, G and H are,

$$U = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho E \\ \rho Y_k \end{bmatrix} \tag{2.44}$$

$$F = \begin{bmatrix} \rho u \\ \rho u^2 + \sigma_{xx} \\ \rho uv + \tau_{xy} \\ \rho Eu + u\sigma_{xx} + v\tau_{yx} + q_x \\ \rho Y_k u - \Gamma(Y_k)_x \end{bmatrix}$$

$$G = \begin{bmatrix} \rho v \\ \rho uv + \tau_{yx} \\ \rho v^2 + \sigma_{yy} \\ \rho Ev + v\sigma_{yy} + u\tau_{xy} + q_y \\ \rho Y_k v - \Gamma(Y_k)_y \end{bmatrix}$$

$$H = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ -w_k \end{bmatrix}.$$

The various stress terms can be written in non-dimensional form as,

$$\sigma_{xx} = p - \lambda(u_x + v_y) - 2\frac{1}{Re}\mu u_x \tag{2.45}$$

$$\tau_{xy} = - \lambda(u_y + v_x) \tag{2.46}$$

$$\sigma_{yy} = p - \lambda(u_x + v_y) - 2\frac{1}{Re}\mu v_y \tag{2.47}$$

$$q_x = - \frac{1}{RePr}\left[\frac{k}{(\gamma - 1)M^2}\frac{\partial T}{\partial x} - \frac{k}{c_p}\Sigma h_k \frac{\partial Y_k}{\partial x}\right] \tag{2.48}$$

$$q_y = - \frac{1}{RePr}\left[\frac{k}{(\gamma - 1)M^2}\frac{\partial T}{\partial y} - \frac{k}{c_p}\Sigma h_k \frac{\partial Y_k}{\partial y}\right] \tag{2.49}$$

$$\lambda = - \frac{2\mu}{3Re} \tag{2.50}$$

$$\Gamma = \frac{1}{RePr}\frac{k}{c_p} \quad (Le=1) \quad . \tag{2.51}$$

Note that λ, M, Re, Pr, Sc and Le are based on the free stream reference quantities.

2.1.4. Integral Form Of The Governing Equations

Instead of solving the partial differential equations it is often desirable to solve the integral forms of these equations using a finite volume method. The integral form of the governing equations can be expressed as,

$$\frac{\partial}{\partial t}\iint_{\Omega} U dxdy + \int_{d\Omega} \mathrm{M}\cdot n \; ds + \iint_{\Omega} H dxdy = 0 \tag{2.52}$$

where **n** is an outward pointing normal vector, Ω is the region of interest and dΩ is the boundary curve. In two dimensions the volume has unit depth. Figure 2-2 shows the nomenclature used. The vector H represents the source term quantities defined above. The second order tensor **M** is defined by

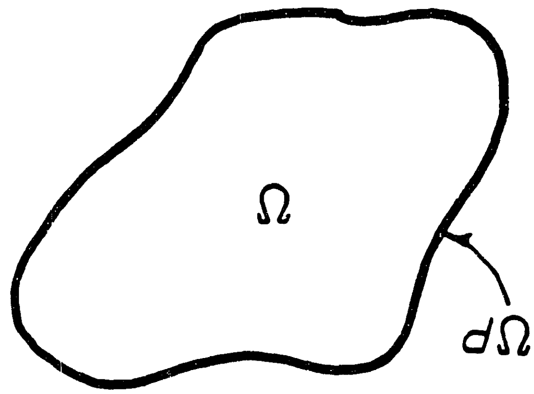


Figure 2-2: Finite Volume Nomenclature

$$M = F\mathbf{i}_x + G\mathbf{i}_y \tag{2.53}$$

where F and G are defined by equations (2.44). Equation (2.52) can be written in Cartesian form to give,

$$\frac{\partial}{\partial t}\iint_{\Omega} U dx dy + \int_{\partial\Omega}(F dy - G dx) + \iint_{\Omega} H dx dy = 0 \quad . \tag{2.54}$$

Throughout the remainder of this thesis the integral form of the governing equations will be used.

2.2. Chemistry Models

Two different chemistry models are considered in this study. The first, a simple diatomic dissociation reaction, is used to assess the characteristics of various numerical integration techniques. The second, a realistic hydrogen - air chemistry model, is used to demonstrate the validity of the developed numerical methods with multi-component[3]

[3]Multi-component implying more than one chemical reaction time scale

chemistry. The H_2-Air model is also used to study two candidate scramjet flame holders.

Before describing the details of these two chemistry models consider the general chemical reaction model. The model will serve to illustrate the character and complexity of finite rate chemistry. Following Williams [58], finite rate chemistry is described by a set of reactions of the following form,

$$\Sigma_{j=1}^{N_s} \lambda'_{ij} C_j \overset{k_{fi}}{\underset{k_{bi}}{\Longleftarrow\!=\!=\!=\!=\!=\!=\!=\!=\!\Longrightarrow}} \Sigma_{j=1}^{N_s} \lambda''_{ij} C_j \qquad (2.55)$$

where i = 1, 2, 3...N_R, j = 1, 2, 3...N_S, the λ's are the stoichiometric coefficients and the C's are the species concentrations. Note C, as used here, is a dimensional species concentration. The reaction rates k_{fj} and k_{bj} are of modified Arrhenius form and are given by expressions of the form,

$$k_i = A_i T^{B_i} e^{-C_i/T} \qquad (2.56)$$

It is the exponential nature of this term that is responsible for many of the difficulties experienced when numerically integrating these equations. These difficulties will be discussed in the next chapter. The rate of change of species j by reaction i is

$$(\dot{C}_j)_i = (\lambda''_{ij} - \lambda'_{ij}) [k_{fi} \pi_{j=1}^{N_s} C_j^{\lambda_{ij}} - k_{bi} \pi_{j=1}^{N_s} C_j^{\lambda_{ij}}] \qquad (2.57)$$

The total rate of change of the concentration of species j by all N_s reactions is then found by summing the contributions from each reaction,

$$\dot{C}_j = \Sigma_{i=1}^{N_r} (\dot{C}_j)_i \qquad (2.58)$$

Finally, the production rate of species j is found from,

$$w_j = \dot{C}_j M_j \tag{2.59}$$

where M_j is the molecular weight of species j. The concentration is related to the mass fraction through the relation,

$$C_j' = \frac{Y_j \rho'}{M_j'} \quad . \tag{2.60}$$

Note there is one reaction rate term, w_j, for each species transport equation.

2.2.1. Simple Dissociation Model

Diatomic dissociation represents one of the simplest chemical reactions known yet retains features leading to the numerical integration difficulties characteristic of finite rate chemistry. Specifically, the model retains the exponential function in the rate term. For small changes in temperature this term can vary by several orders of magnitude and, thus, can create major integration difficulties. The dissociation reaction can be written as

$$O_2 \overset{k_f}{\longleftrightarrow} 2\ O \tag{2.61}$$

where

$$k_f = AT^B e^{-C/T} \quad . \tag{2.62}$$

Only the forward reaction rate is considered here for simplicity. The coefficients A, B and C are given in chapter six. These constants were chosen to produce a difficult numerical test case and do not necessarily represent O_2 dissociation. The reaction rate expressions are given by,

$$w_{O_2} = -\frac{k_f U_{O_2}}{\rho} \tag{2.63}$$

and

$$w_O = -w_{O2} \quad . \tag{2.64}$$

where $U_{O_2} = \rho Y_{O_2} A$. Finally a relation can be written for the sum of the species density fractions, i.e.,

$$Y_{O_2} + Y_O = 1 \quad . \tag{2.65}$$

2.2.2. H_2-Air Combustion Model

The Hydrogen-Air combustion model used in this thesis was proposed by Rogers and Chinitz [43] in 1982. The model was developed to represent H_2 - air combustion kinetics in a scramjet combustor with as few reaction steps and species as possible. The model consists of the following two steps,

$$H_2 + O_2 \underset{k_{b1}}{\overset{k_{f1}}{\longleftrightarrow}} 2\,OH$$

$$2\,OH + H_2 \underset{k_{b2}}{\overset{k_{f2}}{\longleftrightarrow}} 2\,H_2O \tag{2.66}$$

where the forward reaction rate constants k_{f1} and k_{f2} are given by,

$$k_{fi} = A_i(\phi) T^{N_i} exp^{-E_i/RT} \tag{2.67}$$

where $A_i(\phi)$ is a function of the equivalence ratio ϕ. ϕ is defined as the fuel to air ratio divided by the stoichiometric fuel to air ratio. Values of the parameters used in this model are,

$$A_1(\phi) = (8.917\phi + \frac{31.433}{\phi} - 28.950)\times 10^{47} cm^3/mole\cdot s \tag{2.68}$$

$$A_2(\phi) = (2.000 + \frac{1.333}{\phi} - .833\phi)\times 10^{64} cm^6/mole^2\cdot s$$

$$E_1 = 4865 \ cal/mole$$

$$E_2 = 42{,}000. \ cal/mole$$

$$N_1 = -10$$

$$N_2 = -13$$

$$R_u = 1.987 \ \frac{cal}{mole\cdot k}$$

From the law of mass action the backward reaction rates are given by,

$$k_{b1} = \frac{k_{f1}}{Keq_1} \tag{2.69}$$

and

$$k_{b2} = \frac{k_{f2}}{keq_2} \tag{2.70}$$

where keq_1 and keq_2 are the equilibrium values for each reaction.

The model is valid for temperatures between 1000k and 2000k and equivalence ratios between .2 and 2.0. Because the chemistry model is not valid below at temperature of 1000K an ignition temperature must be specified. Typically for H_2 - air combustion the ignition temperature is equal to 1000K. The reaction rates for the various species conservation equations can be written as,

$$w_{O_2} = Aw_{O_2} \ (-k_{f1}C_{H_2}C_{O_2} + k_{b1}(C_{OH})^2) \tag{2.71}$$

$$w_{H_2O} = Aw_{H_2O}\ 2\ (-k_{f2}C_{H_2}(C_{OH})^2 - k_{b2}(C_{H_2O})^2) \tag{2.72}$$

$$W_{H_2} = Aw_{H_2}\ (-k_{f1}C_{H_2}C_{O_2} + k_{b1}(C_{OH})^2 - k_{f2}U_{H_2}(C_{OH})^2 + k_{b2}C_{H_2O}) \tag{2.73}$$

$$W_{OH} = Aw_{OH}\ 2\ (-k_{f1}C_{H_2}C_{O_2} + k_{b1}(C_{OH})^2 - k_{f2}C_{H_2}(C_{OH})^2 + k_{b2}C_{H_2O}) \quad . \tag{2.74}$$

where the C's are the species concentrations (moles/cm^3). Finally to close the equation set a relation can be written for the sum of the species density fractions,

$$Y_{H_2} + Y_{O_2} + Y_{OH} + Y_{H_2O} + Y_{N_2} = 1 \quad . \tag{2.75}$$

Note N_2 is present in the mixture but is assumed to be inert.

2.3. The Coupling Between Fluid Mechanics and Chemistry

The fluid dynamic processes and the chemical processes are coupled in two different ways. The first way occurs through the heat of formation term in the total energy equation (2.5). Since this contribution to the total energy can vary widely it can greatly influence the fluid dynamics. The second type of coupling occurs through the fluid properties, i.e., C_p, C_v, etc. Consider the following equation for the effective C_p,

$$(C_p)_{eff} = \Sigma_{i=1}^{N_s}\ C_{pi}Y_i \tag{2.76}$$

Similar relations exist for $(C_v)_{eff}$ and the other fluid property values. Together these coupling mechanisms can dramatically alter the resulting flow fields. A number of examples of these kinds of interaction were shown by Bussing [5]. Chapters 6, 7 and

8 will consider several examples where heat addition can have a pronounced effect on the flow field.

Chapter 3

NUMERICAL INTEGRATION - STIFF EQUATIONS

A variety of numerical methods can be used to solve the unsteady transport equations described in chapter 2. The questions to be addressed in this chapter are how do these methods compare and what limitations characterize each of these methods. The chapter is broken into three sections where each section assesses a particular method. The numerical integration techniques to be considered include the fully explicit, the fully implicit and the point implicit methods. The limits of each method are determined through a Von Neuman stability analysis. In addition to the stability analysis, a discussion of the available literature will also be given. It will be shown that the numerical integration time step for the explicit scheme is dependent on τ_{fluid} and τ_{chem}, for the point implicit scheme is dependent on τ_{fluid} and for the fully implicit scheme is unrestricted.

Before examining these numerical techniques we shall consider some of the problems associated with the numerical integration of stiff systems. Mathematically speaking, stiffness can be defined by examining the eigenvalues of the Jacobian of the governing equation system. The Jacobian is defined as the matrix formed by differentiating the flux vectors **F** and **G** with respect to the state vector **U**. Stiffness is defined as the ratio of the largest eigenvalue to the smallest eigenvalue. Stiffness can be defined in terms of time scales or lengths scales. For the remainder of this thesis the term stiffness will refer to time scale stiffness, unless otherwise indicated. A useful definition of time stiffness is equal to the ratio of the largest time scale to the smallest time scale, ie,

$$\text{Stiffness} = \tau_{largest}/\tau_{smallest} \; . \tag{3.1}$$

High levels of stiffness can severely degrade the performance of numerical methods, as will be illustrated in this chapter. For the system of equations governing chemically reacting flows, stiffness typically arises from the source terms, H, in the species conservation equations. If the source terms are large they produce rapid temporal and spatial changes in the dependent variables, leading to a range of physical time scales.

For problems involving more than one specie, several time scales can occur. In addition there are fluid dynamic time scales associated with convection and diffusion. Typically for the reacting flows considered in this thesis the stiffness parameter can be as high as 10^6.

To assess the problem of multiple time scales, consider the following example due to Seinfeld et al, [46]. They considered the linear O.D.E. system given by,

$$y_t = Ay \tag{3.2}$$

where $y = [y_1,y_2]^T$, $y(0) = [2,1]^T$, and

$$A = \begin{vmatrix} -500.5 & 499.5 \\ 499.5 & -500.5 \end{vmatrix} \tag{3.3}$$

The solution of equation (3.1) is

$$y_1(x) = 1.5e^{-t} + 0.5e^{-1000t}$$
$$y_2(x) = 1.5e^{-t} - 0.5e^{-1000t} \tag{3.4}$$

where the eigenvalues of A are $\lambda_1 = -1000$ and $\lambda_2 = -1$. Both y_1 and y_2 have a rapidly decaying component corresponding to λ_1 and a much slower decaying component corresponding to λ_2. If we were solving this problem numerically, accuracy would dictate that we advance the solution using very small time steps. But, once the component due to λ_1 decays , we would prefer to advance the solution using larger time steps that would still maintain the desired accuracy. Different numerical schemes have different stability characteristics and not all allow this variation in time step. Therefore, care must be taken to pick a numerical method that will allow the desired choice of time step.

In order to compare the different numerical methods it is necessary to simplify equation (2.39) to a representative model scalar equation. The model equation must be simple enough to be amenable to analytic analysis yet still represent the essential physics. With this in mind the model equation selected is:

$$U_t = - aU_x - U/\tau_{chem} \tag{3.5}$$

where the term U/τ_{chem} represents a typical chemical source term and a is a characteristic convection velocity. We will use this equation to assess the character of each of the methods.

3.1. EXPLICIT NUMERICAL INTEGRATION

Of the methods to be considered here, explicit numerical methods are the simplest. In addition to being the simplest they were the first methods to be applied to problems involving finite rate chemistry. In this section we will show that purely explicit methods suffer from severe time step restrictions when the stiffness is high.

A typical explicit numerical technique for equation (2.39) might be,

$$U_j^{n+1} = U_j^n - \Delta t(F_x^n + H^n) \tag{3.6}$$

The term F^n_x is differenced according to the numerical method chosen. Note that all terms not involving time differencing are evaluated at time level n.

3.1.1. Historical Background of the Explicit Method

Fully explicit methods have found application for many years to problems involving finite rate chemistry. They are generally considered effective when the stiffness parameter is less than approximately ten[4]. Lomax [31, 30] in his 1968 paper discusses the behavior of several explicit techniques and shows where they should be used. In a more recent paper Candel, Daradiha and Esposito [7] applied MacCormack's 1969 scheme to a viscous 2-D chemically reacting flow over steps.

---

[4]The choice of 10 will become clear in the next subsection

3.1.2. Stability Analysis

Two numerical schemes will be analyzed here. The methods chosen represent popular schemes in use today. The methods were proposed by MacCormack [32] and Jameson, Schmidt and Turkel [22]. Both of these methods, as originally proposed by the respective authors, are fully explicit.

First let us consider the MacCormack [32] explicit scheme which consists of a predictor and a corrector step. Applying the MacCormack scheme to the model equation, (3.1), gives for the predictor step,

$$U_j^{\times} = U_j^n - \frac{a\Delta t}{\Delta x}(U_{j+1}^n - U_j^n) - \frac{\Delta t}{\tau_{chem}} U_j^n \tag{3.7}$$

and for the corrector step,

$$U_j^{n+1} = 1/2\ [U_j^n + U_j^{\times\times} - \frac{a\Delta t}{\Delta x}(U_j^{\times} - U_{j-1}^{\times}) - \frac{\Delta t}{\tau_{chem}} U_j^{\times}]\ . \tag{3.8}$$

The predictor and corrector equations can be combined to yield,

$$\begin{aligned} U_j^{n+1} = {} & U_{j-1}^n(1 + \Gamma - D)\Gamma/2 \\ & + U_j^n(1 - \Gamma^2 - D + D^2/2) \\ & + U_{j+1}^n(-1 + \Gamma + D)\Gamma/2 \end{aligned} \tag{3.9}$$

where

$$\Gamma = a\Delta t/\Delta x \tag{3.10}$$

$$D = \Delta t/\tau_{chem} \quad . \tag{3.11}$$

Next we can find the numerical time step restrictions using the Von Neuman stability analysis procedure. We begin by rewritting the U's in terms of their Fourier components,

$$U_j^n = V^n e^{ij\phi}$$

$$U_j^{n+1} = V^{n+1} e^{ij\phi} \tag{3.12}$$

where ϕ is a phase angle and V^n is the amplitude function at time level n. The amplification factor G can be defined with the equation,

$$V^{n+1} = G\ V^n \ . \tag{3.13}$$

For the scheme to be stable [G] $\leq$ 1 , for all values of ϕ. Substituting these expressions into equation (3.9) yields,

$$\begin{aligned} G = {} & e^{-i\phi}(1 + \Gamma - D)\Gamma/2 \\ & + (1 - \Gamma^2 - D + D^2/2) \\ & + e^{+i\phi}(-1 + \Gamma + D)\Gamma/2 \ . \end{aligned} \tag{3.14}$$

Finally this equation can be simplified to,

$$\begin{aligned} G = [& E^2 - 4\ \Gamma^2\ E\ \sin^2(\phi/2) + 4\Gamma^4 \sin^4(\phi/2) \\ & + \sin^2\phi(-\Gamma + D\Gamma/2)^2]^{1/2} \end{aligned} \tag{3.15}$$

where

$$E = 1 + (-\ D + D^2/2) \ . \tag{3.16}$$

Plotting G, as given by equation (3.15), as a function of Γ and D leads to the contour plot shown in figure 3-1. The plot shows graphically that $\Gamma < 1$ and that $D < 2$ need to be satisfied for the scheme to be stable. The first condition is the well known CFL condition. The second condition says that $\Delta t < 2\tau_{chem}$, which can be a severe limitation if the stiffness paremeter is large. Thus we see that the numerical time is directly tied to the chemical time scales. In fact the contours show that the most restrictive case on Γ occurs when $D = 1.0$ i.e., $\Gamma < .9$.

Next consider the stability of the explicit Jameson, Schmidt and Turkel [22] scheme with a source term. The basic Jameson, Schmidt and Turkel scheme consists of four steps which take the following form at time level n:

$$U^{(0)} = U^{n}$$
$$U^{(1)} = U^{(0)} - a_1 \Delta t R U^{(0)}$$
$$U^{(2)} = U^{(0)} - a_2 \Delta t R U^{(1)}$$
$$U^{(3)} = U^{(0)} - a_3 \Delta t R U^{(2)}$$
$$U^{(4)} = U^{(0)} - a_4 \Delta t R U^{(3)}$$
$$U^{n+1} = U^{(4)} \tag{3.17}$$

where

$$RU_j^{(q)} = \frac{a\Delta t}{2\Delta x}(U_{j+1}^{(q)} - U_{j-1}^{(q)}) + \frac{\Delta t}{\tau_{chem}} U_j^{(q)} \tag{3.18}$$

and

$$a_1 = 1/4, \quad a_2 = 1/3, \quad a_3 = 1/2, \quad a_4 = 1 \quad .$$

Let us begin by considering a single stage of the scheme,

$$U_j^{k+1} = U_j^{k} - \Gamma/2(U_{j+1}^{k} - U_{j-1}^{k}) - DU_j^{k} \tag{3.19}$$

where Γ and D are given by equations (3.10) and (3.11). Applying the Von Neumann stability analysis method to this equation yields,

$$V^{k+1} = (1 + z)V^{k} \tag{3.20}$$

with z given by,

$$z = -i\Gamma \sin\phi - D \tag{3.21}$$

where ϕ is a phase angle. The multistage scheme can be written in a like manner as,

$$V^{n+1} = g(z)V^{n} \tag{3.22}$$

where g(z) is the multistage scheme amplification factor. With multistage coefficient values of, $a_1 = 1/4$, $a_2 = 1/3$, $a_3 = 1/2$ and $a_4 = 1$, g(z) is given by the polynomial,

$$g(z) = 1 + z + z^2/2 + z^3/6 + z^4/24 \ . \tag{3.23}$$

g(z) like G must be equal to or less than 1 for the scheme to be stable. Figure 3-2 shows the contour plot of g(z) for various values of Γ and D's. An approximation from the plot shows that the scheme is stable for, $\Gamma + D < 2\sqrt{2}$. As we found with the MacCormack scheme the numerical time step is again dependent upon τ_{chem} which can be very restrictive for stiff problems.

3.2. IMPLICIT NUMERICAL INTEGRATION

Implicit numerical methods offer several advantages over explicit methods. The principle advantage is that the numerical time step can be made independent of the chemical time scales. Implicit methods however are considerably more complicated than explicit methods. The remainder of this section will assess the character of implicit schemes.

A typical implicit numerical technique for equation (2.39) could be,

$$U_j^{n+1} = U_j^{n} - \Delta t(F_x^{n+1} + H^{n+1}) \tag{3.24}$$

where all terms are evaluated at time level n+1.

3.2.1. Historical Background of the Implicit Method

Fully implicit methods have been used extensively for the solution of stiff ordinary differential equations, see references [19, 20, 46, 31, 30]. These techniques have proved effective for solving point chemical kinetic problems[5]. The extension of these methods to partial differential equations has also received considerable attention [25, 2, 37]. For one dimensional problems the approach can yield quadratic convergence. However for multiple space dimensions the performance of this technique can be severely degraded. The degradation is due to the difficulties associated with inverting the large matrices characteristic of fully implicit methods [25].

3.2.2. Stability Analysis

To study the stability of a fully implicit method consider the following example. If the model problem (3.5) is differenced according to the fully implicit backward Euler method we have,

$$U_j^{n+1} = U_j^n - a\frac{\Delta t}{2\Delta x}(U_{j+1}^{n+1} - U_{j-1}^{n+1}) - \frac{\Delta t}{\tau_{chem}}U_j^{n+1} \tag{3.25}$$

which can be rewritten as,

$$\Gamma/2U_{j+1}^{n+1} + (1 + D)U_j^{n+1} - \Gamma/2U_{j-1}^{n+1} = U_j^n \tag{3.26}$$

where Γ and D are given in the previous section. Next performing a Von Neuman stability analysis on this equation gives the following expression for the amplification factor, G,

$$|G| = \frac{1}{((1 + D)^2 + (\sin\phi)^2\Gamma^2)^{1/2}} \tag{3.27}$$

where ϕ is a phase angle. For this scheme to be stable G < 1. It is clear that for any Γ or D the fully implicit scheme is unconditionally stable.

---

[5]Point chemical kinetic means that the fluid is stationary

The fully implicit approach is very efficient for one dimensional problems. However for multiple space dimensions this method requires inverting large matricies [25] or the need to introduce simplifications like approximate factorization [56]. Both of these approaches introduce difficulties which can severely limit the performance of the method.

3.3. POINT IMPLICIT NUMERICAL INTEGRATION

Point implicit methods offer some of the advantages of fully implicit methods without some of the serious disadvantages. We shall see that these methods remove the restriction on the numerical time step due to τ_{chem}.

The point implicit version of the model equation (3.5) can be written in the following form,

$$U_j^{n+1} = U_j^n - \Delta t(F_x^n + H^{n+1}) \tag{3.28}$$

In this case only the chemical source term is evaluated at time level n+1.

3.3.1. Historical Background of the Point Implicit Method

In 1959 Curtiss [10] recognized that one effective strategy for solving stiff systems of ODE's was to solve the equations implicitly or alternatively to evaluate the chemical source terms at time level n+1. For problems involving PDE's several authors [50, 36] have described methods where terms involving spatial gradients are treated explicitly and the chemical source terms are evaluated implicitly (point implicit methods). However, these studied were limited to one spatial dimension. These ideas were extended to multiple space dimensions by a group at Lockheed [29]. This was done while the author's current research effort was in progess. It will be shown in chapter 5 that these techniques are a special case of a general solution method, developed as part of the author's Ph.D. research effort. In addition other authors [42, 53, 51] have developed splitting methods which share many similarities to the point implicit methods. As will be shown in the next subsection these techniques have important advantages over both explicit and fully implicit methods.

3.3.2. Stability Analysis

In this section we will consider variations of the basic MacCormack [32] and Jameson, Schmidt and Turkel [22] schemes. The modifications involve treating the chemical source term implicitly.

First consider the restrictions on the numerical time step for the point implicit MacCormack scheme. Applying the scheme to the model equation, (3.5), gives for the predictor step,

$$U_j^{\times} = U_j^n - \frac{a\Delta t}{\Delta x}(U_{j+1}^n - U_j^n) - \frac{\Delta t}{\tau_{chem}}U_j^{\times} \tag{3.29}$$

and for the corrector step,

$$U_j^{n+1} = 1/2(U_j^n + U_j^{\times} - \frac{a\Delta t}{\Delta x}(U_j^{\times} - U_{j-1}^{\times}) - \frac{\Delta t}{\tau_{chem}}U_j^{n+1}) \tag{3.30}$$

Again if we define $\Gamma = a\Delta t/\Delta x$ and $D = \Delta t/\tau_{chem}$, the combined predictor and corrector equation becomes,

$$\begin{aligned} U_j^{n+1} &= U_{j-1}^n(1 + \Gamma)\Gamma M \\ &+ U_j^n(1 - \Gamma^2)2M \\ &+ U_{j+1}^n(-1 + \Gamma)\Gamma M \end{aligned} \tag{3.31}$$

where M is defined as,

$$M = \frac{1}{2(1 + D/2)(1 + D)} . \tag{3.32}$$

With the U's as defined by equations (3.8) we can apply the Von Neuman stability analysis procedure to equation (3.30) to give the following expression for G,

$$G = M\ [(2\Gamma^2\cos\phi + 2(1 - \Gamma^2))^2 + 4\Gamma^2(\sin\phi)^2]^{1/2}\ . \tag{3.33}$$

Figure 3-3 shows a contour plot of equation (3.33) plotted for the variables Γ and D. The plot shows that the numerical time step, Δt, is given by $\Gamma < 1 + D/2.2$. Thus, the numerical time steps taken by the point implicit MacCormack scheme are favorably dependent of τ_{chem}. Finally it will be shown in chapter 6 that with this method the number of iterations needed to reach steady state is independent of the stiffness parameter.

The stability limits of the point implicit Jameson, Schmidt and Turkel [22] scheme will now be given. Let us again begin by considering a single stage of the scheme. Differencing the model equation using forward time centered space with the chemical source terms treated implicitly yields,

$$U_j^{n+1} = U_j^n - \frac{\Gamma}{2}(U_{j+1}^n - U_{j-1}^n) - DU_j^{n+1} \tag{3.34}$$

where Γ and D are given by equations (3.10) and (3.11). A Von Neumann stability analysis of this equation leads to,

$$V^{k+1} = (1 + z)V^k \tag{3.35}$$

with z given by,

$$z = -\frac{\Gamma i \sin(\phi)}{(1 + D)} \tag{3.36}$$

where ϕ is a phase angle. As was discussed previously the multistage scheme can be expressed in a similar manner,

$$V^{n+1} = g(z)V^n \tag{3.37}$$

where g(z) is given by equation (3.23), must have a magnitude equal to or less than 1 for the scheme to be stable. Figure 3-4 shows the contour plot of g(z) for various values of Γ and D's. The plot shows that the scheme is stable for $\Gamma < 2\sqrt{2} + 2D$. In this case, the numerical time step is dependent upon τ_{fluid} and favorably dependent of τ_{chem}. This scheme along with the point implicit MacCormack scheme offer large advantages compared to fully explicit schemes for stiff problems. Note, unlike fully implicit methods, point implicit methods do not require the inversion of large matricies. In addition, most of the currently used explicit methods can be easily modified to point implicit schemes.

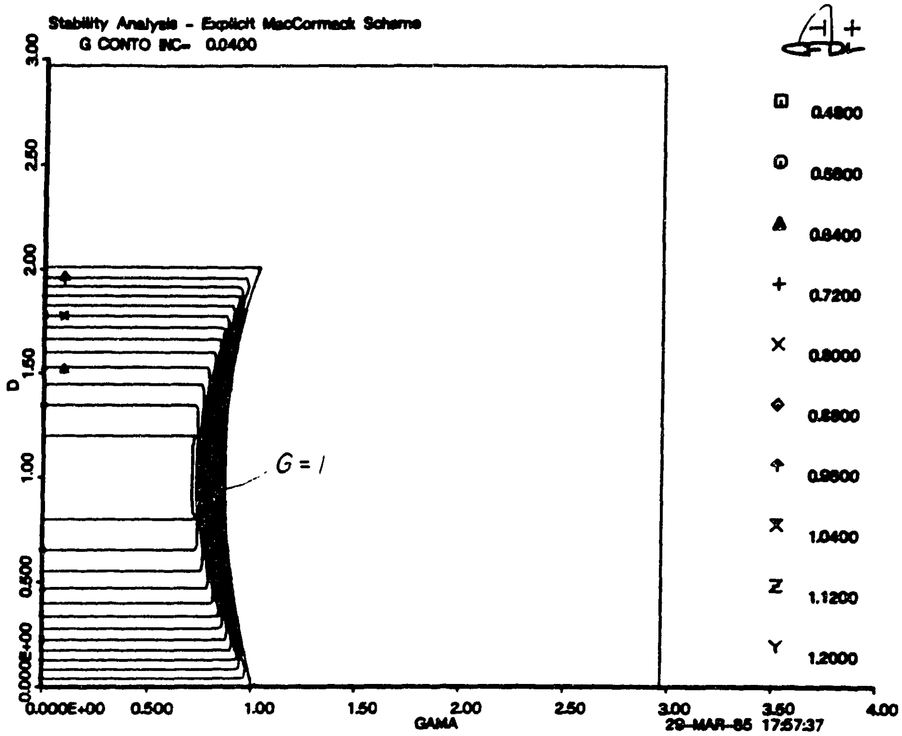


Figure 3-1: Stability Contours for the Explicit MacCormack Scheme

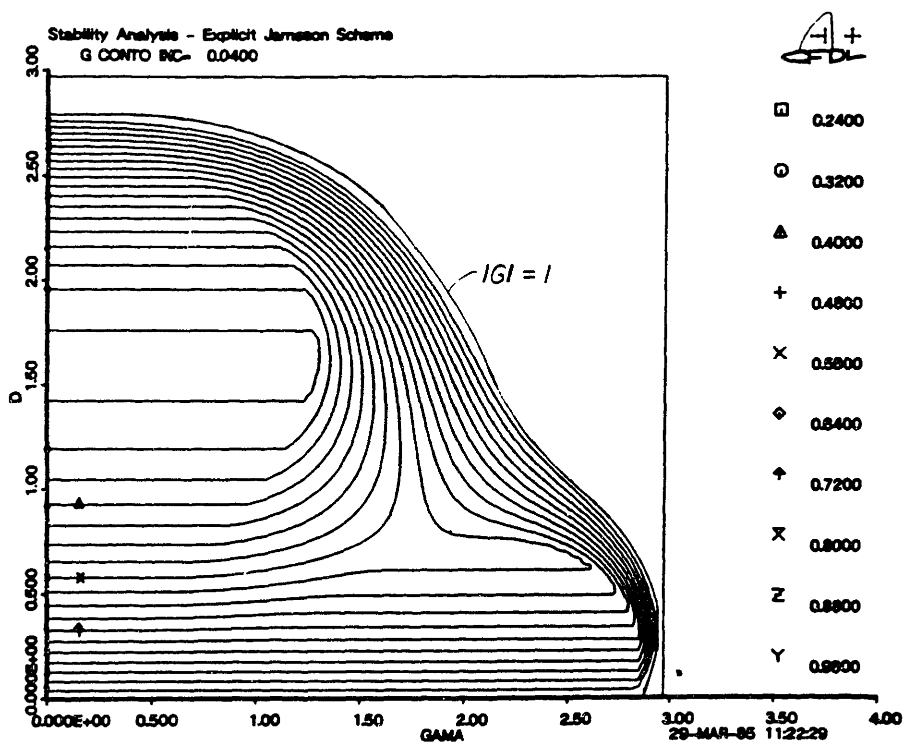


Figure 3-2: Stability Contours for the Explicit Jameson, Schmidt and Turkel Scheme

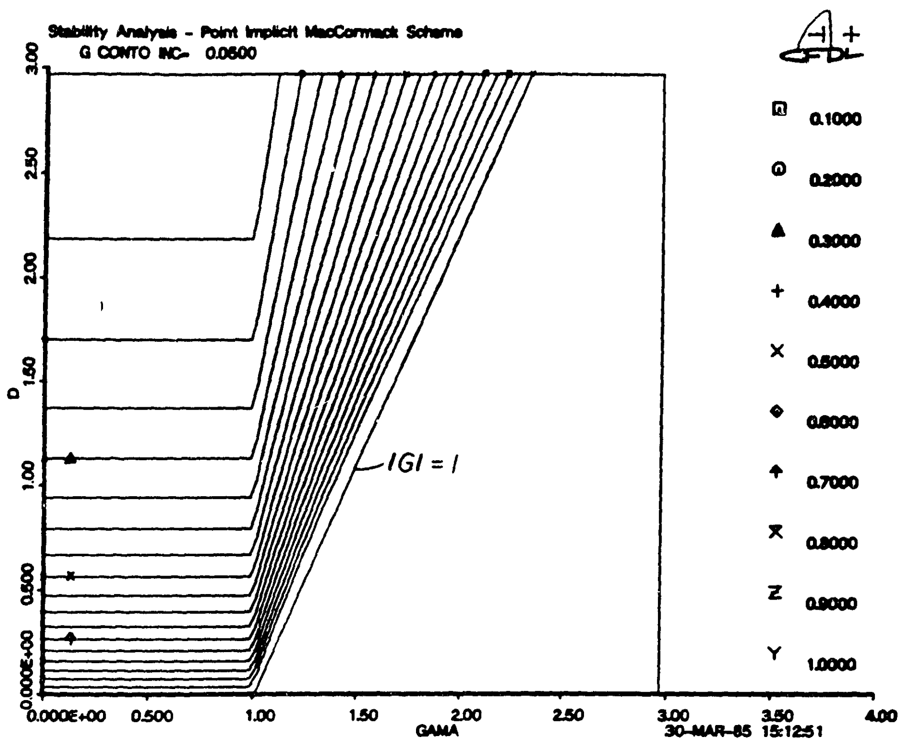


Figure 3-3: Stability Contours for the Point Implicit MacCormack Scheme

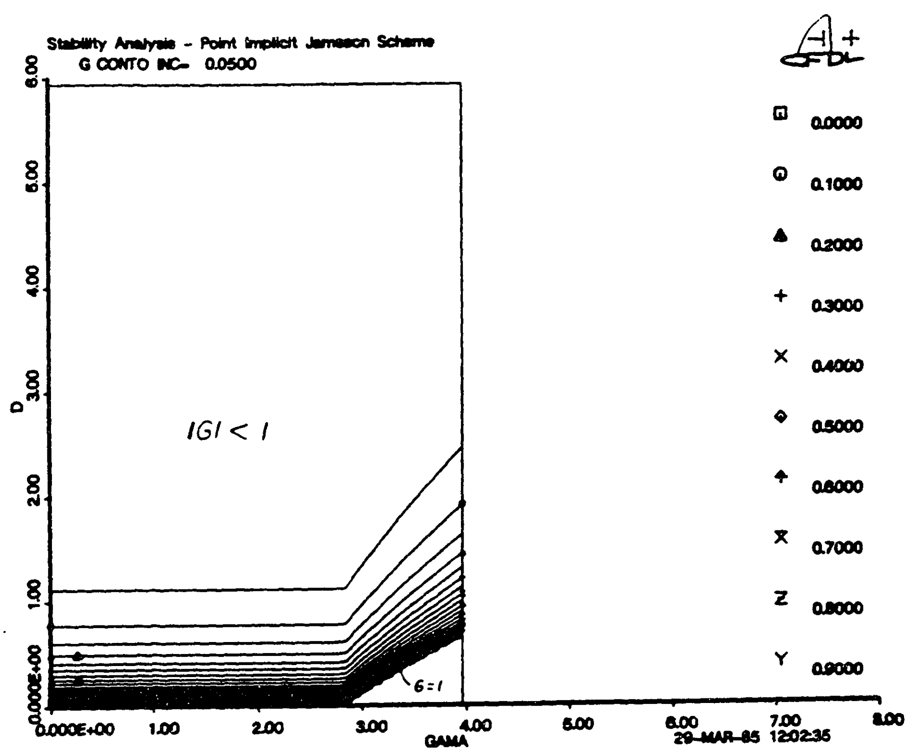


Figure 3-4: Stability Contours for the Point Implicit Jameson, Schmidt and Turkel Scheme

Chapter 4
NUMERICAL METHODS

The integration of equations (2.39), (2.41) or (2.43) can be performed by any of the techniques described in the previous section. The emphasis of this chapter will be the evaluation of the convective fluxes, the diffusive fluxes and the boundary conditions. In this chapter we will also consider several artificial viscosity models that can be used to stabilize and smooth the solution. Two basic numerical methods will be described in this chapter. The techniques considered are the Jameson, Schmidt and Turkel and the MacCormack finite volume schemes. The Jameson, Schmidt and Turkel method is used for all of the 2-D calculations and will therefore be described in considerable detail. The MacCormack scheme is used only in 1-D to test the numerical acceleration ideas and thus it will only be briefly outlined in this thesis. The MacCormack scheme was used for the 1-D calculations since a large part of the code was available from the author's previous work [5]. The Jameson, Schmidt and Turkel method was used for the 2-D Euler and Navier-Stokes equations for two reasons. First, little work had been done with this method in connection with the Navier-Stokes equations and it was felt that this was a new area requiring exploration. Second, the coding of the viscous stress terms is considerably simpler than for the MacCormack method permitting more flexible coding. The techniques to be described here for solving chemically reacting flows apply equally well to both methods.

4.1. Jameson, Schmidt and Turkel Scheme - 1981

In 1981 Jameson, Schmidt and Turkel proposed a new finite volume time stepping scheme for the unsteady Euler equations [22]. For simplicity the scheme will be referred to as the Jameson scheme. The scheme is a modified version of the classical fourth-order Runge-Kutta method for ordinary differential equations. The method requires less computer storage of array quantities than the older Runge-Kutta scheme, a factor particularly important for three-dimensional calculations. The method is fourth-order accurate in time for linear equations and second-order accurate for nonlinear equations [52]. The scheme is second-order accurate in space for both linear and nonlinear problems provided the grid is sufficiently smooth.

The Jameson scheme follows the finite volume formulation discussed briefly in chapter 2. The governing equations written in integral form for a cartesian system (Eguation (2.54)) are,

$$\frac{\partial}{\partial t}\iint_{\Omega} U dxdy + \int_{\partial\Omega}(Fdy - Gdx) + \iint_{\Omega} H dxdy = 0 \quad . \tag{4.1}$$

The computational domain is divided up into quadrilaterals over which equation (4.1) is solved (figure 4-1).

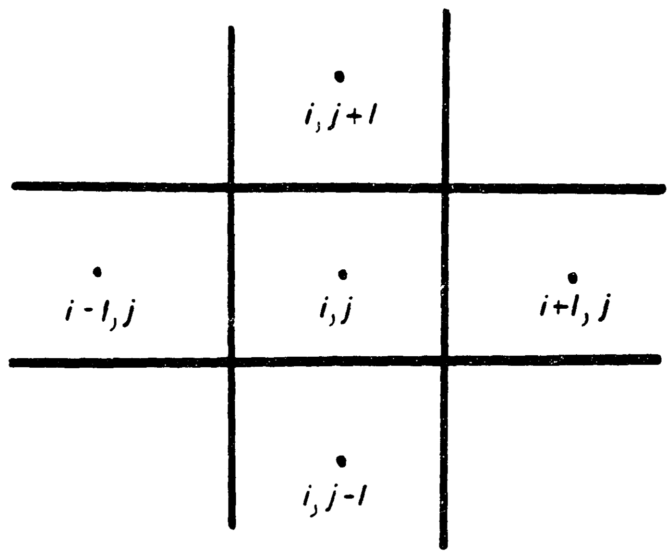


Figure 4-1: Typical Cell Nomenclature

The procedure is equivalent to performing a mass, momentum, energy and species balance for each cell. The Jameson method decouples the temporal and spatial terms producing a system of ordinary differential equations. These equations can be written as

$$\frac{d}{dt}(S_{ij}U_{ij}) + LU_{ij} + S_{ij}H_{ij} = 0 \tag{4.2}$$

where L is a spatial discretization operator, and i and j are the cell indices. The source term, H_{ij}, is evaluated based on properties from within a cell ij and S_{ij} is the area of cell ij. In the Jameson scheme these equations are solved by a multistage time stepping scheme. The details of the time stepping scheme, the spatial difference operator and the boundary cell formulations will be given in the following subsections.

4.1.1. Time Stepping Integrator

The basic Jameson scheme consists of four steps which take the following form at time level n:

$$
\begin{aligned}
U^{(0)} &= U^{n} \\
U^{(1)} &= U^{(0)} - a_1 \Delta t R U^{(0)} \\
U^{(2)} &= U^{(0)} - a_2 \Delta t R U^{(1)} \\
U^{(3)} &= U^{(0)} - a_3 \Delta t R U^{(2)} \\
U^{(4)} &= U^{(0)} - a_4 \Delta t R U^{(3)} \\
U^{(\times)} &= U^{(4)}
\end{aligned}
\tag{4.3}
$$

where

$$
RU^{(q)} = \frac{1}{S} LU^{(q)} + H^{(q)}
\tag{4.4}
$$

and

$$
a_1 = 1/4, \quad a_2 = 1/3, \quad a_3 = 1/2, \quad a_4 = 1 \ .
$$

The quantity L is a spatial discretization operator, S_{ij} is the cell area, Δt is the numerical time step and U_{ij} is the cell averaged state vector.

4.1.2. Smoothing

Jameson's method like most finite difference methods for hyperbolic equations must be smoothed. Smoothing serves two purposes: one to damp out Gibb's phenomena at shocks and two, to provide proper domain of dependence for central difference schemes. Smoothing or numerical damping is therefore essential to achieving converged solutions. At least three methods can be used to smooth the discrete equations (equation (4.3)). The methods can be grouped as co-smoothing, post smoothing and post split smoothing. Each of these methods has received considerable attention and will be described below.

Co-smoothing is probably the most widely used type of smoothing implemented within multistage schemes. Following Jameson [22], equation (4.2) can be rewritten to include smoothing (co-smoothing) as

$$\frac{d}{dt}(S_{ij}U_{ij}) + LU_{ij} - DU_{ij} + S_{ij}H_{ij} = 0 \tag{4.5}$$

where D is an artificial dissipation operator. The term co-smoothing means that numerical dissipation is added on each stage of the multistage scheme. Jameson [22] found that an effective dissipation operator DU could be constructed in the following way:

$$DU = D_xU + D_yU \tag{4.6}$$

where D_xU and D_yU are the contributions from the two coordinate directions and are given by

$$D_xU = d_{i+1/2,j} - d_{i-1/2,j}$$
$$D_yU = d_{i,j+1/2} - d_{i,j-1/2} \quad . \tag{4.7}$$

A typical d on the right hand side of equation (4.7) can be written as, for example,

(4.8)

$$d_{i+1/2,j} = \frac{S_{i+1/2,j}}{\Delta t}[\epsilon^{(2)}_{i+1/2,j}(U_{i+1,j} - U_{i,j})$$

$$- \epsilon^{(4)}_{i+1/2,j}(U_{i+2,j} - 3U_{i+1,j}$$

$$+ 3U_{i,j} - U_{i-1,j})] \quad .$$

The coefficients $\epsilon^{(2)}$ and $\epsilon^{(4)}$ are dependent on the local flow properties and were defined by Jameson as folows:

(4.9)

$$\epsilon^{(2)}_{i+1/2,j} = \kappa^{(2)}\ max(\nu_{i+1,j}\ ,\ \nu_{i,j})\ ,$$

$$\epsilon^{(4)}_{i+1/2,j} = max(0\ ,\ (\kappa^{(4)} - \nu^{(2)}_{i+1/2,j}))\ ,$$

$$\nu_{i,j} = \frac{\Delta P_1}{\pi_2}\ ,$$

$$\Delta P_1 = |\ P_{i+1,j} - 2P_{i,j} + P_{i-1,j}\ |\ ,$$

$$\pi_2 = P_{i+1,j} + 2P_{i,j} + P_{i-1,j} \quad .$$

This is a blending of second and fourth order smoothing with a pressure switch. Through numerical experimentation Jameson found that good choices for $\epsilon^{(2)}$ and $\epsilon^{(4)}$ are 1/4 and 1/256 respectively.

The second class of smoothing methods are the so called post unsplit smoothing schemes. These methods work by operating on the updated state vector computed from a full time stepping cycle, equation (4.3). This post operation filters out unwanted harmonics to produce a smooth solution. The procedure can be written as two additional steps of the multistage time stepping scheme, ie,

(4.10)

$$\Delta U^1 = \sigma D_{xx} U^{\times}$$

$$\Delta U^2 = \sigma D_{yy} U^{\times}$$

where U at time level n+1 is.

$$U^{n+1} = U^{\times} + \Delta U^{1} + \Delta U^{2} \quad . \tag{4.11}$$

$D_{xx}U^{*}$ and $D_{yy}U^{*}$ can be evaluated as follows,

$$D_{xx}U^{\times} = (U_{i+1,j} - 2U_{i,j} + U_{i-1,j})^{\times}$$

$$D_{yy}U^{\times} = (U_{i,j+1} - 2U_{i,j} + U_{i,j-1})^{\times} \tag{4.12}$$

which are second order undivided central differences. The coefficient σ is typically within the range of .05 - .1 .

The last class of smoothing methods are the post split smoothing schemes. Like the post smoothing scheme discussed above this method operates on the state vector after a full multistage cycle. The smoothing operation can be written as,

$$U^{n+1} = (I + \sigma D_{xx})(I + \sigma D_{yy})U^{\times} \tag{4.13}$$

where D_{xx} and D_{yy} are given by equation (4.12). The coefficient σ also falls within the range .05 to .1. Note the blended second and fourth differences method described for the co-smoothing approach could be tried with either the post un-split or post split smoothing methods.

Several researchers have investigated these different approaches. Tong [54] found that all three methods performed well. Powell's [39] investigation showed that for values of σ greater than .2, the post split smoothing method becomes unstable. In the current study all three methods were tried and were all found to produce similar results. Some preliminary investigations were done to assess the various smoothing methods. Little difference was noted between the various methods for $\sigma < 1$. Based upon these investigations it was decided to use the post unsplit and post split methods. Unless otherwise stated the post unsplit smoothing method will be used in the remainder of this thesis. Note, care must be taken to ensure that the numerical smoothing terms do not contaminate the solution above some acceptable level. This point will be addressed in chapter 8 in connection with the real viscous diffusion terms where potential problems can be anticipated.

4.1.3. Evaluation of the Convective Fluxes

The convective part of the spatial discretization operator, L, mentioned above will now be detailed. For a given cell , figure 4-2,

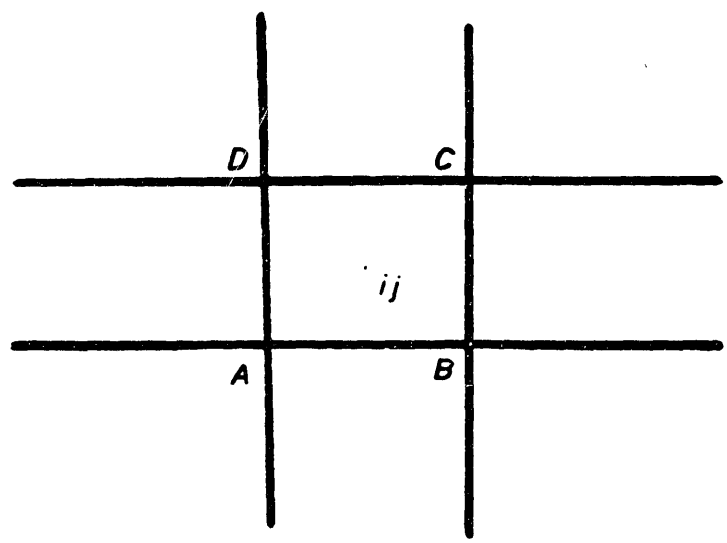


Figure 4-2: Convective Cell Nomenclature

the convective part of the operator $(LU_{i,j})_{convective}$ can be written as,

$$(LU_{ij})_{convective} = (M_{AB} + M_{BC} + M_{CD} + M_{DA})_{convective} \tag{4.14}$$

where the vectors M_{AB}, M_{BC}, M_{CD} and M_{DA} are the fluxes through the sides of the cell. For example the flux M_{BC} can be expressed as,

$$M_{BC} = F_{BC}\Delta y_{BC} - G_{BC}\Delta x_{CB} \tag{4.15}$$

where the quantities F_{BC} and G_{BC} are the mean values of F and G on the cell side BC and the metrics Δx_{BC} and Δy_{BC} are given by

$$\Delta x_{BC} = x_C - x_B$$
$$\Delta y_{BC} = y_C - y_B \quad . \tag{4.16}$$

The convective quantities on the cell faces are found by averaging the value of the quantity in adjacent cells. Mathematically we can express this process as follows (see figure 4-1),

$$\phi_{i,j+1/2} = \frac{1}{2}(\phi_{i,j} + \phi_{i,j+1}) \tag{4.17}$$

where ϕ can represent any cell face quantity. Typically ϕ is set equal to the flux quantities F and G, i.e. the flux quantities are averaged across the cell faces. The same averaging procedure can be used to evaluate the convective quantities on the other three cell faces.

4.1.4. Evaluation of the Viscous Stress Terms

To complete the evaluation of the cell fluxes we need to determine the viscous stresses on each face. We can write the viscous contribution of the spatial discretization operator LU_{ij} as

$$(LU_{ij})_{viscous} = (M_{AB} + M_{BC} + M_{CD} + M_{DA})_{viscous} \quad . \tag{4.18}$$

where the vectors M_{AB}, M_{BC}, M_{CD} and M_{DA} are the fluxes through the sides of the cell. The flux M_{BC} is given by equation (4.15). In order to evaluate the viscous parts of F and G we need to find a way to express these quantities on the cell face i+1/2,j. We can define a typical stress variable as $\partial\phi/\partial\zeta$ where ϕ could be u, v, T or Y_k and ζ is a spatial variable, i.e., x or y. The quantity can be computed by averaging it over an area centered about the cell face where the stress term is being evaluated. For example if we want $\partial\phi/\partial\zeta$ with ϕ = u and ζ = y at the point i+1/2,j then ,

$$\left(\frac{\partial u}{\partial y}\right)_{i+1/2,j} = \frac{1}{S^{\times}}\iint \frac{\partial u}{\partial y}dS \tag{4.19}$$

where S* is the ghost cell area centered about the point i+1/2,j (see figure 4-3).

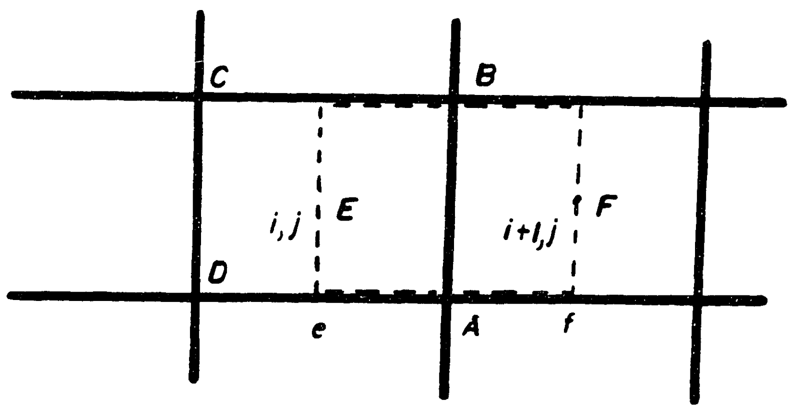


Figure 4-3: Ghost Cell Viscous Stress Term Nomenclature

The ghost cell area S^* for the point i+1/2,j is equal to the average of cell areas $S_{1,j}$ and $S_{1,j+1}$. Now using Green's theorem the area integral in equation (4.19) can be rewritten as,

$$\left(\frac{\partial u}{\partial y}\right)_{i+1/2,j} = -\frac{1}{S^{\times}}\int u dx \tag{4.20}$$

which, for the notation given in figure 4-3 can be differenced to give

$$\left(\frac{\partial u}{\partial y}\right)_{i+1/2,j} = -\frac{1}{S^{\times}}(u_A \Delta x_A + u_F \Delta x_F - u_B \Delta x_B - u_E \Delta x_E) \quad . \tag{4.21}$$

The quantities u_A, u_F, u_B and u_E are the values of u at the points A, F, B and E. The term Δx_A can be written as,

$$\Delta x_A = x(f) - x(e) \quad . \tag{4.22}$$

Similar expressions can be written for Δx_F, Δx_B and Δx_E. The viscosity coefficient on face i+1/2,j can be found by averaging it over the two bounding cells i.e.,

$$\mu_{i+1/2,j} = \frac{\mu_{i,j} + \mu_{i+1,j}}{2} \quad . \tag{4.23}$$

In a similar way expressions for the remaining viscous flux quantities and coefficients can be written as well as the viscous fluxes on the other three faces. This approach for determining the viscous fluxes has also been used successively by Swanson and Turkel, reference [52]. Peyret's book [38] on computational fluid dynamics also describes this approach.

4.1.5. Boundary Cell Formulations

In the previous sections we considered only the interior cells. In this section we will consider how the different boundary cells are evaluated. At least four different boundary cells can be written for the problems of interest. These include supersonic inflow and outflow, a slip wall and a non-slip wall boundary condition. In addition at a wall the temperature and species concentrations can be set equal to a specific value or their respective normal derivatives set to specific values.

Figure 4-4 shows a typical inflow boundary cell. To determine the inflow quantities we need to construct the flux vector on face 4. To do this we need only base the face 4 flux quantities on the free stream conditions. For example F_1 on face 4 is computed as follows:

$$(F_1)_{face4} = (\rho u)_{freestream} \quad . \tag{4.24}$$

The supersonic outflow boundary conditions can be implemented as follows.

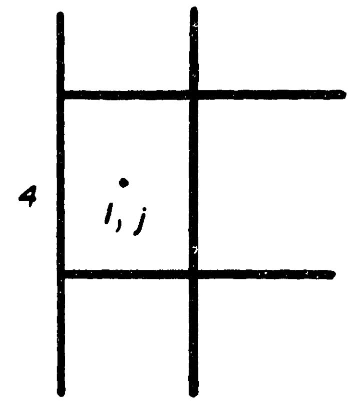


Figure 4-4: Inflow Boundary Nomenclature

From characteristic theory the supersonic outflow boundary quantities on face 2, see figure 4-5, are extrapolated from the interior points. For viscous flows it is assumed that we are far enough downstream so that the streamwise and crossflow diffusion terms are small compared to the convective terms and can be neglected in the supersonic part of the flow. The same exit boundary conditions are applied near the wall.

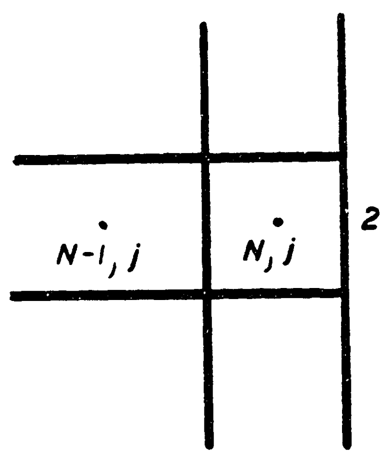


Figure 4-5: Outflow Boundary Nomenclature

As an example, the flux quantity, F_1 on face 2 can be computed by first order extrapolation from the interior points as;

$$(F_1)_{face2} = (\rho u)_{N-1,j} \quad . \tag{4.25}$$

With the flux quantities known on face 2, a standard flux balance is performed on cell N,j. The slip wall boundary conditions apply to inviscid flows. Specifically, for these flows the normal wall velocity is equal to zero, see figure 4-6.

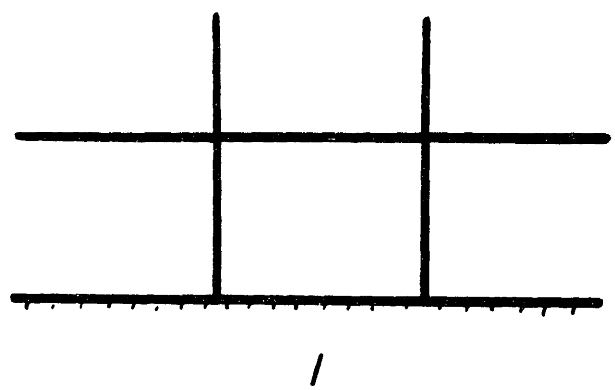


Figure 4-6: Slip Wall Boundary Condition Nomenclature

The zero wall flux condition implies that,

$$V_n \, \Delta S = 0 \tag{4.26}$$

where V_n is the normal wall velocity and ΔS is the wall cell surface area. This equation can be rewritten in Cartesian coordinates as follows,

$$V_n \, \Delta S = u \Delta y - v \Delta x \quad . \tag{4.27}$$

Thus for inviscid flows the net flux through face 1 are

$$\int_1 (F_1 dy - G_1 dx) = 0$$

$$\int_1 (F_2 dy - G_2 dx) = P_1 \Delta y_1$$

$$\int_1 (F_3 dy - G_3 dx) = - P_1 \Delta x_1 \qquad (4.28)$$

$$\int_1 (F_4 dy - G_4 dx) = 0$$

$$\vdots$$

$$\int_1 (F_8 dy - G_8 dx) = 0$$

where the various quantities are defined by equations (4.15) and (4.16). The quantities Δx_1 and Δy_1 are the projections of the wall on the cartesian axis. For a locally normal system the wall pressure can be computed with the following expressions

$$P_1 = P_n + \left(\frac{\partial P}{\partial n}\right) \Delta n \qquad (4.29)$$

$$\left(\frac{\partial P}{\partial n}\right) = - \frac{1}{\rho_n RC}\left((\rho u)_n^2 + (\rho v)_n^2\right)$$

where RC is the radius of curvature of the wall. P_n, ρ_n, $(\rho u)_n$ and $(\rho v)_n$ are evaluated at a distance of Δn normal to the wall. Figure 4-7 shows the details of the wall pressure evaluation.

Finally consider the no slip wall boundary conditions. First, break the wall fluxes into their inviscid and viscous parts. The inviscid flux terms through face 1 are the same as those given by equation (4.28). For viscous problems we have an additional wall boundary condition, i.e. that the velocity component tangent to the wall must also equal zero. As a result of this additional constraint the viscous flux vector is non zero. The wall shear stress quantities, i.e., the viscous wall fluxes, can be determined quite simply. The basic wall cell nomenclature is shown in figure 4-8. For example the derivative $\partial u/\partial y$ can be computed on face 2 as (figure 4-9),

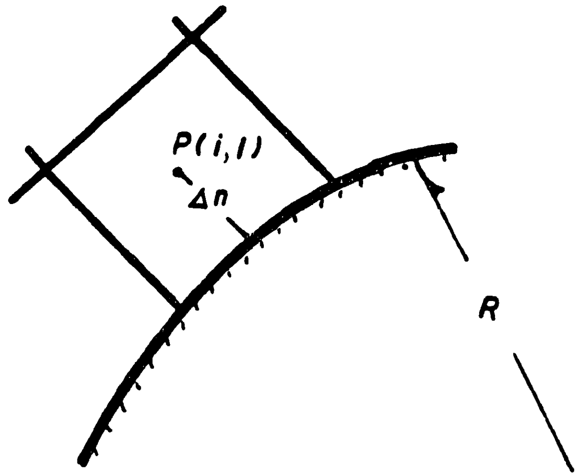


Figure 4-7: Wall Pressure Evaluation Nomenclature

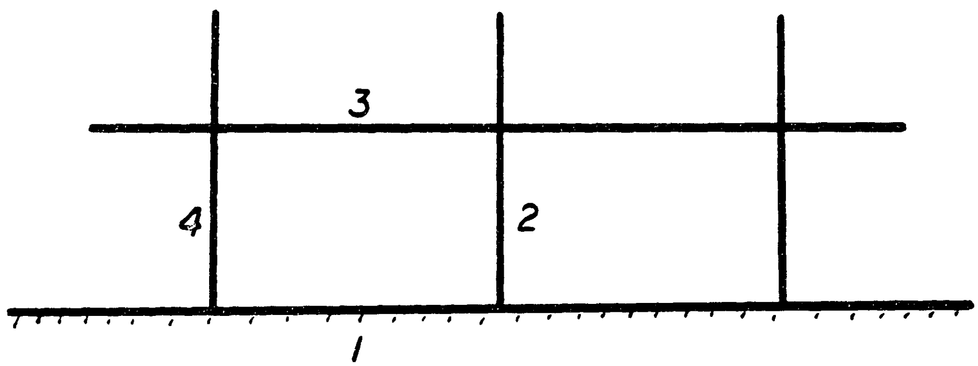


Figure 4-8: Basic Boundary Cell Nomencleature

$$\frac{\partial u}{\partial y} = -\frac{u_B \Delta x_B - u_C \Delta x_C - u_D \Delta x_D}{S_2} \quad . \tag{4.30}$$

The wall temperature derivative $\partial T/\partial y$ on face 2 can be computed from the following,

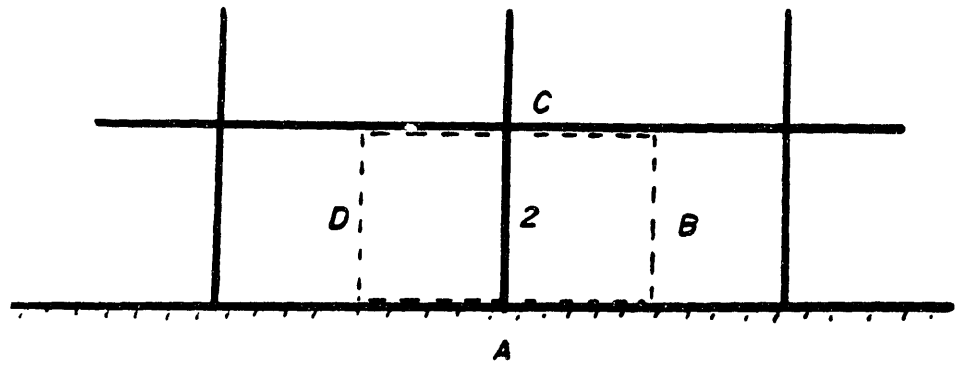


Figure 4-9: Face 2 Boundary Cell Nomencleature

$$\frac{\partial T}{\partial y} = -\frac{T_W \Delta x_A + T_B \Delta x_B - T_C \Delta x_C - T_D \Delta x_D}{S_2} \qquad (4.31)$$

where $T_W = .5\ (T(i,j) + T(i+1,j))$ for an adiabatic wall and $T_W = T_{specified}$ for a specified wall temperature. These quantities can be computed in a like manner for face 4. For face 1, figure 4-10, the u and T derivatives with respect to y are given by,

$$\frac{\partial u}{\partial y} = -\frac{-u_C \Delta x_A - u_C \Delta x_C}{S_1} \qquad (4.32)$$

and

$$\frac{\partial T}{\partial y} = -\frac{T_A \Delta x_A + T_B \Delta x_B - T_{i,j} \Delta x_C - T_D \Delta x_D}{S_1} \qquad (4.33)$$

where T_A, T_B and $T_D = T(i,j)$ for an adiabatic wall. If the wall temperature is specified then T_B and $T_D = T_{specified}$ and

$$T_A = T_{i,j} + (T_{specified} - T_{i,j}) \times 2 \ . \tag{4.34}$$

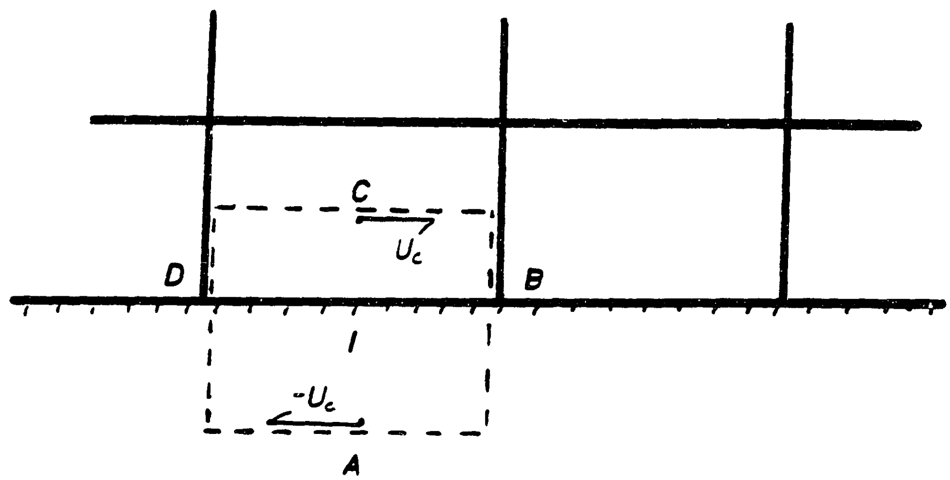


Figure 4-10: Face 1 Boundary Cell nomencleature

Similar expressions can be written for the other viscous quantities. The viscous stress terms can then be computed with equations (4.18) and (4.15).

4.1.6. Stability Analysis

In chapter 3 we considered the stability of a number on numerical methods. In this section the stability of the Jameson scheme applied to the linear 2-D inviscid and the 1-D viscous model equations will be assessed. In chapter 3 we found that the amplification factor, G, for the Jameson scheme can be written as,

$$G = 1 + z + 1/2z^2 + 1/6z^3 + 1/24z^4 \tag{4.35}$$

where z is the amplification factor of a single stage of the Jameson scheme.

The stability of the 2-D problem can be assessed by analyzing the model equation,

$$U_t = -a(U_x + U_y) \tag{4.36}$$

where "a" is a characteristic convection velocity. Performing a Von Neumann stability analysis we find that z equals

$$z = -i(\Gamma_x sin(\theta_x) + \Gamma_y sin(\theta_y)) \tag{4.37}$$

where $\Gamma_x = a\Delta t/\Delta x$, $\Gamma_y = a\Delta t/\Delta y$ and $i = \sqrt{-1}$. θ_x and θ_y are the phase angles in the x and y directions. Substituting equation (4.37) into equation (4.35) shows that for scheme to be stable , $\Gamma_x+\Gamma_y<2\sqrt{2}$, which is straight line on a Γ_x vs Γ_y plot. Thus Δt must satisfy the following equation,

$$\Delta t < \frac{2\sqrt{2}}{(a/\Delta x + a/\Delta y)} \quad . \tag{4.38}$$

The 1-D viscous model equation can be written as,

$$U_t = -aU_x + \mu' U_{yy} \tag{4.39}$$

where μ' is a viscosity coefficient. A Von Neumann stability yields,

$$z = -4Q sin^2(\theta/2) - i\Gamma sin\theta \tag{4.40}$$

with $Q = \mu'\Delta t/\Delta y^2$ and $\Gamma = a\Delta t/\Delta x$. Combining equations (4.40) and (4.35) and plotting, figure 4-11, tells us that Δt can be approximated with the following equation

$$\Delta t < \frac{1}{[\mu' 2/(\sqrt{2}\Delta y^2) + a/(2\sqrt{2}\Delta x)]} \quad . \tag{4.41}$$

This equation can be rewritten in terms of cell Reynolds number, $Re_c=a\Delta y/\mu'$, as follows;

$$\Delta t < \frac{2\sqrt{2}\Delta x}{a} \frac{1}{4(\Delta x/\Delta y) \;/\; Re_c \;+\; 1} \quad . \tag{4.42}$$

Thus the numerical time step, Δt, is affected by viscosity when $4(\Delta x/\Delta y)/Re_c$ is of order 1 or greater. Note that a, the characteristic velocity, is always positive.

4.2. MacCormack Scheme - 1969

In 1969 MacCormack [32] proposed a fully explicit numerical integration scheme. Over the past 14 years this method has been used extensively [45, 13, 57] and gained a reputation for being simple and robust. In this thesis the MacCormack scheme is applied to the quasi 1-D Euler equations to test various numerical acceleration techniques.

The inviscid finite volume formulation of the MacCormack scheme can be written in the following way. If the equations are written in integral form, equation (2.54), we can discretize these equations to get

$$\frac{d}{dt}(US)_{i,j} + \Sigma_{K=1}^{4} M_K \cdot \mathbf{n}_K S_K + HS_{ij} = 0 \quad . \tag{4.43}$$

The predictor step of the MacCormack scheme can be expressed as,

$$\begin{aligned} U^{\times}_{i,j} = U^{n}_{i,j} - \Delta t/S_{i,j}[& (M^{n}_{i,j} \cdot \mathbf{n}_2 S_2 + M^{n}_{i-1,j} \cdot \mathbf{n}_4 S_4) \\ & + (M^{n}_{i,j} \cdot \mathbf{n}_3 S_3 + M^{n}_{i,j-1} \cdot \mathbf{n}_1 S_1) \\ & + S_{i,j} H^{n}_{i,j}] \end{aligned} \tag{4.44}$$

and the corrector step written as,

$$\tilde{U}^{n+1}_{i,j} = 1/2[U^{n}_{i,j} + U^{\times}_{i,j} - \Delta t/S_{i,j}((M^{\times}_{i+1,j}\cdot\mathbf{n}_2 S_2 + M^{\times}_{i,j}\cdot\mathbf{n}_4 S_4) + (M^{n}_{i,j+1}\cdot\mathbf{n}_3 S_3 + M^{\times}_{i,j}\cdot\mathbf{n}_1 S_1)) + S_{i,j}H^{\times}_{i,j}] \quad . \tag{4.45}$$

If the dot products are expanded, the elements of these equations can be found. For example we can write the first element of the right hand side of equation (4.44)

$$M^{n}_{i,j}\cdot\mathbf{n}_2 S_2 = F^{n}_{i,j}\Delta y_2 - G^{n}_{i,j}\Delta x_2 \tag{4.46}$$

where similar expressions can be written for the other product terms.

The evaluation of the convective fluxes and the boundary fluxes are the same as those given above for the Jameson scheme. A stability analysis of the 2-D model equation (no chemical source term), equation (4.36), using the MacCormack method leads to the following numerical time step restriction:

$$\Delta t \leq \frac{1}{(a/\Delta x + a/\Delta y)} \quad . \tag{4.47}$$

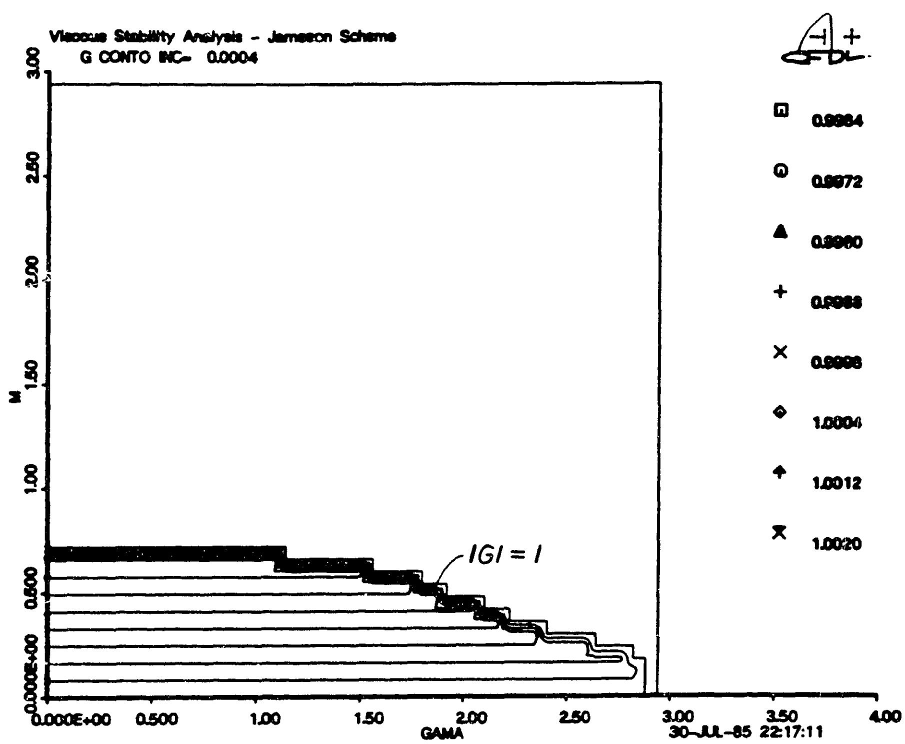


Figure 4-11: Stability Analysis of The 1 - D Convection Diffusion Equation - Jameson, Schmidt and Turkel Scheme

Chapter 5
ACCELERATION TECHNIQUES

A variety of techniques have been developed to accelerate the unsteady equations to steady state. These techniques include the constant CFL condition, residual smoothing, enthalpy damping, multigridding and a chemical time scale preconditioner. The first three of these techniques were discussed by Jameson, Schmidt and Turkel [22] in connection with the Euler equations and by Swanson and Turkel [52] in connection with the Navier Stokes equations. In this chapter we shall consider a chemical time scale preconditioner, the constant CFL condition and multigridding as ways of accelerating the chemically reacting transport equations to steady state.

5.1. Chemical Time Scale Preconditioning

In chapter 3 it was shown that purely explicit schemes suffer from severe time step restrictions if the stiffness is large. In this section a technique will be described which removes the stiffness restriction. If only the steady state solution is desired then the number of iterations required by the technique to achieve convergence is independent of the level of stiffness. This characteristic is particularly important when the stiffness level in a given problem varies by several orders of magnitude. The technique is not time accurate if numerical time steps are taken which are larger than the transient time scales characteristic of the problem. However, if the time steps are chosen to be of the order of the characteristic time scales of interest, the method is time accurate.

5.1.1. Heuristic Description of Time Scaling

If only the steady state solution is desired then the time history can be modified to remove the stiffness associated with the chemical time scales. To see how this can be done consider the following . Figure 5-1 outlines the paths taken by a typical fluid and species quantity with real time. The figure shows that the species quantity under goes a rapid change in its density fraction while the fluid quantity evolves much more slowly. It is the great disparity in the slopes of the two curves which is

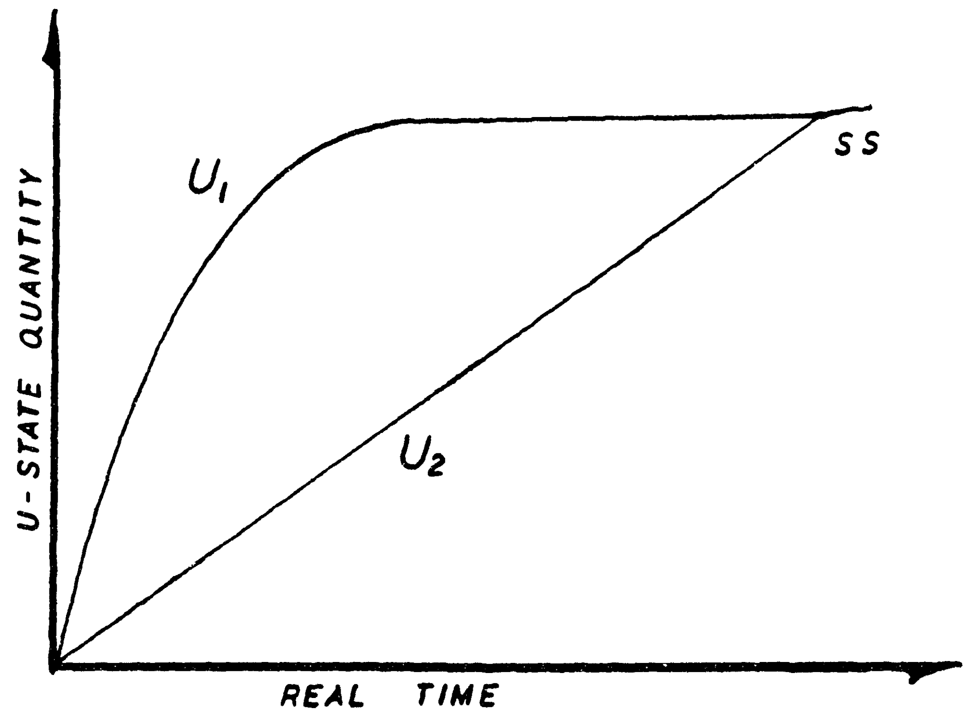


Figure 5-1: Real Time Behavior Of A Typical Fluid And Species Quantity

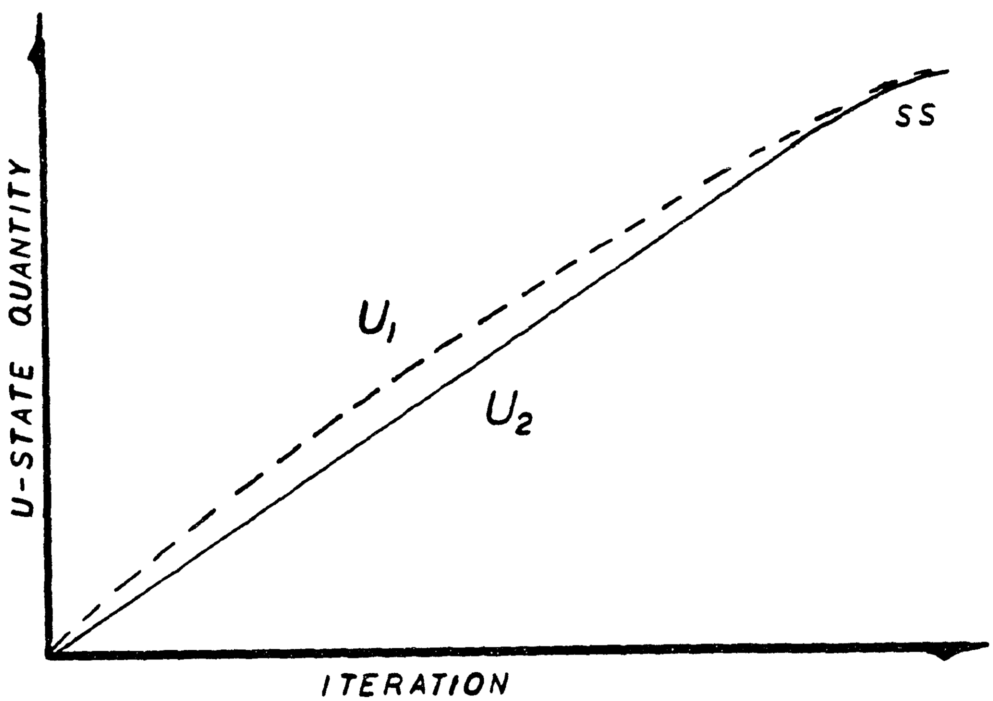


Figure 5-2: Pseudo Time Behavior Of A Typical Fluid And Species Quantity

responsible for the stiffness in the problem. If, as shown in figure 5-2, the two quantities could be marched together in pseudo time, then the fast processes which require small time steps would not hold up the slower processes which could be marched at larger time steps. It turns out that the governing equations,

$$\frac{\partial U}{\partial t} = - \frac{\partial F}{\partial x} - \frac{\partial G}{\partial y} - H \tag{5.1}$$

can be modified to reflect this desired pseudo time behavior by rewritting them as,

$$\mathbf{S} \frac{\partial U}{\partial t} = - \frac{\partial F}{\partial x} - \frac{\partial G}{\partial y} - H \tag{5.2}$$

where **S** is a preconditioning matrix whose purpose is to normalize the various time scales to be of the same order. The "pseudo time" history of the state quantities given by equation (5.2) might be very different from those given by equation (5.1), but both satisfy the same steady state equation. Note if the problem is stiff and **S** is chosen to remove this stiffness then the technique is no longer time accurate. If **S** is constructed correctly the chemical time scales can be made approximately equal to the fluid time scales and the chemical stiffness can be removed from the problem. The scaling matrix could have a variety of forms, i.e., diagonal, triangular, full, etc. In the next subsection we will consider what the elements of the scaling matrix should look like and how they can be constructed.

The method can be made time accurate by choosing numerical time steps small enough to reduce the error at any time step to some acceptable level. One way to do this is to set the numerical time step to some fraction of the time scale of the transient process of interest. The method can also be made time accurate by simply choosing **S** to be the identity matrix. It turns out that the numerical method can be coded to handle both time accurate or pseudo time calculations with only minor changes to the code structure.

5.1.2. Derivation of the Scaling Matrix

As mentioned above the S matrix can take on a variety of forms. To determine the form of the S matrix and specifically what the matrix elements should contain, consider the following. If the time stiffness is to be removed from the problem then the matrix S should in some way contain the chemical time scales. The desired time scale character can be seen by considering the species equation without the convective term, i.e.,

$$\frac{dU_Y}{dt} = -H = -\frac{kU_Y}{\rho} . \tag{5.3}$$

Integrating this equation yields,

$$U_Y = Ce^{-kt/\rho} = Ce^{-t/\tau_{chem}} \tag{5.4}$$

where $\tau_{chem} = \rho/k$. Thus we see that U_Y evolution is dependent on the chemical time scale τ_{chem} through the exponential term. If H is differentiated with respect to U_Y as,

$$\frac{\partial H}{\partial U_Y} = \frac{k}{\rho} = \frac{1}{\tau_{chem}} \tag{5.5}$$

we find that this derivative is also related to the chemical time scale, ie, is equal to the inverse of the chemical time scale, τ_{chem}. This would suggest that the matrix S should contain elements like $\partial H/\partial U$. It should be pointed out that the derivative $\partial H/\partial U$ for a system of equations becomes a Jacobian.

In chapter 2 we considered several numerical integration methodologies. In particular we looked at the point implicit method where the spatial gradient terms are treated explicitly and the chemical source terms are treated implicitly. The point implicit method can be written as,

$$\frac{U^{n+1} - U^{n}}{\Delta t} = -\left(\frac{\partial F}{\partial x}\right)^{n} - H^{n+1} \tag{5.6}$$

where U, F and H are given in chapter 2. To solve this equation H^{n+1} must be linearized. It is convenient to linearize H^{n+1} using the Newton method,

$$H^{n+1} = H^{n} + \left(\frac{\partial H}{\partial U}\right)^{n}\Delta U + O(\Delta U^{2}) \tag{5.7}$$

where $\Delta U = U^{n+1} - U^{n}$. If the linearized form of H^{n+1}, equation (5.7), is substituted into equation (5.6) as,

$$\frac{U^{n+1} - U^{n}}{\Delta t} = -\left(\frac{\partial F}{\partial x}\right)^{n} - \left(H^{n} + \left(\frac{\partial H}{\partial U}\right)^{n}(U^{n+1} - U^{n})\right) \tag{5.8}$$

which can be rewritten as,

$$\left[I - \Delta t\left(\frac{\partial H}{\partial U}\right)^{n}\right]\left(\frac{U^{n+1} - U^{n}}{\Delta t}\right) = -\left(\frac{\partial F}{\partial x}\right)^{n} - H^{n} \tag{5.9}$$

This equation could also be written as,

$$PI\left(\frac{U^{n+1} - U^{n}}{\Delta t}\right) = -\left(\frac{\partial F}{\partial x}\right)^{n} - H^{n} \tag{5.10}$$

where PI is equal to,

$$PI = \left[I + \Delta t\left(\frac{\partial H}{\partial U}\right)^{n}\right] \quad . \tag{5.11}$$

The matrix PI can be expressed as,

$$PI = \begin{vmatrix} 1 + \Delta tA_{11} & + \Delta tA_{12} & \cdot & \cdot & \cdot & \\ + \Delta tA_{21} & 1 + \Delta tA_{22} & & & & \\ \cdot & & \cdot & \cdot & & \\ & \cdot & & & \cdot & + \Delta tA_{N-1,N} \\ & & & & + \Delta tA_{N,N-1} & 1 + \Delta tA_{NN} \end{vmatrix} \quad (5.12)$$

where the $A_{i,j} = \partial H_i / \partial U_j$ are inversely proportional to the various time scales. Comparing equation 5.2 with equation 5.10 we see that the matrix **PI** can be interpreted as a possible preconditioner of equation 5.1. Note that this matrix contains terms of the form, $\Delta t \partial H / \partial U$ considered necessary to rescale the equations in time. Thus the implicit treatment of the chemical source terms leads to one possible form of the desired scaling matrix.

Another type of preconditioner considered, was to choose an S matrix containing only diagonal elements. With only diagonal elements, U^{n+1}, can be obtained directly without the need to invert the S matrix. As a result considerable computational work per iteration could be saved. However, dropping the off diagonal terms leads to inaccuracies in correctly time scaling the equations. If the inaccuracies are large, i.e. predicting changes in U which are larger than those given by the point implicit preconditioner, then the scheme will tend to be numerically unstable. Only a limited investigation was made with this type of preconditioner. The results indicate that with a diagonal preconditioner, where the diagonal terms are of the following form,

$$1 + 2\Delta t \frac{\partial H_k}{\partial U_k} \quad (5.13)$$

convergence was obtained in approximately twice as many iterations compared to the full point implicit preconditioner. It is possible, although not shown, that with enough species that the diagonal preconditioner could become more computational efficient than the full point implicit preconditioner. In this these no attempt was made to determine where the crossover might occur or even if one exists. In addition it might be possible to find approximations for the Jacobian derivatives which could reduce the computational work required to compute the S matrix. For example Pratt [40] has suggested using exponentials to approximate the Jacobian derivatives.

Finally, another technique was developed which possesses some of the attributes of preconditioning the equations to effectively time scale them. This method will be referred to as Local Time Cycling (LTC). In the LTC method the equations are solved explicitly but numerically updated only in certain regions of the computational domain depending upon the total accumulated time for each particular cell. If at the beginning of the calculation all cells begin with zero accumulated time , then the total accumulated time is sum of all the numeral time steps that the equations in a given cell have been advanced at since the beginning of the calculation. The LTC method would prove useful if the equations are stiff in part of the domain. The stiffness could limit the numerical time step taken in these cells and hold back the the longer time scale physics there compared to the rest of the domain. The method is implemented is implemented as follows. The first step comsists of one explicit iteration of all the equations over the complete global domain, i.e., one global iteration. If, after the global iteration the accumulated time in any given cell is less than in its neighboring cells then the equations in that particular cell are integrated in time until their accumulated time equals the accumulated time in the neighboring cells. It might be necessary to advance the equations in a given cell several times, local time cycling, and repeat the process every global iteration. The method has the advantage that it does not require the evaluation of the Jacobian derivatives. However, there is additional computational work associated with locally solving the equations and in determining which parts of the domain should be local time cycled.

With the exception of part of chapter 6, the full point implicit preconditioner will be used to time scale the equations in the remainder of this thesis.

5.1.3. Mathematical Representation of Time Scaling

The scaling matrix S derivied in the previous subsection can be shown to possess the desired time scaling character needed to remove the chemical time scale stiffness. To understand how this comes about consider the following O_2 dissociation reaction,

$$O_2 \longleftrightarrow 2O \tag{5.14}$$

for a compressible flow in a variable area duct. Using the quasi 1-D assumption and assuming no diffusion the system of equations being solved is,

$$\frac{\partial U}{\partial t} = -\frac{\partial F}{\partial x} - H \tag{5.15}$$

where,

$$U = \begin{vmatrix} \rho A \\ \rho u A \\ \rho E A \\ \rho Y_{O_2} A \end{vmatrix} \tag{5.16}$$

$$F = \begin{vmatrix} \rho u A \\ \rho u^2 A + pA \\ \rho H u A \\ \rho Y_{O_2} u A \end{vmatrix} \tag{5.17}$$

$$H = \begin{vmatrix} 0 \\ -\ p\ dA/dx \\ 0 \\ -\ w_{O_2} A \end{vmatrix} \tag{5.18}$$

The reaction rate, w_{O_2}, corresponding to equation (5.14) can be represented as[6],

---

[6]Only the forward rate is considered to simplify the example

$$H_{O_2} = Aw_{O_2} = kY_{O_2}A = kU_{O_2}/\rho \tag{5.19}$$

and

$$H_\rho = 0 \tag{5.20}$$

where A is the cross section area of the duct, and

$$k = \rho/\tau_{chem} \quad . \tag{5.21}$$

The effective time steps used by each equation within the system of equations can be illustrated by using the point implicit MacCormack method. See chapter 3 for a description of the MacCormack method. The scaling matrix S for the quasi one dimensional problem reduces to the following,

$$S = \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \Delta t \partial H_{O2}/\partial U_\rho & 0 & 0 & 1 + \Delta t \partial H_{O2}/\partial U_{O_2} \end{vmatrix} . \tag{5.22}$$

Note that only the chemical source terms, w, of the vector **H** are treated implicitly. The point implicit predictor equation for the continuity equation can be written as,

$$\Delta U_\rho = \Delta t \ [Res]^n_\rho \tag{5.23}$$

where,

$$[Res]^n_\rho = -\frac{(F^n_{\rho\, i+1} - F^n_{\rho\, i})}{\Delta x} \quad . \tag{5.24}$$

ΔU_ρ represents the change in the ρA and Δt is the numerical time step. In a similar way an expression for the change in $\rho Y_{O_2} A$ is given by,

$$\Delta U_{O_2} = \frac{\Delta t}{1+\Delta t \partial H_{O_2}/\partial U_{O_2}} [Res]^n_{O_2} \tag{5.25}$$

$$[Res]^n_{O_2} = -\frac{(F_{O_2}{}^n_{i+1} - F_{O_2}{}^n_{i})}{\Delta x} - H^n_{O_2} - \frac{\partial H_{O_2}}{\partial U_\rho}\Delta U_\rho \quad . \tag{5.26}$$

With equations (5.19), (5.20) and (5.21) equations (5.23) and (5.25) become,

$$\Delta U_\rho = \Delta t \; [Res]^n_\rho \tag{5.27}$$

and

$$\Delta U_{O_2} = \frac{\Delta t}{1 + \Delta t/\tau_{chem}} [Res]^n_{O_2} \quad . \tag{5.28}$$

Now if we choose,

$$\Delta t = \tau_{fl} \qquad (CFL = 1) \tag{5.29}$$

where

$$\tau_{fl} = \frac{\Delta x}{a} \tag{5.30}$$

and if $\tau_{fluid} \gg \tau_{chem}$ (Stiff) then equations (5.27) and (5.28) become,

$$\Delta U_\rho = \tau_{fluid} \; [Res]^n_\rho \tag{5.31}$$

and

$$\Delta U_{O_2} \simeq \tau_{chem} \, [Res]^{n}_{O_2} \quad . \tag{5.32}$$

So far we have considered a reaction model with only a forward rate. If we include both the forward and backward rates and consider the situation where the convective terms are small compared to the chemical source terms the conservation equation for O_2 can be written as,

$$\Delta U_{O_2} \simeq \tau_{chem} \, H^{n}_{O_2} \tag{5.33}$$

$$\Delta U_{O_2} \simeq \tau_{chem} \, [\, \frac{U_{eq} - U^{n}_{O_2}}{\tau_{chem}} \,] \tag{5.34}$$

$$\Delta U_{O_2} \simeq U_{eq} \tag{5.35}$$

where U_{eq} is the local equilibrium value of U. In this case U is always advanced to its local equilibrium value over one iteration. This is equivalent to advancing the state quantity U at a time scale which would take U^n to U_{eq}. If the convective term becomes important later on in the calculation, then U is advanced at the local chemical reaction time scale. Thus in general time scaling the equations is equilvalent to advancing each state quantity at its own characteristic rate. For example the fluid quantities are marched on the fluid time scales, τ_{fluid}, while the species quantities are marched at their respective reaction rate time scales, τ_{chem}, or their fluid time scales, τ_{fl}, which ever is smallest.

If large numerical time steps are taken then the solution is no longer time accurate but as we will see in the next chapter the convergence rate is dramatically improved. Note the time scaling method can be made time accurate by reducing the numerical time step everywhere to the physics time scales of interest, i.e., setting $\Delta t = \tau_{chem}$ for example.

The preconditioning procedure is applied only to those regions of the domain where the stiffness level is greater than approximately one. This implies that wherever the stiffness level is greater than one the point implicit preconditioner is used and wherever the stiffness lever is less than one no preconditioner is used. The preconditioner used with the Roger's and Chinitz H_2-air chemistry model is given in appendix 1.

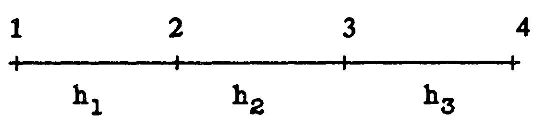
5.2. Local Time Stepping

In the local time stepping/constant CFL condition procedure, Γ is fixed everywhere which results in a variable $\Delta t = \Gamma \Delta x/a$ for each cell. Therefore, the solutions are not time accurate but convergence is achieved much more quickly. A constant CFL condition is a preconditioner procedure, similar to the chemical time scaling preconditioner described above, which reduces the stiffness of the discrete system of equations arising from widely differing mesh spacing (Δx) or eigenvalues (a). The paper by Eriksson and Rizzi [17] illustrates this for the Euler equations.

To show how the constant CFL condition helps improve the convergence rate consider the following example by Murman [33]. If we solve the equation,

$$U_t + aU_x = 0 \tag{5.36}$$

over the one dimensional domain defined by,



using the Lax Wendroff method,

$$U_j^{n+1} = U_j^n - a\Delta t U_x^n + a^2 \frac{\Delta t^2}{2} U_{xx}^n \tag{5.37}$$

where

$$U_{x_j} = \frac{2}{h_{j-1} + h_{j+1}}\left(\frac{U_{j+1} + U_j}{2} - \frac{U_j + U_{j-1}}{2}\right) \tag{5.38}$$

and

$$U_{xx_j} = \frac{2}{h_{j-1} + h_{j+1}}\left(\frac{U_{j+1} - U_j}{h_{j+1}} - \frac{U_j - U_{j-1}}{h_{j-1}}\right) \tag{5.39}$$

then the problem reduces to solving a system of algebraic equations. Note h_1, h_2 and h_3 are the mesh spacings Δx. The system can be written as,

$$\begin{matrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{matrix}^{n+1} = \left|\begin{matrix} 1 & 0 & 0 & 0 & U_1 \\ a_{21} & a_{22} & a_{23} & 0 & U_2 \\ 0 & a_{32} & a_{33} & a_{34} & U_3 \\ 0 & 0 & 0 & 1 & U_4 \end{matrix}\right|^{n} \tag{5.40}$$

subject to the boundary conditions,

$$U_1^n = U_1^e \tag{5.41}$$

and

$$U_4^{n+1} = U_3^n \ . \tag{5.42}$$

The coefficients in equation (5.40) are of the form,

$$a_{22} = -\frac{a^2 \Delta t^2}{h_3 h_1} \ . \tag{5.43}$$

where similar expressions exist for the remaining coefficients. A convenient way to characterize the convergence behavior of systems like those given by equation (5.40) is to look at their eigenvalues. The eigenvalues are a measure of how well conditioned the system is. For equation (5.40) the eigenvalues, μ, are given as,

$$\mu_{1,2} = 1\ ,\ 1 \tag{5.44}$$

$$\mu_{3,4} = 1/2[(a_{22} + a_{33}) \pm ((a_{22}+a_{33})^2 - 4(a_{22}a_{33} - a_{23}a_{32}))^{1/2}] \ . \tag{5.45}$$

If we now consider a specific example with h_1=1/2, h_2=1/3 and h_3=1/6 then if Δt is kept constant, as in a time accurate calculation, than μ_3=.544 and μ_4=.122. If instead the CFL number is kept constant then, μ_3=.299 and μ_4=.226. It turns out that the convergence rate is governed by the ratio of the eigenvalues μ_3 and μ_4. The larger the ratio the poorer the convergence rate. We see that for the constant Δt case the ratio μ_3/μ_4=4.46 while for the constant CFL case μ_3/μ_4=1.2. Thus the constant CFL condition preconditions the equations to remove the stiffness introduced by spatial discretization. Note the boundary eigenvalues μ_1 and μ_2 are not included in the ratio test since they are due to the boundaries.

5.3. Multigridding Stiff equations

The time scaling technique developed for the base solvers was also found to be extendable to multigrid methods. Many multigrid techniques have been proposed over the last several years following the ideas of Brandt [3]. The methods proposed by Ni [35] and Jameson [22] are of interest as they apply to the transport equations under consideration here. However these multigrid schemes suffer from the same type of stability limitation characteristics as purely explicit methods. This occurs because the coarse grid calculations or multigrid calculations are done explicitly and must follow the same explicit stability restrictions discussed previously for explicit schemes. For multigrid methods to be of use, time rescaling is necessary to remove the stiffness imposed by the source terms. To the best of the author's knowledge this is the first attempt to use multigrid methods to solve the transport equations with stiff source terms. The remainder of this section will discuss these issues and focus on the use of the Ni multiple-grid method for the solution of flows involving finite rate chemistry.

The time scaled coarse grid Ni multiple-grid technique will be demonstrated in this thesis only in one dimension. It is discussed here to show how the chemical time scaling technique can be applied to multigridding methods.

5.3.1. Ni Multiple Grid Method

In 1981 Ni [35] proposed a multiple-grid scheme for solving the Euler equations. Since then Johnson [23] and Chima [9] have extended Ni's method to the full and thin-layer Navier-Stokes equations. Johnson [23] also showed that the Ni fine grid solver could be replaced with MacCormack's scheme. The Ni multiple-grid accelerator would remain unchanged. Ni reported a five fold improvement in efficiency with the Euler equations and Johnson noted a three fold improvement with the Navier-Stokes equations. In addition the Ni scheme has also been extended to include both grid embedding [55] and adaptive grid embedding [12]. If this scheme is extended to flow problems involving finite rate chemistry, then stability problems can be expected since the Ni scheme is explicit and suffers from the same stability limitationt as the MacCormack scheme. It can be expected that the multiple grid accelerator would suffer from these same restrictions due to the added source terms. If the stiff multiple grid equations could be rescaled in time to allow all the state quantities to evolve at the same rate then the limitations due to stiffness could be eliminated from the multiple grid accelerator.

The basic Ni [35] multiple grid algorithm will now be outlined. Starting with:

$$\frac{\partial U}{\partial t} = -\frac{\partial F}{\partial x} \tag{5.46}$$

the Ni method can be described as having 5 steps.

Step 1

The first step involves computing the fine grid correction ΔU_I. The fine grid correction can be determined as follows,

$$\delta U_I = ((\delta U_I)_A + (\delta U_I)_B) \tag{5.47}$$

where

$$(\delta U)_A = 1/2(\Delta U_A + \frac{\Delta t}{\Delta x}\Delta F_A) \tag{5.48}$$

$$(\delta U)_B = 1/2(\Delta U_B - \frac{\Delta t}{\Delta x}\Delta F_B) \tag{5.49}$$

$$\Delta U_A = \frac{\Delta t}{\Delta x}(F^n_{I-1} - F^n_I) \tag{5.50}$$

$$\Delta U_B = \frac{\Delta t}{\Delta x}(F^n_I - F^n_{I+1}) \tag{5.51}$$

$$\Delta F = (\frac{\partial F}{\partial U})\Delta U \tag{5.52}$$

and $\partial F/\partial U$ is the Jacobian of F and U. The cell arrangement and labeling are given in Figure 5-3.

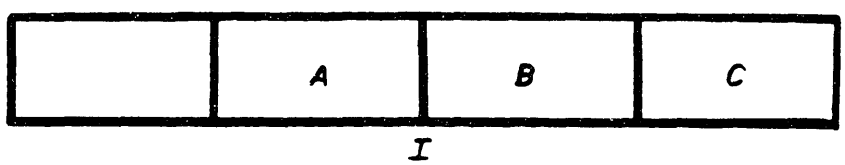


Figure 5-3: Fine Cell Nonenclature

Step 2

Step 2 involves transfering corrections from the fine grid to the next coarser grid. The procedure can be written as,

$$\Delta U^{2h} = T_{h}^{2h} \ \delta U^{h} \tag{5.53}$$

where T is an operator which transfers to each control volume of the coarse grid the correction δU^{h} of the centered fine grid point.

Step 3

It is in this step where the coarse grid calculations are performed. Basically the coarse grids are found by removing every other grid point from the next finer grid. Figure 5-4 shows an example of the 2h grid used here to illustrate a coarse grid calculation.

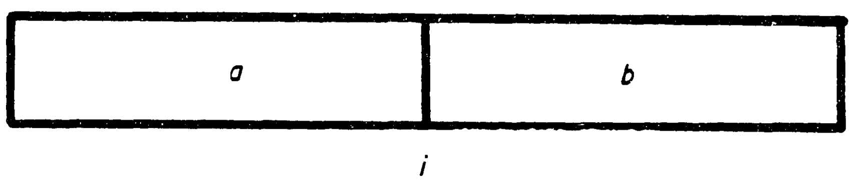


Figure 5-4: Coarse Cell Nonenclature

The idea here is to propagate fine grid changes more quickly out of the domain. The coarse grid correction can be computed as follows,

$$\delta U_{i}^{2h} = ((\delta U_{i})_{a} + (\delta U_{i})_{b}) \tag{5.54}$$

$$(\delta U_{i})_{a} = 1/2 \ (\Delta U_{a}^{2h} + (\frac{\Delta t}{\Delta x})^{2h} \ \Delta F_{a}^{2h}) \tag{5.55}$$

$$(\delta U_{i})_{b} = 1/2 \ (\Delta U_{b}^{2h} - (\frac{\Delta t}{\Delta x})^{2h} \ \Delta F_{b}^{2h}) \tag{5.56}$$

where the quantities are as defined in Step 1.

Step 4

This step involves interpolating the coarse grid corrections back down to the finest level. The process can be represented by,

$$\delta U_i^{2h} = I_{2h}^{h} \; \delta U_i^{2h} \tag{5.57}$$

where I_{2h}^{h} is a linear interpolation operator.

Step 5

The last step updates the state variables with the corrections from all grid levels. The process can be written as,

$$U_i^{n+1} = U_i^{n} + \delta U_i^{h} + \delta U_i^{2h} + \delta U_i^{4h} + \dots + \delta U_i^{kh} \tag{5.58}$$

where k is the coarsest level chosen. These five steps represent a single Ni multiple grid cycle.

5.3.2. Point Implicit Ni Multiple Grid Method

If the Ni scheme is to be used to solve stiff equations, like those discussed previously, then specific modifications to the scheme need to be made. The first change involves rewriting equation (5.46) to include the source terms, i.e., equation (2.39). The second major change involves removing the chemical time scale dependence by time scaling. These modifications will be described with the 5 step format used to outline the basic Ni scheme.

Step 1

As noted by Johnson [23] the Ni fine grid solver can be replaced by other Lax Wendroff schemes, specifically the 1969 MacCormack method. Since the author had already developed a point implicit version of the MacCormack's scheme it was decided to use this instead of an equivalent point implicit Ni fine grid solver to compute the fine grid correction. Thus, we will only outline the changes needed to remove the stiffness from the Ni coarse grid accelerator.

Step 2

Same as the basic Ni scheme.

Step 3

The inclusion of the chemical source terms modifies equations (5.54), (5.55) and (5.56) to,

$$\delta U_i^{2h} = ((\delta U_i)_a + (\delta U_i)_b) \tag{5.59}$$

$$(\delta U_i)_a = 1/2(\Delta U_a^{2h} + (\frac{\Delta t}{\Delta x})^{2h} \Delta F_a^{2h} + \Delta t \Delta H_a^{2h}) \tag{5.60}$$

$$(\delta U_i)_b = 1/2(\Delta U_b^{2h} - (\frac{\Delta t}{\Delta x})^{2h} \Delta F_b^{2h} - \Delta t \Delta H_b^{2h}) \tag{5.61}$$

where,

$$\Delta F^{2h} = (\frac{\partial F}{\partial U}) \Delta U^{2h} \tag{5.62}$$

$$\Delta H^{2h} = (\frac{\partial H}{\partial U}) \Delta U^{2h} \quad . \tag{5.63}$$

Now since,

$$\frac{\partial H}{\partial U} \simeq \frac{1}{\tau_{chem}} \tag{5.64}$$

(see section on time scaling) equations (5.60) and (5.61) can be stiff if $\tau_{chem} \ll \Delta t_{f1}$. Thus if the multiple grid procedure is to be of use these equations must be time scaled, i.e., ΔH^{2h} treated implicitly. The simplest formulation which seems to work is to set ΔU^{2h} in equation (5.63) equal to δU_I, i.e.,

$$\Delta U_a^{2h} = (\delta U_I)_a \tag{5.65}$$

$$\Delta U_b^{2h} = (\delta U_I)_b \quad . \tag{5.66}$$

It was found best not to recompute $\partial H/\partial U$ on the coarse grid levels but instead to evaluate it by area averaging $\partial H/\partial U$ computed on the finest grid. For example for cell b (2h level),

$$\left(\frac{\partial H}{\partial U}\right)_b^{2h} = \frac{1}{S_B + S_C}\left[\left(\frac{\partial H}{\partial U}\right)_B^h \times S_B + \left(\frac{\partial H}{\partial U}\right)_C^h \times S_C\right] \tag{5.67}$$

where S is the cell area. Basing $(\partial H/\partial U)^{2h}$ on the finest level is necessary as the chemical time scales, τ_{chem}, can be sensitive to temperature. If $\partial H/\partial U$ is recomputed on each coarse level the temperature used would be an average value over many fine cells. This average temperature could produce a very different chemical reaction behavior which would not be consistant with the fine grid predictions.

Thus rewriting equations (5.60) and (5.61) with the source term treated implicitly leads to:

$$SN(\delta U_I)_a = 1/2\left(\Delta U_a^{2h} + \frac{\Delta t}{\Delta x}\Delta F_a^{2h}\right) \tag{5.68}$$

$$SN(\delta U_I)_b = 1/2\left(\Delta U_b^{2h} - \frac{\Delta t}{\Delta x}\Delta F_b^{2h}\right) \tag{5.69}$$

where SN is given as,

$$SN = \begin{vmatrix} 1 - \Delta tA_{11} & - \Delta tA_{12} & \cdot & \cdot & \cdot & \\ - \Delta tA_{21} & 1 - \Delta tA_{22} & & & & \\ \cdot & & \cdot & \cdot & \cdot & \\ \cdot & & \cdot & & & - \Delta tA_{N-1,N} \\ & & & - \Delta tA_{N,N-1} & & 1 - \Delta tA_{NN} \end{vmatrix} \tag{5.70}$$

and $A_{ij} = 1/2\ \partial H_i/\partial U_j$ for equation (5.68) and $A_{ij} = -1/2\ \partial H_i/\partial U_j$ for equation (5.69). The matrix SN represents a preconditioner similar to the preconditioners derived earlier is this chapter.

Step 4

Same as the basic Ni solver.

Step 5

Same as the basic Ni solver.

Chapter 6

1-D FLOW WITH CHEMISTRY

The next three chapters will consider the application of the solution method and acceleration techniques to a variety of one and two-dimensional problems. The first section of this chapter will address convergence characteristics of the solution methods while the second section will validate the method's ability to predict hydrogen - air combustion.

6.1. 1-D Method Validation

In this section, the method and acceleration techniques will be validated using the quasi-one-dimensional Euler equations ((2.39), (2.40)) with simple modified model of diatomic oxygen dissociation, equation (2.35). A symmetric converging/diverging nozzle is used as the test geometry. The configuration was chosen because it produces the necessary high static temperature conditions to trigger oxygen dissociation. The duct area distribution is shown in figure 6-1. The point implicit version of the MacCormack scheme [32] is used for all of the calculations considered in this chapter.

Let us begin by considering the validation of the non-reacting flow field. The test conditions are given in table 6-1. The quasi-one-dimensional numerical calculations are compared with quasi-one-dimensional theory [47]. Figures 6-2, 6-3 and 6-4 show the comparisons between the computed and theoretical distributions of pressure, temperature and Mach number. The results show the excellent agreement between theory and computation. Note the Mach number is everywhere supersonic.

If the dissociation reaction is allowed to occur, ie, by turning on the chemical reaction routine, than a different flow field is produced. Two cases with very different stiffness ratios will be considered. The flow field properties and reaction rate parameters are given in tables 6-2 and 6-3. The cases differ in the value of the constants used in the reaction rate term. The stiffness level for case 1 is approximately 10 while case 2 is approximately 1000. The constants were picked to assess the solution method's ability to handle problems with widely different levels of stiffness[7]. Since the reaction rate values were picked for numerical reasons, the results do not necessarily model real oxygen dissociation.

The calculated property distributions for case 1 , are shown in figures 6-5 , 6-6, 6-7, 6-8 and 6-9. The figures show the distributions of pressure, temperature, Mach number, species and the oxygen dissociation rate through the channel. From these figures it is clear that the reaction is triggered at the x=.55 location. For this model reaction dissociation becomes important when the temperature reaches approximately 4200°K. The reaction takes place over 12 cells, roughly 12% of the channel length. The reaction rate plot 6-9 shows that the reaction zone is also confined to these cells. Note that the reaction rate increases from zero to a maximum of approximately 3000 in only 7 cells. The species distributions plot shows that O_2 changes completely to O over the reaction zone. It is useful to consider a numerical time step distribution plot of the fluid and dissociation time scales through the channel, figure 6-10. The figure shows that the chemical time scales drops from approximately 10^{-2} to 10^{-3} while the fluid time scale remains essentially constant. The effect is to create a "time well" where the chemical time scale is smaller than the local fluid time scale. The local fluid time scale is defined by $\tau_{f1}=\Delta x/(u+c)$. Note that the ordinate is scaled as the log of the time scales. If the problem is solved explicitly, then all of the state quantities are advanced at the smaller of the two time scales, ie, the chemical time scale within the well and the fluid time scale outside of the well. If the chemical time scale preconditioner is used, then each state quantity is advanced at its own characteristic rate. In particular, the fluid quantities would follow the fluid time scale curve while the species quantity, O_2, would follow the chemical time scale curve.

If the reaction rate constant is increased by two orders of magnitude, case 2, then the pressure, temperature, Mach number, species distribution and reaction rate are as given in figures 6-11, 6-12, 6-13, 6-14 and 6-15. The obvious difference between this case and case 1 is the thickness of the reaction zone. Here the zone is roughly 3 cells thick or 3% of the channel length. This is expected since the reaction is two orders of magnitude more active. Note also that the reaction zone is sharp with little

---

[7]See chapter 3 for a definition of stiffness

or no wiggles[8]. The numerical time step distributions plot, figure 6-16, shows that the minimum chemical time scales are three orders smaller than the fluid time scales. The time history plot is a convenient way to determine the local stiffness.

It is interesting to note that although the oxygen is dissociating, which is an endothermic process, the fluid static temperature rises through the reaction zone. This character is clearly shown in figures 6-6 and 6-12. Although heat is absorbed, the temperature rises because the fluid properties, c_p and c_v etc, for O_2 and O are very different. In fact γ goes from 1.33 for O_2 to 1.2 for O which for this case has a much greater effect on the flow field than the heat absorbed. Note that the values of c_p and c_v were picked to produce the numerical behaviour shown here and do not necessarly represent real O_2 or O gas properities.

Having discussed the physical properties of the non-reacting and reacting problems, we will now address the question of convergence behavior using the different acceleration techniques. Convergence of the computed solution is defined as a three orders of magnitude reduction of the global density residual, where the residual is equal to the unsteady term of equation (2.39). Figure 6-17 shows the convergence histories for the non-reacting problem. The figure shows that if $\Delta t = \Delta t_{min}$ convergence is obtained in 280 iterations. If the CFL number is kept constant convergence is obtained in 270 iterations and only 120 iterations with the Ni multiple grid accelerator. For this problem Δt does not vary much from cell to cell and thus only the Ni multiple grid accelerator improves the convergence rate significantly. For problems where Δt varies more widely the constant CFL condition would also help improve the convergence rate.

If the reaction is allowed to proceed as in case 1 then the convergence behavior with the various acceleration techniques is as shown in figure 6-18. If no acceleration techniques are used, $\Delta t = \Delta t_{min}$, then convergence is very slow, approximately 10^4 iterations. If the constant CFL condition is used then the iteration count drops to 10^3. With chemical time scale preconditioning the iteration number becomes 280 and with the time scaled Ni multiple grid accelerator it takes only 120 iterations to reach convergence. A similar set of curves exists for case 2, figure 6-19. The only difference

---

[8] The damping used was second order post smoothing [28]

is that the $\Delta t = \Delta t_{min}$ and the constant CFL runs take roughly two orders of magnitude longer to converge. The chemical time scaled preconditioned techniques remain essentially unchanged.

Thus figures 6-18 and 6-19 show that with chemical time scale preconditioning the number of iterations needed to get to steady state is independent of the level of chemical stiffness.

Finally, for the diatomic dissociation problems considered here, the LTC method mentioned in chapter 5, proves to be approximately two to ten times more computationally expensive than the point implicit method.

6.2. 1-D H_2-Air Combustion

In this section the solution method will be validated with a realistic hydrogen - air chemistry model. The chemistry model was first proposed by Rogers and Chinitz [44] and was discussed in chapter 2. Since experimental data for hydrogen - air combustion is not readily available the computed flow field is compared with a computation performed by Drummond [15]. Drummond uses a spectral method to evaluate the spatial gradient terms and a time integrator based on the scheme proposed in this thesis.

The test geometry consists of a symmetric straight walled diverging nozzle. The geometry area distribution is given in figure 6-20. Without chemical reaction the pressure, temperature and Mach number distributions are as shown in figures 6-21, 6-22 and 6-23. From these figures it is clear that the Mach number is everywhere supersonic. Table 6-4 lists the inflow conditions and parameters used for the calculation. If the H_2 - Air reaction is allowed to proceed, ie, by turning on the chemistry model, then several interesting phenomenon occur. First the reaction begins at the first station since the inflow static temperature is higher than the fuel ignition temperature. This effect is evident in the H_2O/OH species plot, figure 6-24, which shows the density fractions H_2O and OH undergoing considerable change at the first station. In particular, we see that the density fraction of OH undergoes a step change from 0.0 to a maximum value of .019 across the first cell. It will be shown shortly that this is due to the high level of stiffness associated with the OH formation reaction, ie, the second step of the Roger's chemistry model. The H_2O density fraction however changes with x much more smoothly. In fact, these behaviors can be seen in the numerical time step distribution plot, figure 6-25. The plot shows that there are three distinct time scales, one fluid and two chemistry time scales. If we recall that the Roger's chemistry model consists of two steps,

$$H_2 + O_2 \longleftrightarrow 2OH$$

$$H_2 + 2OH \longleftrightarrow 2H_2O$$

then the two chemistry time scales correspond to the time scales associated with these two equations. In fact the rapid change noted for the density fraction OH, implies that the first reaction occurs very fast, ie, is very stiff. From figure 6-25 the time scale for the first step of the Roger's model is 10^6 faster than the local fluid time scale. The figure shows that the time scale for the second reaction step is approximately equal to the local fluid time scale. It should be pointed out that the first step of the chemistry model could have been replaced by an equilibrium step model where the density fraction of OH is related algebraically to the density fractions of H_2 and O_2. Although the first step of the reaction is very stiff and could be treated as an equilibrium step it was retained within the partial differential equation system to maintain the numerical difficulty.

The species plot, figure 6-24, shows that for this example the reaction zone consists of two regions. The first region is associated with the stiff OH reaction, which shows this part of the reaction zone to be smaller than one cell width. The second, associated with H_2O formation, is roughly 30 cells in width or 30% of the channel length. These zones are important because they govern where energy is added or removed from the flow. In this particular case the formation of H_2O results in the liberation of energy which as we will see in the next two chapters, can significantly alter the flow field. The reaction zone thickness can also be used to compute the reaction time or equivalently the heat release time. The H_2 reaction/heat release time scale can be approximated by,

$$\tau_{hr} = L_{reaction\ zone} / V$$

where $L_{reaction\ zone}$ is the reaction zone thickness and V is the mean flow velocity through the reaction zone. For the particular example described here $L_{reaction\ zone}$=.3m and V=1300m/sec which implies that $\tau_{hr}=2*10^{-4}$ seconds. As we will see in the next two chapters the fluid time scales will have to be comparable or longer then the heat release time scale for heat release effects to influence the flow field. This occurs because it is the H_2O reaction which is exothermic and the rate of heat release is governed by τ_{H_2O}.

Figure 6-24 also shows the corresponding distributions that would be produced if the reactions were modeled as equilibrium reactions. Thus one has to be careful to solve a non-equilibrium flow, like the one computed here, with finite rate chemistry and not equilibrium chemistry.

Convergence for this calculation was achieved in approximately 700 iterations. If the equations had not been preconditioned (time scaled) then convergence would have taken in excess of 10^8 iterations. The distributions of pressure, temperature and Mach number are given in figures 6-26, 6-27 and 6-28. The figures show that the reaction has altered the flow field.

Finally it is interesting to compare the *real* time behavior, figure 6-29, to the *pseudo* time behavior, figure 6-30, for the density fractions of OH and H_2O at the mid-point of the channel. The real time history figure was provided by Drummond [16]. The real time plot shows that OH ignites in 10^{-11} seconds while H_2O doesn't ignite until 10^{-5} seconds. In pseudo time both density fractions ignite at iteration 1. In addition the slopes of the two pseudo time curves are approximately equal indicating that each is marched at their own characteristic reaction rate (time scales), in agreement with the time scale preconditioning method proposed in this thesis. Probably the most important result is that the steady state reached by the two methods is the same, ie, Y_{OH}=.21 and Y_{H_2O}=.074. Therefore, although different transient paths are taken the steady state reached using a time accurate method or the time scaling preconditioning method given here are the same.

Properties	Values	Dimensions
P_{∞}	6.6×10^4	N/m^2
T_{∞}	1200.	°K
M_{∞}	6.	
c_p	1000.	J/kg·°K
c_p	718.	J/kg·°K
L	.213	m
Grid	129	
CFL	.9	

Table 6-1: Table Of Flow Data - Validation

Properties	Values	Dimensions
P_∞	6.6×10^4	N/m^2
T_∞	1200.	°K
M_∞	6.	
c_{pO2}	1040.	J/kg·°K
c_{pO}	780.	J/kg·°K
c_{vO2}	600.	J/kg·°K
c_{vO}	500.	J/kg·°K
Hf_{O2}	0.0	J/kg
Hf_{O}	1.0×10^5	J/kg
L	.213	m
$w = A\ T^{-B}\ e^{-C/T}$		
A	2.0 $\times10^{12}$	
B	-1	
C	80	
Grid	129	
CFL	.9	

Table 6-2: Table Of Flow Data - Case 1

Properties	Values	Dimensions
P_∞	6.6×10^4	N/m^2
T_∞	1200.	°K
M_∞	6.	
c_{pO2}	1040.	J/kg·°K
c_{pO}	780.	J/kg·°K
c_{vO2}	600.	J/kg·°K
c_{vO}	500.	J/kg·°K
Hf_{O2}	0.0	J/kg
Hf_O	1.0×10^5	J/kg
L	.213	m
$w = A\ T^{-B}\ e^{-C/T}$		
A	10^{14}	
B	-1	
C	80	
Grid	129	
CFL	.9	

Table 6-3: Table Of Flow Data - Case 2

Properties	Values	Dimensions
P_∞	1.01×10^5	N/m^2
T_∞	1500.	°K
$T_{Ignition}$	1000.	°K
ϕ	.1	
M_∞	1.5	
c_{pH2O}	17160.	J/kg·°K
c_{pOH}	1181.	J/kg·°K
c_{pH2}	2854.	J/kg·°K
c_{pO2}	2041.	J/kg·°K
c_{pN2}	1285.	J/kg·°K
c_{vH2O}	17160.	J/kg·°K
c_{vOH}	1181.	J/kg·°K
c_{vH2}	2854.	J/kg·°K
c_{vO2}	2041.	J/kg·°K
c_{vN2}	1285.	J/kg·°K
Hf_{H2O}	-1.44×10^7	J/kg
Hf_{OH}	2.3×10^6	J/kg
Hf_{H2}	0.0	J/kg
Hf_{O2}	0.0	J/kg
Hf_{N2}	0.0	J/kg
L	2.0	m
Grid	129	
CFL	.9	

Table 6-4: Table Of Flow Data - H2-Air Chemistry

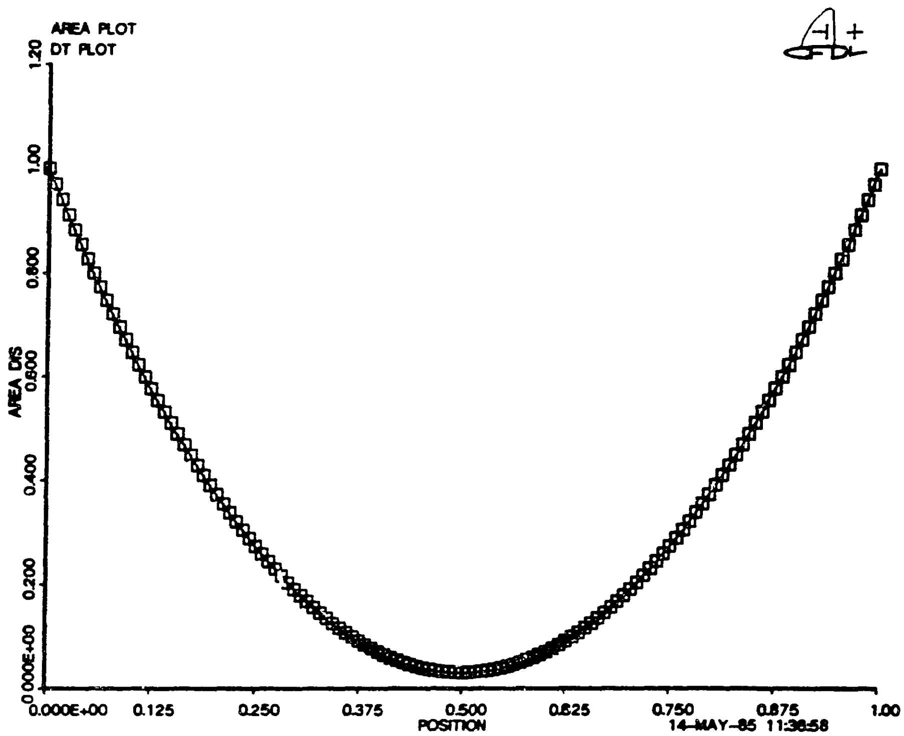


Figure 6-1: One Dimensional Area Distribution

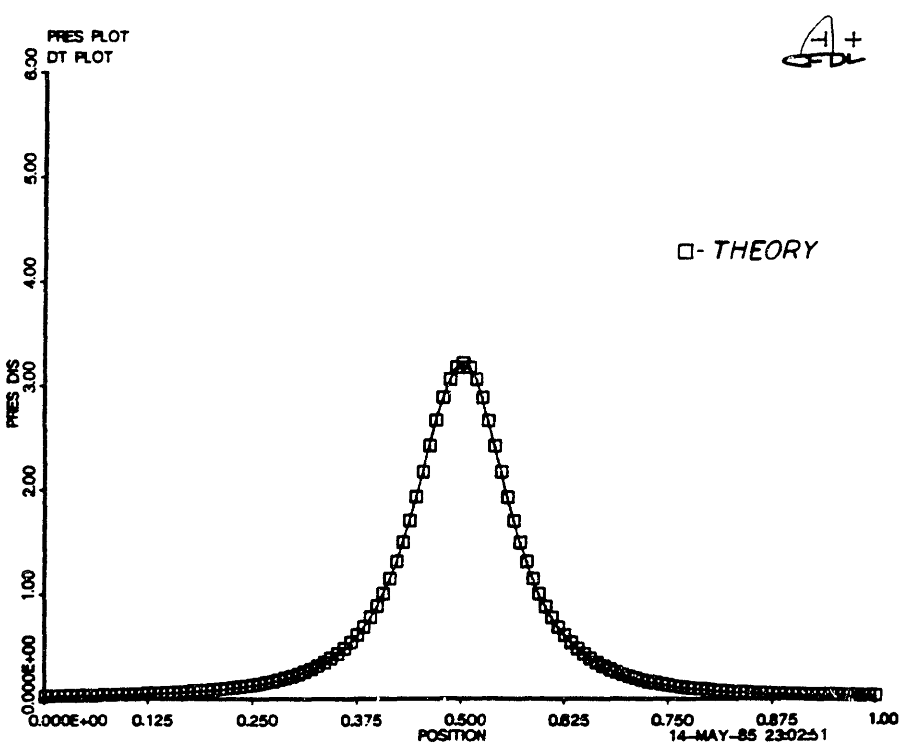


Figure 6-2: 1 - D Non Reacting Pressure Plot

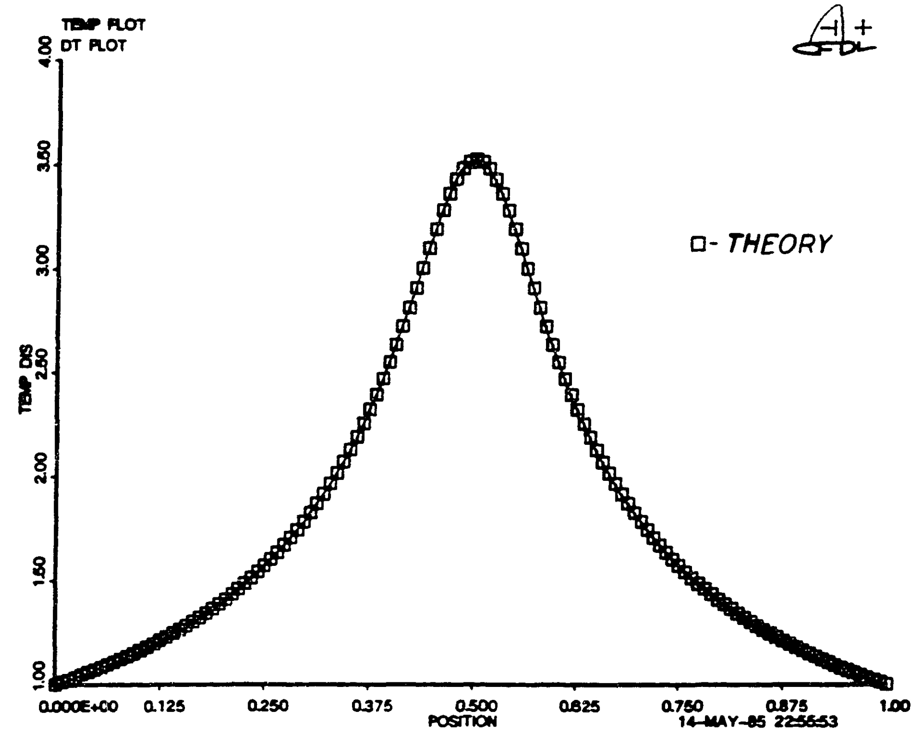


Figure 6-3: 1 - D Non Reacting Temperature Plot

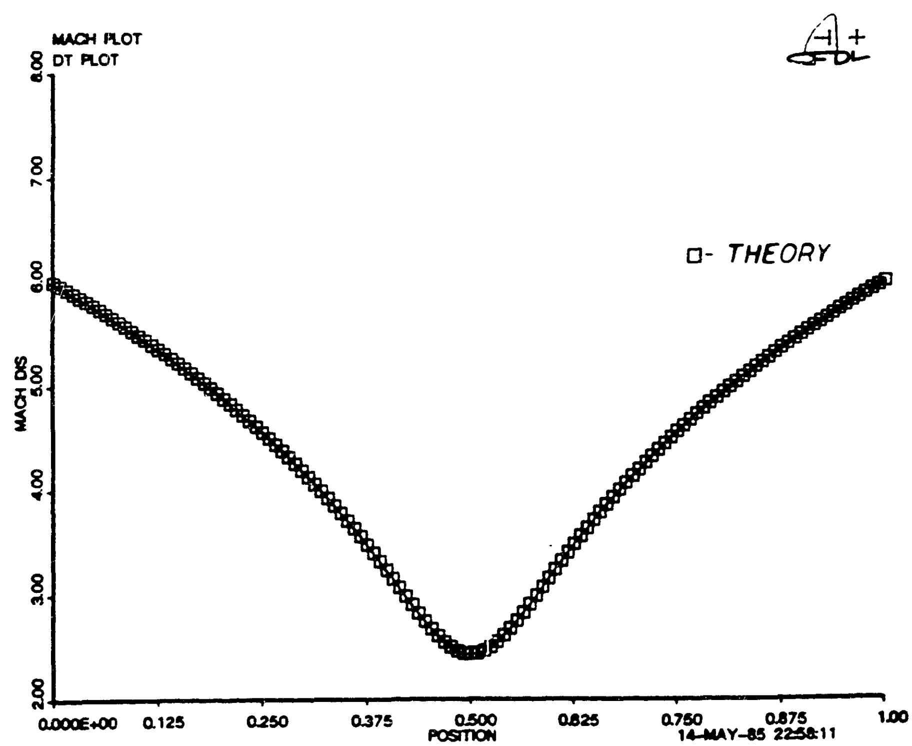


Figure 6-4: 1 - D Non Reacting Mach Number Plot

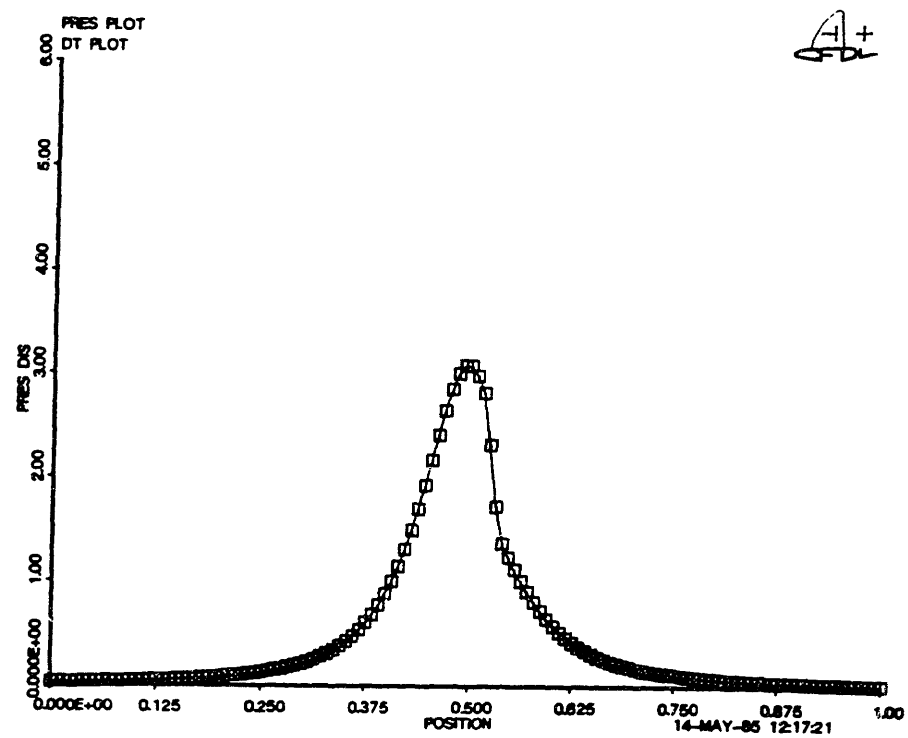


Figure 6-5: 1 - D Reacting Pressure Plot - Case 1

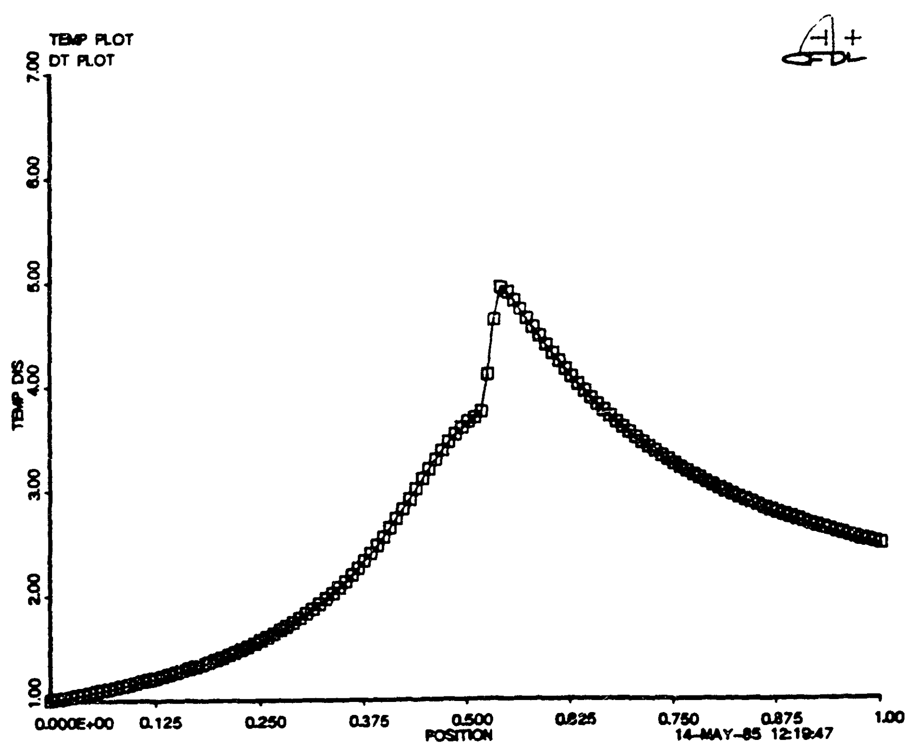


Figure 6-6: 1 - D Reacting Temperature Plot - Case 1

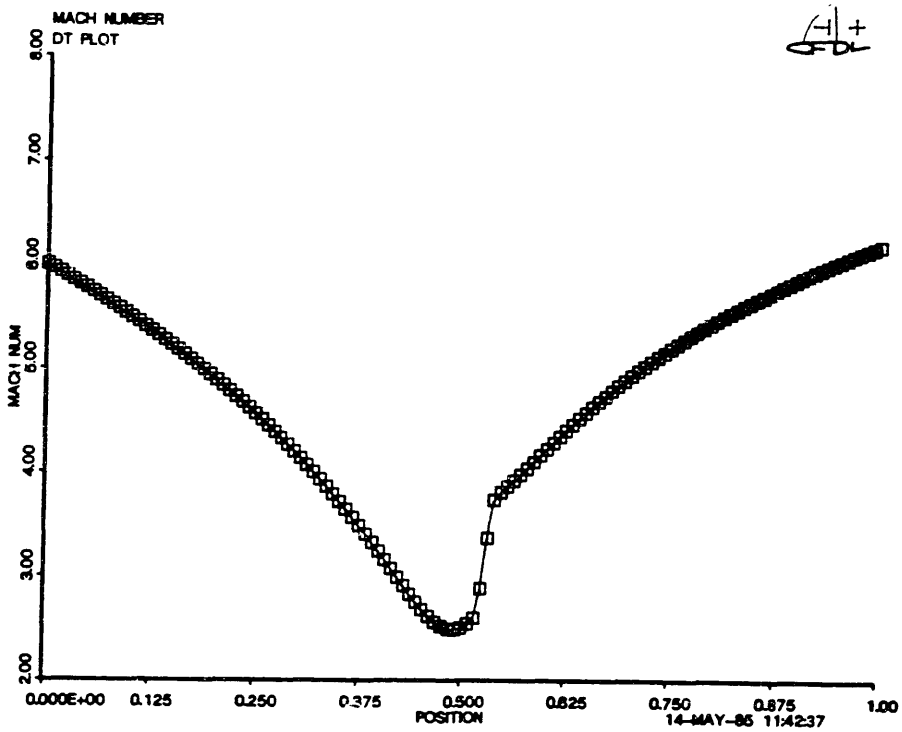


Figure 6-7: 1 - D Mach Number Plot - Case 1

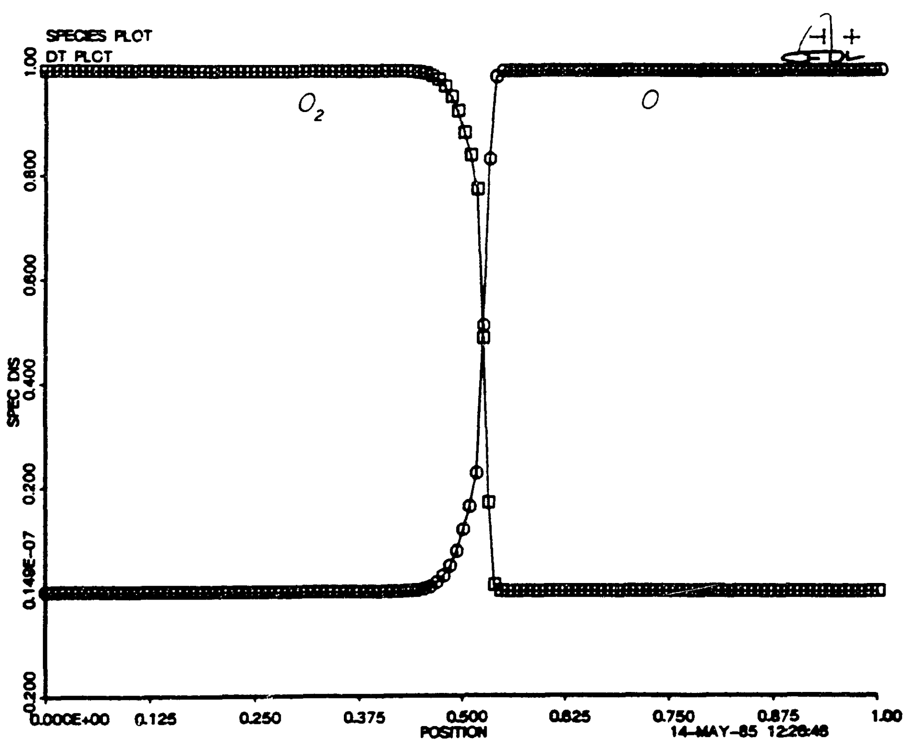


Figure 6-8: 1 - D O2 and O Species Plot - Case 1

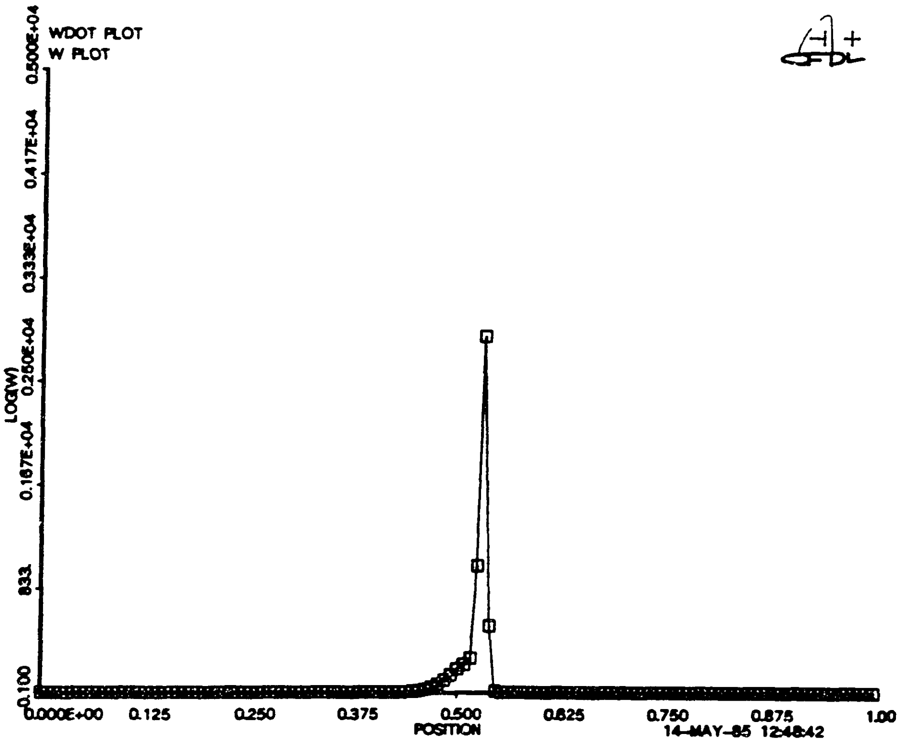


Figure 6-9: 1 - D Reaction Rate Plot - Case 1

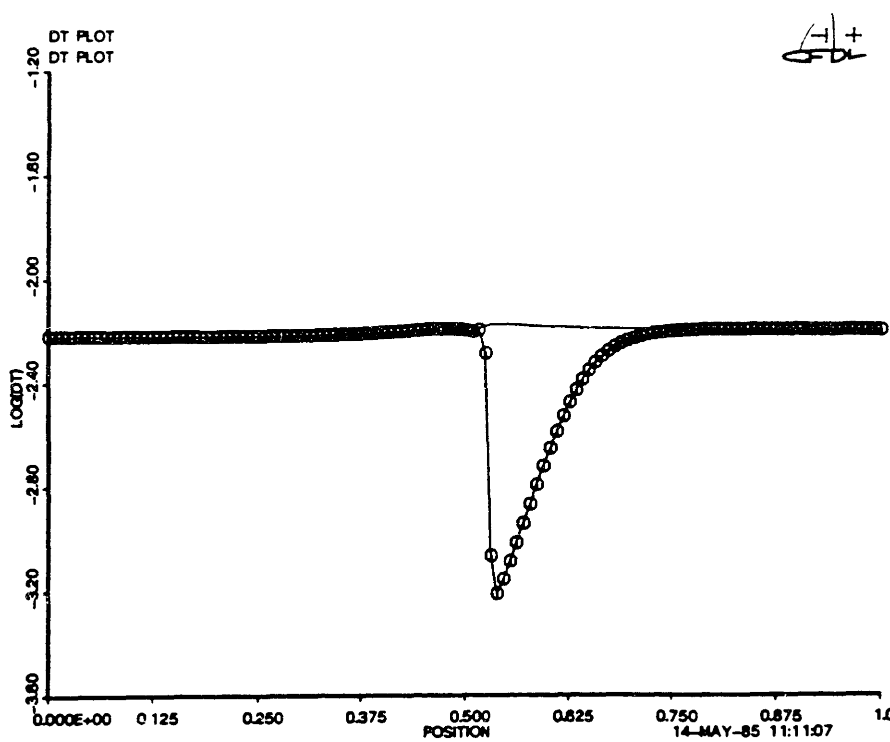


Figure 6-10: 1 - D Reacting Time Scale Plot - Case 1

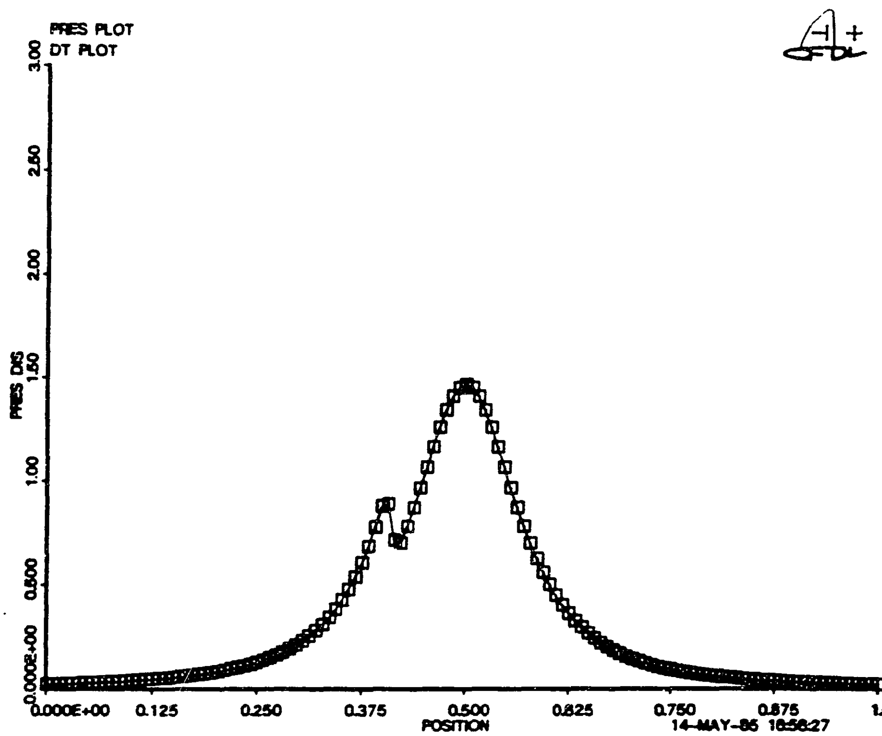


Figure 6-11: 1 - D Reacting Pressure Plot - Case 2

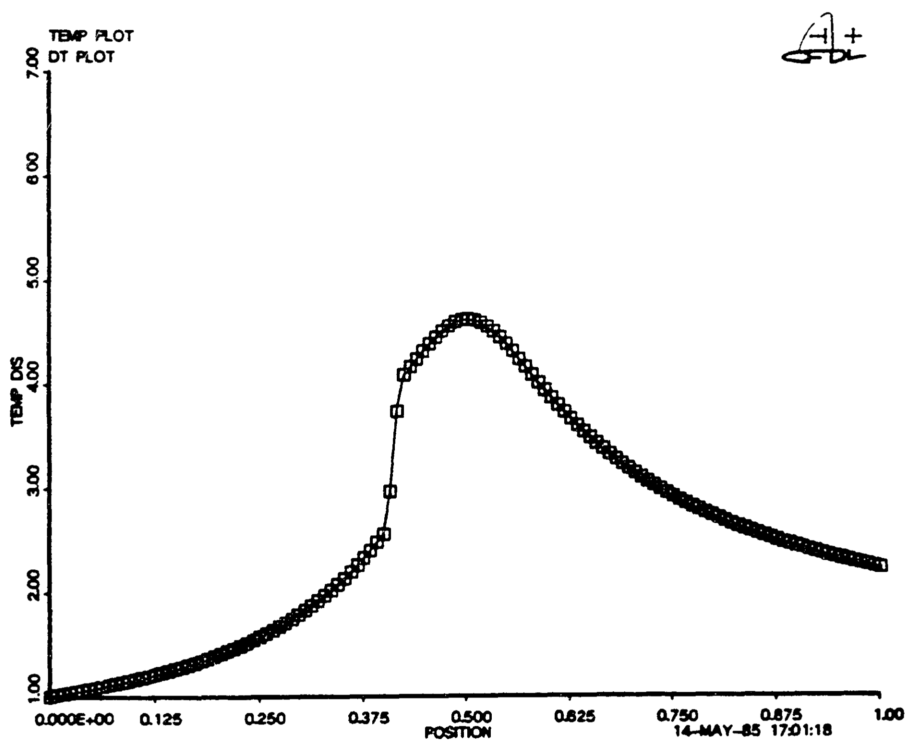


Figure 6-12: 1 - D Reacting Temperature Plot - Case 2

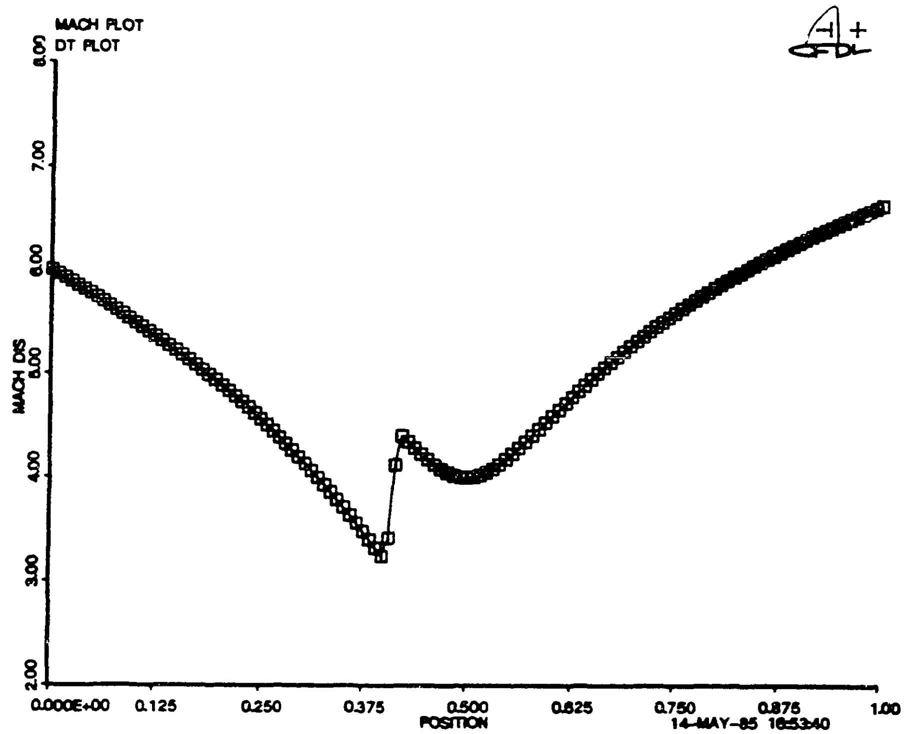


Figure 6-13: 1 - D Mach Number Plot - Case 2

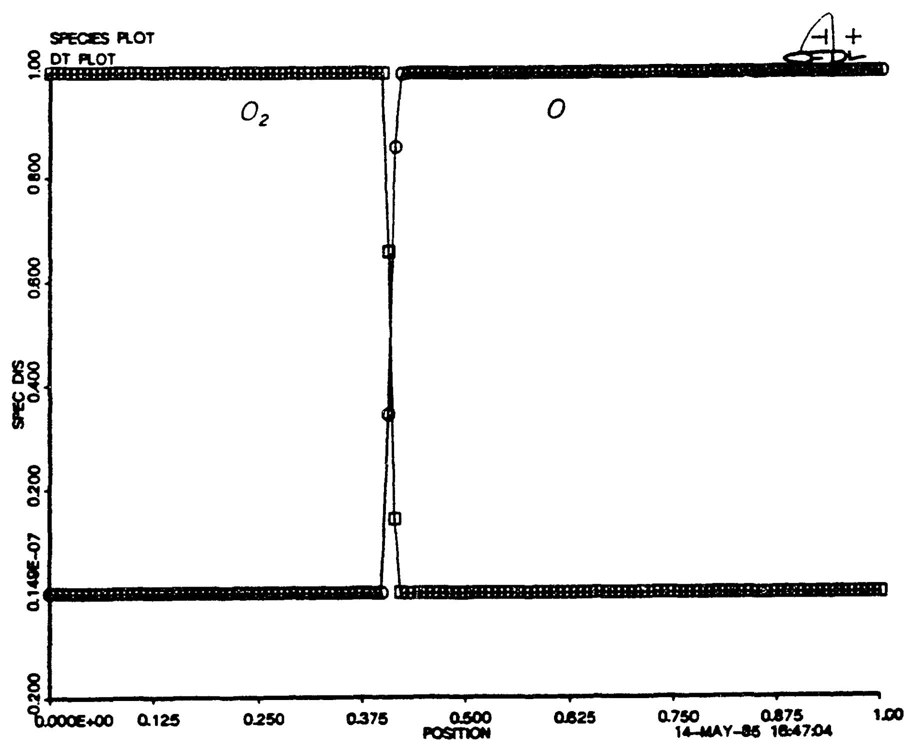


Figure 6-14: 1 - D O2 and O Species Plot - Case 2

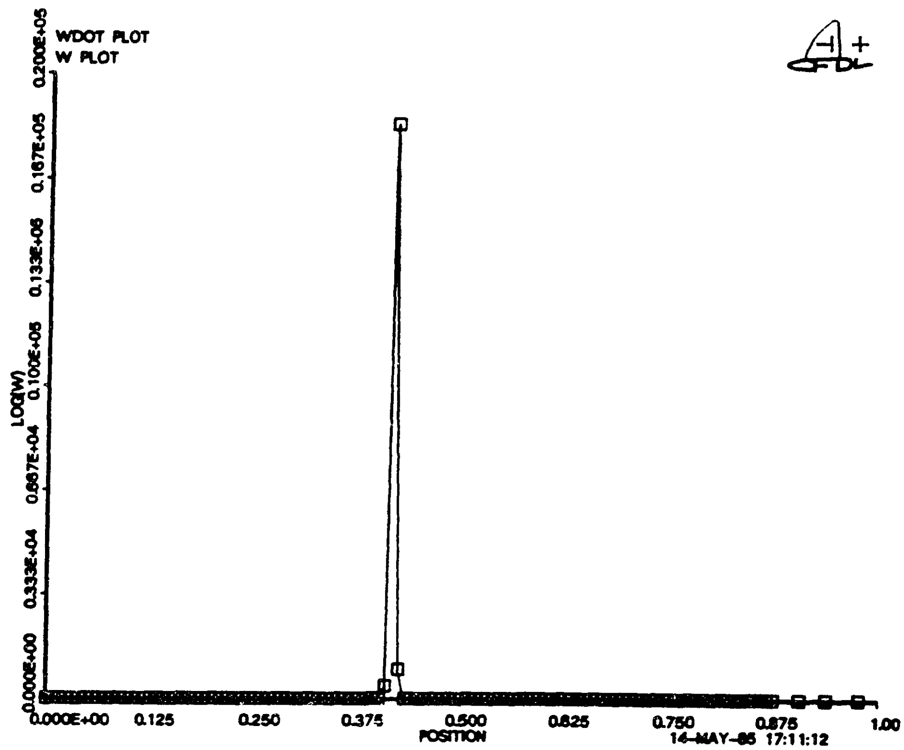


Figure 6-15: 1 - D Reaction Rate Plot - Case 2

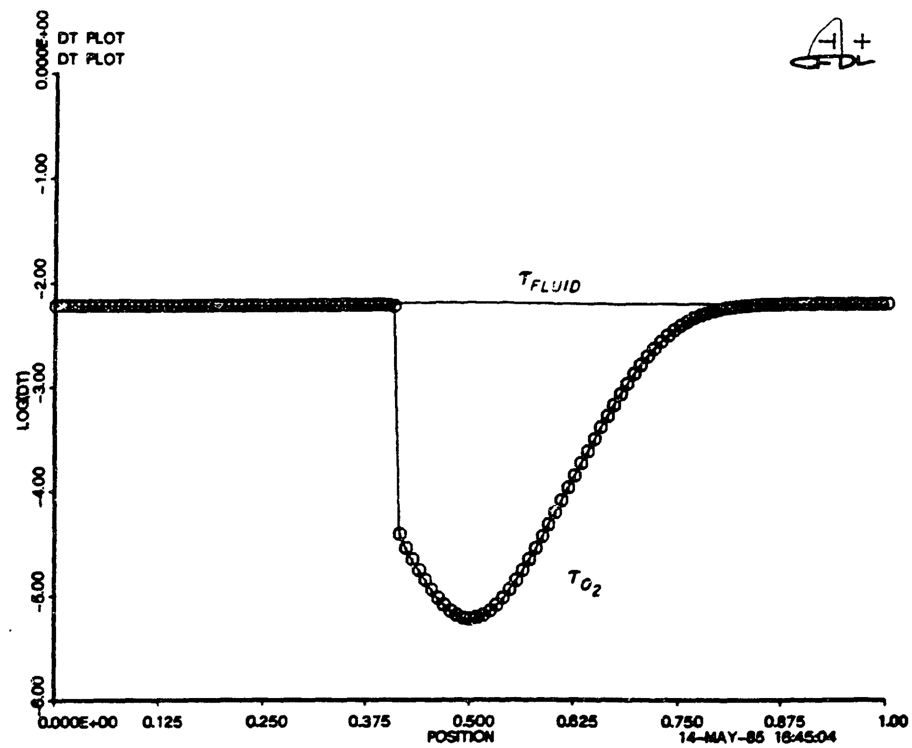


Figure 6-16: 1 - D Reacting Time Scale Plot - Case 2

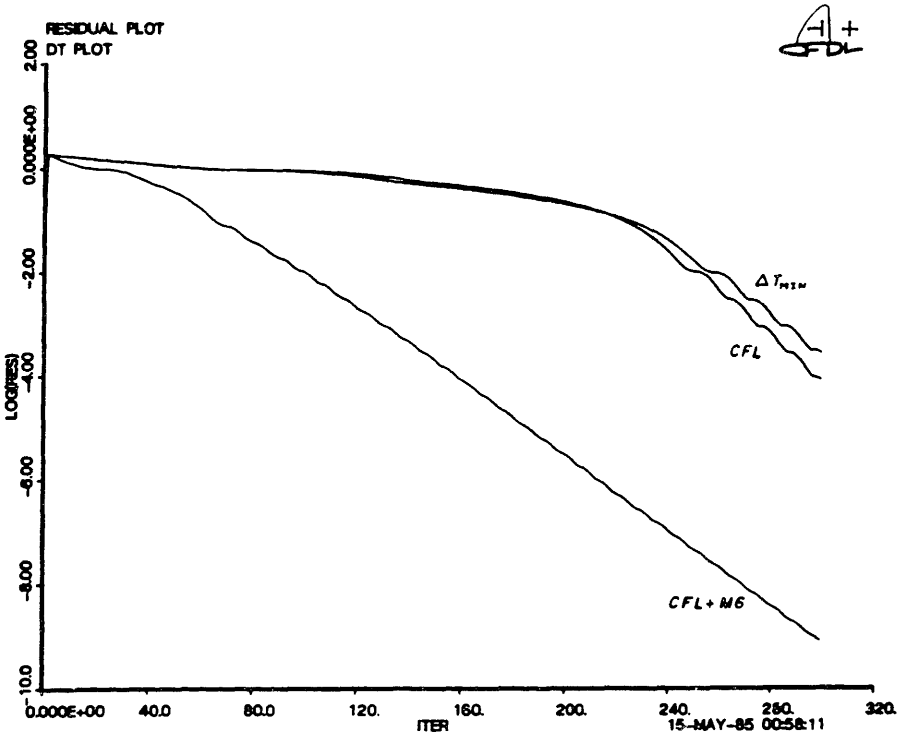


Figure 6-17: 1 - D Non-Reacting Convergence History Plot

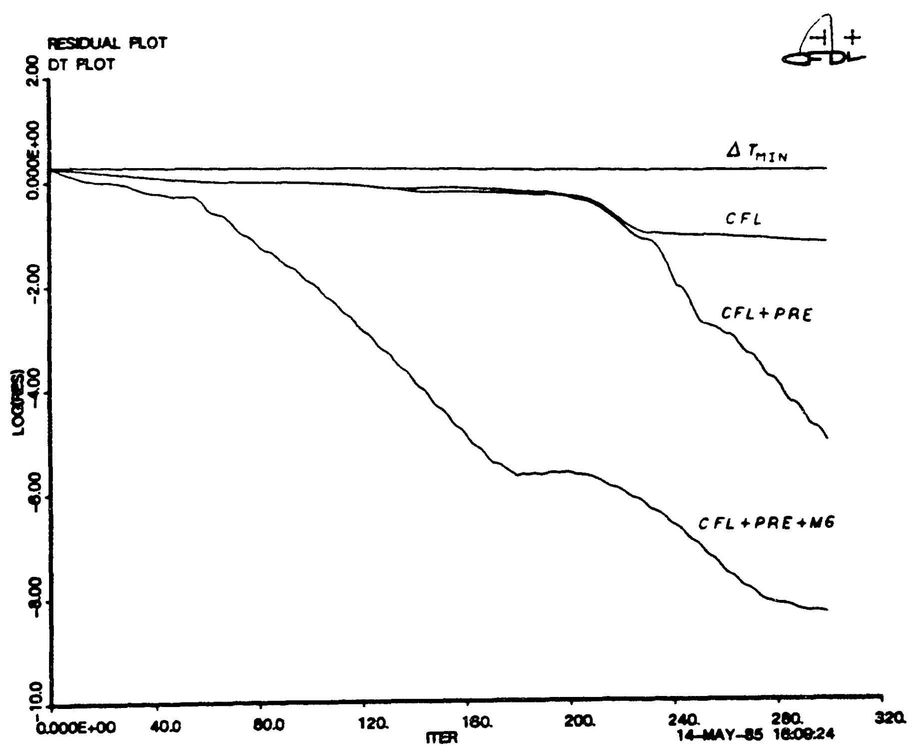


Figure 6-18: 1 - D Reacting Convergence History Plot - Case 1

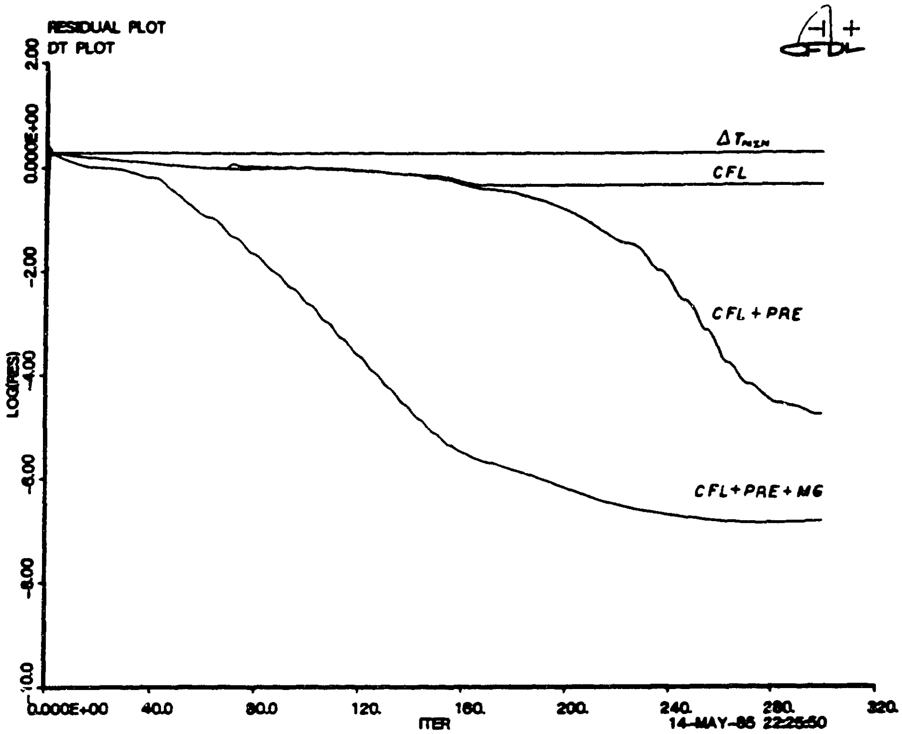


Figure 6-19: 1 - D Reacting Convergence History Plot - Case 2

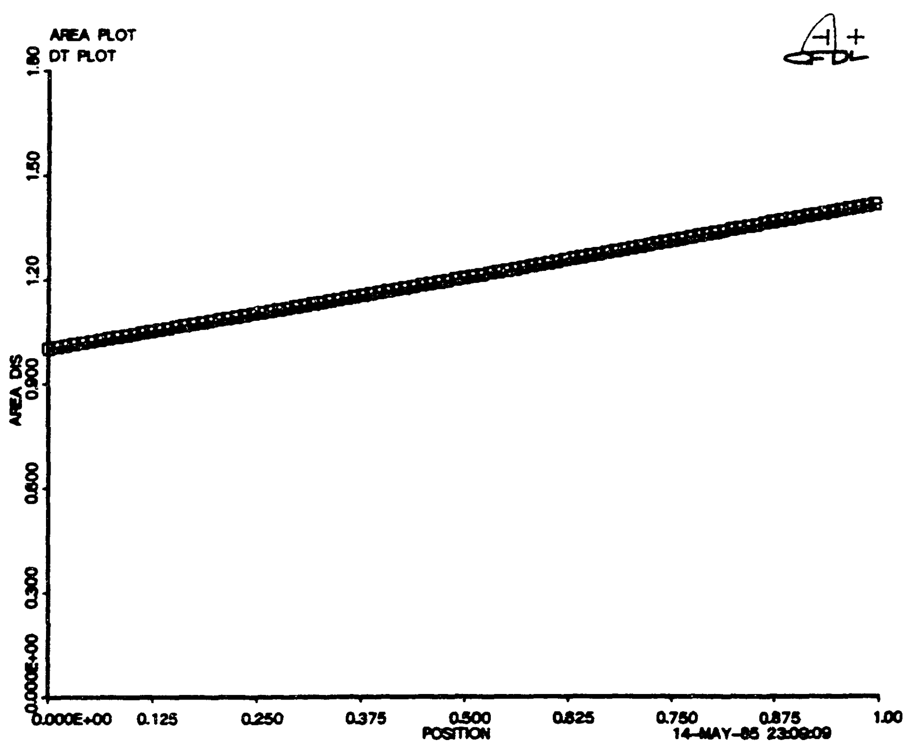


Figure 6-20: 1 - D Validation Test Geometry

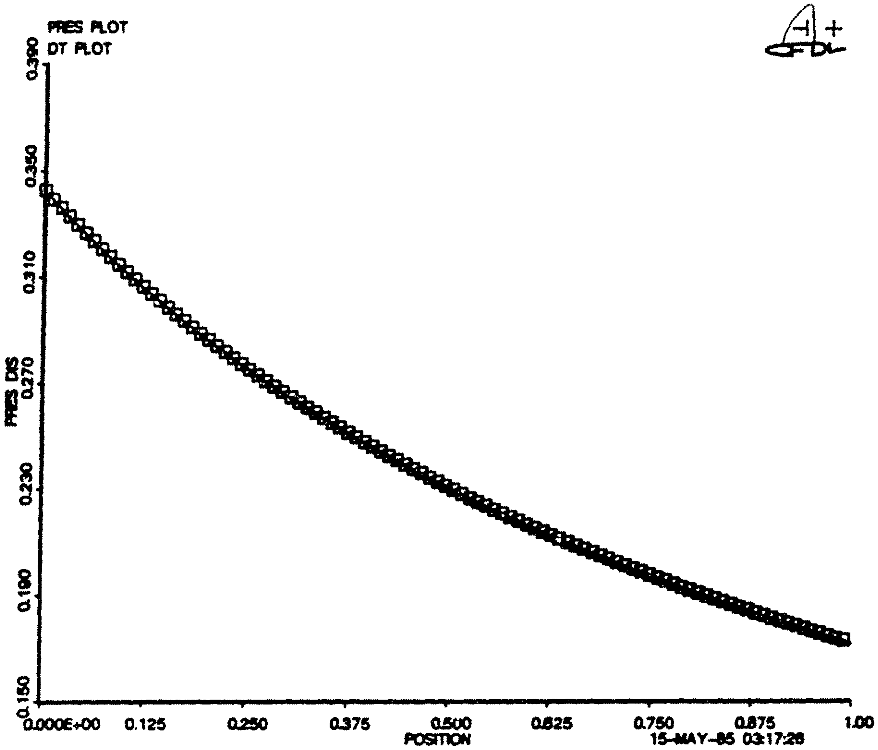


Figure 6-21: 1 - D Non Reacting Pressure Plot

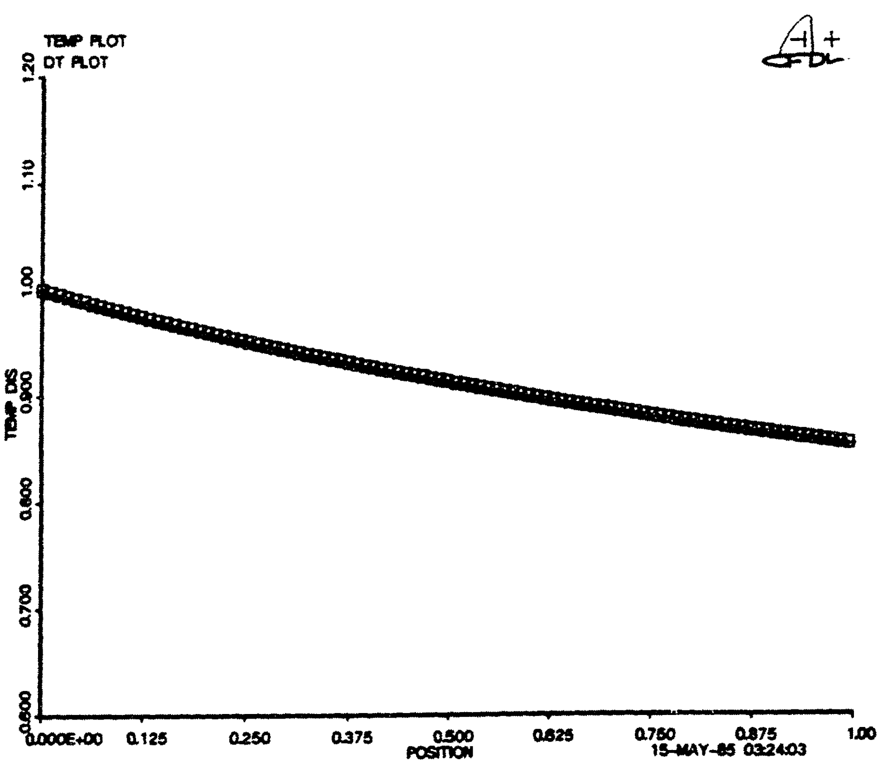


Figure 6-22: 1 - D Non-Reacting Temperature Plot

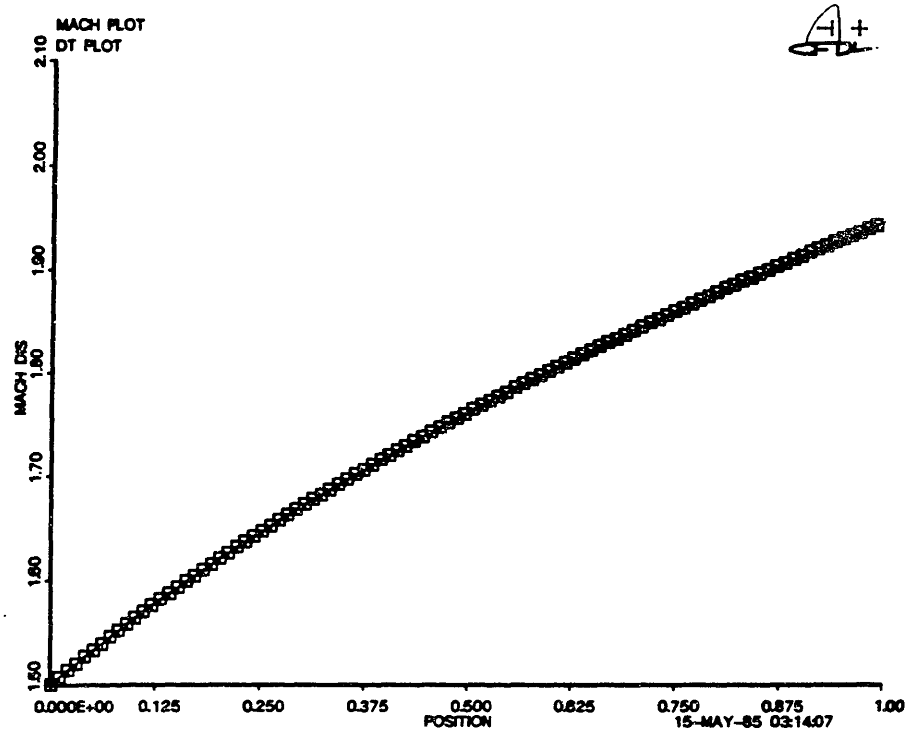


Figure 6-23: 1 - D Non Reacting Mach Number Plot

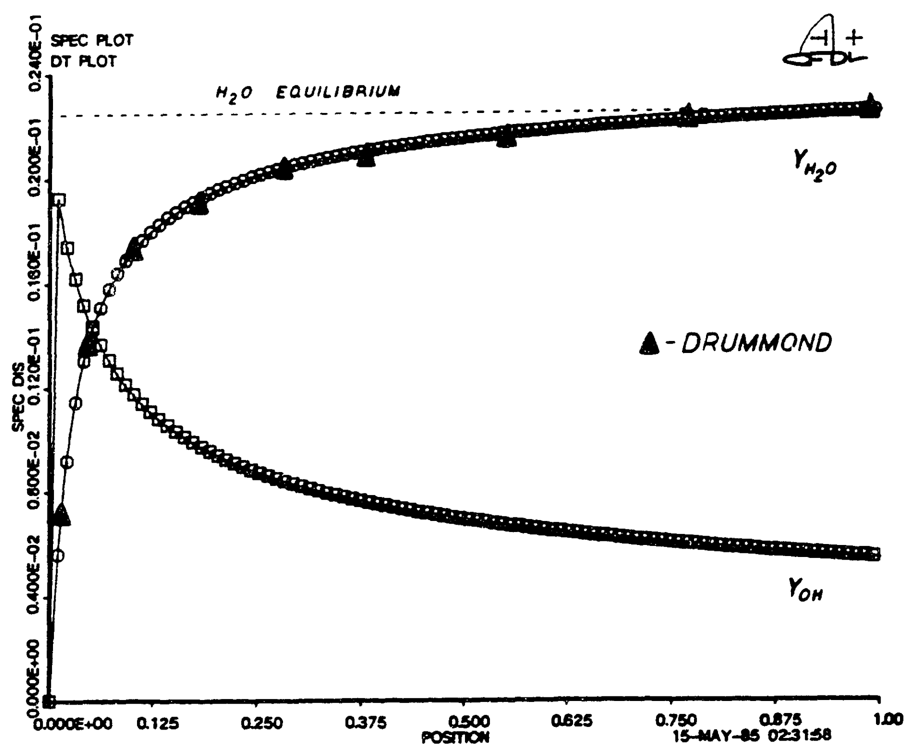


Figure 6-24: 1 - D Reacting H2O and OH Species Plot

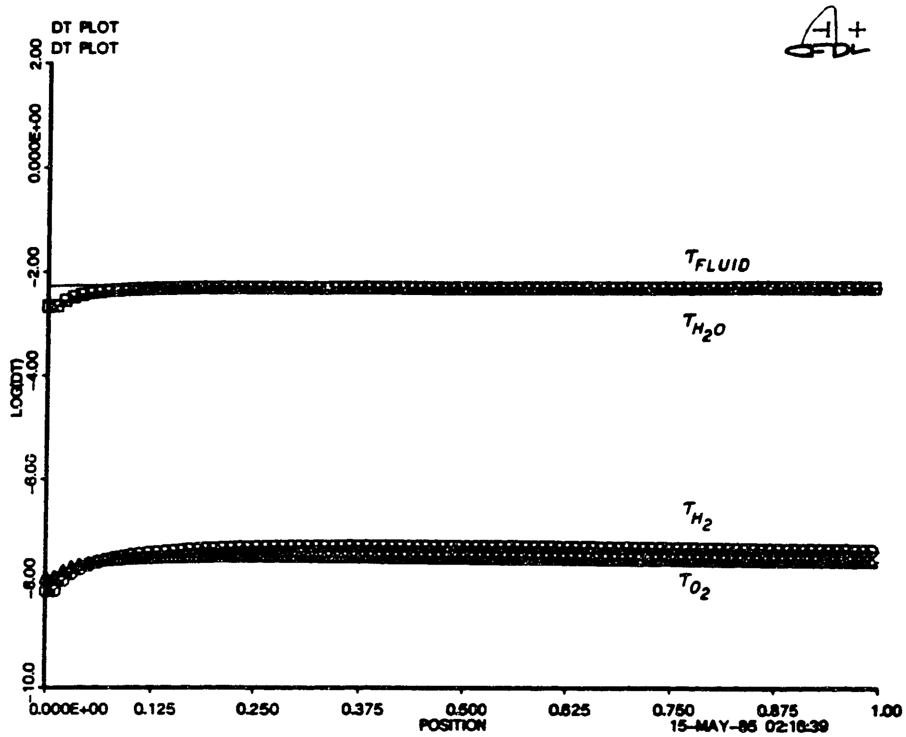


Figure 6-25: 1 - D Reacting Time Scale Plot

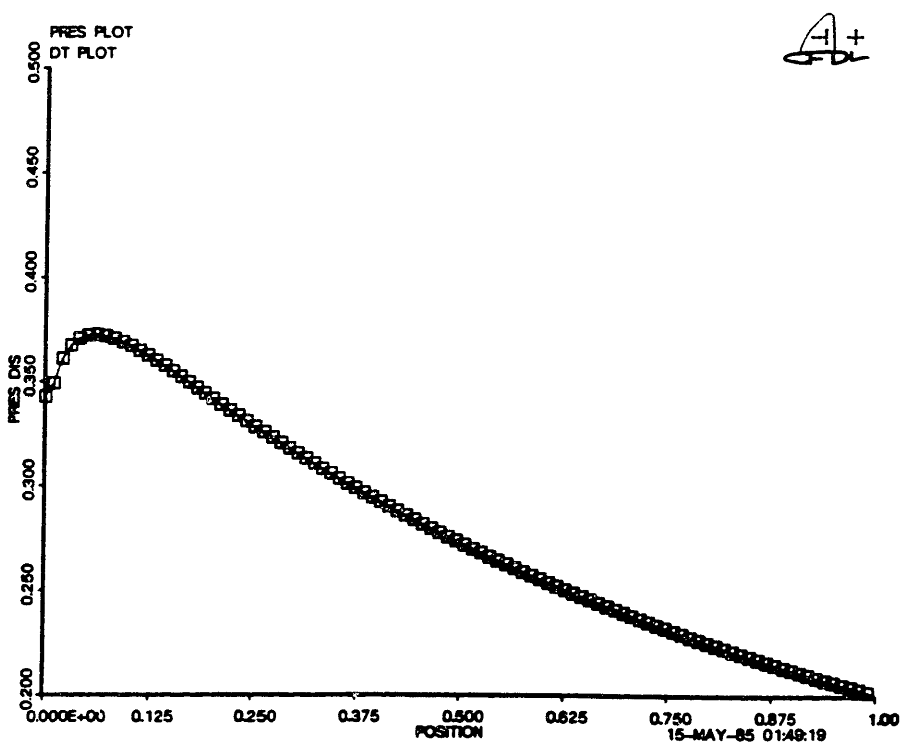


Figure 6-26: 1 - D Reacting Pressure Plot

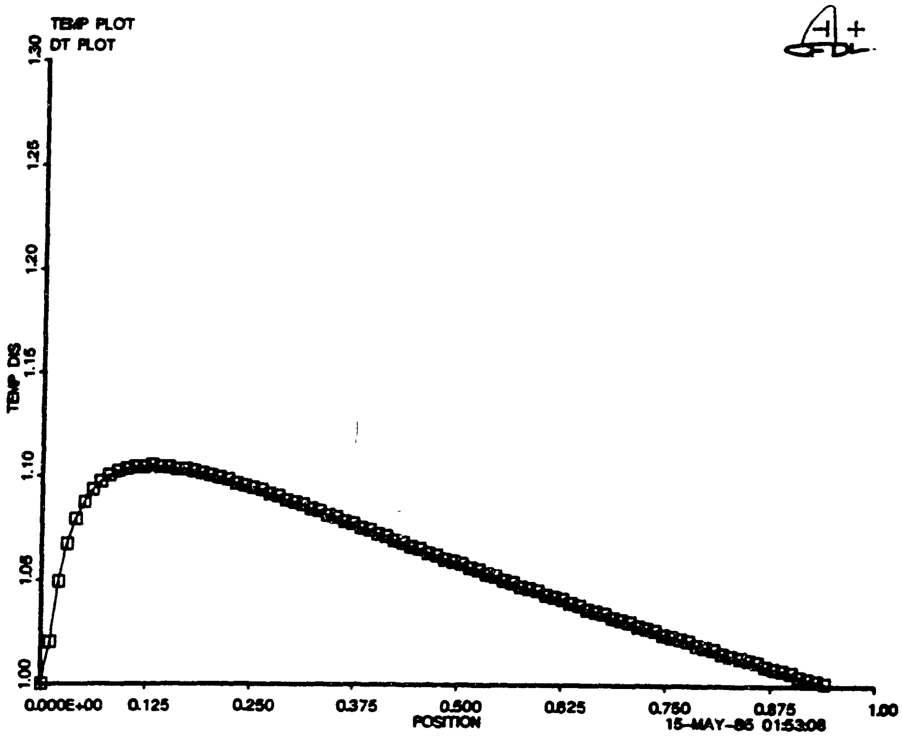


Figure 6-27: 1 - D Reacting Temperature Plot

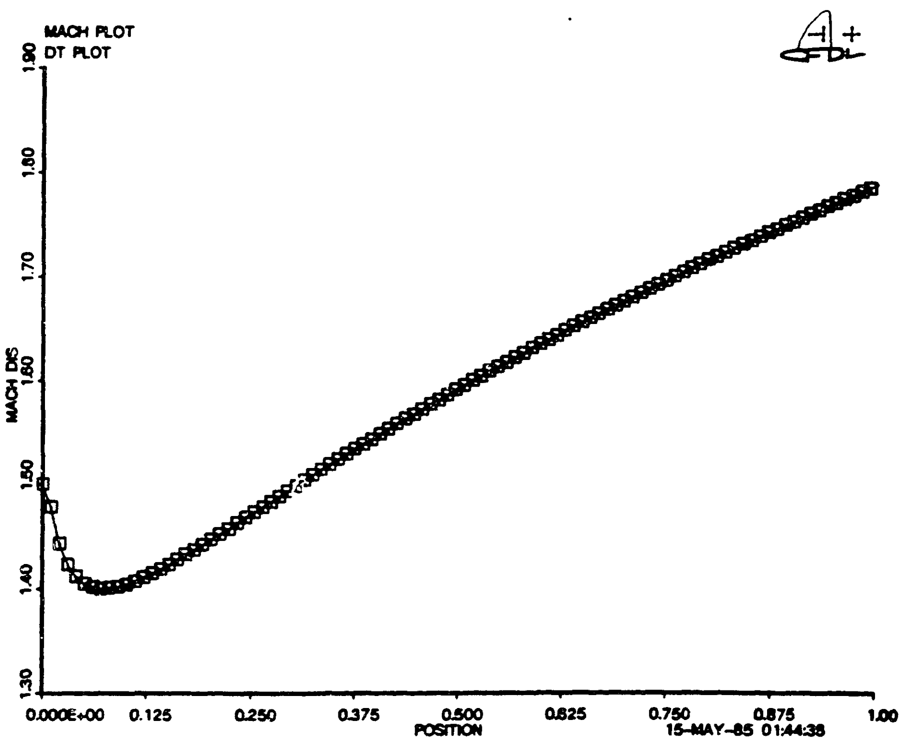


Figure 6-28: 1 - D Reacting Mach Number Plot

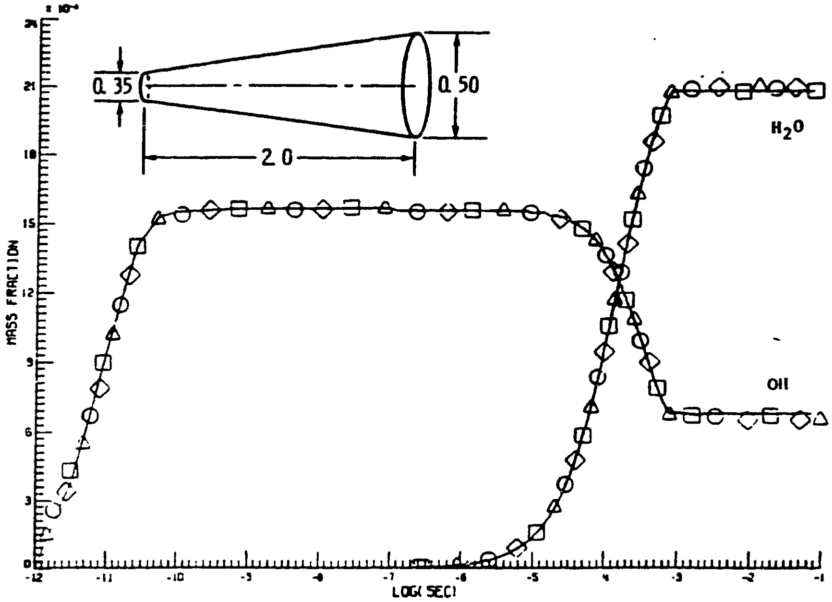


Figure 6-29: Real Time H2O/OH Distributions At The Mid Point Of The Channel

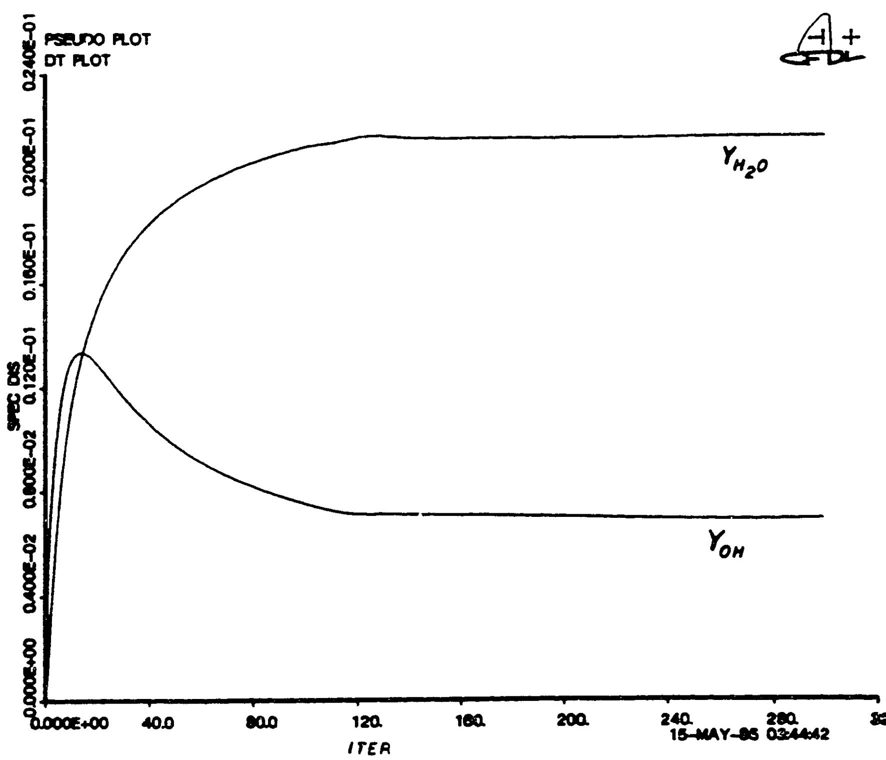


Figure 6-30: Pseudo Time H2O/OH Distributions At The Mid Point Of The Channel

Chapter 7

2-D INVISCID FLOW WITH H_2-AIR CHEMISTRY

In this chapter a special class of flame holders will be investigated. The test geometry is shown in figure 7-1. The geometry was choosen because it can generate an oblique shock of sufficient strength to ignite and hold a reaction zone. The flow field will be simulated using the two dimensional Euler equations with the Roger's and Chinitz [44] chemistry model. Neglecting the viscous diffusion terms means that there are no viscous diffusion flame effects in this problem. For the inviscid calculations the combustion zone will be refered to as a reaction zone. In the next chapter, where the viscous terms are included in the computations, the combustion zone will be refered to as a flame. The remainder of this chapter will be devoted to studying the effects of combustion and heat release on the flow field. In particular it will be shown how the flow field character can be changed from a fully supersonic flow to one with embedded subsonic zones or to a fully subsonic flow field. The ability to predict all three types of flows are of interest in scramjet design.

The two dimensional equations (equations (2.41) and (2.42)) were solved using the chemical time preconditioned Jameson, Schmidt and Turkel scheme with the CFL number kept constant. See chapters four and five for a description of these techniques. The computer written for this problem is given in appendix 2.

7.1. 2-D Inviscid Flow No Reaction

The non-reacting flow field can be validated by comparing the computed solution to two dimensional inviscid theory. For the geometry given by figure 7-1 and the fluid property data given in table 7-1 the computed non reacting pressure, temperature and Mach number contour distributions are shown in figures 7-2, 7-3 and 7-4. The figures show the primary oblique shock, the reflected oblique shock and the expansion fan. A useful validation is to compare the computed to the theoretical upper wall pressure. The comparison is shown in figure 7-5. With the exception of some smearing of the shock, which is characteristic of finite difference schemes, the computed solution compares well with the theoretical solution. The convergence history plot, 7-6, shows that the unsteady term, $\partial U/\partial t$, was reduced to 10^{-12} in approximately 500 iterations. Note the curve levels off after 500 iterations corresponding to the Cyber 205 machine zero. For the 60 $\times$ 60 grid calculation the computer time required per iteration was approximately 1 CPU second. The computations were performed on the Langley Cyber 205 computer. The run times could be substantially reduced if the code were vectorized. However for this thesis no attempt was made to vectorize the code.

7.2. 2-D Inviscid Flow with H_2-Air Chemistry

The flame holding properties of the ramp geometry could have an important application to scramjet engine design. In this section we will show how these devices can anchor a flame (Reaction zone). In addition the heat release produced by the reaction zone can change the character of the local flow field. For example the release of heat can produce locally embedded subsonic zones or with higher levels of heat release the flow can become thermally choked. A convenient way of classifying these different situations is to group them together on a ϕ vs channel inlet Mach number map. ϕ is equal to the fuel to air ratio divided by the stoichiometric fuel to air ratio. ϕ and M are good plotting variables since they indirectly represent thermal and kinetic energy. ϕ and M were chosen as plotting variables instead of the thermal and kinetic energy because they are known before the calculation. The map can be used as a design tool in deciding what type of flow field will be produced for a given ϕ and channel inlet Mach number.

7.2.1. 2-D Heat Release

In this subsection three different examples of heat release over ramps will be investigated. The examples include the case where the reaction zone thickness is small compared to the geometric length scale, where the reaction zone thickness is of the order of the geometric length scale and the case where the primary oblique shock temperature rise is reduced to below the fuel ignition temperature by the expansion fan. Note for the first two cases the primary oblique shock is not affected by the expansion fan. In all of the cases considered, the mixture entering the channel is premixed hydrogen and air.

We will begin by considering how the character of the non-reacting flow field can be changed with chemical reaction. The first of the cases mentioned above will be used to illustrate the different flow fields possible. The calculations use the geometry shown in figure 7-1. Table 7-2 outlines the fluid data used in the calculations. The reference non reacting pressure, temperature and Mach number distributions are shown in figures 7-7, 7-8 and 7-9. These figures show clearly that the flow is entirely supersonic throughout the channel. If chemical reaction is allowed to occur with ϕ = .1 then the pressure, temperature and Mach number distributions are modified to those shown in figures 7-10, 7-11 and 7-12. The figures show that the flow is still supersonic and that the primary oblique shock has moved forward approximately 10% of the channel length. Note that the shock reflection of the lower wall is still regular. Figures 7-13 and 7-14 show the species contours produced by chemical reaction. Both figures verify the original idea that the reaction is triggered by the oblique shock. The figures also show that the OH reaction zone is thin compared to the H_2O reaction zone. Finally the solution converged in approximately 600 iterations, figure 7-15.

Increasing ϕ to .24 leads to a flow field fundamentally different from the one considered above. Figures 7-16, 7-17, 7-18, 7-19 and 7-20 show the contours of pressure, temperature, Mach number, H_2O density fraction and the OH density fraction respectively. The flow field is different from the previous case in that it contains an embedded subsonic zone. The subsonic zone is located behind the Mach stem formed at the base of the oblique shock. The minimum Mach number produced for this case is .9. The Mach stem has moved foward by 50% of the channel length compared to the non-reacting case. The embedded subsonic can be made to fill most of the channel by increasing ϕ to .35 . In this case the minimum Mach number is reduced to .79 . The contours of Mach number and H_2O density fraction for this case are given by figures 7-21 and 7-22. The size of the subsonic zone increases with increasing heat release. Finally the figures show that the reaction zone coincides with the shock location.

If we increase the value of ϕ still further then another class of interesting flow fields can be produced. These flow fields are characterized by heat addition levels high enough to thermally choke the flow [5]. It appears that thermal choking for this example occurs when the embedded subsonic zones extends from the lower wall to the upper wall. Thermal choking creates a normal shock in the channel. The normal shock is unstable due to the converging area of the channel and is pushed out of the front end of the channel. Under these conditions the flow is entirely subsonic. Figure 7-23 shows the Mach contours for the evolution stage just before the normal shock would be forced out of the inlet. In this simulation the inflow boundary conditions were held fixed and thus these figures do not represent a real solution but serve to illustrate the consequences of thermal choking. It is interesting to note that the quasi one dimensional equations could also be used to predict the onset of thermal choking since the flow is essentially one dimensional. See Bussing [5] for examples of such a calculation. Note, there could be a stable transition solution for the case where the flow just becomes thermally choked. However this case was not seen computationally.

Consider now the second class of ramp problems discussed in the beginning of this section. So far we have looked at cases where the reaction zone thickness was much smaller than the channel length. For the previous examples

$$L_{\text{reaction zone}} / L_{\text{channel length}} = .1 \quad .$$

If the channel length is reduced but the flame thickness is kept constant then a different flow behavior can be expected. If we redo case one with ϕ=.1, but reduce the geometric length scales to 10% of their original values then the following results. The results for this second case will now be discussed. The channel bump length is now .1 L_B , where L_B is the case one channel bump length. Note, in the original case the primary oblique shock was shifted foward by 10% of the channel length. The contours of pressure, temperature, Mach number, H_2O density fraction and OH density fraction are shown in figures 7-24, 7-25, 7-26, 7-27 and 7-28. The figures show that the pressure, temperature and Mach number contours look the same as those of the non-reacting case. In this case heat release takes place far downstream and has no effect on the fluid mechanics. Note the reaction zone thicknesses are now an order of magnitude thicker. Thus with the same level of ϕ as in case one a different flow field is produced. A thicker reaction zones implies that heat addition is delayed and takes place further downstream from the point of ignition. Thus care must be taken to match the reaction zone length scale and the channel length scale to produce the desired behavior.

The last of the three cases mentioned at the begining of this section will now be discussed. Unlike the previous two cases, this case will assess the the effect of the expansion fan interacting with the primary oblique shock. In this case the interaction could reduce the static temperature rise across the shock sufficiently to extinguish the reaction. To produce this kind of interaction we need to shorten the ramp used in the previous examples. Figure 7-29 shows the new geometry. Note the geometric length scale is the same as that used in case one. Using the same flow conditions described for case one, and with a ϕ=.1, the contours of pressure, temperature, Mach number, H_2O density fraction, OH density fraction, H_2O reaction rate and OH reaction rate are given in figures 7-30, 7-31, 7-32, 7-33, 7-34, 7-35 and 7-36. These figures show that the reaction zone is limited to a small region of the flow and does not traverse the full channel. Figure 7-35 and 7-36 show where the reaction is actually occuring.

7.2.2. Ramp Heat Release Characterization Map

In the previous section it was shown that chemical reaction in a ramp geometry can produce a variety of fundamentally different flow fields. If they could be characterized in a way such that all possible flow fields could be represented on a single figure then, the designer would have a quick way to design these devices. It is suggested that one way to represent the various flow fields is to map them according to the type of flow field, on a ϕ vs channel inlet Mach number plot. With these variables the suggested map for the ramp channel geometry is given by figure 7-37. The map can be divided into different regions identified as follows: region 1 corresponds to fully supersonic flow with no chemical reaction, region 2 corresponds to fully supersonic flow with chemical reaction, region 3 corresponds to supersonic flow with an embedded subsonic region and chemical reaction, regions 4 and 5 are transition cases and region 6 corresponds to the case of thermally choked flow. Chemical reaction does not occur below a minimun inlet Mach number. The minumum inlet Mach number dictates that the static temperature rise across the oblique shock is not high enough for ignition to occur. The map shows there are at least four distinct regions where the flow is fundamentally different, i.e., 1, 2, 3 and 6. The different flow fields generated for case one are represented by the line AB. The results from case two indicate that the position of the lines dividing the various regions will tend to move towards the $M=M_{min}$ line as the ratio of the flame thickness to the ramp length is increased. Similarly, case three suggests that as the ratio of ξ/L (channel height fixed) is decreased the lines dividing the various regions will also tend to move towards the $M=M_{min}$ line. Another possible choice for the abscissa variable is the heat release due to chemical reaction (HR). ϕ was chosen over HR as the abscissa variable because ϕ is known before the calculation is performed whereas HR is not.

The map is intended to suggest a general way of laying out the possible flow fields. A specific map would probably be needed for each fuel, ramp geometry and set of inflow conditions.

Properties	Values	Dimensions
P_∞	1.0×10^5	N/m^2
T_∞	900.	°K
u Velocity	1500	m/s
v Velocity	0	m/s
M_∞	2.5	
c_p	1000.	J/kg·°K
c_v	714.	J/kg·°K
L	1.0	m
Grid	60 × 60	
CFL	1.5	
σ_v	.1	

Table 7-1: Table Of Fluid Data - Validation $\gamma = 1.4$

Properties	Values	Dimensions
P_∞	1.0×10^5	N/m^2
T_∞	900.	°K
$T_{Ignition}$	1000.	°K
ϕ	.1	
u Velocity	1500	m/s
v Velocity	0	m/s
M_∞	2.5	
c_{pH2O}	17160.	J/kg·°K
c_{pOH}	1181.	J/kg·°K
c_{pH2}	2854.	J/kg·°K
c_{pO2}	2041.	J/kg·°K
c_{pN2}	1285.	J/kg·°K
c_{vH2O}	17160.	J/kg·°K
c_{vOH}	1181.	J/kg·°K
c_{vH2}	2854.	J/kg·°K
c_{vO2}	2041.	J/kg·°K
c_{vN2}	1285.	J/kg·°K
Hf_{H2O}	-1.44×10^7	J/kg
Hf_{OH}	2.3×10^6	J/kg
Hf_{H2}	0.0	J/kg
Hf_{O2}	0.0	J/kg
Hf_{N2}	0.0	J/kg
L	1.0	m
Grid	60×60	
CFL	1.5	
σ_v	.1	

Table 7-2: Table Of Fluid Data - H_2 Air

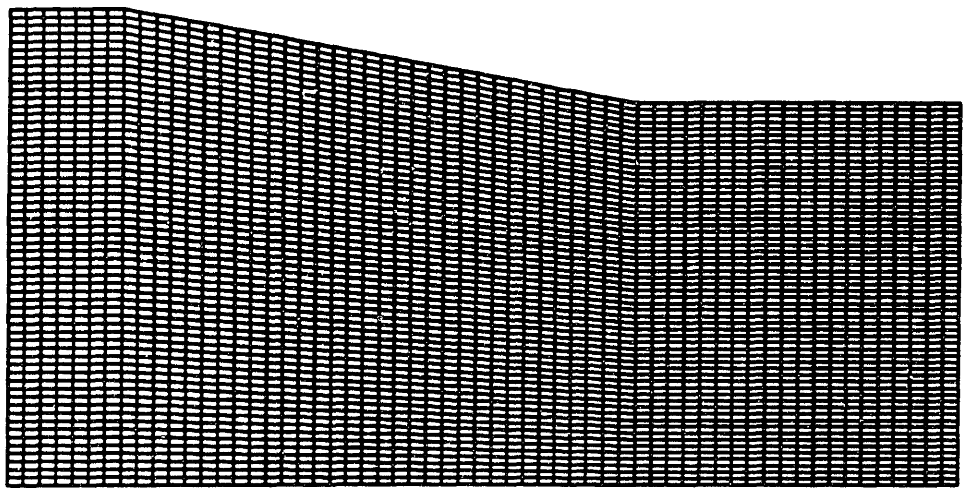


Figure 7-1: Ramp Channel Test Geometry

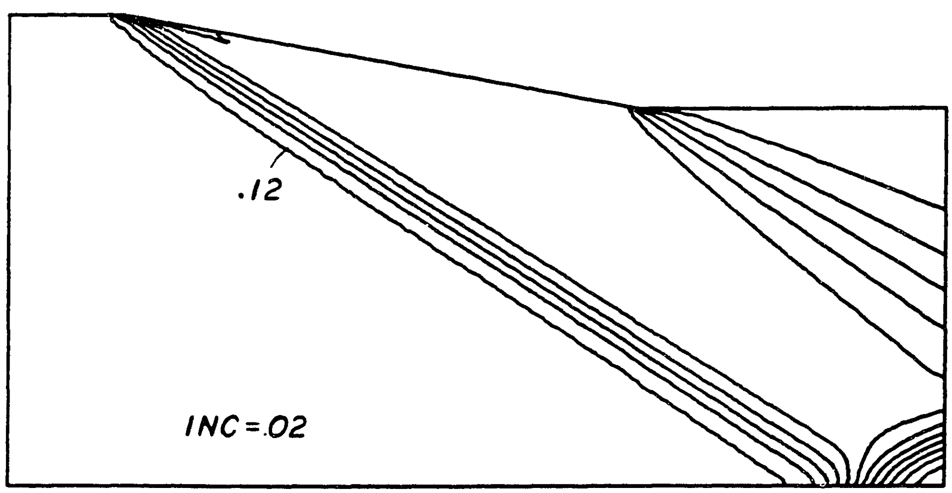


Figure 7-2: Non Reacting Pressure Contour PLot - Gama = 1.4

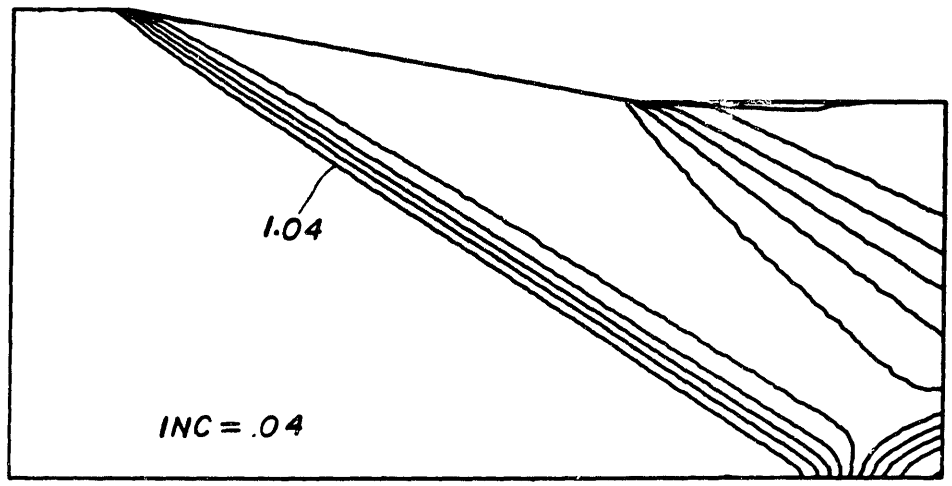


Figure 7-3: Non Reacting Temperature Contour Plot - Gama = 1.4

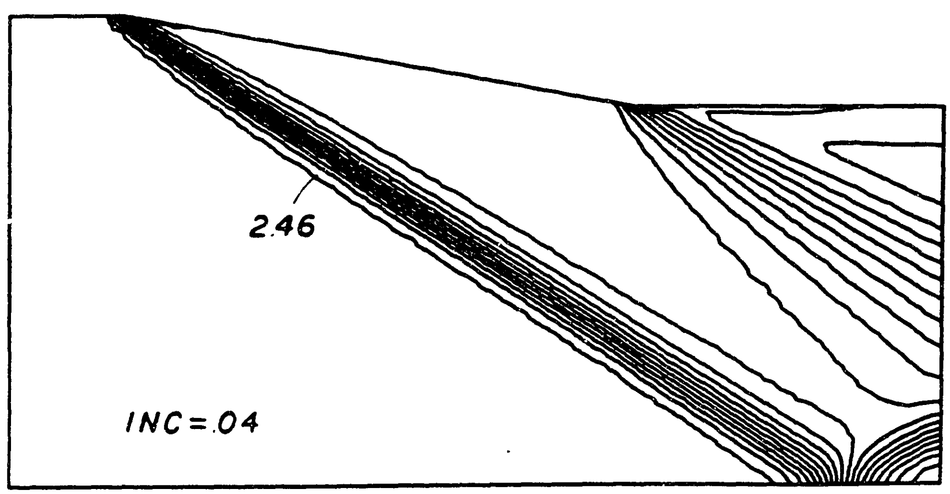


Figure 7-4: Non Reacting Mach Number Contour Plot - Gama = 1.4

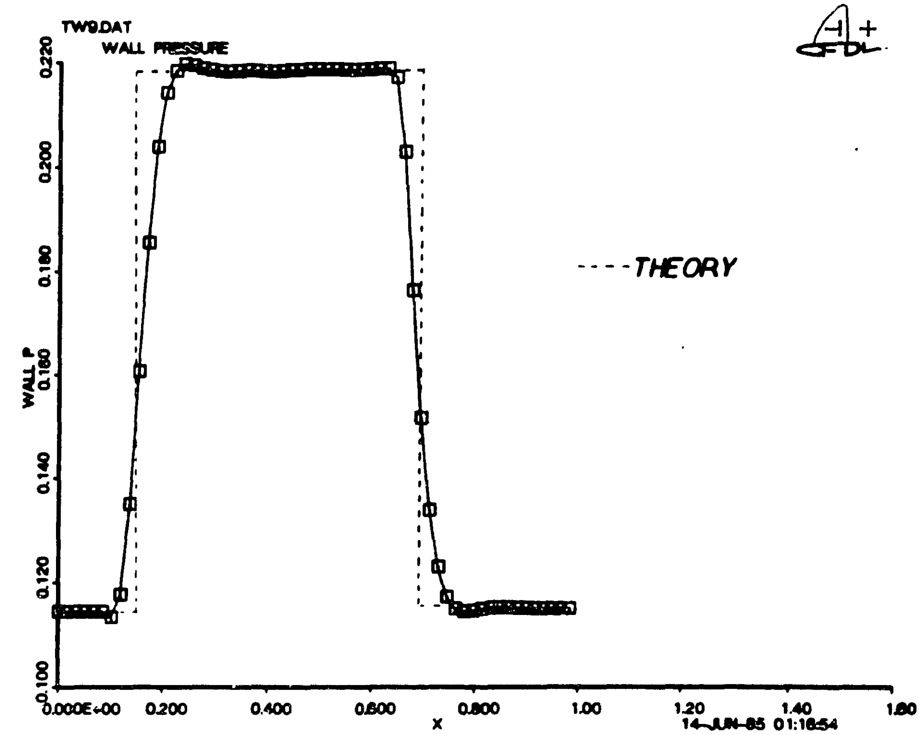


Figure 7-5: Comparison of computed and Theoritical Upper Wall Pressures-Gama=1.4

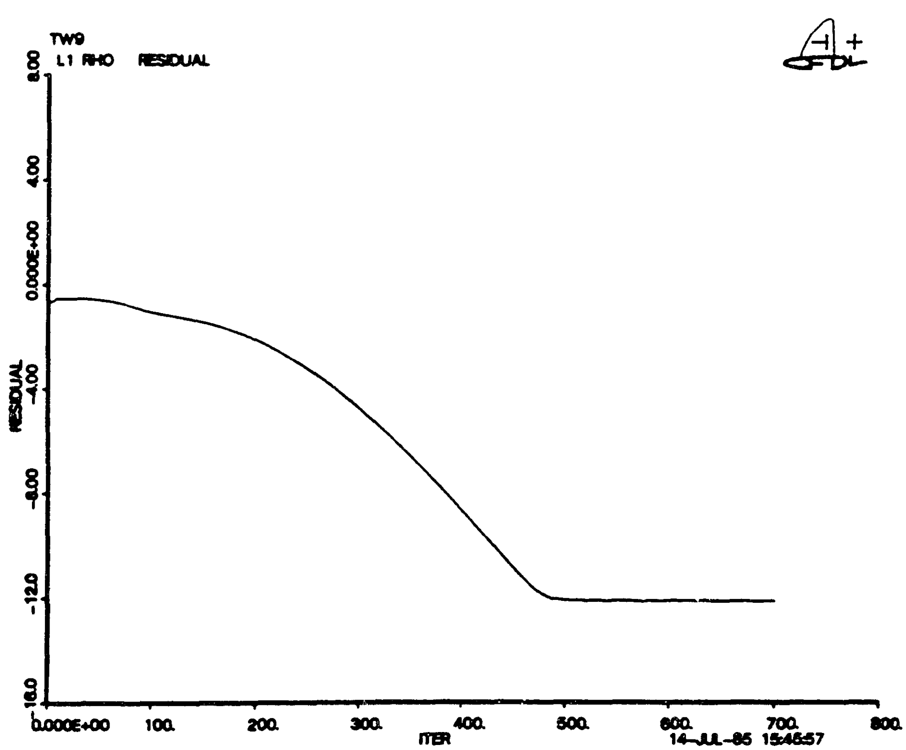


Figure 7-6: Non Reacting Convergence History Plot - Gama = 1.4

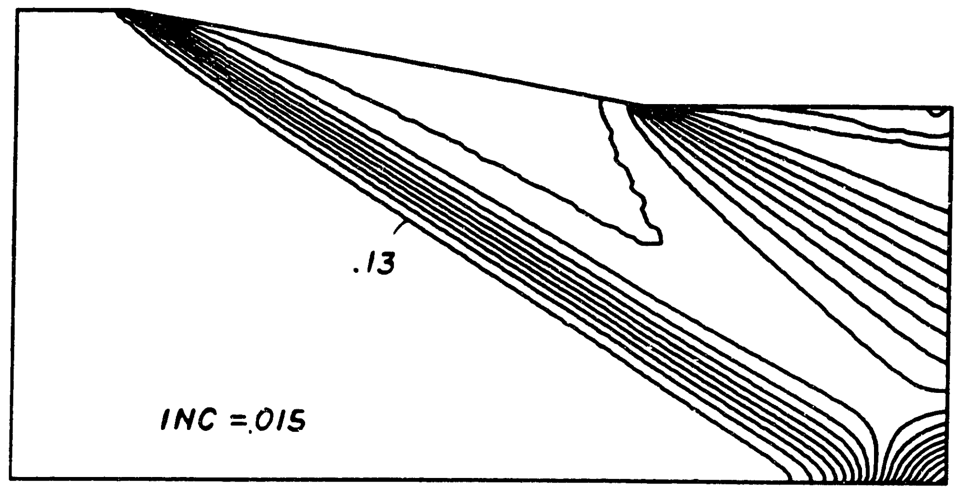


Figure 7-7: Non Reacting Pressure Contour Plot - Phi = .1

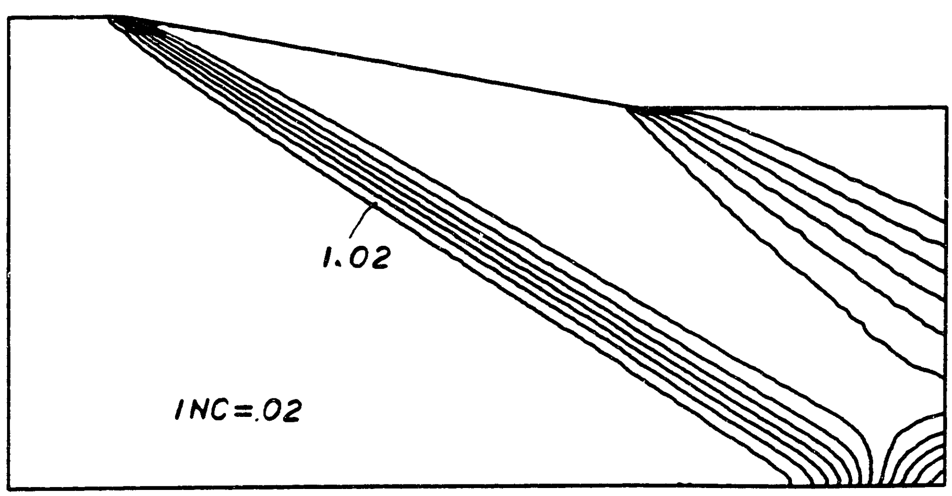


Figure 7-8: Non-Reacting Temperature Contour Plot - Phi = .1

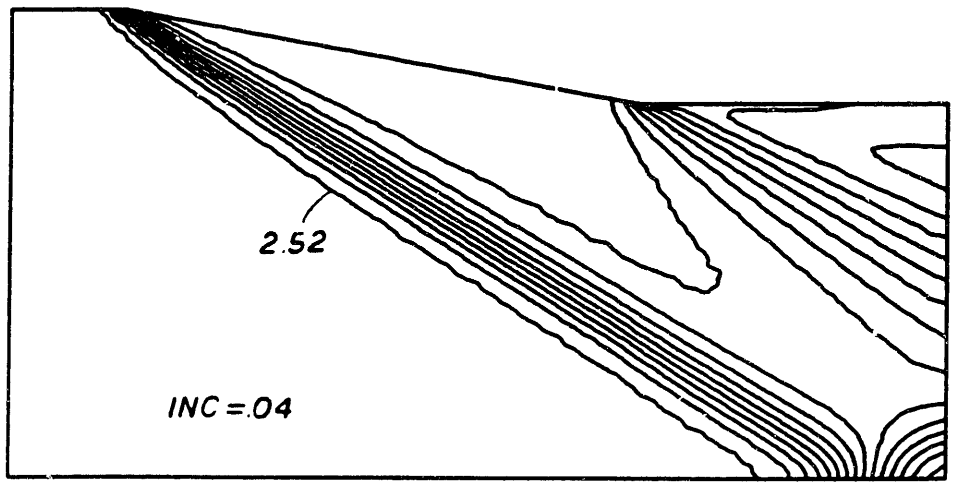


Figure 7-9: Non-Reacting Mach Number Contour Plot - Phi = .1

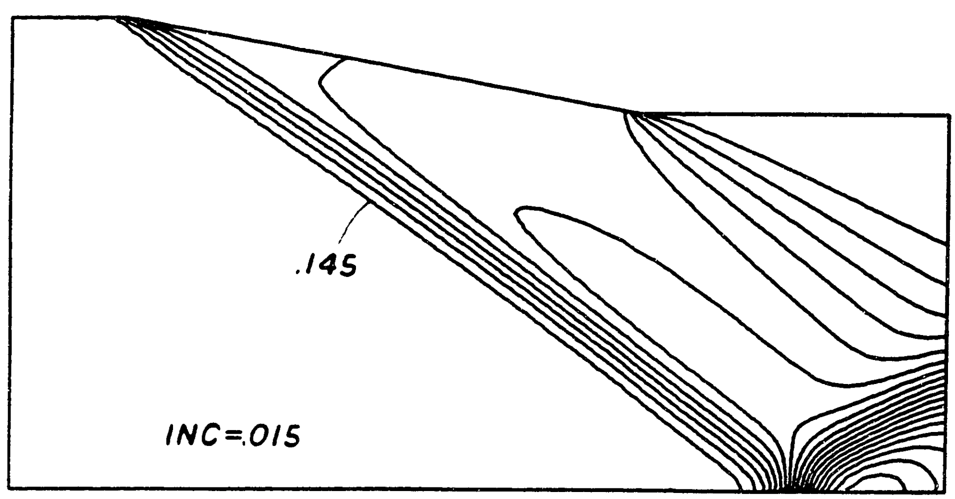


Figure 7-10: Reacting Pressure Contour Plot - Phi = .1

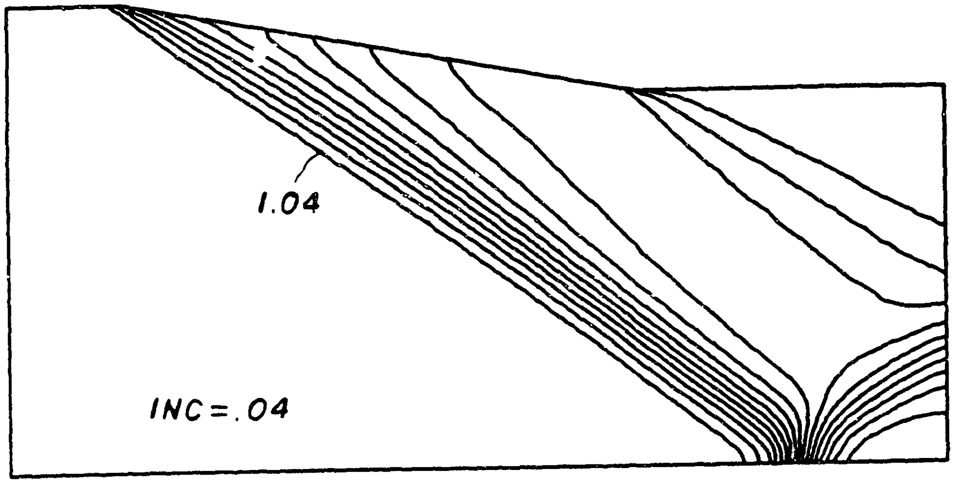


Figure 7-11: Reacting Temperature Contour Plot - Phi = .1

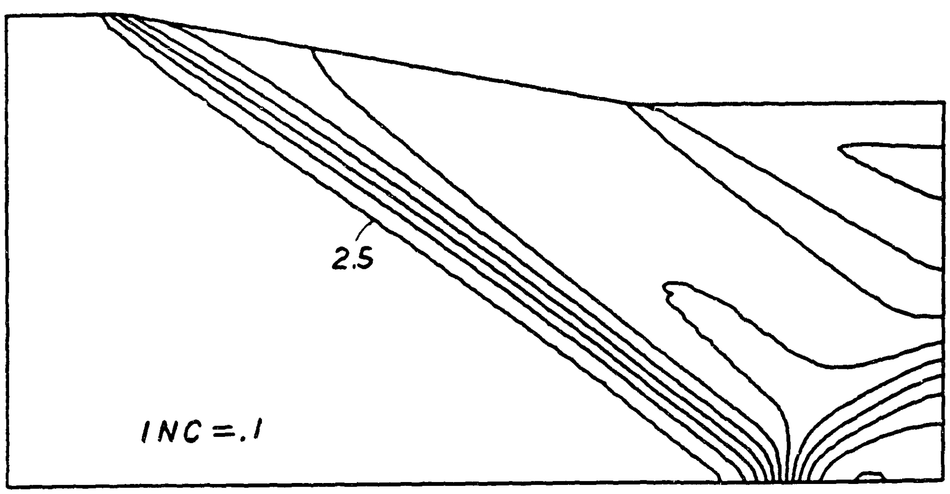


Figure 7-12: Reacting Mach Number Contour Plot - Phi = .1

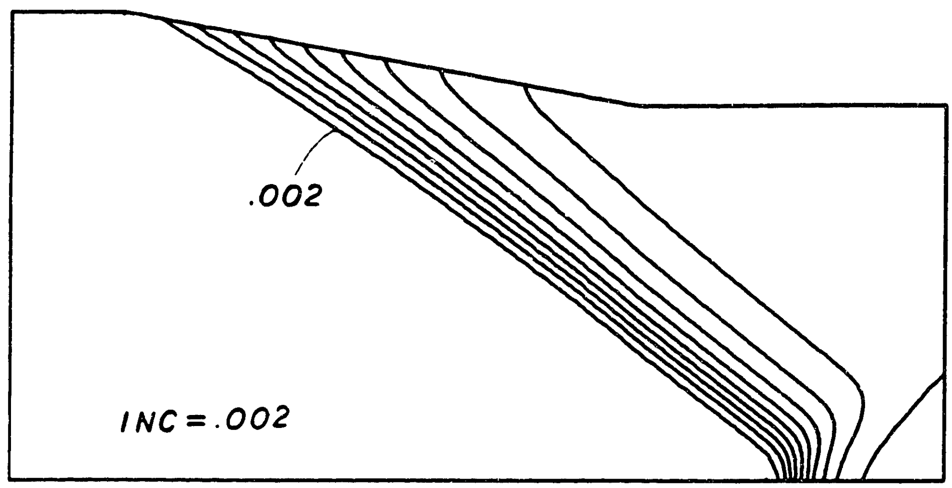


Figure 7-13: Reacting H2O Density Fraction Contour Plot - Phi = .1

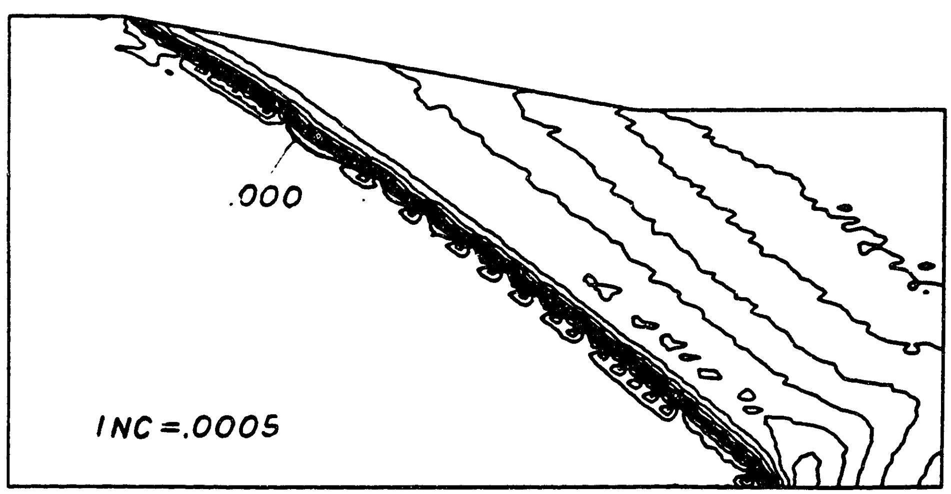


Figure 7-14: Reacting OH Density Fraction Contour Plot - Phi = .1

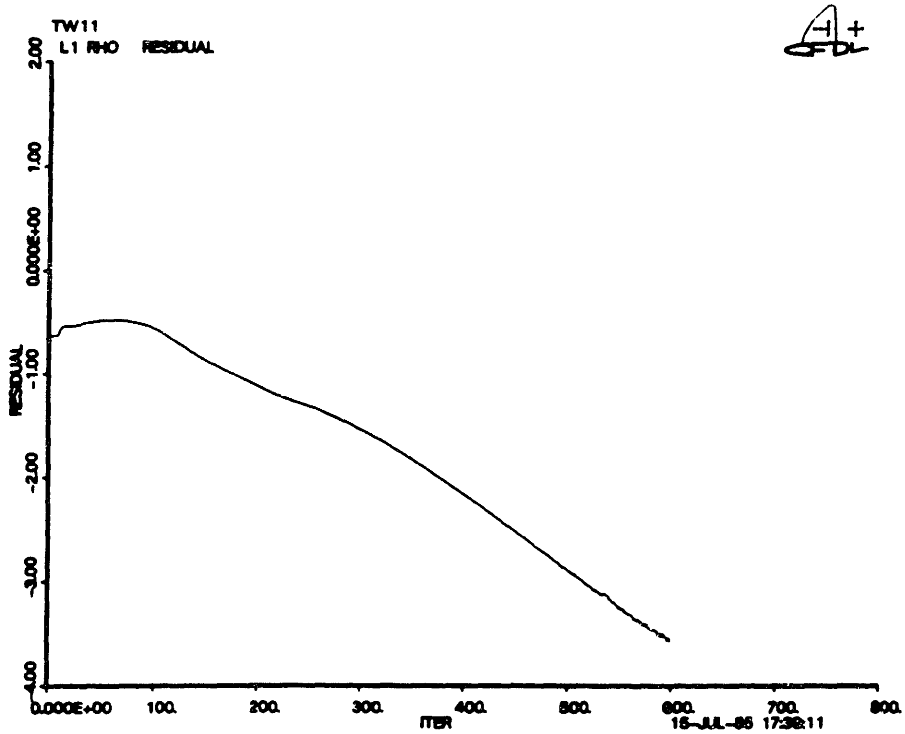


Figure 7-15: Reacting Convergence History Plot - Phi = .1

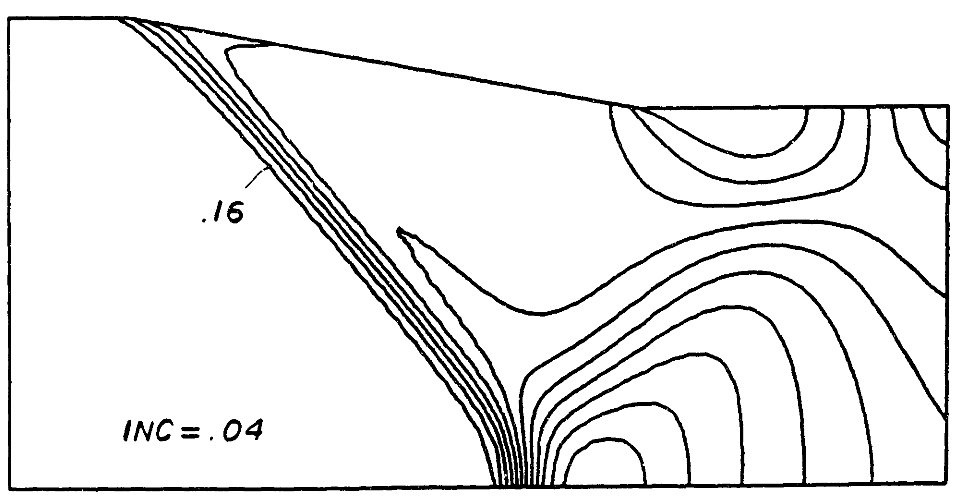


Figure 7-16: Reacting Pressure Contour Plot - Phi = .24

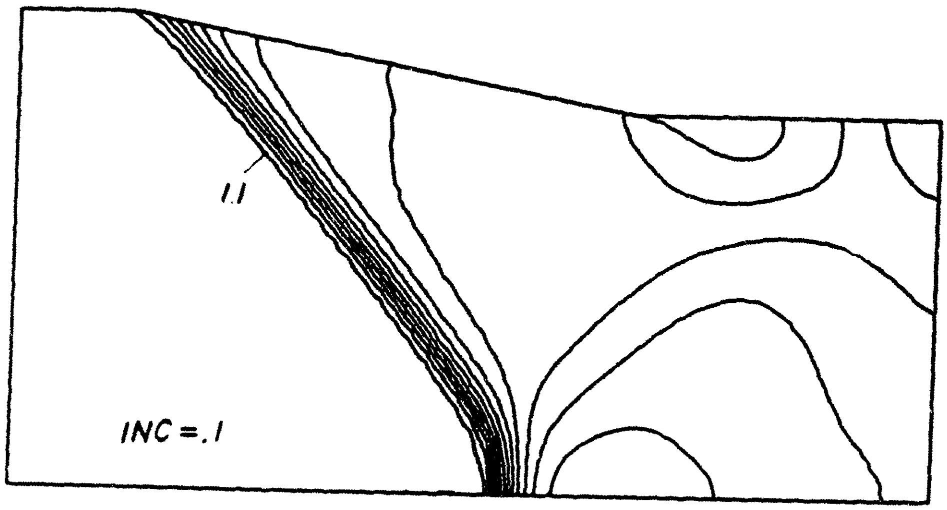


Figure 7-17: Reacting Temperature Contour Plot - Phi = .24

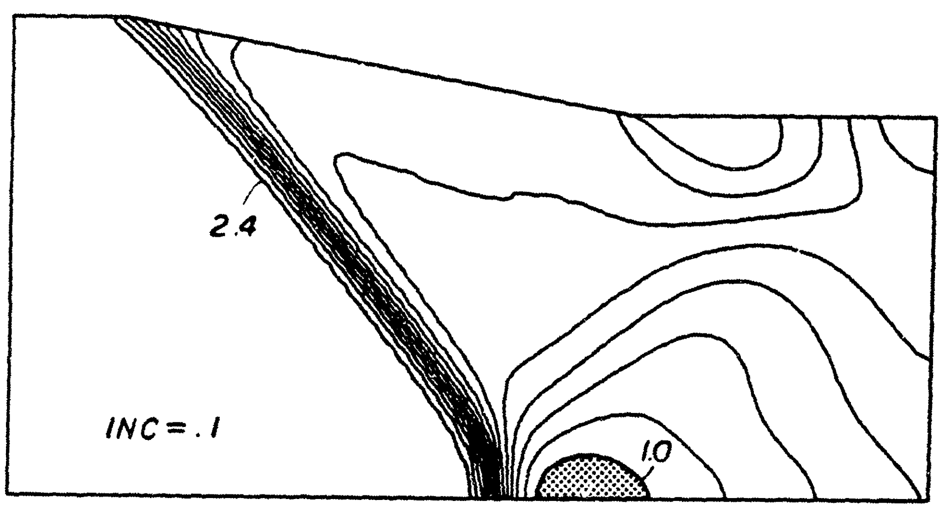


Figure 7-18: Reacting Mach Number Contour Plot - Phi = .24

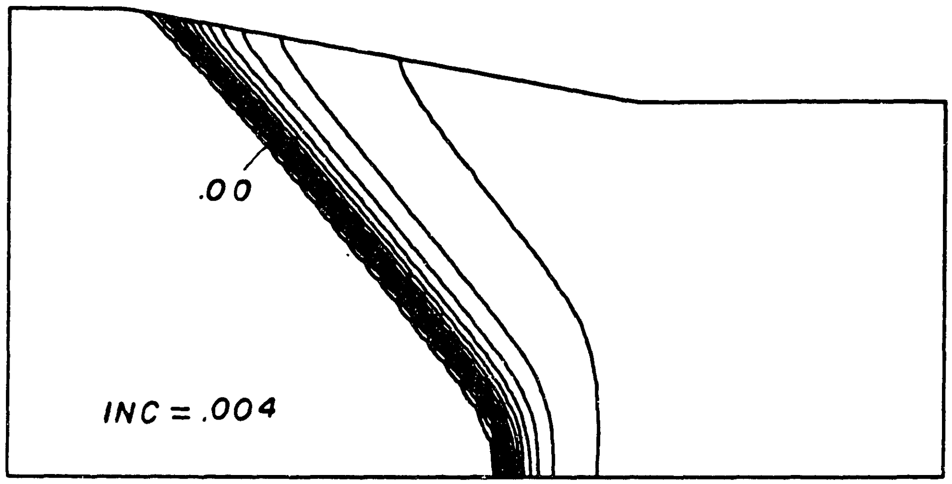


Figure 7-19: Reacting H2O Density Fraction Contour Plot - Phi = .24

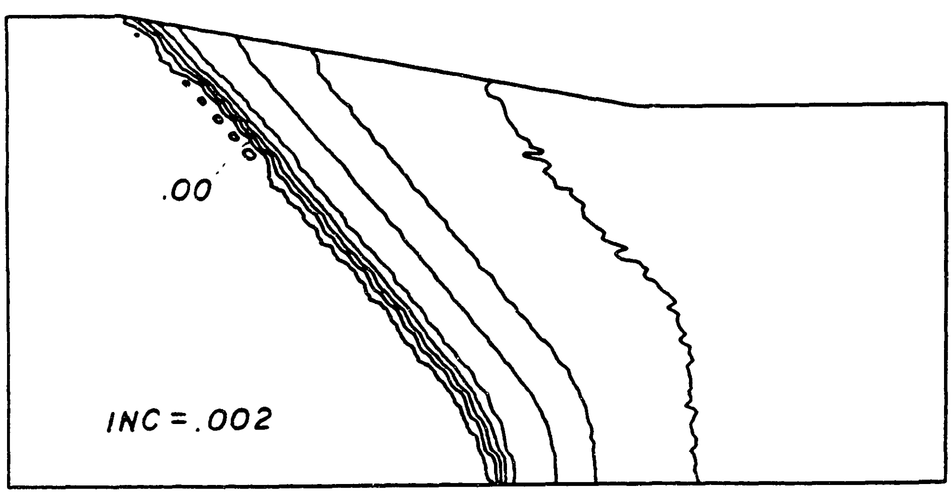


Figure 7-20: Reacting OH Density Fraction Contour Plot - Phi = .24

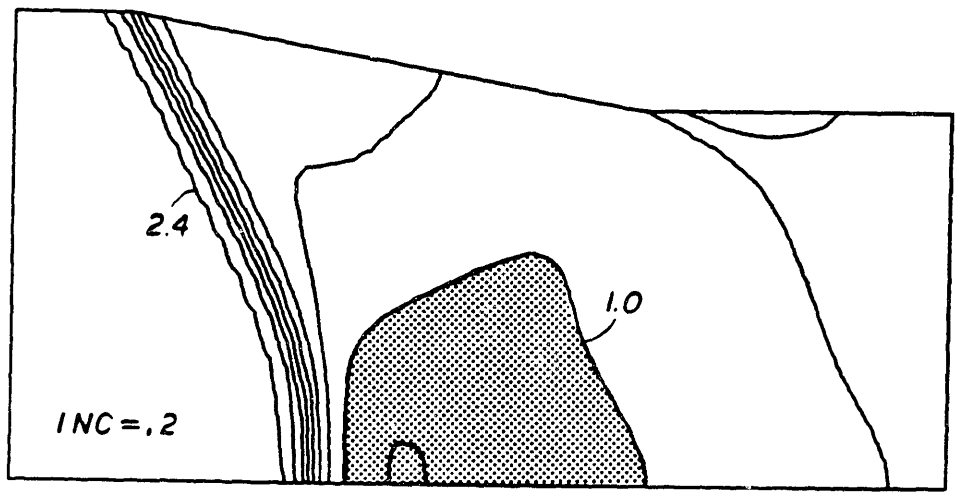


Figure 7-21: Reacting Mach Number Contour Plot - Phi = .35

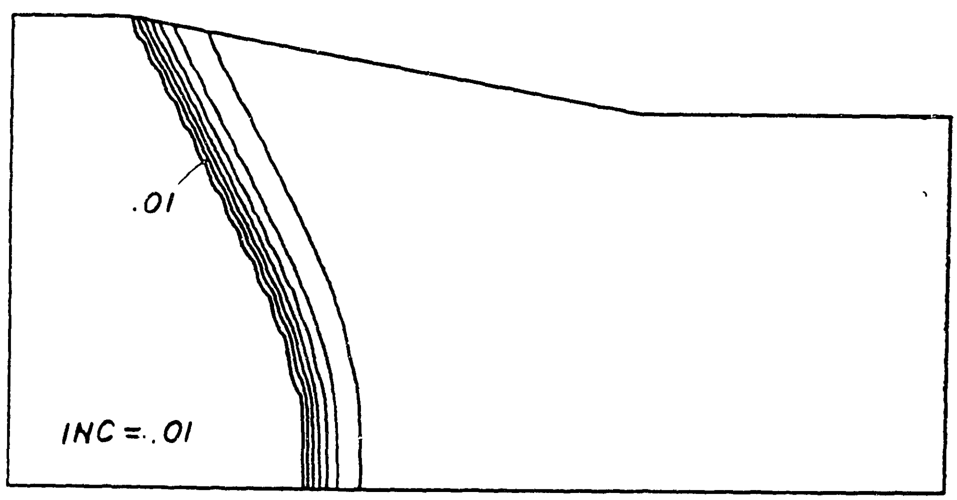


Figure 7-22: Reacting H_2O Density Fraction Contour Plot - Phi = .35

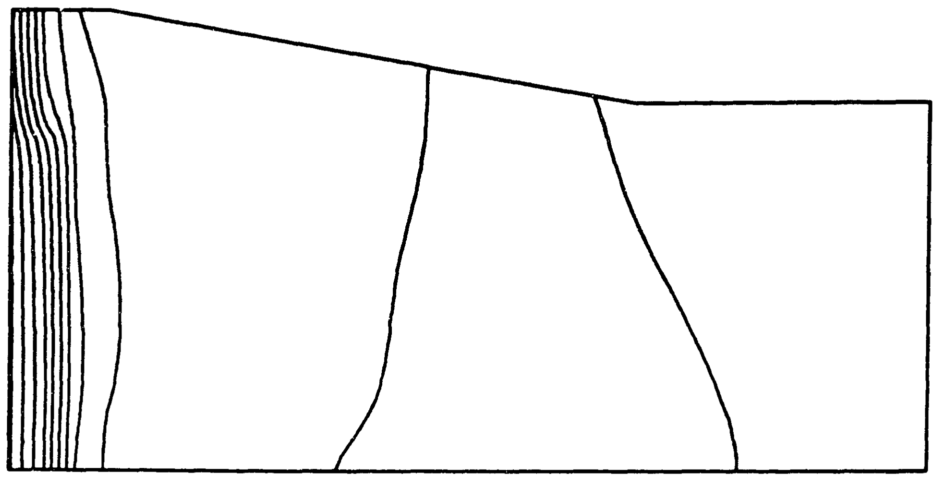


Figure 7-23: Reacting Mach Number Contour Plot - Phi = .5

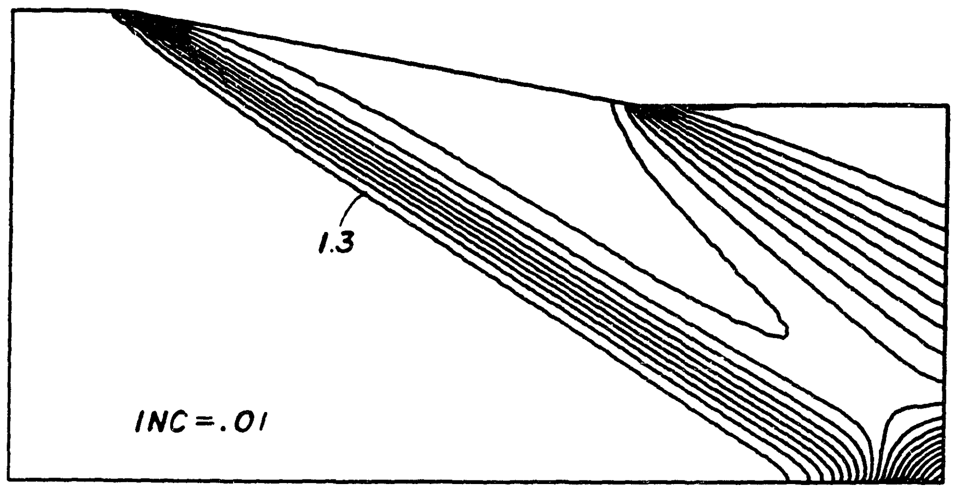


Figure 7-24: Reacting Pressure Contour Plot - Phi = .1

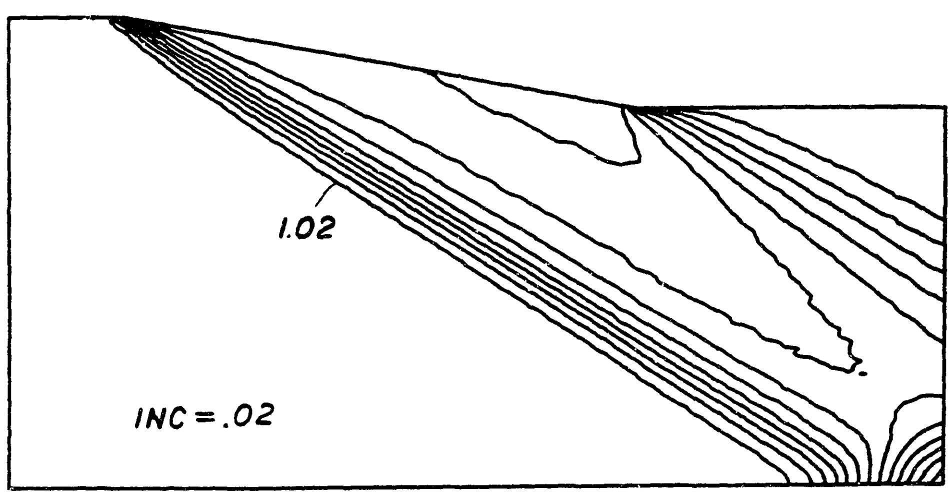


Figure 7-25: Reacting Temperature Contour Plot - Phi = .1

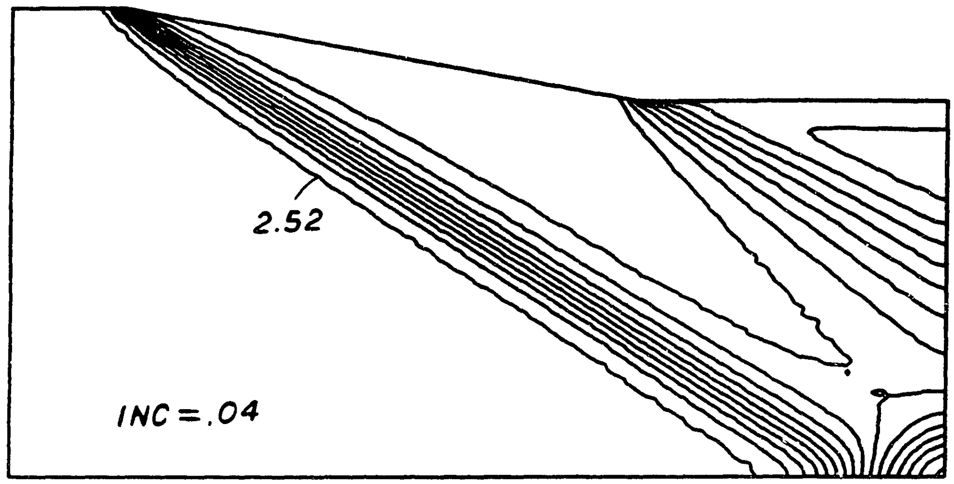


Figure 7-26: Reacting Mach Number Contour Plot - Phi = .1

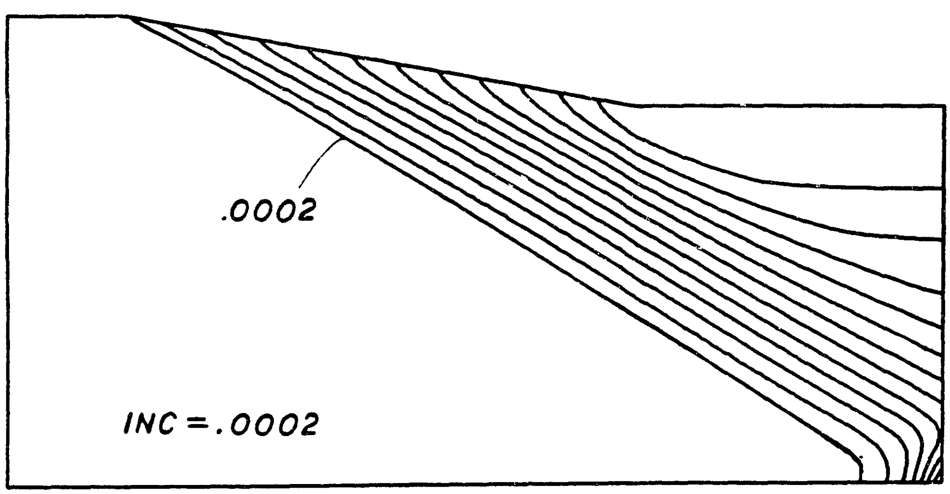


Figure 7-27: Reacting H2O Density Fraction Contour Plot - Phi = .1

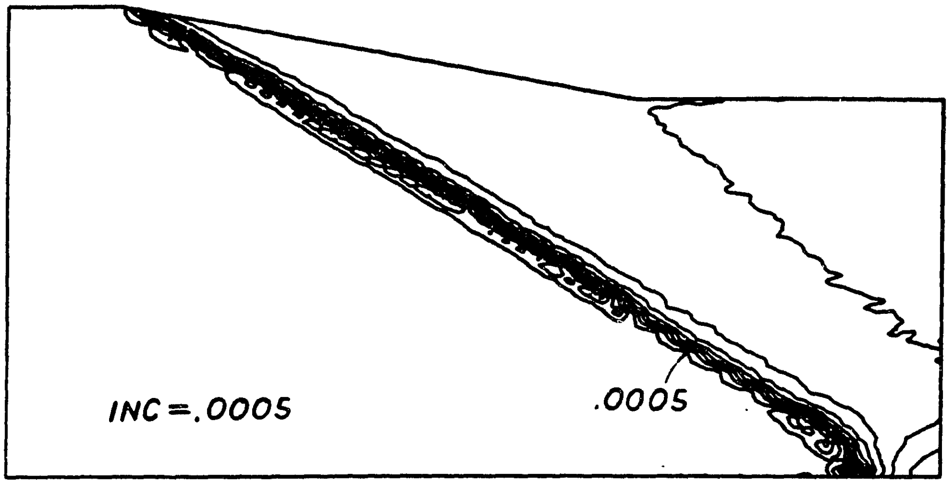


Figure 7-28: Reacting OH Density Fraction Contour Plot - Phi = .1

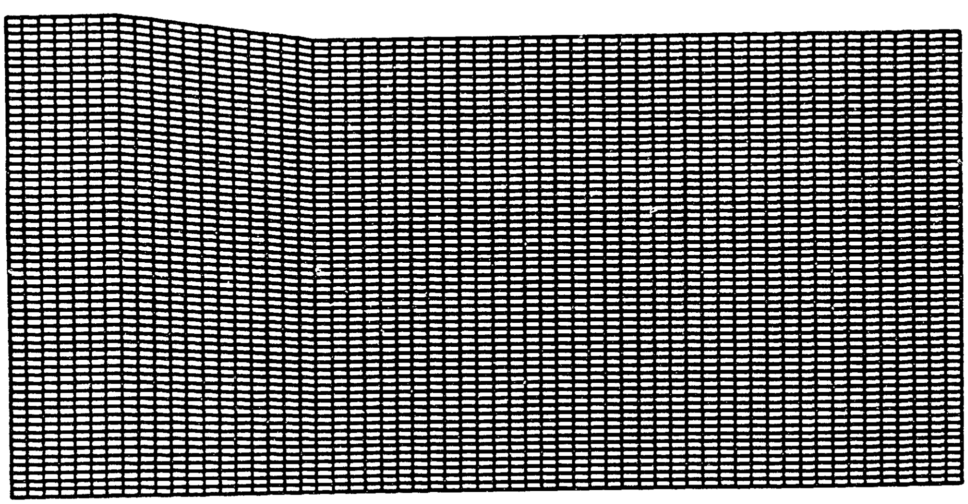


Figure 7-29: Ramp Geometry - Short Case

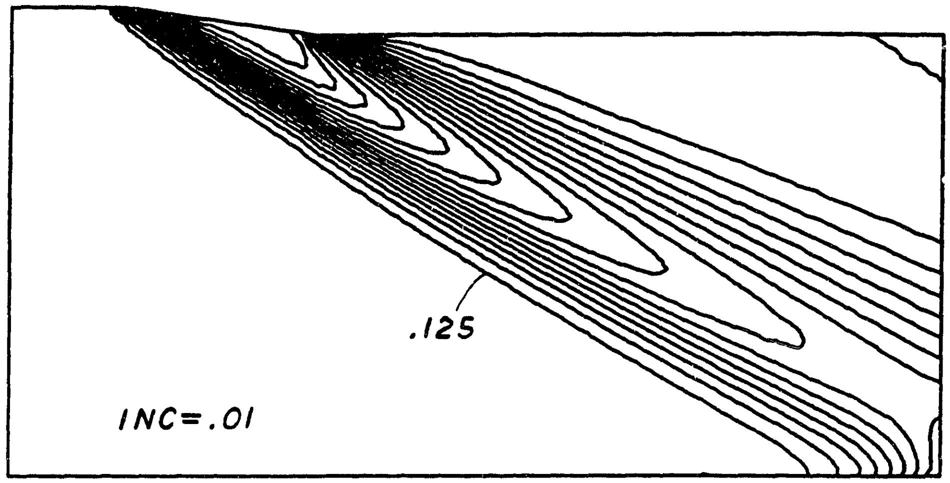


Figure 7-30: Reacting Pressure Contour Plot - Phi = .1

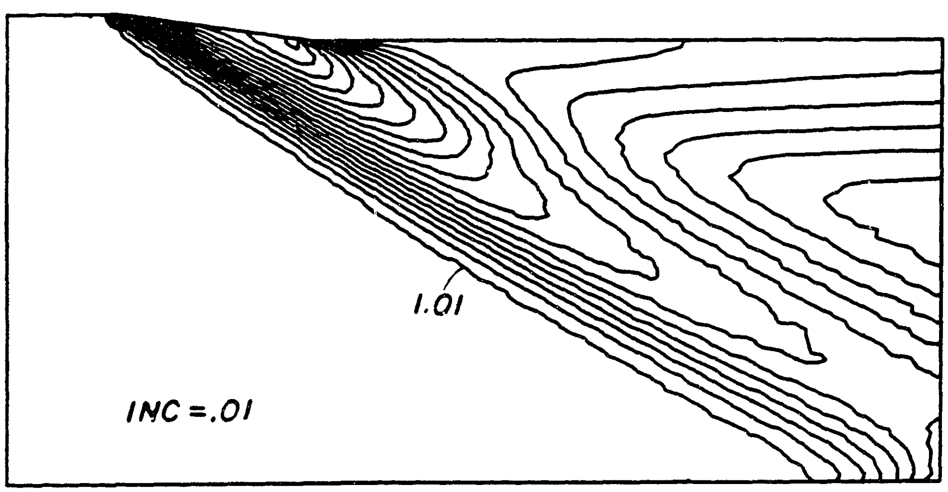


Figure 7-31: Reacting Temperature Contour Plot - Phi = .1

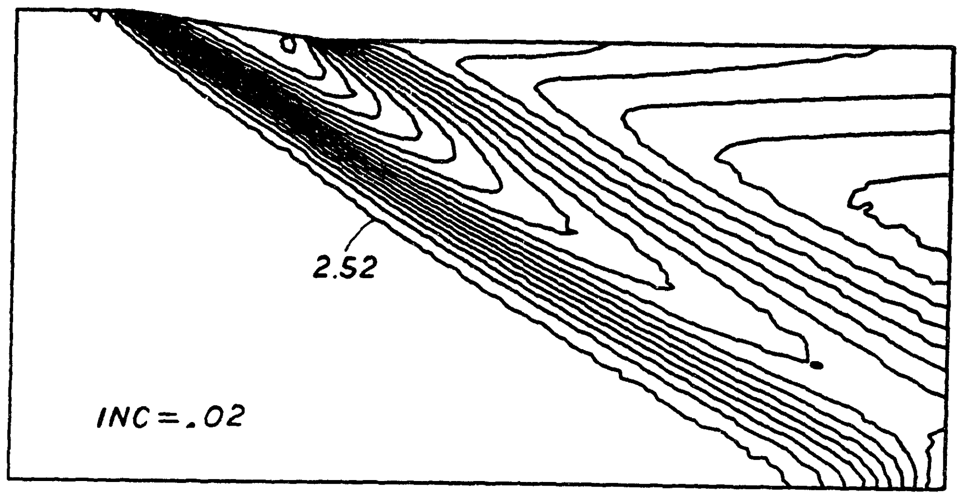


Figure 7-32: Reacting Mach Number Contour Plot - Phi = .1

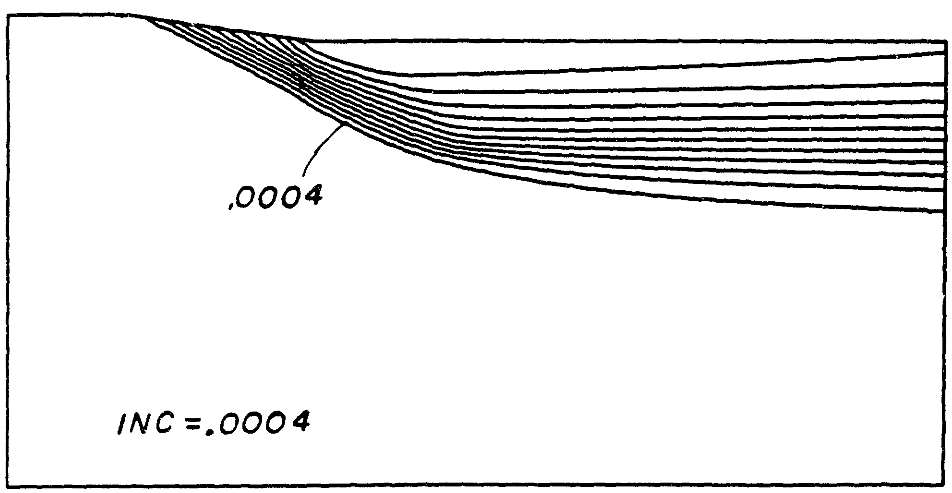


Figure 7-33: Reacting H2O Density Fraction Contour Plot - Phi = .1

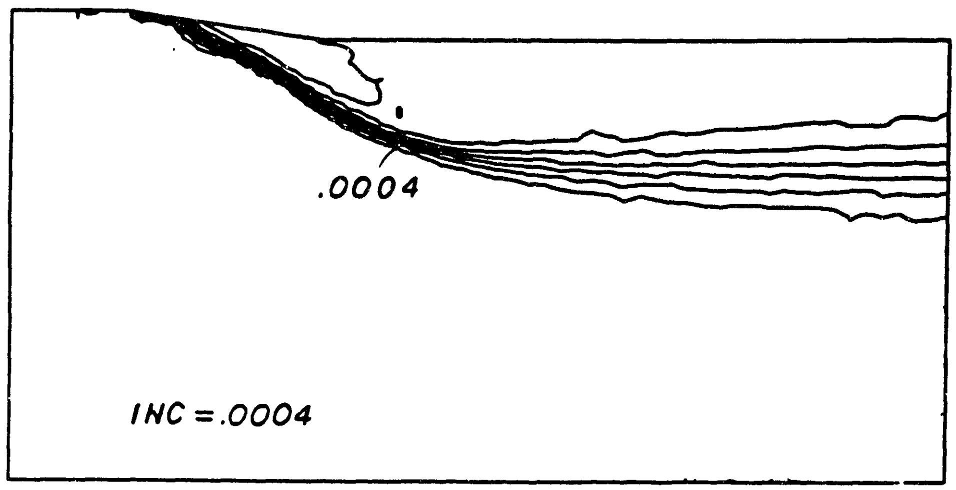


Figure 7-34: Reacting OH Density Fraction Contour Plot - Phi = .1

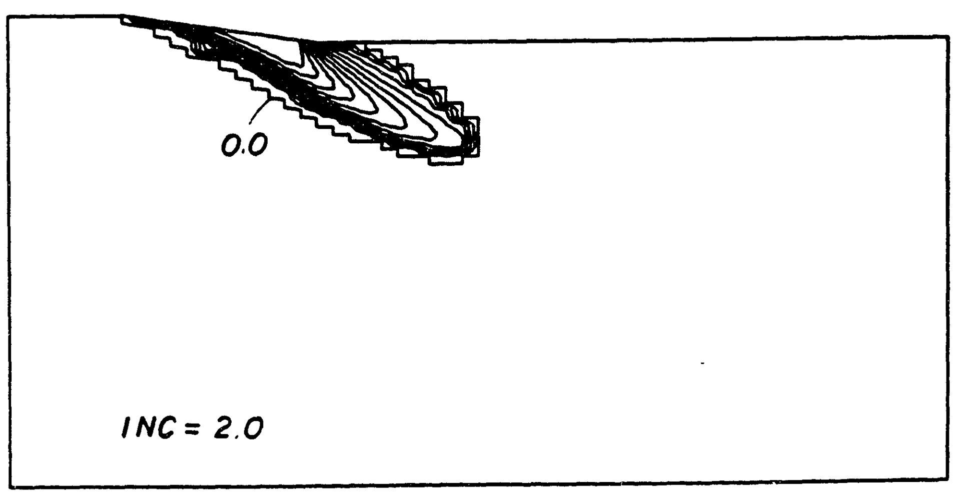


Figure 7-35: H_2O Rate Contour Plot - Phi = .1

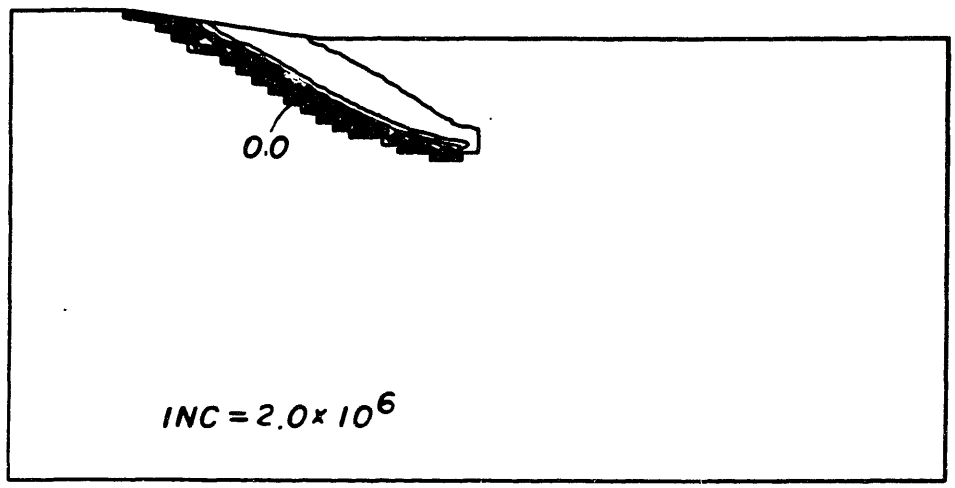


Figure 7-36: OH Rate Contour Plot - Phi = .1

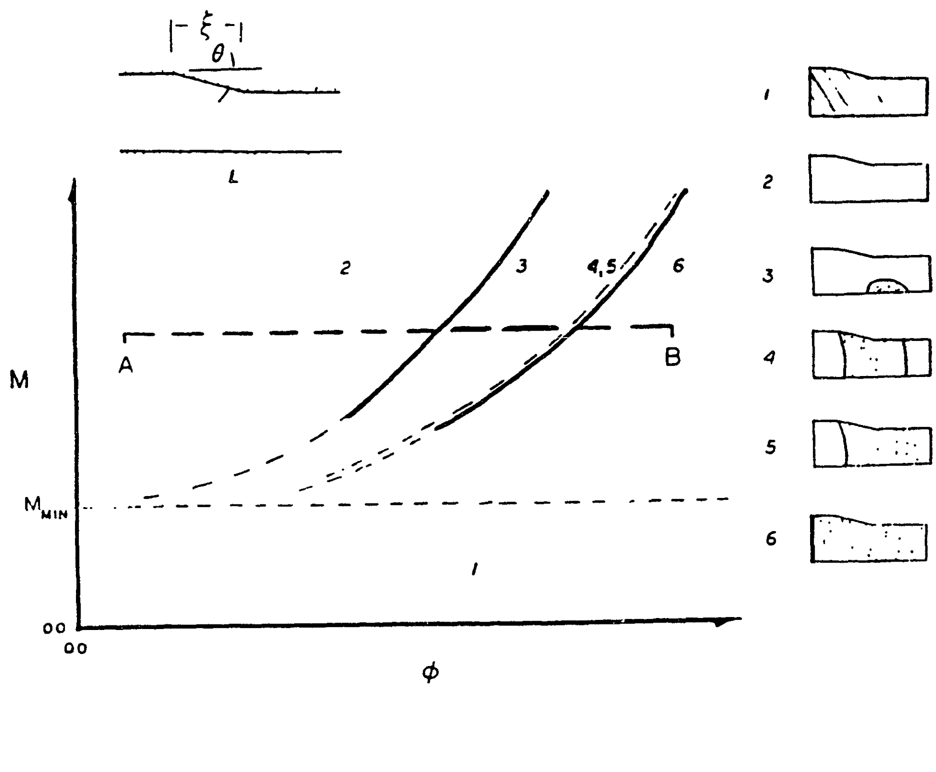


Figure 7-37: 2 - D Ramp Heat Release Map

Chapter 8

2-D VISCOUS FLOW WITH H_2-AIR CHEMISTRY

Flame holding occurs when a chemical reaction is initiated in a flow field and a stable flame propagates downstream from that point. In a viscous flow near a solid wall, flame holding can occur in two ways. The first way is to have a flow with a maximum temperature which is above the ignition temperature. In this case combustion will always occur. In the second situation the maximum temperature reached in the flow is everywhere below the fuel ignition temperature. For burning to occur an ignition source must be provided. However, only under certain situations can a stable flame holder be produced. The next section will illustrate some of these different cases.

One example of a potential supersonic flame holder where viscous effects are important is a rearward facing step. These devices operate by creating a hot spot behind the step from which heat and radicals can diffuse into the unburned mixture. If the fluid maximum temperature is above the fuel ignition temperature chemical reaction will always occur. If the mixture is premixed the step offers little benefit in this case since the boundary layer before and after the step will also be reacting. For this case a stable flame holder could be produced as long as the flow is not thermally choked. If the fluid maximum temperature is everwhere below the fuel ignition temperature, the step can provide a large volume of hot gas from which heat and radicals can diffuse out of to ignite the unburned gas mixture comming over the step. To get the flow burning in this case, an ignition source must be provided. For a stable flame holder to exist the heat produced in the recirculation zone must be equal to the heat which diffuses out of the recirculation zone. This implies that a fluid particle must remain in the recirculation zone long enough for chemical reaction to occur, i.e., $\tau_r > \tau_{H2O}, \tau_{OH}$. τ_r is the time a fluid particle remains in the recirculation zone and τ_{H2O} and τ_{OH} are chemical reaction time scales for the H_2-air chemistry model.

In the next section an attempt will be made to characterize the rearward facing step geometry. The full 2 - D laminar Navier-Stokes equations with H_2 air chemistry will be solved using the chemical time scaled preconditioned Jameson, Schmidt and

Turkel scheme with the constant CFL condition. For these calculations the flow over the step was assumed to be premixed. The computer code written for this problem is given in appendix 3.

8.1. Effect Of Artificial Viscosity/Numerical Smoothing

Most finite differences schemes require the addition of artificial viscosity to stabilize the scheme and remove unwanted wiggles in the solution near shocks. If we are only interested in solving the Euler equations then adding artificial viscosity tends to smear out shocks. The amount of smearing can be controlled by adjusting the artificial viscosity coefficients.

Solving the Navier-Stokes equations with added artificial viscosity can lead to errors if special care is not taken to ensure that the real viscous terms are not overwhelmed by the artificial viscosity terms. In particular, to get an accurate solution, the artificial viscous stress fluxes must be small in comparison to the sum of the real viscous stresses plus the convective fluxes. A typical artificial viscosity term can be written as,

$$(D_{xx}\phi)_{atr} \simeq \sigma_v \frac{\Delta x^2}{\Delta t} \frac{\partial^2 \phi}{\partial x^2} \tag{8.1}$$

while a typical real viscous stress term is given by,

$$(D_{xx}\phi)_{real} \simeq \frac{1}{Re} \frac{\partial}{\partial x} \mu \frac{\partial \phi}{\partial x} \tag{8.2}$$

which can be approximated as,

$$\simeq \frac{\mu}{Re} \frac{\partial^2 \phi}{\partial x^2} \quad . \tag{8.3}$$

ϕ could be a velocity component, the temperature or a species density fraction. For regions of the flow field where the real viscous stress terms dominate over the convective terms a relation can be derivied from equations (8.1) and (8.2) indicating how small the artificial terms must be in order that they do not contaminate the solution. Specifically,

$$\sigma_v \frac{\Delta x^2}{\Delta t} \ll \frac{\mu}{Re} \tag{8.4}$$

which implies that σ_v and Δx must be choosen carefully to get accurate solutions. In fact it was found that the artificial viscosity fluxes had to be at least an order of magnitude less than the real fluxes to prevent them from contaminating the solution. It was found best to set σ_v=.05 and vary Δx so that equation (8.4) is satisfied. The choice of σ_v was determined from experience gained while computing laminar flat boundary layers.

8.2. Viscous Flow Validation

Two test cases will be used to validate the method for the Navier-Stokes equations. The first case consists of supersonic flow over a flat plate. The second case is for supersonic flow over a rearward facing step. The flat plate results are compared with another set of computations while the rearward facing results are compared with experiment.

The simplest way to test a solution method for the Navier-Stokes equations is to solve for the flow over a flat plate. The computations were started impulsively and time marched to convergence. A temperature equal to the stagnation temperature was specified at the wall. For the fluid properties given in table 8-1, and the mesh distribution shown in figure 8-1, the computed profiles of u, T and v are given by figures 8-2, 8-3 and 8-4. The profiles were measured at a distance of x=L, where x is the distance measured from the plate leading edge and L equals two thirds of the length of the plate. The Reynolds number based on the total plate length, 3/2L, is 1500. The profiles are compared with a calculation performed by Carter [8]. The comparison shows excellent agreement. Note the kink in the profile is due the bow shock emanating from the leading edge of the plate. It was found best to run the calculation with an artificial viscosity coefficient of .05 . In some preliminary investigations it was observed that if a coefficient larger than .05 were used, say .1, we would begin to see some contamination of the flow field due to artificial viscosity.

These values correspond to artificial diffusion terms which are at least an order of magnitude smaller than the sum of the real convection and diffusion terms. The solution converged in approximately 2000 iterations, figure 8-5. Convergence is based on a four order reduction in the continuity equation residual.

The second validation test case involved comparing the computed wall pressure behind a rearward facing step with experimental data. The data was taken from an experiment performed by Jakubowski and Lewis [21]. They studied supersonic flow over rearward facing steps for various step heights. The test data used in this comparison is given in table 8-2. The solid walls were assumed to be adiabatic and the upper wall is assumed to be a symmetry plane. The inflow boundary conditions are set equal to the free stream conditions. The inflow temperature and velocity profiles are assumed to be constant across the inflow boundary. The computational grid is shown in figure 8-6. In constructing a grid care must be taken to ensure that the errors associated with the artificial viscosity terms do not contaminate the computed solutions. The computed pressure, temperature and Mach contours are given by figures 8-7, and 8-8. The velocity vector plot, 8-9, shows clearly the viscous layers. Figure 8-10 is an enlargement of the recirculation zone and shows that the flow is in rotation. The computations give the reattachment point as being one step height downstream of the step, in agreement with Jakubowski and Lewis's experimental observations. Note the boundary layer coming over the step is of the order of one step height, again in agreement with Jakubowski and Lewis's experimental observations. Finally the computed and experimental wall pressures behind the step are given in figure 8-11. The figure shows that reasonable agreement was obtained between the computation and the experiment. The differences between the experiment and the computation could be partly due to the fact that the gas mixture in the experiment was not standard air but a dissociated mixture which could not be adequately accounted for in the present computer code. In addition the code did not take into account the variation of c_p and c_v with temperature. The code considers the various thermodynamic properties to be dependent only the species density fractions. Convergence for this example was obtained in approximately 2000 iterations (figure 8-12).

Given these two validation cases a high degree of confidence in the solution method and code ability to solve the Navier-Stokes equations has been established.

8.3. Heat Release

The computational mesh used for all of the remaining rearward facing step calculations is shown in figure 8-13, while the fluid and property data are given in table 8-3. Note the lower boundary is modeled as a solid wall while the upper boundary is modeled as a symmetry line. In order to reduce the computational domain size an incompressible Blasius flat plate boundary layer profile was specified at the inlet. For this example the solid walls were assumed to be adabatic and non-catalytic.

The flow over a rearward facing step is strongly dependent on the ratio of the boundary layer thickness to the step height, Ω. Three situations were investigated to assess the effect of varying Ω. The three situations include $\Omega >> 1$, $\Omega \simeq 1$ and $\Omega << 1$. In the first case, $\Omega >> 1$, the amount of flame holding provided by the step is small compared to the flame holding provided by a simple flat plate boundary layer. A flat plate can hold a flame if the maximum temperature reached in the boundary layer is greater than the fuel ignition temperature. In this case the step has little or no effect on flame holding and will not be considered further here. For $\Omega \simeq 1$, the boundary layer thickness was chosen to be equal to the height of the step, which is typical for scramjet applications. Finally, the boundary layer thickness was set to zero for the case when $\Omega << 1$.

The flow is also strongly affected by the ratio of the fluid maximum temperature to the fuel ignition temperature, Σ. If Σ is greater than one the fuel and air autoignite wherever the static temperature is above the fuel ignition temperature. A Σ less than one corresponds to the situation where the fluid maximum temperature is everywhere below the fuel ignition temperature. In this case, for chemical reaction to occur, an ignition source must be provided. Numerically this implies forcing the temperature behind the step to be above the ignition temperature for a period of time and then removing this constraint once the fuel has started burning.

Examples of a non-reacting temperature, Mach number and velocity field, with $\Omega \simeq 1$, $\Sigma < 1$ and $\phi = .5$, are shown in figure 8-14, 8-15 and 8-16. A blow up of the region behind the step shows clearly the recirculation region , figure 8-17. Figures 8-18 and 8-19 show that, with combustion, chemical reaction occurs in the boundary layers and behind the step. The amount of chemical reaction produced by the step is comparable to that produced by the boundary layer coming over the step. The maximum temperature reached behind the step is approximately 1850K. If ϕ is increased to 1 the flow thermally chokes. Due to resource limations the residuals were only reduced by 2.5 orders of magnitude for all of the reacting step calculations.

Using the same flow conditions just discussed but with a step height equal to .15 m (Re, Pr, γ and M free stream same as above), Ω=0 and ϕ=.5, the following is generated. First the non-reacting temperature, Mach number and velocity field distribution is given in figure 8-20, 8-21, 8-22 and 8-23. With combustion, figures 8-24 and 8-25 , chemical reaction occurs only behind the step. The maximum temperature reached was approximately 2350K. In this case chemical reaction is limited to the region behind the step. Thus if $\Sigma>1$ the step serves only to increase the size of the burning region compared to a a simple boundary layer. The increased reaction zone size could increase the rate at which heat and radicals are transferred to the unburned fluid and thus increase the rate of flame spreading away from the flame holding region.

So far all of the cases considered have assumed that $\Sigma>1$. If $\Sigma<1$ the steady state solution will depend on the initial conditions. To compute the stable reacting solution, if one exists, requires that an ignition source be provided. To ignite the fuel, a hot spot is created behind the step. The assumption here is that the release of heat will raise the local static temperature above the fuel ignition temperature and sustain a stable burning process. This means that a stable flame can only be produced if the fuel concentration is high enough to insure a sufficient level of heat release. For the problem under consideration the non-reacting maximum temperature is approximately 1350K. As a demonstration that multiple steady states are possible consider increasing the fuel ignition temperature to 1450K, Ω<<1. Note the fuel ignition temperature is usually a constant but is varied here to illustrate a point. With a ϕ=1 and the chemistry model turned on, two different steady states can be arrived at by varying the initial conditions. The two steady states include a non-reacting and a reacting solution. If the starting temperature was chosen to be below T_{ig} than the non-reacting solution was obtained. If a hot spot was set up behind the step, than the reacting steady state solution was obtained. With the step height used in the first reacting example (h=.025m) only the non-reacting steady state is obtained. However, if the step height is increased by a factor of 6 (Re, Pr, γ and M free stream kept constant), two solutions can be obtained depending upon the initial conditions. Figures 8-26, 8-27, 8-28 and 8-29 show the temperature , the velocity field, the H_2O reaction rate source term and the H_2O density fraction contours for the reacting situation. Note the reattachment point has moved to approximately 1.5 step heights downstream of the step compared to 1 for the non-reacting case, figure 8-22. The maximum temperature reacted in the flow field was 2860K.

8.4. Rearward Facing Step Characterization Map

It is suggested that some of the reacting rearward facing step flows can be characterized on a ϕ vs Tmax/Tig map, figure 8-30. The map breaks up the possible flow fields into three regions, reacting (1), non-reacting/reacting (2) and non-reacting (3). The figures indicate that for a Σ less than 1, there is a region on the map where the steady state is dependent upon the initial conditions chosen. In this region at least two different steady state solutions can be generated, reacting and non-reacting. The non-reacting steady state is arrived at by choosing an initial temperature distribution which is everywhere below Tig. The reacting steady state, if one exists, is found by starting the calculation with a hot spot (T>Tig) behind the step. For the other two regions only one steady state exists.

Properties	Values	Dimensions
P_∞	7.0	N/m^2
T_∞	216.	°K
u Velocity	882	m/s
v Velocity	0	m/s
M_∞	3.	
Re_L	1000.	
Pr	.72	
c_p	1000.	J/kg·°K
c_v	714.	J/kg·°K
L	.15	m
Grid	60 × 60	
CFL	1.	
σ_v	.05	

Table 8-1: Flat Plate Test Data - Validation

Properties	Values	Dimensions
P_∞	744.	N/m^2
T_∞	1046.	°K
u Velocity	2491.	m/s
v Velocity	0.	m/s
M_∞	4.	
Re_H	29200.	
Pr	.72	
c_p	1149.	J/kg·°K
c_p	877.	J/kg·°K
L	.175	m
h	.0051	m
Grid	51 × 31	
CFL	1.	
σ_v	.05	

Table 8-2: Rearward Facing Step Test Data - Validation

Properties	Values	Dimensions
P_∞	100000.	N/m^2
T_∞	900.	°K
$T_{Ignition}$	1300.	°K
ϕ	.1	
u Velocity	1200.	m/s
v Velocity	0.	m/s
M_∞	2.	
Re_H	230000.	
Pr	1.	
c_{pH2O}	17160.	J/kg·°K
c_{pOH}	1181.	J/kg·°K
c_{pH2}	2854.	J/kg·°K
c_{pO2}	2041.	J/kg·°K
c_{pN2}	1285.	J/kg·°K
c_{vH2O}	17160.	J/kg·°K
c_{vOH}	1181.	J/kg·°K
c_{vH2}	2854.	J/kg·°K
c_{vO2}	2041.	J/kg·°K
c_{vN2}	1285.	J/kg·°K
Hf_{H2O}	-1.44×10^7	J/kg
Hf_{OH}	2.3×10^6	J/kg
Hf_{H2}	0.0	J/kg
Hf_{O2}	0.0	J/kg
Hf_{N2}	0.0	J/kg
L	.025	m
h	.005	m
Grid	51 × 31	
CFL	1.	
σ_v	.05	

Table 8-3: Rearward Facing Step Test Data

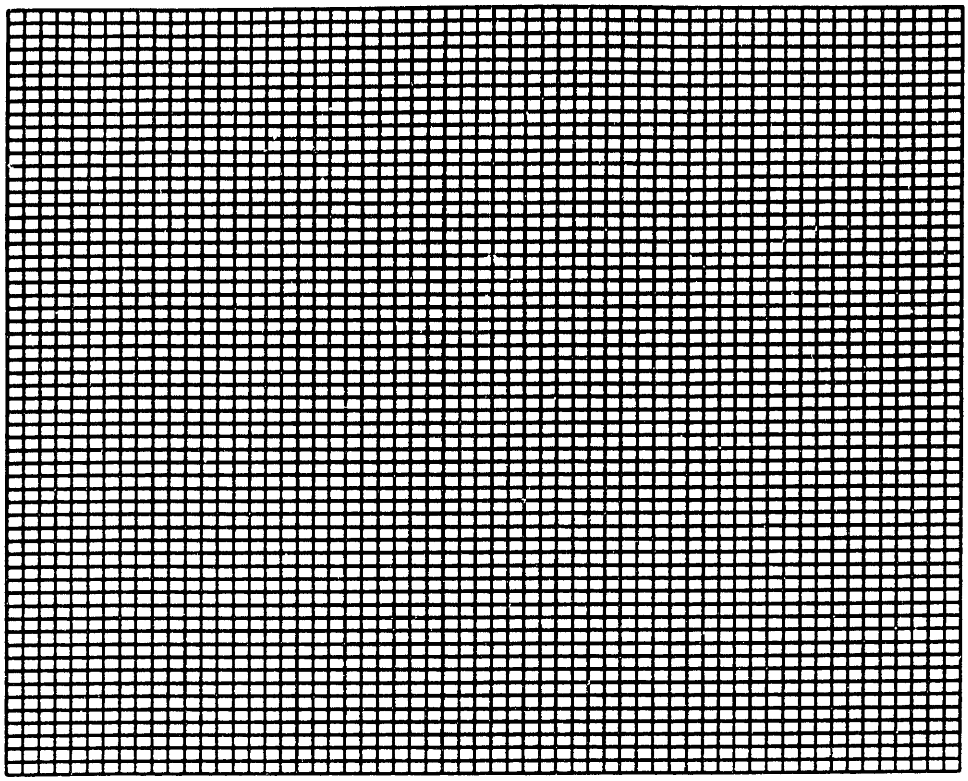


Figure 8-1: Flat Plate Mesh

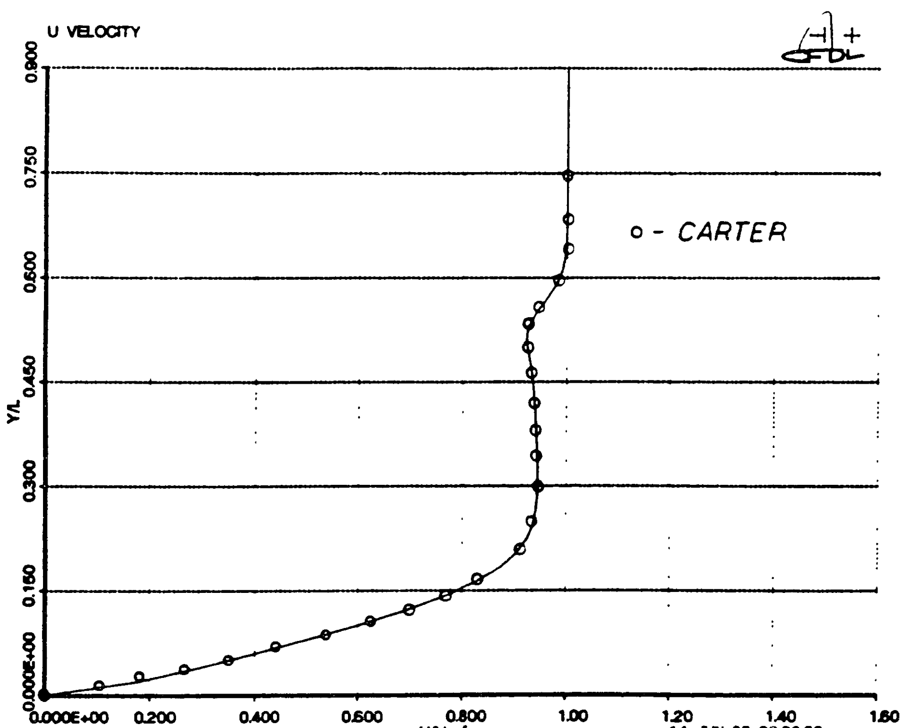


Figure 8-2: u Velocity Component Profile

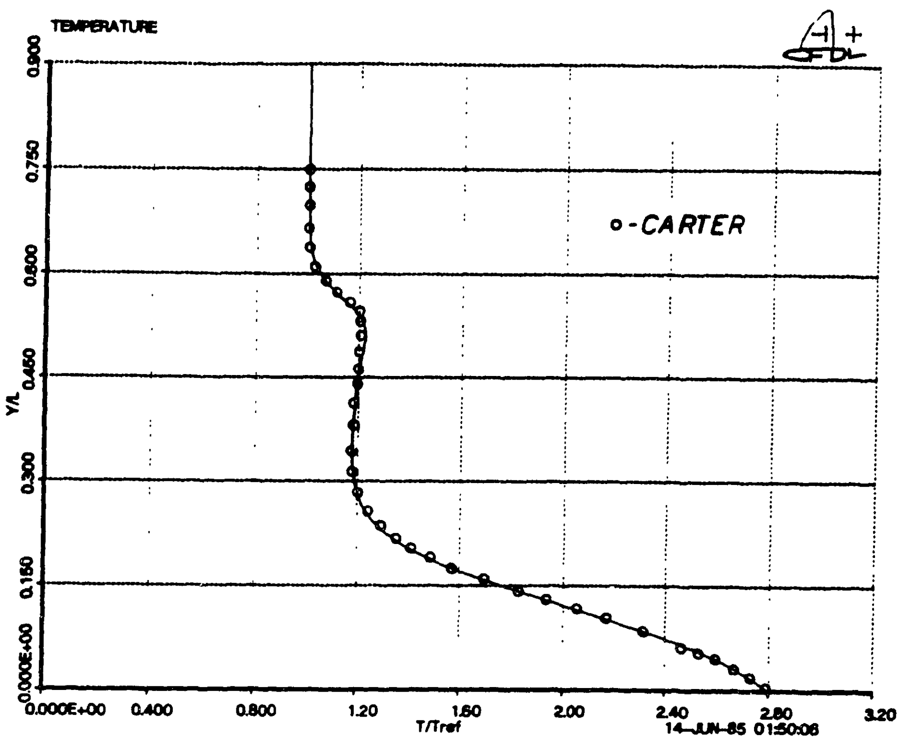


Figure 8-3: Temperature Profile

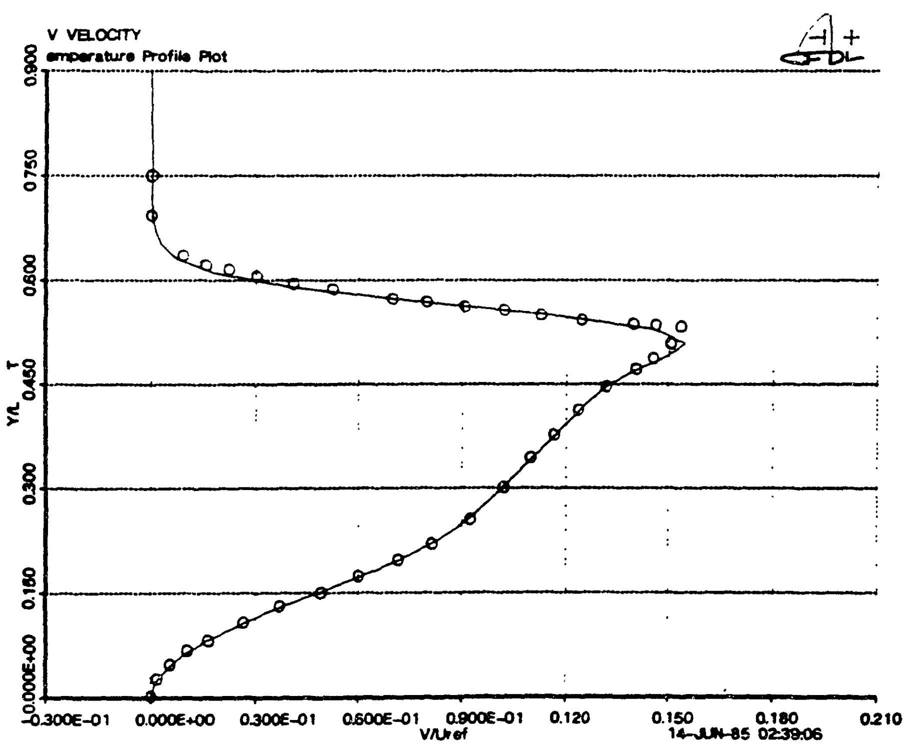


Figure 8-4: v Velocity Component Profile

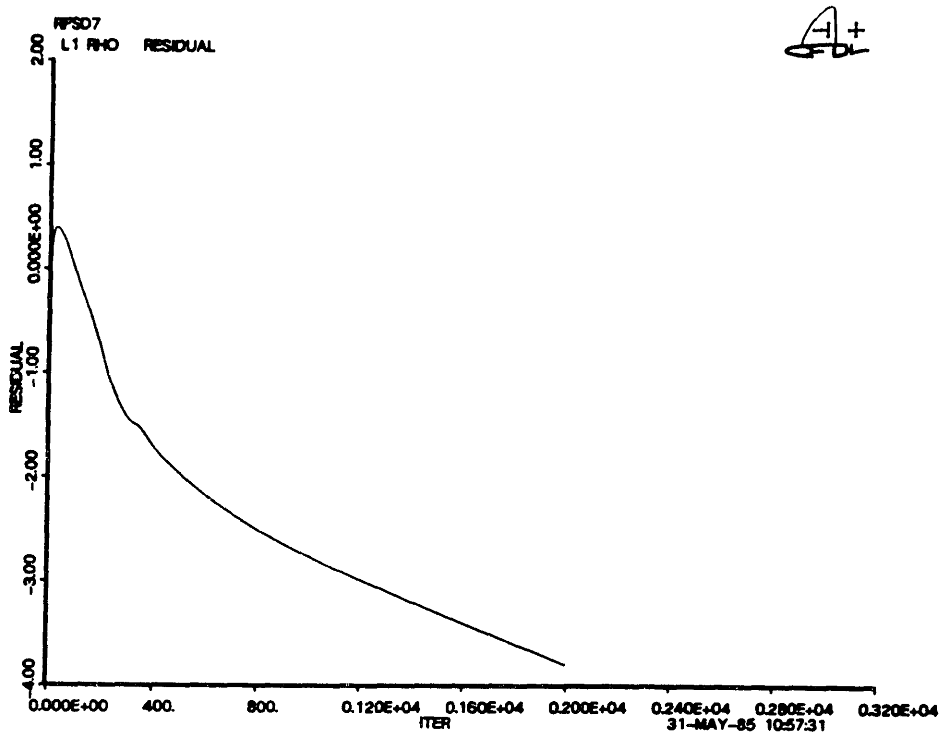


Figure 8-5: Flat Plate Convergence History

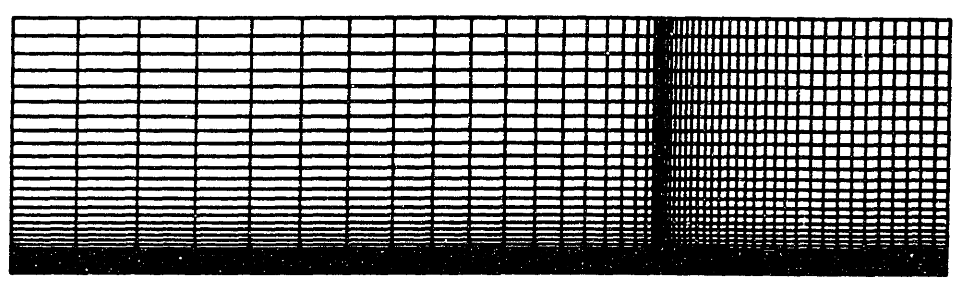


Figure 8-6: Rearward Facing Step Mesh - Validation

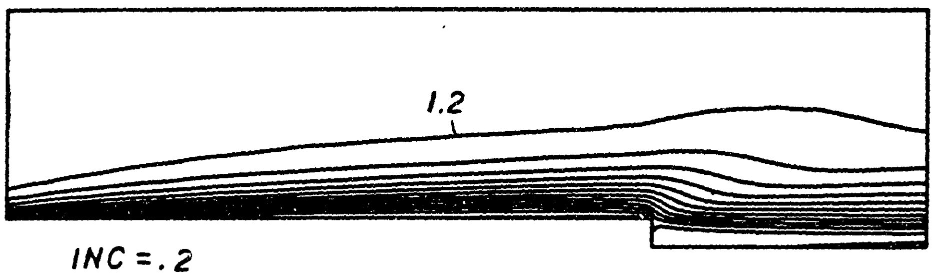


Figure 8-7: Temperature Contours - Validation

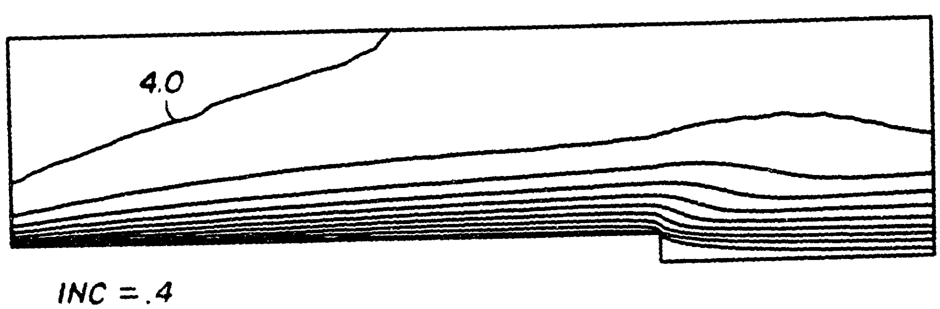


Figure 8-8: Mach Number Contours - Validation

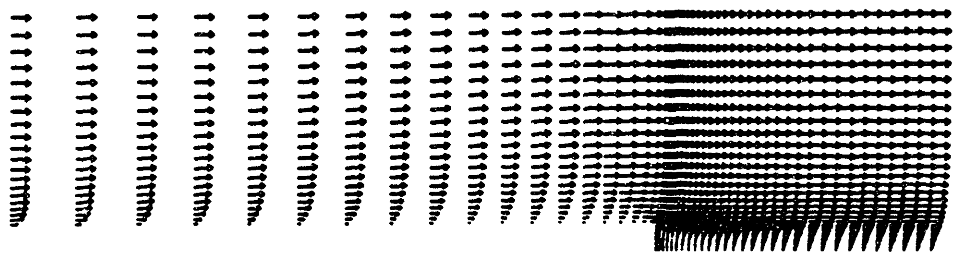


Figure 8-9: Velocity Vector Plot - Validation

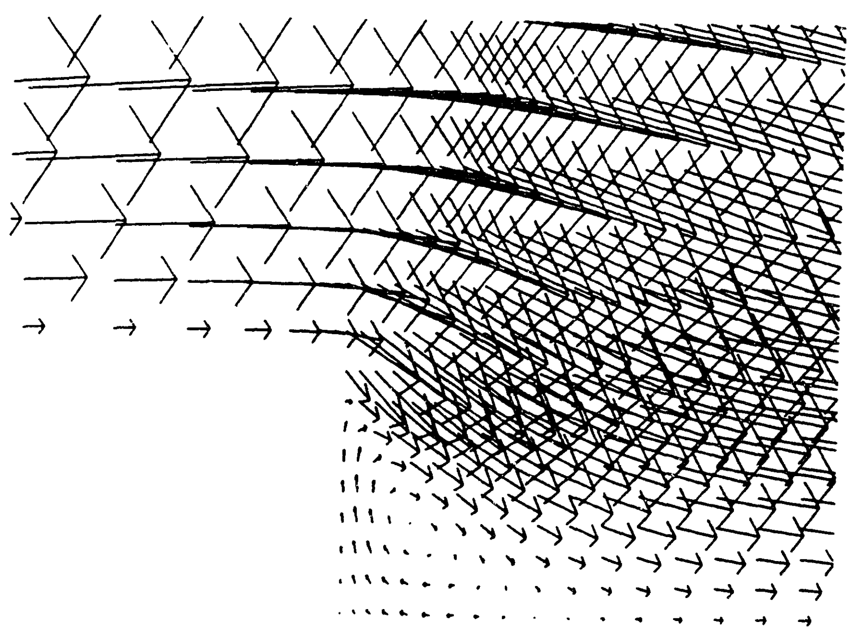


Figure 8-10: Recirculation Zone Velocity Vector Plot - Validation

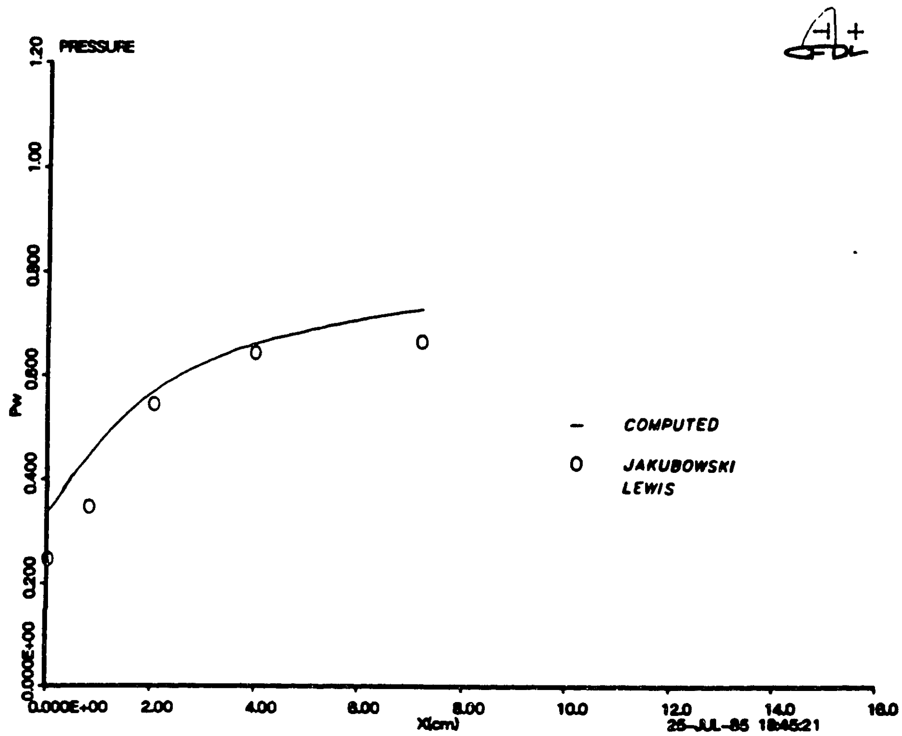


Figure 8-11: Comparison Between Computed and Experimental Upper Wall Pressure

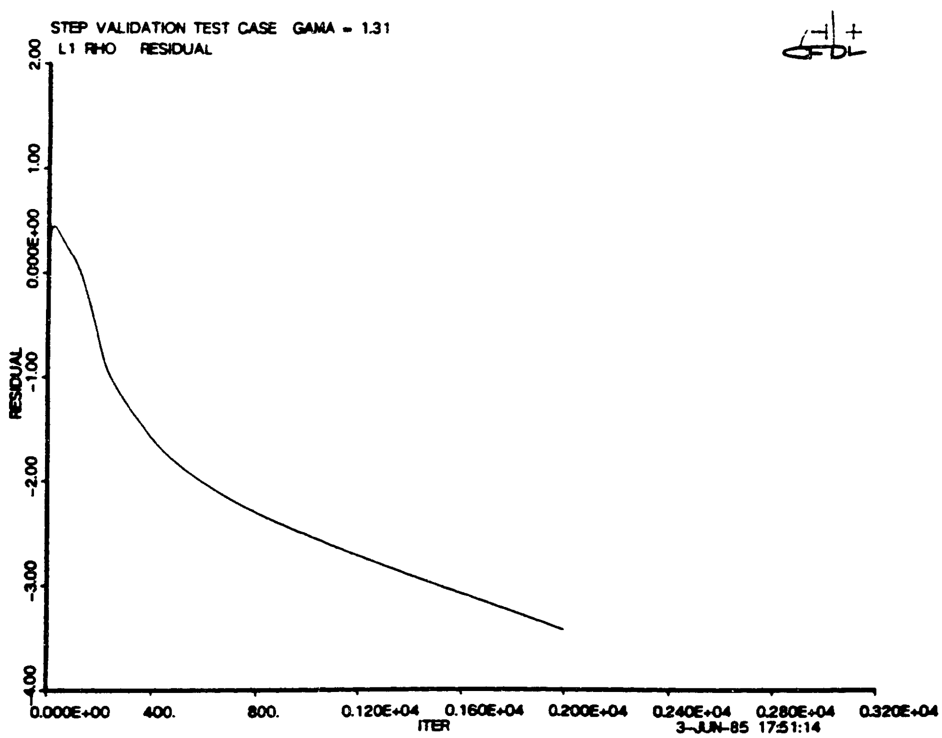


Figure 8-12: Rearward Facing Step Convergence History

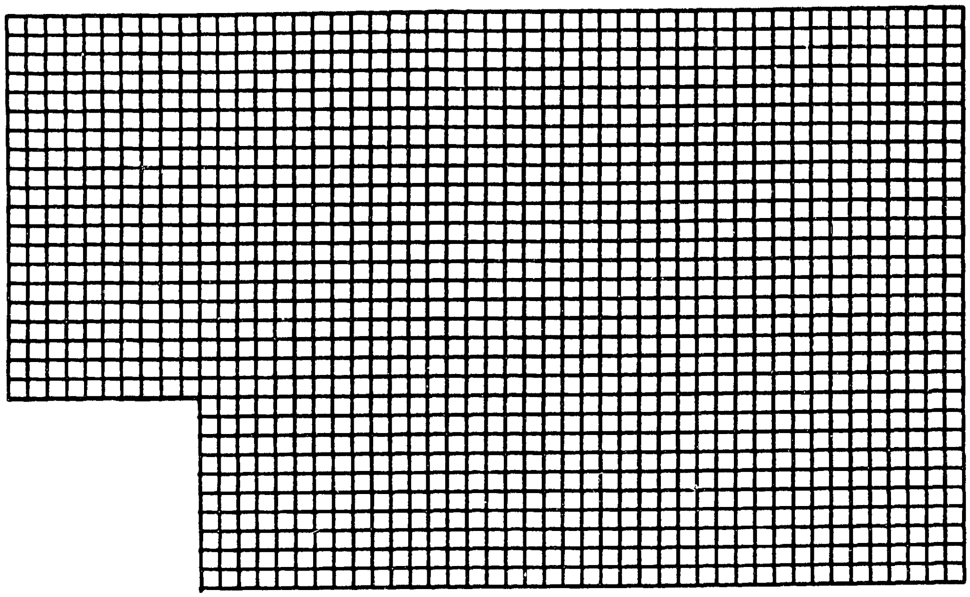


Figure 8-13: Computational Grid

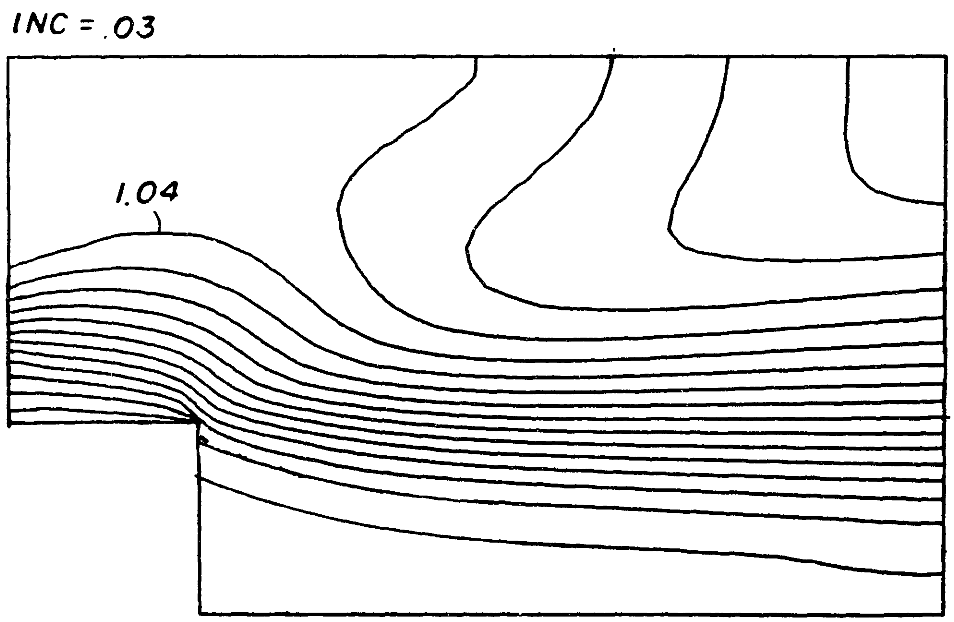


Figure 8-14: Non-Reacting Temperature Contours

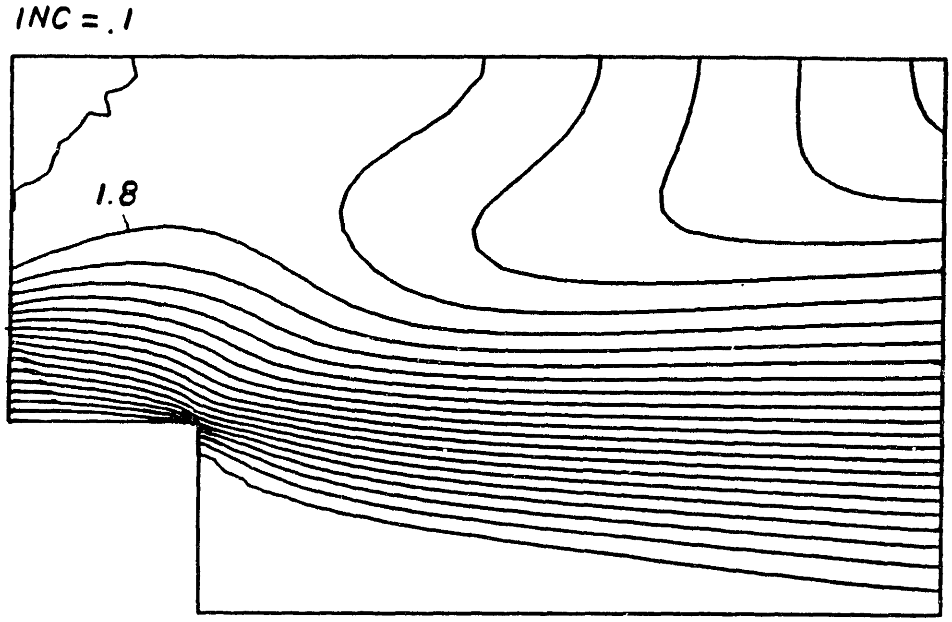


Figure 8-15: Non-Reacting Mach Number Contours

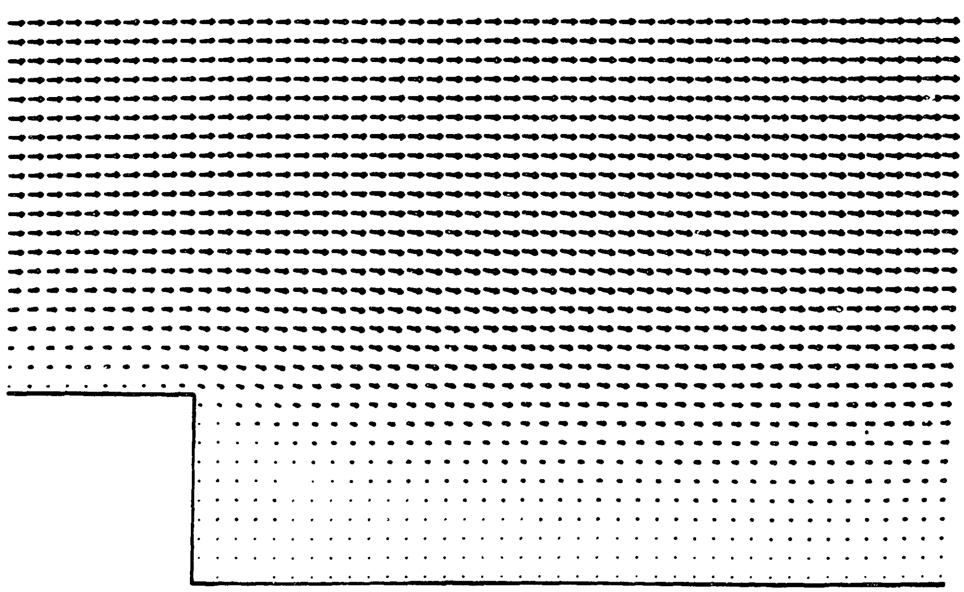


Figure 8-16: Non-Reacting Velocity Vector Distribution

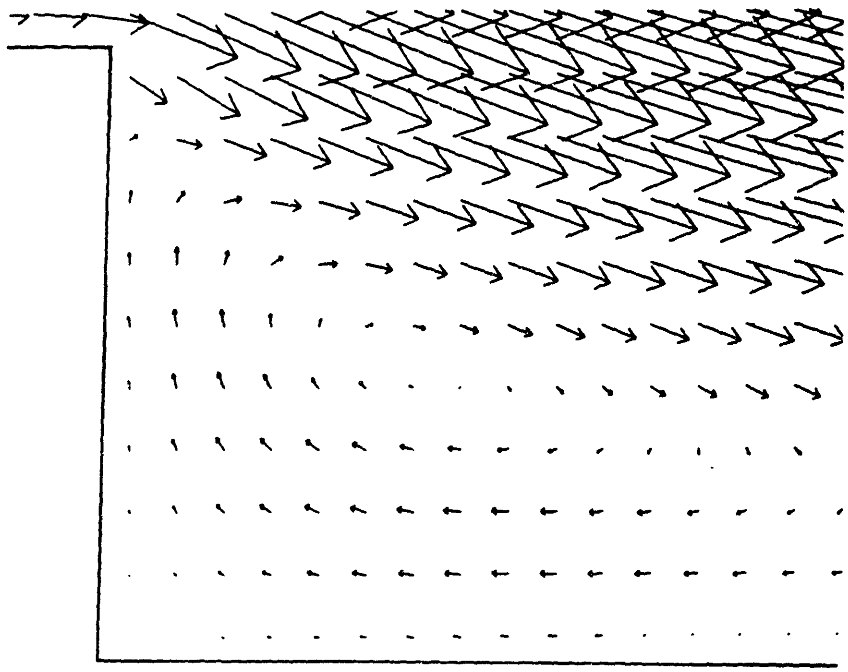


Figure 8-17: Blowup Non-Reacting Recirculation Zone

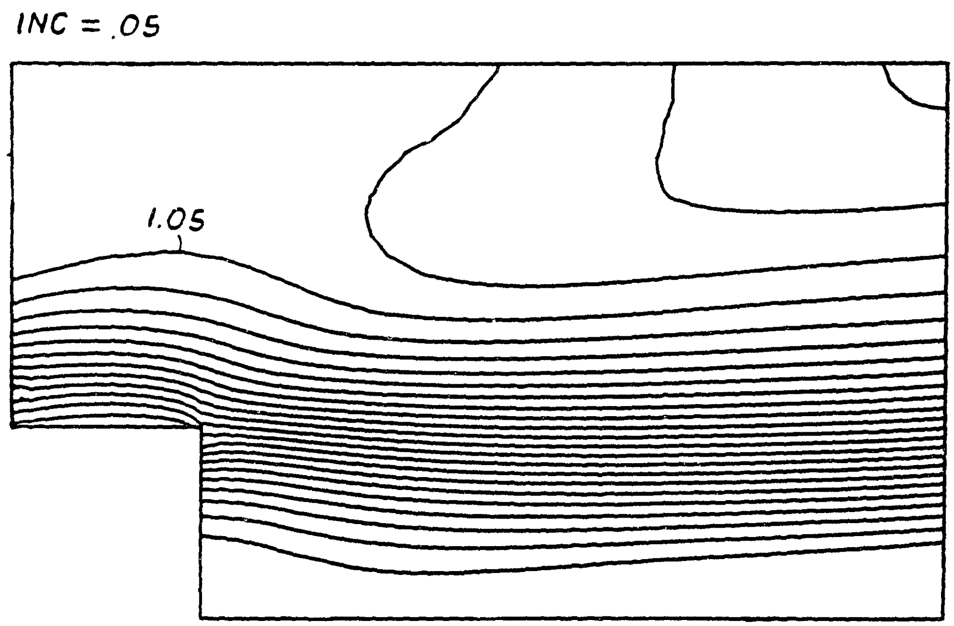


Figure 8-18: Reacting Temperature Contours

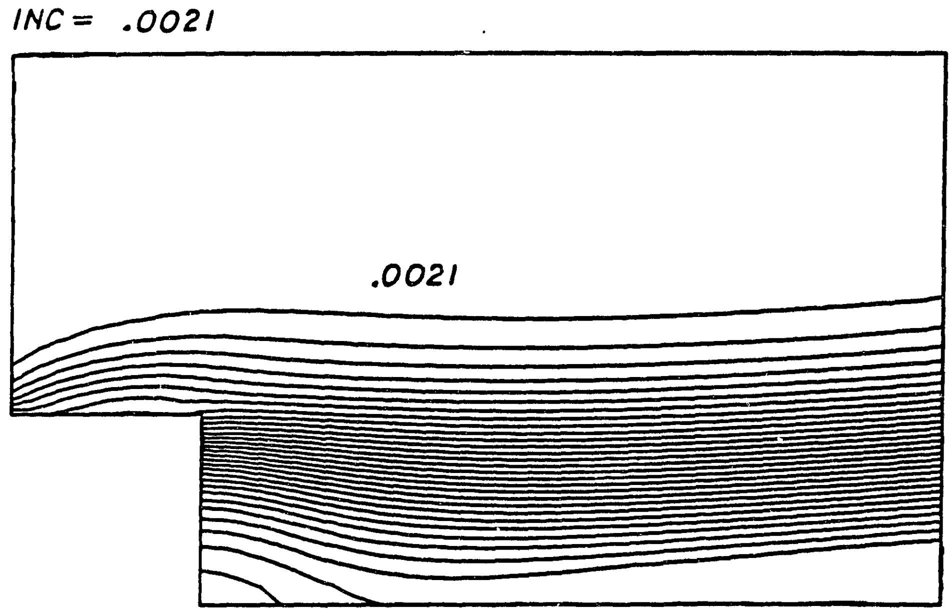


Figure 8-19: Reacting Y_{H_2O} Contours

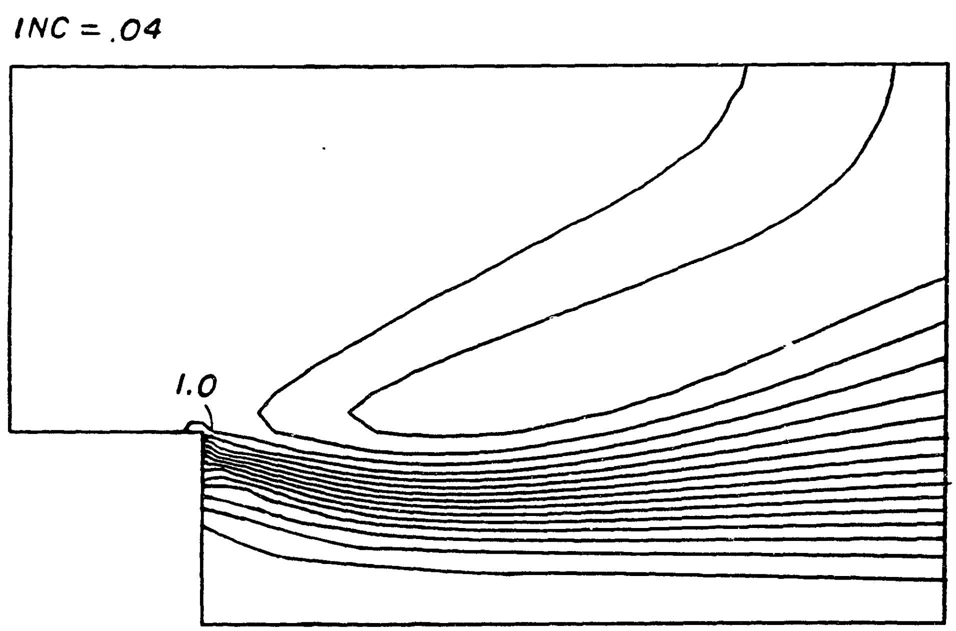


Figure 8-20: Non-Reacting Temperature Contours

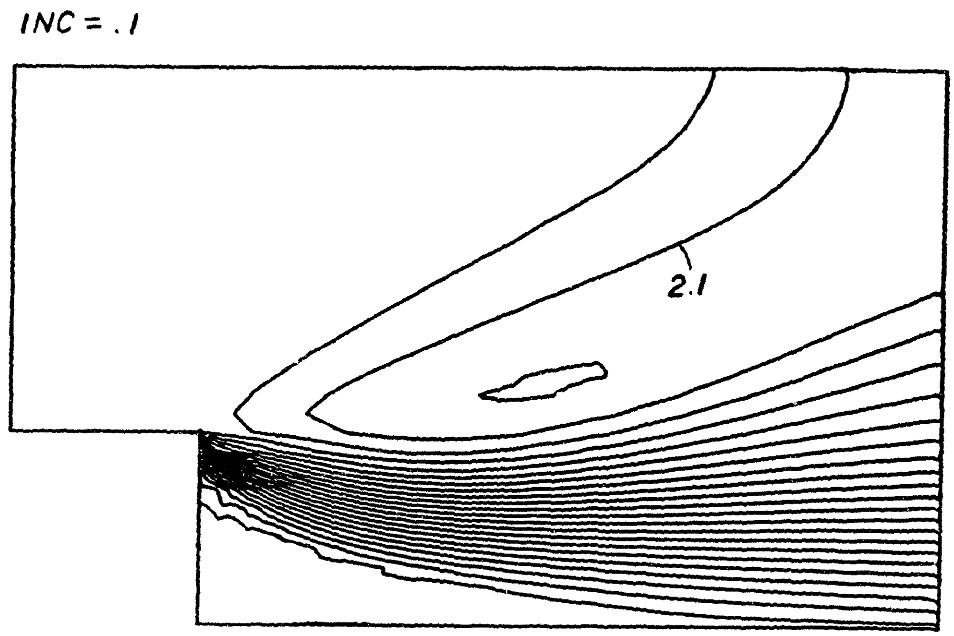


Figure 8-21: Non-Reacting Mach Number Contours

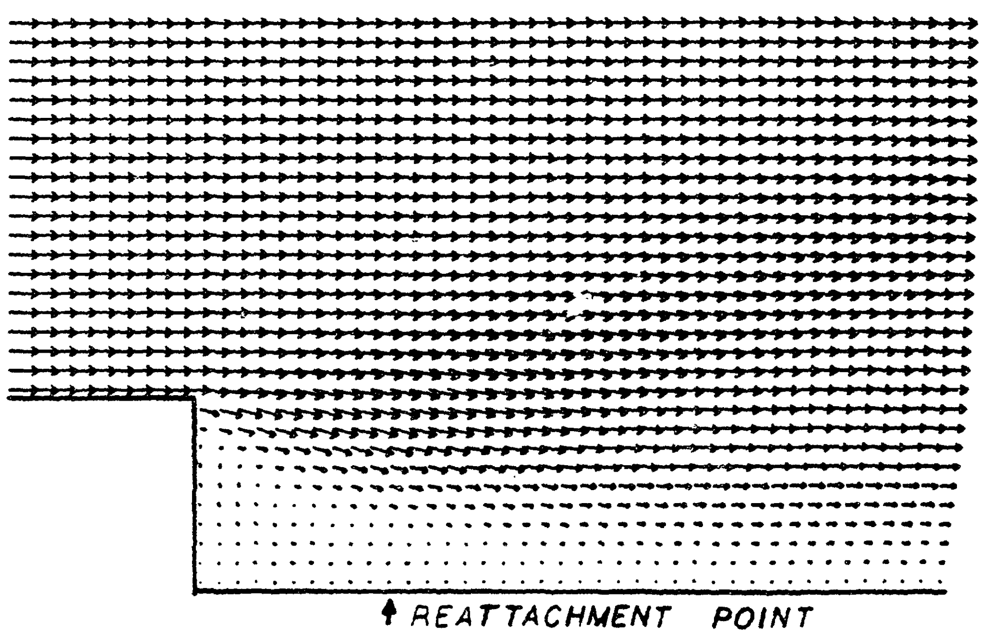


Figure 8-22: Non-Reacting Velocity Vector Distribution

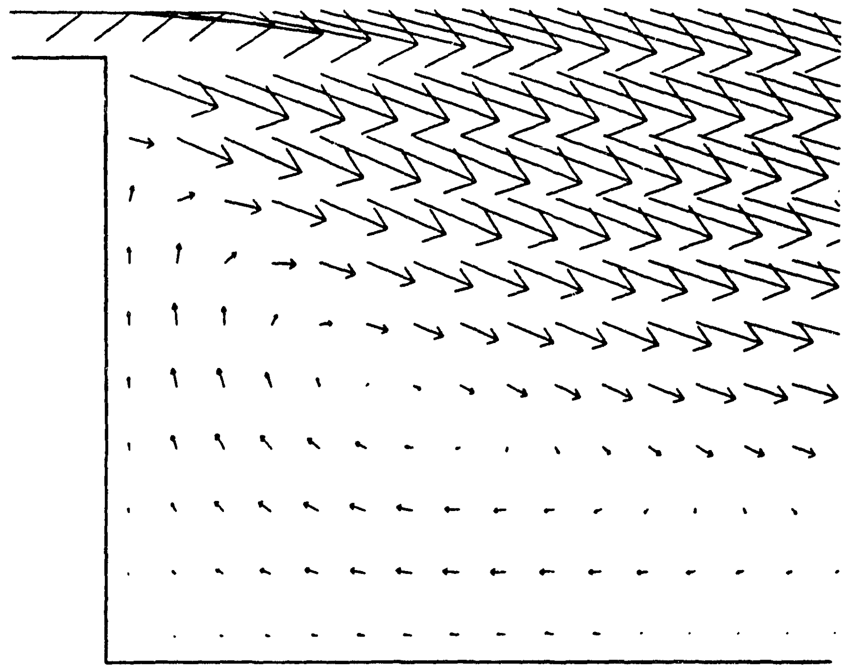


Figure 8-23: Blowup Non-Reacting Recirculation Zone

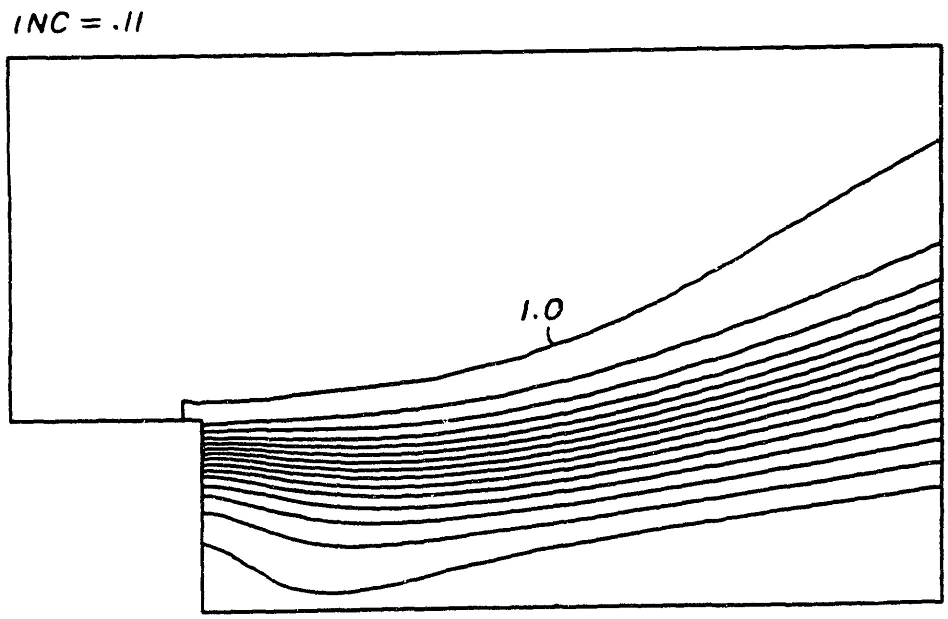


Figure 8-24: Reacting Temperature Contours

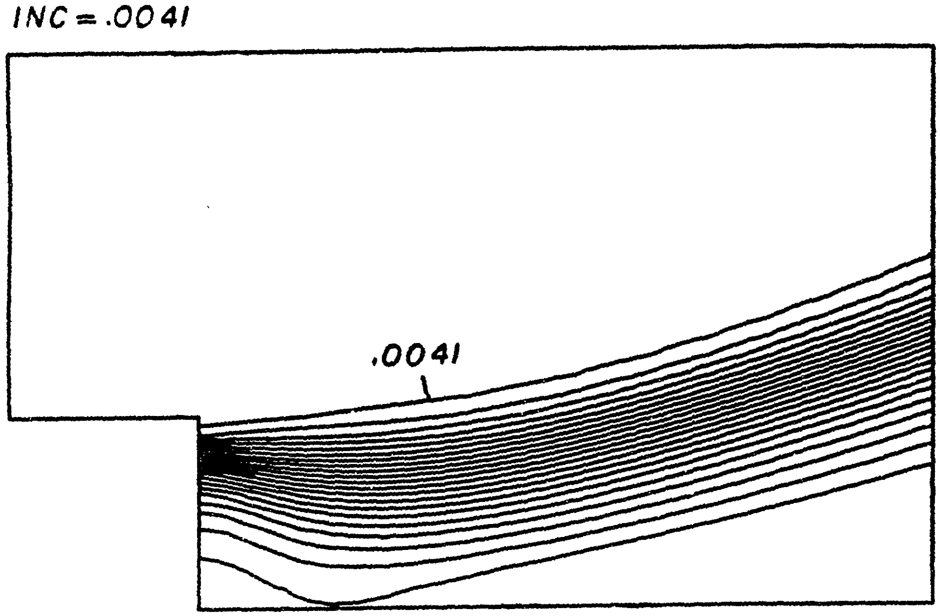


Figure 8-25: Reacting Y_{H_2O} Contours

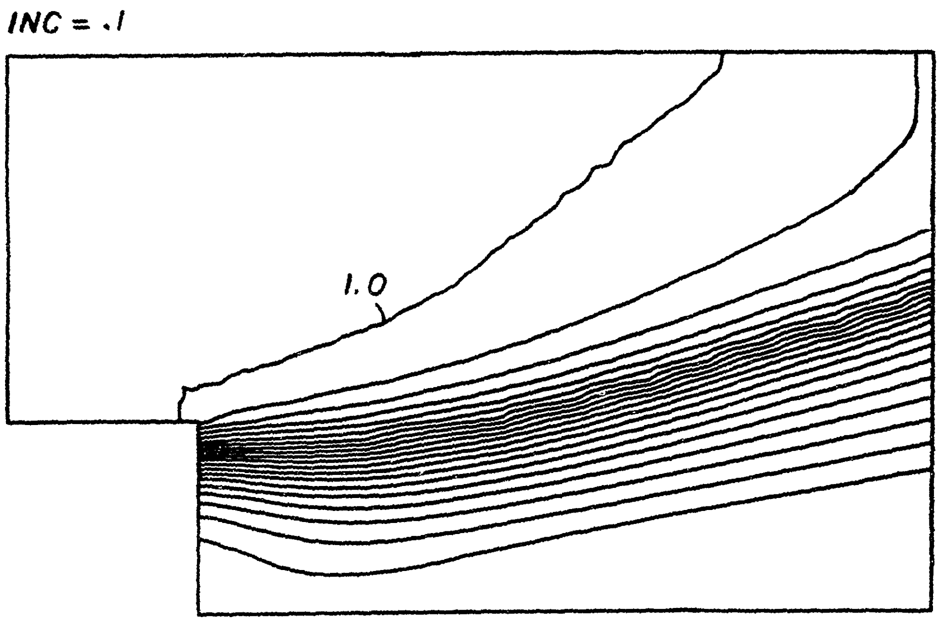


Figure 8-26: Reacting Temperature Contours

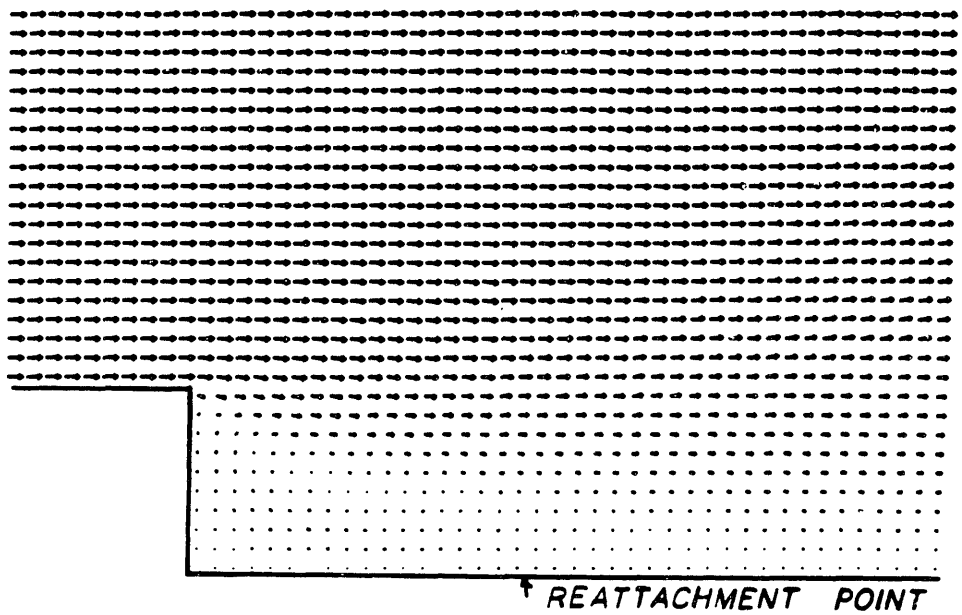


Figure 8-27: Reacting Velocity Vector Distribution

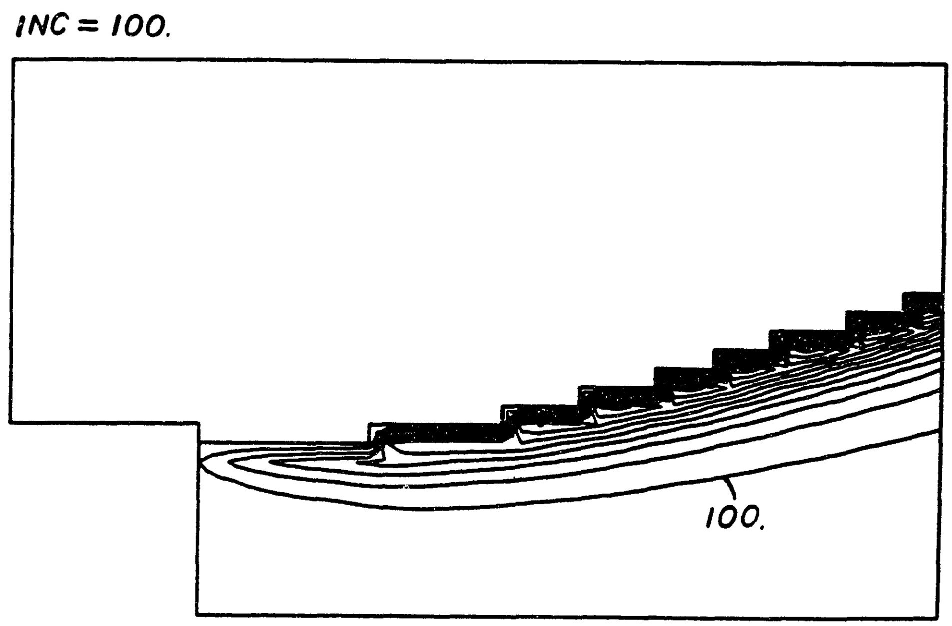


Figure 8-28: H_2O Reaction Rate Source Term

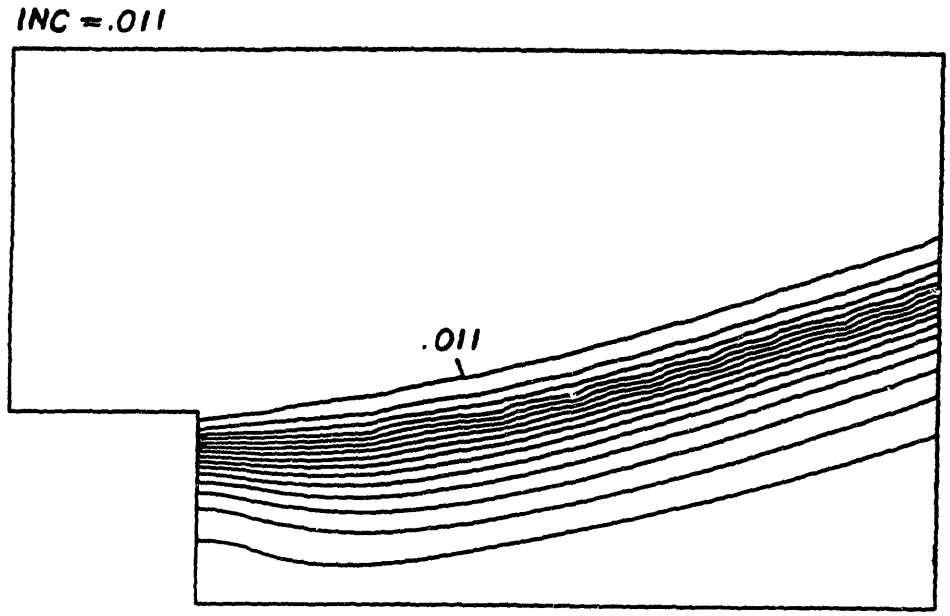


Figure 8-29: Reacting Y_{H_2O} Contours

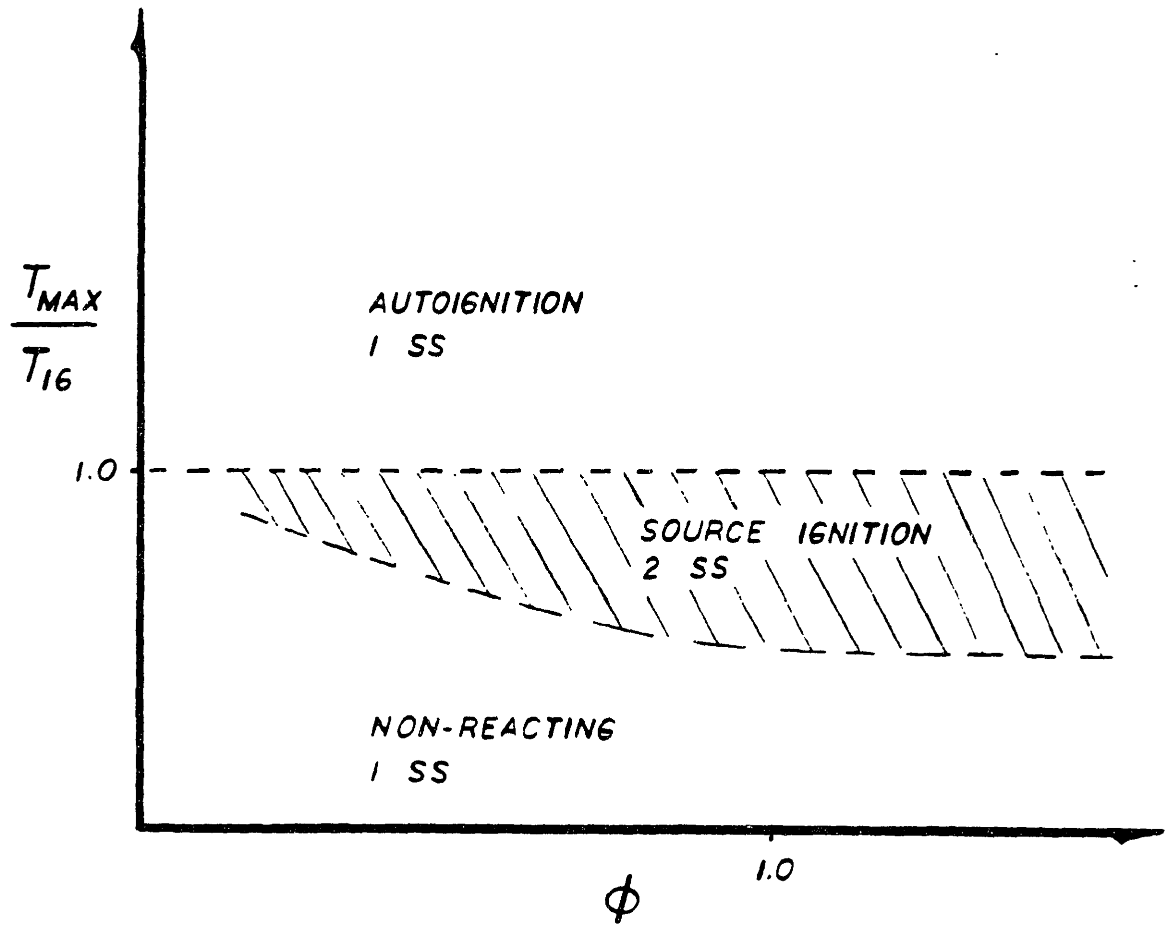


Figure 8-30: Rearward Facing Step Characterization Map

Chapter 9
CONCLUSION

Several important contributions to the field of computational fluid dynamics and chemically reacting flows associated with scramjet flame holders can be drawn from this study and summarized as follows:

Computational Fluid Dynamics

1. The time stiffness associated with stiff finite rate chemistry can be removed from the Navier-Stokes equations by preconditioning the unsteady equations.

2. If only the steady state is desired, the amount of computational work was found to be independent of the level of chemical stiffness.

3. The preconditioner chosen can be shown to be equivalent to advancing each state quantity at its own characteristic rate.

4. The point implicit method can be shown to be a special case of a more general preconditioning method.

5. The method can be made time accurate by choosing numerical time steps which are less than the time scales of interest.

6. The preconditioning technique is shown to be extendable to the Ni multiple grid method.

Scramjet Flame Holders

1. Two candidate scramjet flame holders were studied using the 2-D Euler and Navier-Stokes equations with an H_2-air chemistry model.

2. With the ramp flame holder geometry at least three different flow fields could be produced. These included fully supersonic flow, supersonic flow with an embedded subsonic zone and a thermally choked flow.

3. It was suggested that the behavior of 2-D compression ramps could be characterized and the results plotted on an inlet ϕ vs inlet Mach number map. The map serves to identify what type of flow will be produced for a given combination of ϕ and inlet Mach number.

4. The onset of thermal choking could be delayed by shortening the ramp wedge or by increasing the ratio of the flame thickness to the ramp length.

5. Reacting flows over rearward facing steps were found to be strongly dependent on the ratio of the boundary layer thickness coming over the step to the step height and to the ratio of the fluid maximum temperature to the fue ignition temperature.

6. It was suggested that some of the possible rearward facing step flow fields could be plotted on a ϕ vs Tmax/Tig map.

Each of these points will be discussed in detail in the following sections. In addition several recommendations will be made about possible future work. The results of this thesis have been published in two recent AIAA papers [4, 6].

9.1. Numerical Methods

The numerical integration of the equations governing chemical reacting flows can be very expensive unless special care is taken to remove the time stiffness limitations characteristic of these problems. A variety of numerical methods have been developed to solve these equations including both explicit and implicit techniques. Purely explicit methods are limited and become very inefficient when the time scales associated with chemical reaction become small compared to the fluid time scales. Implicit methods overcome these limitations but lose their high performance when applied to problems with more than one space dimension. A preconditioning procedure is developed which successfully removes the stiffness limitations due to chemical reactions and whose performance does not degrade when applied to problems with more than one space dimension. In addition the preconditioning technique can be applied to multiple grid acceleration methods which can further improve the efficiency of the scheme. The basic idea is to modify the original unsteady equations,

$$\frac{\partial U}{\partial t} = - \frac{\partial F}{\partial x} - \frac{\partial G}{\partial y} - H$$

to

$$S \frac{\partial U}{\partial t} = - \frac{\partial F}{\partial x} - \frac{\partial G}{\partial y} - H$$

where U, F, G, and H are the state quantities, both fluxes and source terms respectively. S is a preconditioning matrix which is used to rescale the equations in time so that each state quantity, U, is numerically advanced at its own characteristic rate. Specifically this means that a fluid quantity is marched at a fluid time scale and a species quantity at its own chemical rate time scale. Note both equations satisfy the same steady state equation. If only the steady state solution is desired then advancing each state quantity at its own characteristic rate produces a time inaccurate solution but one that converges very quickly to the steady state. With this particular technique the number of iterations needed to achieve steady state is found to be independent of the level of stiffness. Stiffness is defined as the ratio of the convective fluid time scale (based on the numerical cell dimension), to the chemical reaction time scale,

$$S = \frac{\tau_{fluid}}{\tau_{chem}} \quad .$$

For the problems of interest S can vary from 10 to 10^6, which can translate to substantial computational work savings. If a time accurate solution is desired then one needs only to reduce the numerical time step to the time scale of interest.

The preconditioning concepts were also shown to be extendable to the Ni multiple grid acceleration scheme. In addition the use of a local CFL number can improve the rate of convergence of the scheme.

The preconditioning procedure is applied to the MacCormack and Jameson, Schmidt and Turkel finite volume schemes. The preconditioned methods are applied to one and two dimensional problems with simple O_2 dissociation and a two step H_2 - air chemistry model.

9.2. 2-D Scramjet Flame Holders

Two candidate scramjet flame holders were assessed in this thesis. The first flame holder considered uses an oblique shock wave to trigger and hold a flame. It was found that with various levels of heat addition the flow field character could be changed from an entirely supersonic flow, to one containing embedded subsonic zones. If the level of heat addition is increased still further the flow becomes thermally choked and the shock system is forced out of the front of the inlet. A variety of parameters, including the ratio of the flame thickness to the channel geometric length , ϕ and inlet channel Mach number were varied and the results were plotted on a ϕ vs inlet Mach number map. This suggested map can identify the flow regime generated for a given set of flow conditions. ϕ is the ratio of the fuel to air ratio to the stoichiometric fuel to air ratio.

The second class of flame holders considered were rearward facing steps. Rearward facing steps act as flame holders by creating a hot region from which a flame can be anchored. It was found that reacting flows over rearward facing steps could produce a variety of different flow fields. Is is suggested that some of the different possible flow fields could be plotted on a ϕ vs Tmax/Tig map.

9.3. Recommendations For Future Work

Based on this study several recommendations can be made about possible future work. These recommendations can be summarized as follows:

1. Improved damping scheme
2. Real gas effects
3. Grid embedding
4. Turbulence modeling
5. A more accurate H_2 - air chemistry model
6. Extend the method to 3 - D

Each recommendation is explained in detail below.

Improved Damping Scheme

The 2-D viscous results indicate the importance of ensuring that the artificial viscosity terms are not a significant fraction of the sum of the convective plus the real viscous terms. This condition is particularly important when dealing with reacting flows where the species gradients can be large. It is suggested that an artificial viscosity scheme like the one proposed by Jameson be used. He used a switch on the second order smoothing term related to the pressure. It might prove advantageous to base the switch on the species concentration when damping the species equations. This could prove helpful since a thin reaction zone might be produced, which could have only a small pressure gradient. In this case a switch based on pressure would be small in value which could be insufficient for effective damping.

Real Gas Effects

The non-reacting rearward facing step calculation indicates the importance of modeling the real behavior of the gases. In particular modeling C_p and C_v as functions of species gradients, temperature and pressure and not just species gradients, as was done here, can greatly improve the solution accuracy. This occurs because the solution is very sensitive to all of these quantities. This information can be obtained from the NASA gas tables [34] where the fluid properties are given as functions of pressure and temperature in tabular form or in the form of parametric equations. Either form could be used effectively to model the real gas behavior of the fluid.

Grid Embedding

In recent years grid embedding ([55], [12], [11]) has received considerable attention. The advantages of this approach is that grid points are added only where needed and points are not wasted in regions where nothing happens. The disadvantages are that the technique tends to be complex and to have a complicated data base. With these limitations in mind the technique seems to offer the potential of significant work and storage savings compared to non-embedding techniques [11]. The embedding techniques would be particulary useful in the recirculation region behind rearward facing steps where high accuracy is required to predict flame holding.

Turbulence Modeling

Extending the present work to include turbulence effects is important since most scramjet flows are turbulent. For most scramjet applications Sindir [49] showed that the least complex turbulence model that can adaquately describe these flows is the k - ϵ turbulence model. He found that the simpler models, i.e. algebraic, could not reliably predict the flow behind geometries like rearward facing steps. In addition with chemical reaction these discrepancies are expected to become greater.

Sindir found that when he solved the fluid transport and k - ϵ turbulence equations he encountered a stiffness problem similar to the one characteristic of the reacting species equation. The preconditioning procedure developed as part of this thesis can be used to overcome this problem. In this case the source terms represent the production of turbulent kinetic energy and the dissipation rate of turbulent kinetic energy. The k - ϵ turbulence transport equations can be written in conservative form and solved using the techniques described in chapters 4 and 5.

Care must be taken to insure that the effect of turbulence on the chemical reaction rate is properely accounted for. Turbulence adds a random fluctuation that produces a temperature and species fluctuation which can significantly increase the burning rate. This interaction is particularly important when the turbulence time scales are of the same order of the chemical reaction time scales.

A More Accurate H_2 - Air Chemistry Model

The H_2 - Air chemistry model used in the thesis was developed by Rogers and Chinitz [44]. The model is a simplified version of a more accurate 8 step model and is intended to model scramjet combustor flow. Using the 8 step reaction model gets around the problem of having to specify the ignition temperature, since the model is valid below the ignition temperature, and provides a more accurate reaction detail important in flame holding.

The implementation of this model will follow the method described at the beginning of this thesis. The only difficulties would be the need to add two new species transport equations for the O and H radicals. In addition the matricies which need to be solved become 10 * 10 instead of the 8 * 8 used here, which implies some additional work in computing the jacobians.

Extend The Method To 3-D

Three dimensional effects can have a strong effect on the overall flow field. Using the 2-D equations to predict the flow over a rearward facing step allows information to propagate from the recirculation zone to the outer-flow only by diffusion. In 3-D secondary flow fields could arise augmenting the diffusion mechanism with 3-D convective mechanisms. Thus 3-D effects could increase the exchange of information between the recirculation zone and the outer-flow leading to more effective flame holding.

In reality fuel is injected through circular holes producing highly 3-D flow fields. It is believed that the recirculation behind these jets can produce stable flame holding. Therefore the ability to compute 3-D nonreacting/reacting flow fields would have many useful applications.

Appendix 1 H_2-Air Preconditioner

The preconditioning matrix **S** is given by,

$$\mathbf{S} = [I + \Delta t \frac{\partial H}{\partial U}] \tag{1}$$

where $\partial H/\partial U$ is the Jacobian matrix of the source term with respect to the state quantities U. Note the dimension of **S** is equal to the number of state quantities U. For the 2-D Navier-Stokes equations with the Roger's and Chinitz H_2-air chemistry model, the number of required state quantities is 8. In this case the Jacobian consits of 64 elements. However, many of these elements are equal to zero. The **S** matrix used with the Roger's and chinitz chemistry model will now be derivied.

The state quantities U are given as $U = [\rho, \rho u, \rho v, \rho E, \rho Y_{H_2}, \rho Y_{O_2}, \rho Y_{H_2O}, \rho Y_{N_2}]$ while the source terms H are $H = [0, 0, 0, 0, w_{H_2}, w_{O_2}, w_{H_2O}, 0]$. The three non-zero source terms are,

$$w_{H_2} = Aw_{H_2} [- \frac{k_{f1} U_5 U_6}{Aw_{H_2} Aw_{O_2}} + \frac{k_{b1} (U_{OH})^2}{Aw_{OH}^2} - \frac{k_{f2} U_5 U_{OH}^2}{Aw_{H_2} Aw_{OH}^2} + \frac{k_{b2} U_7^2}{Aw_{H_2O}^2}] \tag{2}$$

$$w_{O_2} = Aw_{O_2} [- \frac{k_{f1} U_5 U_6}{Aw_{H_2} Aw_{O_2}} + \frac{k_{b1} (U_{OH})^2}{Aw_{OH}^2}] \tag{3}$$

$$w_{H_2O} = Aw_{H_2O}\, 2\, [\frac{k_{f2}U_5U_{OH}^2}{Aw_{H_2}Aw_{OH}^2} - \frac{k_{b2}U_7^2}{Aw_{H_2O}^2}] \tag{4}$$

where $U_{OH} = U_1 - U_5 - U_6 - U_7 - U_8$ and $H_5 = -w_5$, $H_6 = -w_6$ and $H_7 = -w_7$. With the U's and H's the Jacobian elements can be evaluated as follows,

$$\frac{\partial H_n}{\partial U_m} = 0 \tag{5}$$

for n = 1,2,3,4,8 and m = 1,2,3,4,5,6,7,8. Only rows 5, 6 and 7 of the matrix S contribute non-zero elements to the Jacobian. For example the contributions to row 5 from the Jacobian are,

$$\frac{\partial H_5}{\partial U_1} = 2\ Aw_{H_2}\ [\frac{k_{b1}U_{OH}}{Aw_{OH}^2} - \frac{k_{f2}U_5U_{OH}}{Aw_{H_2}Aw_{OH}^2}\ ] \tag{6}$$

$$\frac{\partial H_5}{\partial U_2} = \frac{\partial H_5}{\partial U_3} = 0 \tag{7}$$

$$\frac{\partial H_5}{\partial U_4} = \frac{\partial H_5}{\partial T}\,\frac{1}{\partial U_4/\partial T} \tag{8}$$

$$= Aw_{H_2}\ [- \frac{(\partial k_{f1}/\partial T)U_5U_6}{Aw_{H_2}Aw_{OH}^2} + \frac{(\partial k_{b1}/\partial T)(U_{OH})^2}{Aw_{OH}^2} - \frac{(\partial k_{f2}/\partial T)U_5U_{OH}^2}{Aw_{H_2}Aw_{OH}^2} + \frac{(\partial k_{b2}/\partial T)(U_7)^2}{Aw_{H_2O}^2}]\ \frac{1}{\partial U_4/\partial T} \tag{9}$$

$$\frac{\partial H_5}{\partial U_5} = Aw_{H_2} [- \frac{k_{f1}U_6}{Aw_{H_2}Aw_{O_2}} - 2\frac{k_{b1}U_{OH}}{Aw_{OH}^2} - \frac{k_{f2}U_{OH}^2}{Aw_{H_2}Aw_{OH}^2} + 2\frac{k_{f2}U_5U_{OH}}{Aw_{H_2}Aw_{OH}^2}] \tag{10}$$

$$\frac{\partial H_5}{\partial U_6} = Aw_{H_2} [- \frac{k_{f1}U_5}{Aw_{H_2}Aw_{O_2}} - 2\frac{k_{b1}U_{OH}}{Aw_{OH}^2} + 2\frac{k_{f2}U_5U_{OH}}{Aw_{H_2}Aw_{OH}^2}] \tag{11}$$

$$\frac{\partial H_5}{\partial U_7} = 2\ Aw_{H_2} [- \frac{k_{b1}U_{OH}}{Aw_{OH}^2} + \frac{k_{f2}U_5U_{OH}}{Aw_{H_2}Aw_{OH}^2} + \frac{2k_{b2}U_7}{Aw_{H_2O}}] \tag{12}$$

$$\frac{\partial H_5}{\partial U_8} = 2\ Aw_{H_2} [- \frac{-k_{b1}U_{OH}}{Aw_{OH}^2} + \frac{k_{f2}U_5U_{OH}}{Aw_{H_2}Aw_{OH}^2}] \tag{13}$$

where,

$$\frac{\partial k_{f1}}{\partial T} = \frac{A_1(\phi)T^{-11}}{T_\infty^{10}} [-10 + \frac{E_1}{R_u T T_\infty}] \exp(-E_1/R_u T T_\infty) \tag{14}$$

$$\frac{\partial k_{f2}}{\partial T} = \frac{A_2(\phi)T^{-14}}{T_\infty^{13}} [-13 + \frac{E_1}{R_u T T_\infty}] \exp(-E_1/R_u T T_\infty) \tag{15}$$

$$\frac{\partial k_{b1}}{\partial T} = \frac{1}{K_{eq1}} \frac{\partial k_{f1}}{\partial T} - \frac{k_{f1}}{K_{eq1}^2} \frac{\partial K_{eq1}}{\partial T} \tag{16}$$

$$\frac{\partial k_{b2}}{\partial T} = \frac{1}{K_{eq2}} \frac{\partial k_{f2}}{\partial T} - \frac{k_{f2}}{K_{eq2}^{2}} \frac{\partial K_{eq2}}{\partial T} \tag{17}$$

$$\frac{\partial U_4}{\partial T} = \rho c_v \frac{c_{v\infty} T_\infty}{U_\infty^{2}}. \tag{18}$$

Similarly, expressions for the elements of row 6 and 7 can be derived in the same manner.

Appendix 2 2-D Inviscid Code

```
      PROGRAM NSFRC(UNIT6=IFRCD)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                   C
C                                                                   C
C                   -----------------------                         C
C                   \                     \                         C
C                   \      I F R C        \                         C
C                   \                     \                         C
C                   -----------------------                         C
C                                                                   C
C                                                                   C
C                                                                   C
C            (  INVISCID - FINITE RATE CHEMISTRY   )                C
C                                                                   C
C                                                                   C
C                                                                   C
C            2-D INVISCID NON-EQUILIBRIUM CHEMISTRY SOLVER          C
C            -------------------------------------------            C
C                                                                   C
C            THE PROGRAM IS PRESENTLY SET UP TO SOLVE               C
C            FOR SUPERSONIC FLOW IN CHANNELS AND OVER               C
C            FORWARD FACING RAMPS. THE PROGRAM CAN ALSO            C
C            HANDLE FLOWS WITH EMBEDDED SUBSONIC ZONES.             C
C            THE SCHEME IS BASED ON AN EXTENED FINITE VOLUME        C
C            MULTISTEP EULER EQUATION METHOD PROPOSED BY            C
C            JAMESON( JUNE 1983).                                   C
C                                                                   C
C            THE EQUATIONS SOLVED ARE THE 2 - D EULER               C
C            EQUATIONS COUPLED WITH FINITE RATE CHEMISTRY.          C
C            THE ROGERS AND CHINITZ H2-AIR CHEMISTRY                C
C            MODEL IS MODEL THE COMBUSTION PROCESS                  C
C                                                                   C
C            THE SOURCE TERMS ARE INTEGRATED IMPLICITLY             C
C            WHILE THE SPATIAL DERIVATIES ARE INTEGRATED            C
C            EXPLICITLY (POINT IMPLICIT METHOD).                    C
C                                                                   C
C                                                                   C
C            CP AND CV ARE CURRENTLY DEPENDENT ONLY ON              C
C            THE SPECIES DENSITY FRACTIONS.                         C
C                                                                   C
```

```
C                                                                            C
C                          WRITTEN AT THE                                    C
C                MASSACHUSETTS INSTITUTE OF TECHNOLOGY                       C
C                           DATE STARTED                                     C
C                            MARCH 1984                                      C
C                                BY                                          C
C                                                                            C
C                        THOMAS   BUSSING                                    C
C                                                                            C
C               THE DETAILS OF THE EQUATIONS AND NUMERICAL                   C
C               METHOD USED ARE GIVEN IN THOMAS BUSSING                      C
C               MIT PH-D THESIS (AUGUST 1985)                                C
C                                                                            C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      COMMON/VAR0/U(8,60,60,2),FI(8,60,60),GI(8,60,60),AH(8,60,60)
      COMMON/VAR00/FV(4,8,60,60),GV(4,8,60,60)
      COMMON/VAR1/UVEL(60,60),VVEL(60,60),PRES(60,60),TEMP(60,60)
      COMMON/VAR111/YH2(60,60),YO2(60,60),YH2O(60,60),YOH(60,60)
      COMMON/VAR2/DEN(60,60),SOUND(60,60),AINTE(60,60),AMACH(60,60)
      COMMON/VAR3/ENTHP(60,60),VIS(60,60),YYN2(60,60),CPND(60,60)
      COMMON/VAR4/DXX,DYY,X(62,62),Y(62,62),AREA(60,60)
      COMMON/VAR5/DT(60,60)
      COMMON/VAR6/NX,NXX,NXXX,NY,NYY,NYYY,IRES,IEQ
      COMMON/VAR7/IVIS,NITER,NOITER
      COMMON/VAR8/P1,T1,AM1,VISL,U1,V1,AK1,CV,CP,R,GAMA,DEN1,E1
      COMMON/VAR9/P11,C11,E11
      COMMON/VAR10/DFH2,DFO2,DFH2O,DFOH,DFN2
      COMMON/VAR11/CPH2,CPO2,CPH2O,CPOH,CPN2,CVH2,CVO2,CVH2O,CVOH,CVN2
      COMMON/VAR12/CONH2,CONO2,CONH2O,CONOH,CONN2
      COMMON/VAR13/COND,CFL,DCOFF,AL,VELO1
      COMMON/VAR14/RES(8,60,60)
      COMMON/VAR15/ALPHA(4)
      COMMON/VAR16/DXE(4,60,60),DXN(4,60,60),DXW(4,60,60),DXS(4,60,60)
      COMMON/VAR17/DYE(4,60,60),DYN(4,60,60),DYW(4,60,60),DYS(4,60,60)
      COMMON/VAR18/DD(2,8,60,60)
      COMMON/VAR20/NSX,NSXB,NSXA,NSY,NSYB,NSYA
      COMMON/VAR21/ACOM
      COMMON/VAR22/TRIGTEMP,PHI
      COMMON/VAR23/REN,PR,FACT1,LAMB,SDIFF
      OPEN(UNIT=6,NAME='IFRCD.DAT',TYPE='UNKNOWN',FORM='FORMATTED')
C
C ---   STEP UP INPUT DATA
C
      CALL INPUT
C
C ---   STEP UP-INITALIZE
C
      CALL INIT
      CALL GRID
C
```

```
      DO 1 I        = 1 , NITER
      NOITER        = I
C
C --- MULTISTEP INTEGRATION - FOUR STEPS
C
      IA  = 1
C
      CALL PROPINV
      CALL STAB
      CALL FLUX
      CALL SOURCE
      CALL NSSOLVE(IA)
C
      IA  = 2
C
      CALL PROPINV
      CALL STAB
      CALL FLUX
      CALL SOURCE
      CALL NSSOLVE(IA)
C
      IA  = 3
C
      CALL PROPINV
      CALL STAB
      CALL FLUX
      CALL SOURCE
      CALL NSSOLVE(IA)
C
      IA  = 4
C
      CALL PROPINV
      CALL STAB
      CALL FLUX
      CALL SOURCE
      CALL NSSOLVE(IA)
C
C ---  POST SPLIT SMOOTHING
C
      CALL DAMPX
      CALL DAMPY
C
C ---  DETERMINE IF STEADY STATE HAS BEEN ACHIEVED
C
      CALL CONV
      IF(IRES.EQ.0.AND.I.GT.50) GO TO 100
      DO 2 IY        = 1 , NYY
      DO 2 IX        = 1 , NXX
      DO 2 IZ        = 1 , IEQ
      U(IZ,IX,IY,1) = U(IZ,IX,IY,2)
```

```
2       CONTINUE
1       CONTINUE
100     CONTINUE
C
C ---     DUMP OUTPUT DATA TO DATA FILES
C
        CALL OUT
C
C ---     COMPUTE AND PRINT OUT GROSS COMBUSTION PARAMETERS
C
        CALL PARAM
        STOP
        END
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
        SUBROUTINE INPUT
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C ---   LOAD INPUT DATA
C
        COMMON/VAR0/U(8,60,60,2),FI(8,60,60),GI(8,60,60),AH(8,60,60)
        COMMON/VAR00/FV(4,8,60,60),GV(4,8,60,60)
        COMMON/VAR1/UVEL(60,60),VVEL(60,60),PRES(60,60),TEMP(60,60)
        COMMON/VAR111/YH2(60,60),YO2(60,60),YH2O(60,60),YOH(60,60)
        COMMON/VAR2/DEN(60,60),SOUND(60,60),AINTE(60,60),AMACH(60,60)
        COMMON/VAR3/ENTHP(60,60),VIS(60,60),YYN2(60,60),CPND(60,60)
        COMMON/VAR4/DXX,DYY,X(62,62),Y(62,62),AREA(60,60)
        COMMON/VAR5/DT(60,60)
        COMMON/VAR6/NX,NXX,NXXX,NY,NYY,NYYY,IRES,IEQ
        COMMON/VAR7/IVIS,NITER,NOITER
        COMMON/VAR8/P1,T1,AM1,VISL,U1,V1,AK1,CV,CP,R,GAMA,DEN1,E1
        COMMON/VAR9/P11,C11,E11
        COMMON/VAR10/DFH2,DFO2,DFH2O,DFOH,DFN2
        COMMON/VAR11/CPH2,CPO2,CPH2O,CPOH,CPN2,CVH2,CVO2,CVH2O,CVOH,CVN2
        COMMON/VAR12/CONH2,CONO2,CONH2O,CONOH,CONN2
        COMMON/VAR13/COND,CFL,DCOFF,AL,VELO1
        COMMON/VAR14/RES(8,60,60)
        COMMON/VAR15/ALPHA(4)
        COMMON/VAR16/DXE(4,60,60),DXN(4,60,60),DXW(4,60,60),DXS(4,60,60)
        COMMON/VAR17/DYE(4,60,60),DYN(4,60,60),DYW(4,60,60),DYS(4,60,60)
        COMMON/VAR18/DD(2,8,60,60)
        COMMON/VAR20/NSX,NSXB,NSXA,NSY,NSYB,NSYA
        COMMON/VAR21/ACOM
        COMMON/VAR22/TRIGTEMP,PHI
        COMMON/VAR23/REN,PR,FACT1,LAMB,SDIFF
C
C ---     READ INPUT DATA("INPUT.DAT")
C
C ---   NITER = MAXIMUN NUMBER OF ITERATIONS
C
        NITER   = 100
```

```
C
C ---  NX, NY NUMBER OF POINTS IN THE X AND Y DIRECTIONS
C
      NX      = 60
      NY      = 60
C
C ---  NSX , NSY POINTS WHERE STEP ENDS
C
      NSX     = 11
      NSY     = 11
C
C ---  IEQ - NUMBER OF TRANSPORT EQUATIONS SOLVED
C
      IEQ     = 8
C
C ---  FOR PURELY INVISCID CALCULATION   CFL = 2. (APPROXIMATELY)
C ---  FOR A VISCOUS CALCULATION         CFL = .5 (APPROXIMATELY)
C
      CFL     = 1.
C
C ---  ARTIFICIAL VISCOSITY COEFFICIENT  INVISCID DCOFF = .1
C ---  ARTIFICIAL VISCOSITY COEFFICIENT  VISCOUS  DCOFF = .05
C
      DCOFF   = .1
C
C ---  RESCONV  CONVERGENCE CRITERIA
C
      RESCONV = .0001
C
C ---  ALPHA(1,2,3,4) CONSTANTS USED BY THE TIME INTEGRATOR
C
      ALPHA(1) = .25
      ALPHA(2) = .33
      ALPHA(3) = .5
      ALPHA(4) = 1.
C
C ---   FOR INVISCID CALCULATION    IVIS = 0
C ---   FOR VISCOUS  CALCULATION    IVIS = 1
C
      IVIS     = 0
C
C ---  P1 FREE STREAM PRESSURE (N/M**2)
      P1       = 100000.
C
C ---  T1 FREE STREAM TEMPERATURE (K)
C
      T1       = 900.
C
C ---  U1 FREE STREAM U VELOCITY (M/S)
C
```

```
      U1        = 1500.
C
C ---  V1 FREE STREAM V VELOCITY (M/S)
C
      V1        = 0.0
C
C ---  AL TEST SECTION LENGTH
C
      AL        = 1.
C
C ---  VISL   VISCOSITY AT FREE STREAM TEMPERATURE (N/M**2S)
C
      VISL      = 5.0E-5
C
C ---  COND   THERMAL DIFFUSIVITY AT FREE STREAM TEMPERATURE (W/MK)
C
      COND      = 2.4E-2
C
C ---  CP, CV    SPECIFIC HEATS AT FREE STREAM TEMPERATURE (J/KGK)
C
      CP        = 1000.
      CV        = 718.
C
C ---  GAS HEATS OF FORMATION AT ZERO DEGREES KELVIN (J/KG)
C
      DFH2      = 0.0
      DFO2      = 0.0
      DFH2O     = -1.44E+7
      DFOH      = 2.3E+6
      DFN2      = 0.0
C
C ---  GAS SPECIFIC HEATS AT CONSTANT PRESSURE (J/KG*K)
C
      CPH2      = 17160.
      CPO2      = 1181.
      CPH2O     = 2854.
      CPOH      = 2041
      CPN2      = 1285.
C
C ---  GAS SPECIFIC HEATS AT CONSTANT VOLUME (J/KG*K)
C
      CVH2      = 13000.
      CVO2      = 921.2
      CVH2O     = 2390.
      CVOH      = 1552.
      CVN2      = 988.
C
C ---  INFLOW SPECIES DENSITY FRACTIONS
C
      CONH2     = .002619
```

```
      CONO2    = .2095
      CONH2O   = .0
      CONOH    = .0
      CONN2    = .787881
C
C ---  IF NO CHEMICAL REACTION IS DESIRED ACOM = 0
C ---  IF CHEMICAL REACTION IS DESIRED    ACOM = 1
C
      ACOM     = 0.
C
C ---  TRIGGER TEMPERATURE - ABOVE THIS TEMPERATURE CHEMICAL REACTION
C                            OCCURS
C                          - BELOW THIS TEMPERATURE THE CHEMISTRY IS
C                            FROZEN
      TRIGTEMP = 1000.
C
C
C*****************************************************************
C
C
C ---  WRITE OUT THE INPUT QUANTITIES TO THE OUTPUT DATA FILE  "NSFRCD"
C
      WRITE(6,10) NX , NY , NITER , IEQ
      WRITE(6,11) NSX , NSY
      WRITE(6,12) CFL , DCOFF
      WRITE(6,13) RESCONV
      WRITE(6,14) ALPHA(1),ALPHA(2),ALPHA(3),ALPHA(4)
      WRITE(6,15) IVIS
      WRITE(6,16) P1 , T1 , U1 , V1 ,AL
      WRITE(6,14) VISL , COND , CP , CV
      WRITE(6,16) DFH2 , DFO2 , DFH2O , DFOH , DFN2
      WRITE(6,16) CPH2 , CPO2 , CPH2O , CPOH , CPN2
      WRITE(6,16) CVH2 , CVO2 , CVH2O , CVOH , CVN2
      WRITE(6,16) CONH2 , CONO2 , CONH2O , CONOH , CONN2
      WRITE(6,17) ACOM , TRIGTEMP , PHI
10    FORMAT(1X,4(I4))
11    FORMAT(1X,2(I4))
12    FORMAT(1X,2(E10.4))
13    FORMAT(1X,E10.4)
14    FORMAT(1X,4(E10.4))
15    FORMAT(1X,I4)
16    FORMAT(1X,5(E10.4))
17    FORMAT(1X,3(E10.4))
      RETURN
      END
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      SUBROUTINE INIT
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C ---  GENERATE INITIAL CONDITIONS
```

```
C
      COMMON/VAR0/U(8,60,60,2),FI(8,60,60),GI(8,60,60),AH(8,60,60)
      COMMON/VAR00/FV(4,8,60,60),GV(4,8,60,60)
      COMMON/VAR1/UVEL(60,60),VVEL(60,60),PRES(60,60),TEMP(60,60)
      COMMON/VAR111/YH2(60,60),YO2(60,60),YH2O(60,60),YOH(60,60)
      COMMON/VAR2/DEN(60,60),SOUND(60,60),AINTE(60,60),AMACH(60,60)
      COMMON/VAR3/ENTHP(60,60),VIS(60,60),YYN2(60,60),CPND(60,60)
      COMMON/VAR4/DXX,DYY,X(62,62),Y(62,62),AREA(60,60)
      COMMON/VAR5/DT(60,60)
      COMMON/VAR6/NX,NXX,NXXX,NY,NYY,NYYY,IRES,IEQ
      COMMON/VAR7/IVIS,NITER,NOITER
      COMMON/VAR8/P1,T1,AM1,VISL,U1,V1,AK1,CV,CP,R,GAMA,DEN1,E1
      COMMON/VAR9/P11,C11,E11
      COMMON/VAR10/DFH2,DFO2,DFH2O,DFOH,DFN2
      COMMON/VAR11/CPH2,CPO2,CPH2O,CPOH,CPN2,CVH2,CVO2,CVH2O,CVOH,CVN2
      COMMON/VAR12/CONH2,CONO2,CONH2O,CONOH,CONN2
      COMMON/VAR13/COND,CFL,DCOFF,AL,VELO1
      COMMON/VAR14/RES(8,60,60)
      COMMON/VAR15/ALPHA(4)
      COMMON/VAR16/DXE(4,60,60),DXN(4,60,60),DXW(4,60,60),DXS(4,60,60)
      COMMON/VAR17/DYE(4,60,60),DYN(4,60,60),DYW(4,60,60),DYS(4,60,60)
      COMMON/VAR18/DD(2,8,60,60)
      COMMON/VAR20/NSX,NSXB,NSXA,NSY,NSYB,NSYA
      COMMON/VAR21/ACOM
      COMMON/VAR22/TRIGTEMP,PHI
      COMMON/VAR23/REN,PR,FACT1,LAMB,SDIFF
C
C ---  DETERMINE REMAINING INPUT UNKNOWNS
C
      CP    = CONH2 * CPH2  + CONO2 * CPO2 + CONH2O * CPH2O
     1                      + CONOH * CPOH + CONN2  * CPN2
      CV    = CONH2 * CVH2  + CONO2 * CVO2 + CONH2O * CVH2O
     1                      + CONOH * CVOH + CONN2  * CVN2
      R     = CP - CV
      GAMA  = CP / CV
      DHEATF= CONH2 * DFH2 + CONO2 * DFO2 + CONH2O * DFH2O
     1                     + CONOH * DFOH + CONN2  * DFN2
C
C --- PHI IS THE FUEL EQUIVALENCE RATIO
C
      PHI   = (CONH2 / CONO2) * 8.
C
      DEN1  = P1/(R*T1)
      VELO1 = SQRT(U1**2 + V1**2)
      E1    = CV*T1+.5*VELO1**2 + DHEATF / VELO1**2
C
C ---  DETERMINE  NON-DIMENSIONAL VARIABLES
C
      P11        = P1/(DEN1*VELO1**2)
      E11        = CV*T1/VELO1**2 + DHEATF /VELO1**2 + .5
```

```
      C11         = (SQRT(GAMA*R*T1))/VELO1
C
C ---   COMPUTE THE FREE STREAM NON-DIMENSIONAL VARIABLES
C
      REN         = DEN1 * VELO1 * AL / VISL
      PR          = VISL * CP / COND
      AM1         = VELO1/SQRT(GAMA*R*T1)
      FACT        = 1.0 / ((GAMA - 1.0)*AM1**2)
      LAMB        = -.6666 / REN
      SDIFF       = 1.0 / (REN * PR)
C
      DO 1 J      = 1 , NY
      DO 1 I      = 1 , NX
      PRES(I,J)   = P11
      TEMP(I,J)   = 1.0
      UVEL(I,J)   = U1 / VELO1
      VVEL(I,J)   = V1 / VELO1
      YH2(I,J)    = CONH2
      YO2(I,J)    = CONO2
      YH2O(I,J)   = CONH2O
      YOH(I,J)    = CONOH
      YYN2(I,J)    = CONN2
      VELO        = UVEL(I,J)**2 + VVEL(I,J)**2
      AINTE(I,J)  = E11
      ENTHP(I,J)  = CP*T1/(VELO1**2)+.5*VELO + DHEATF / VELO1**2
      SOUND(I,J)  = C11
      AMACH(I,J)  = AM1
      CPND(I,J)   = 1.0
      DEN(I,J)    = 1.0
      U(1,I,J,1)  = 1.0
      U(2,I,J,1)  = UVEL(1,J)
      U(3,I,J,1)  = VVEL(I,J)
      U(4,I,J,1)  = E11
      U(5,I,J,1)  = CONH2
      U(6,I,J,1)  = CONO2
      U(7,I,J,1)  = CONH2O
      U(8,I,J,1)  = CONN2
      U(1,I,J,2)  = 1.0
      U(2,I,J,2)  = UVEL(1,J)
      U(3,I,J,2)  = VVEL(I,J)
      U(4,I,J,2)  = E11
      U(5,I,J,2)  = CONH2
      U(6,I,J,2)  = CONO2
      U(7,I,J,2)  = CONH2O
      U(8,I,J,2)  = CONN2
      FI(1,I,J)   = UVEL(1,J)
      FI(2,I,J)   = UVEL(1,J)**2 + P11
      FI(3,I,J)   = UVEL(I,J)*VVEL(I,J)
      FI(4,I,J)   = (P11 + E11)*UVEL(1,J)
      FI(5,I,J)   = UVEL(1,J) * CONH2
```

```
      FI(6,I,J)  = UVEL(1,J) * CONO2
      FI(7,I,J)  = UVEL(1,J) * CONH2O
      FI(8,I,J)  = UVEL(1,J) * CONN2
      GI(1,I,J)  = VVEL(I,J)
      GI(2,I,J)  = UVEL(I,J)*VVEL(I,J)
      GI(3,I,J)  = VVEL(I,J)**2 + P11
      GI(4,I,J)  = (P11 + E11 )*VVEL(I,J)
      GI(5,I,J)  = VVEL(1,J) * CONH2
      GI(6,I,J)  = VVEL(1,J) * CONO2
      GI(7,I,J)  = VVEL(1,J) * CONH2O
      GI(8,I,J)  = VVEL(1,J) * CONN2
      AH(1,I,J)  = 0.0
      AH(2,I,J)  = 0.0
      AH(3,I,J)  = 0.0
      AH(4,I,J)  = 0.0
      AH(5,I,J)  = 0.0
      AH(6,I,J)  = 0.0
      AH(7,I,J)  = 0.0
      AH(8,I,J)  = 0.0
      DO 1 K     = 1 , IEQ
      FV(1,K,I,J) = 0.0
      FV(2,K,I,J) = 0.0
      FV(3,K,I,J) = 0.0
      FV(4,K,I,J) = 0.0
      GV(1,K,I,J) = 0.0
      GV(2,K,I,J) = 0.0
      GV(3,K,I,J) = 0.0
      GV(4,K,I,J) = 0.0
1     CONTINUE
C
C ---    ZERO OUT ALL VISCOUS LENGTH ARRAYS
C
      DO 2 K = 1 , 4
      DO 2 I = 1 , NX
      DO 2 J = 1 , NY
      DXE(K,I,J) = 0.0
      DXN(K,I,J) = 0.0
      DXW(K,I,J) = 0.0
      DXS(K,I,J) = 0.0
      DYE(K,I,J) = 0.0
      DYN(K,I,J) = 0.0
      DYW(K,I,J) = 0.0
      DYS(K,I,J) = 0.0
2     CONTINUE
      RETURN
      END
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      SUBROUTINE GRID
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
```

```
C ---  GENERATE THE PHYSICAL GRID
C
       COMMON/VAR0/U(8,60,60,2),FI(8,60,60),GI(8,60,60),AH(8,60,60)
       COMMON/VAR00/FV(4,8,60,60),GV(4,8,60,60)
       COMMON/VAR1/UVEL(60,60),VVEL(60,60),PRES(60,60),TEMP(60,60)
       COMMON/VAR111/YH2(60,60),YO2(60,60),YH2O(60,60),YOH(60,60)
       COMMON/VAR2/DEN(60,60),SOUND(60,60),AINTE(60,60),AMACH(60,60)
       COMMON/VAR3/ENTHP(60,60),VIS(60,60),YYN2(60,60),CPND(60,60)
       COMMON/VAR4/DXX,DYY,X(62,62),Y(62,62),AREA(60,60)
       COMMON/VAR5/DT(60,60)
       COMMON/VAR6/NX,NXX,NXXX,NY,NYY,NYYY,IRES,IEQ
       COMMON/VAR7/IVIS,NITER,NOITER
       COMMON/VAR8/P1,T1,AM1,VISL,U1,V1,AK1,CV,CP,R,GAMA,DEN1,E1
       COMMON/VAR9/P11,C11,E11
       COMMON/VAR10/DFH2,DFO2,DFH2O,DFOH,DFN2
       COMMON/VAR11/CPH2,CPO2,CPH2O,CPOH,CPN2,CVH2,CVO2,CVH2O,CVOH,CVN2
       COMMON/VAR12/CONH2,CONO2,CONH2O,CONOH,CONN2
       COMMON/VAR13/COND,CFL,DCOFF,AL,VELO1
       COMMON/VAR14/RES(8,60,60)
       COMMON/VAR15/ALPHA(4)
       COMMON/VAR16/DXE(4,60,60),DXN(4,60,60),DXW(4,60,60),DXS(4,60,60)
       COMMON/VAR17/DYE(4,60,60),DYN(4,60,60),DYW(4,60,60),DYS(4,60,60)
       COMMON/VAR18/DD(2,8,60,60)
       COMMON/VAR20/NSX,NSXB,NSXA,NSY,NSYB,NSYA
       COMMON/VAR21/ACOM
       COMMON/VAR22/TRIGTEMP,PHI
       COMMON/VAR23/REN,PR,FACT1,LAMB,SDIFF
C
       NXMAX= NX + 1
       NYMAX= NY + 1
       NXX  = NX-1
       NXXX = NX - 2
       NYY  = NY-1
       NYYY = NY - 2
       NSXB = NSX - 1
       NSXA = NSX + 1
       NSYB = NSY - 1
       NSYA = NSY + 1
C
C       ********************************
C ---   INPUT THE MESH DISTRIBUTION HERE
C       ********************************
C
C ---   X COORDINATE FORMULATION
C
       DXX      = 1.0 / NXX
       DYY      = .5  / NYY
       X(1,1)   = 0.0
       DO 1 I   = 2 , NXMAX
1      X(I,1)   = X(I-1,1)+DXX
```

```
      DO 2 J    = 2 , NYMAX
      DO 2 I    = 1 ,NXMAX
2     X(I,J)    = X(I,1)
C
C ---  Y COORDINATE FORMULATION
C
      DO 4 I    = 1 , NXMAX
      Y(I,1)    = 0.0
      DO 4 J    = 2 , NYMAX
      IF(I.LE.8)  DHY = .5
      IF(I.GT.8.AND.I.LE.40) DHY = .5 - .1*(X(I,J)-X(8,J))/
     *                          (X(40,J)-X(8,J))
      IF(I.GT.40) DHY = .4
      Y(I,J)           = Y(I,J-1) + DHY/NYY
4     CONTINUE

C       **********************************************
C ---   REMAINDER OF GRID GENERATION PROCESS AUTOMATIC
C       **********************************************
C
C ---   DETERMINE THE AREA OF EACH CELL
C
      DO 10 J = 1 , NY
      DO 10 I = 1 , NX
      A1        = (X(I+1,J+1) - X(I,J))   * (Y(I,J+1)   - Y(I+1,J))
      A2        = (X(I,J+1)   - X(I+1,J)) * (Y(I+1,J+1) - Y(I,J))
      AREA(I,J) = (ABS(A1) + ABS(A2)) / 2.0
10    CONTINUE
      DO 20 J      = 1 , NYY
      DO 20 I      = 1 , NXX
C
C ---  PROJECTIONS OF VISCOUS CELL EAST SIDE
C
      XE1          = X(I+2,J)   - X(I+2,J+1)
      XE2          = X(I+1,J)   - X(I+1,J+1)
      XN1          = X(I+2,J+1) - X(I+1,J+1)
      XN2          = X(I+1,J+1) - X(I,J+1)
      XW1          = X(I+1,J+1) - X(I+1,J)
      XW2          = X(I,J+1)   - X(I,J)
      XS1          = X(I,J)     - X(I+1,J)
      XS2          = X(I+1,J)   - X(I+2,J)
      YE1          = Y(I+2,J)   - Y(I+2,J+1)
      YE2          = Y(I+1,J)   - Y(I+1,J+1)
      YN1          = Y(I+2,J+1) - Y(I+1,J+1)
      YN2          = Y(I+1,J+1) - Y(I,J+1)
      YW1          = Y(I+1,J+1) - Y(I+1,J)
      YW2          = Y(I,J+1)   - Y(I,J)
      YS1          = Y(I,J)     - Y(I+1,J)
      YS2          = Y(I+1,J)   - Y(I+2,J)
      DXE(1,I,J)         = - .5 * (XE1 + XE2)
```

```
      DXN(1,I,J)          = - .5 * (XN1 + XN2)
      DXW(1,I,J)          = - .5 * (XW1 + XW2)
      DXS(1,I,J)          = - .5 * (XS1 + XS2)
      DYE(1,I,J)          = - .5 * (YE1 + YE2)
      DYN(1,I,J)          = - .5 * (YN1 + YN2)
      DYW(1,I,J)          = - .5 * (YW1 + YW2)
      DYS(1,I,J)          = - .5 * (YS1 + YS2)
      DYS(1,I,J)          = - .5 * (YS1 + YS2)
20    CONTINUE
C
C ---  PROJECTIONS OF VISCOUS CELL WEST SIDE
C
      DO 21 J      = 1 , NYY
      DO 21 I      = 2 , NXX
      XE1         = X(I+1,J)   - X(I+1,J+1)
      XE2         = X(I,J)     - X(I,J+1)
      XN1         = X(I+1,J+1) - X(I,J+1)
      XN2         = X(I,J+1)   - X(I-1,J+1)
      XW1         = X(I,J+1)   - X(I,J)
      XW2         = X(I-1,J+1) - X(I-1,J)
      XS1         = X(I-1,J)   - X(I,J)
      XS2         = X(I,J)     - X(I+1,J)
      YE1         = Y(I+1,J)   - Y(I+1,J+1)
      YE2         = Y(I,J)     - Y(I,J+1)
      YN1         = Y(I+1,J+1) - Y(I,J+1)
      YN2         = Y(I,J+1)   - Y(I-1,J+1)
      YW1         = Y(I,J+1)   - Y(I,J)
      YW2         = Y(I-1,J+1) - Y(I-1,J)
      YS1         = Y(I-1,J)   - Y(I,J)
      YS2         = Y(I,J)     - Y(I+1,J)
      DXE(3,I,J)          = - .5 * (XE1 + XE2)
      DXN(3,I,J)          = - .5 * (XN1 + XN2)
      DXW(3,I,J)          = - .5 * (XW1 + XW2)
      DXS(3,I,J)          = - .5 * (XS1 + XS2)
      DYE(3,I,J)          = - .5 * (YE1 + YE2)
      DYN(3,I,J)          = - .5 * (YN1 + YN2)
      DYW(3,I,J)          = - .5 * (YW1 + YW2)
      DYS(3,I,J)          = - .5 * (YS1 + YS2)
21    CONTINUE
C
C ---  PROJECTIONS OF VISCOUS CELL WEST SIDE FIRST CELL
C
      I           = 1
      DO 22  J    = 1 , NYY
      XE1         = X(I+1,J)   - X(I+1,J+1)
      XE2         = X(I,J)     - X(I,J+1)
      XN1         = X(I+1,J+1) - X(I,J+1)
      XN2         = XN1
      XW1         = X(I,J+1)   - X(I,J)
      XW2         = XW1
```

```
      XS2          = X(I,J)     - X(I+1,J)
      XS1          = XS2
      YE1          = Y(I+1,J)   - Y(I+1,J+1)
      YE2          = Y(I,J)     - Y(I,J+1)
      YN1          = Y(I+1,J+1) - Y(I,J+1)
      YN2          = YN1
      YW1          = Y(I,J+1)   - Y(I,J)
      YW2          = YW1
      YS2          = Y(I,J)     - Y(I+1,J)
      YS1          = YS2
      DXE(3,I,J)          = - .5 * (XE1 + XE2)
      DXN(3,I,J)          = - .5 * (XN1 + XN2)
      DXW(3,I,J)          = - .5 * (XW1 + XW2)
      DXS(3,I,J)          = - .5 * (XS1 + XS2)
      DYE(3,I,J)          = - .5 * (YE1 + YE2)
      DYN(3,I,J)          = - .5 * (YN1 + YN2)
      DYW(3,I,J)          = - .5 * (YW1 + YW2)
      DYS(3,I,J)          = - .5 * (YS1 + YS2)
22    CONTINUE
C
C ---  PROJECTIONS OF VISCOUS CELL NORTH FACE
C
      DO 30 J      = 1 , NYY
      DO 30 I      = 1 , NXX
      XE1          = X(I+1,J)     - X(I+1,J+1)
      XE2          = X(I+1,J+1)   - X(I+1,J+2)
      XN1          = X(I+1,J+1)   - X(I,J+1)
      XN2          = X(I+1,J+2)   - X(I,J+2)
      XW1          = X(I,J+2)     - X(I,J+1)
      XW2          = X(I,J+1)     - X(I,J)
      XS1          = X(I,J+1)     - X(I+1,J+1)
      XS2          = X(I,J)       - X(I+1,J)
      YE1          = Y(I+1,J)     - Y(I+1,J+1)
      YE2          = Y(I+1,J+1)   - Y(I+1,J+2)
      YN1          = Y(I+1,J+1)   - Y(I,J+1)
      YN2          = Y(I+1,J+2)   - Y(I,J+2)
      YW1          = Y(I,J+2)     - Y(I,J+1)
      YW2          = Y(I,J+1)     - Y(I,J)
      YS1          = Y(I,J+1)     - Y(I+1,J+1)
      YS2          = Y(I,J)       - Y(I+1,J)
      DXE(2,I,J)         = - .5 * (XE1 + XE2)
      DXN(2,I,J)         = - .5 * (XN1 + XN2)
      DXW(2,I,J)         = - .5 * (XW1 + XW2)
      DXS(2,I,J)         = - .5 * (XS1 + XS2)
      DYE(2,I,J)         = - .5 * (YE1 + YE2)
      DYN(2,I,J)         = - .5 * (YN1 + YN2)
      DYW(2,I,J)         = - .5 * (YW1 + YW2)
      DYS(2,I,J)         = - .5 * (YS1 + YS2)
30    CONTINUE
C
```

```
C ---   PROJECTIONS OF VISCOUS CELL SOUTH SIDE
C
        DO 50 J       = 2 , NYY
        DO 50 I       = 1 , NXX
        XE1          = X(I+1,J-1)   - X(I+1,J)
        XE2          = X(I+1,J)     - X(I+1,J+1)
        XN1          = X(I+1,J)     - X(I,J)
        XN2          = X(I+1,J+1)   - X(I,J+1)
        XW1          = X(I,J+1)     - X(I,J)
        XW2          = X(I,J)       - X(I,J-1)
        XS1          = X(I,J)       - X(I+1,J)
        XS2          = X(I,J-1)     - X(I+1,J-1)
        YE1          = Y(I+1,J-1)   - Y(I+1,J)
        YE2          = Y(I+1,J)     - Y(I+1,J+1)
        YN1          = Y(I+1,J)     - Y(I,J)
        YN2          = Y(I+1,J+1)   - Y(I,J+1)
        YW1          = Y(I,J+1)     - Y(I,J)
        YW2          = Y(I,J)       - Y(I,J-1)
        YS1          = Y(I,J)       - Y(I+1,J)
        YS2          = Y(I,J-1)     - Y(I+1,J-1)
        DXE(4,I,J)         = - .5 * (XE1 + XE2)
        DXN(4,I,J)         = - .5 * (XN1 + XN2)
        DXW(4,I,J)         = - .5 * (XW1 + XW2)
        DXS(4,I,J)         = - .5 * (XS1 + XS2)
        DYE(4,I,J)         = - .5 * (YE1 + YE2)
        DYN(4,I,J)         = - .5 * (YN1 + YN2)
        DYW(4,I,J)         = - .5 * (YW1 + YW2)
        DYS(4,I,J)         = - .5 * (YS1 + YS2)
50      CONTINUE
C
C ---    LOWER WALL BOUNDARY CELL FLUX
C
        J            = 1
        DO 60 I      = 1 , NXX
        XE2          = X(I+1,J)     - X(I+1,J+1)
        XE1          = XE2
        XN1          = X(I+1,J)     - X(I,J)
        XN2          = X(I+1,J+1)   - X(I,J+1)
        XW1          = X(I,J+1)     - X(I,J)
        XW2          = XW1
        XS1          = -XN1
        XS2          = -XN2
        YE2          = Y(I+1,J)     - Y(I+1,J+1)
        YE1          = YE2
        YN1          = Y(I+1,J)     - Y(I,J)
        YN2          = Y(I+1,J+1)   - Y(I,J+1)
        YW1          = Y(I,J+1)     - Y(I,J)
        YW2          = YW1
        YS1          = -YN1
        YS2          = -YN2
```

```
      DXE(4,I,J)        = - .5 * (XE1 + XE2)
      DXN(4,I,J)        = - .5 * (XN1 + XN2)
      DXW(4,I,J)        = - .5 * (XW1 + XW2)
      DXS(4,I,J)        = - .5 * (XS1 + XS2)
      DYE(4,I,J)        = - .5 * (YE1 + YE2)
      DYN(4,I,J)        = - .5 * (YN1 + YN2)
      DYW(4,I,J)        = - .5 * (YW1 + YW2)
      DYS(4,I,J)        = - .5 * (YS1 + YS2)
60    CONTINUE
      RETURN
      END
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      SUBROUTINE FLUX
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C --- INDIVIDUAL STEP OF MULTISTEP INTEGRATOR
C
      COMMON/VAR0/U(8,60,60,2),FI(8,60,60),GI(8,60,60),AH(8,60,60)
      COMMON/VAR00/FV(4,8,60,60),GV(4,8,60,60)
      COMMON/VAR1/UVEL(60,60),VVEL(60,60),PRES(60,60),TEMP(60,60)
      COMMON/VAR111/YH2(60,60),YO2(60,60),YH2O(60,60),YOH(60,60)
      COMMON/VAR2/DEN(60,60),SOUND(60,60),AINTE(60,60),AMACH(60,60)
      COMMON/VAR3/ENTHP(60,60),VIS(60,60),YYN2(60,60),CPND(60,60)
      COMMON/VAR4/DXX,DYY,X(62,62),Y(62,62),AREA(60,60)
      COMMON/VAR5/DT(60,60)
      COMMON/VAR6/NX,NXX,NXXX,NY,NYY,NYYY,IRES,IEQ
      COMMON/VAR7/IVIS,NITER,NOITER
      COMMON/VAR8/P1,T1,AM1,VISL,U1,V1,AK1,CV,CP,R,GAMA,DEN1,E1
      COMMON/VAR9/P11,C11,E11
      COMMON/VAR10/DFH2,DFO2,DFH2O,DFOH,DFN2
      COMMON/VAR11/CPH2,CPO2,CPH2O,CPOH,CPN2,CVH2,CVO2,CVH2O,CVOH,CVN2
      COMMON/VAR12/CONH2,CONO2,CONH2O,CONOH,CONN2
      COMMON/VAR13/COND,CFL,DCOFF,AL,VELC1
      COMMON/VAR14/RES(8,60,60)
      COMMON/VAR15/ALPHA(4)
      COMMON/VAR16/DXE(4,60,60),DXN(4,60,60),DXW(4,60,60),DXS(4,60,60)
      COMMON/VAR17/DYE(4,60,60),DYN(4,60,60),DYW(4,60,60),DYS(4,60,60)
      COMMON/VAR18/DD(2,8,60,60)
      COMMON/VAR20/NSX,NSXB,NSXA,NSY,NSYB,NSYA
      COMMON/VAR21/ACOM
      COMMON/VAR22/TRIGTEMP,PHI
      COMMON/VAR23/REN,PR,FACT1,LAMB,SDIFF
      DO   1  J     = 2 , NYYY
      DO   1  I     = 2 , NXXX
      X1            = X(I+1,J+1) - X(I+1,J)
      X2            = X(I,J+1)   - X(I+1,J+1)
      X3            = X(I,J)     - X(I,J+1)
      X4            = X(I+1,J)   - X(I,J)
      Y1            = Y(I+1,J+1) - Y(I+1,J)
      Y2            = Y(I,J+1)   - Y(I+1,J+1)
```

```
      Y3            = Y(I,J)     - Y(I,J+1)
      Y4            = Y(I+1,J)   - Y(I,J)
      DO   2  K     = 1 , IEQ
      F1            = .5 * (FI(K,I+1,J) + FI(K,I,J))
      F2            = .5 * (FI(K,I,J+1) + FI(K,I,J))
      F3            = .5 * (FI(K,I-1,J) + FI(K,I,J))
      F4            = .5 * (FI(K,I,J-1) + FI(K,I,J))
      G1            = .5 * (GI(K,I+1,J) + GI(K,I,J))
      G2            = .5 * (GI(K,I,J+1) + GI(K,I,J))
      G3            = .5 * (GI(K,I-1,J) + GI(K,I,J))
      G4            = .5 * (GI(K,I,J-1) + GI(K,I,J))
      FFLUXEN       = (F1 + FV(1,K,I,J))*Y1 + (F2 + FV(2,K,I,J))*Y2
      FFLUXWS       = (F3 + FV(3,K,I,J))*Y3 + (F4 + FV(4,K,I,J))*Y4
      GFLUXEN       = (G1 + GV(1,K,I,J))*X1 + (G2 + GV(2,K,I,J))*X2
      GFLUXWS       = (G3 + GV(3,K,I,J))*X3 + (G4 + GV(4,K,I,J))*X4
      RESID         = (FFLUXEN + FFLUXWS) - (GFLUXEN + GFLUXWS)
      RES(K,I,J)    = RESID / AREA(I,J)
2     CONTINUE
1     CONTINUE
C
C      ---------------------------
C ---  INFLOW  BOUNDARY  EVALUATION
C      ---------------------------
C
      I             = 1
      DO   10 J     = 2 , NYYY
      X1            = X(I+1,J+1) - X(I+1,J)
      X2            = X(I,J+1)   - X(I+1,J+1)
      X3            = X(I,J)     - X(I,J+1)
      X4            = X(I+1,J)   - X(I,J)
      Y1            = Y(I+1,J+1) - Y(I+1,J)
      Y2            = Y(I,J+1)   - Y(I+1,J+1)
      Y3            = Y(I,J)     - Y(I,J+1)
      Y4            = Y(I+1,J)   - Y(I,J)
      DO   11 K     = 1 , IEQ
      F1            = .5 * (FI(K,I+1,J) + FI(K,I,J))
      F2            = .5 * (FI(K,I,J+1) + FI(K,I,J))
      IF(K.EQ.1)F3 = U1/VELO1
      IF(K.EQ.2)F3 = (U1/VELO1)**2 + P11
      IF(K.EQ.3)F3 = (U1 * V1)/VELO1**2
      IF(K.EQ.4)F3 = E11 * U1/VELO1 + U1*P11/VELO1
      IF(K.EQ.5)F3 = CONH2  * U1/VELO1
      IF(K.EQ.6)F3 = CONO2  * U1/VELO1
      IF(K.EQ.7)F3 = CONH2O * U1/VELO1
      IF(K.EQ.8)F3 = CONN2  * U1/VELO1
      F4            = .5 * (FI(K,I,J-1) + FI(K,I,J))
      G1            = .5 * (GI(K,I+1,J) + GI(K,I,J))
      G2            = .5 * (GI(K,I,J+1) + GI(K,I,J))
      IF(K.EQ.1)G3 = V1/VELO1
      IF(K.EQ.2)G3 = V1 * U1/VELO1**2
```

```
      IF(K.EQ.3)G3 = V1**2/VELO1**2 + P11
      IF(K.EQ.4)G3 = E11    * V1/VELO1 + V1*P11/VELO1
      IF(K.EQ.5)G3 = CONH2  * V1/VELO1
      IF(K.EQ.6)G3 = CONO2  * V1/VELO1
      IF(K.EQ.7)G3 = CONH2O * V1/VELO1
      IF(K.EQ.8)G3 = CONN2  * V1/VELO1
      G4           = .5 * (GI(K,I,J-1) + GI(K,I,J))
      FFLUXEN      = (F1 + FV(1,K,I,J))*Y1 + (F2 + FV(2,K,I,J))*Y2
      FFLUXWS      = (F3 + FV(3,K,I,J))*Y3 + (F4 + FV(4,K,I,J))*Y4
      GFLUXEN      = (G1 + GV(1,K,I,J))*X1 + (G2 + GV(2,K,I,J))*X2
      GFLUXWS      = (G3 + GV(3,K,I,J))*X3 + (G4 + GV(4,K,I,J))*X4
      RESID        = (FFLUXEN + FFLUXWS) - (GFLUXEN + GFLUXWS)
      RES(K,I,J)     = RESID / AREA(I,J)
11    CONTINUE
10    CONTINUE
C
C ---  INFLOW BOUNDARY UPPER CORNER CELL
C
      I            = 1
      J            = NYY
      X1           = X(I+1,J+1) - X(I+1,J)
      X2           = X(I,J+1)   - X(I+1,J+1)
      X3           = X(I,J)     - X(I,J+1)
      X4           = X(I+1,J)   - X(I,J)
      Y1           = Y(I+1,J+1) - Y(I+1,J)
      Y2           = Y(I,J+1)   - Y(I+1,J+1)
      Y3           = Y(I,J)     - Y(I,J+1)
      Y4           = Y(I+1,J)   - Y(I,J)
      DO   12 K    = 1 , IEQ
      F1           = .5 * (FI(K,I+1,J) + FI(K,I,J))
      F2           = 0.0
      IF(K.EQ.2)F2 = PRES(1,NYY)
      IF(K.EQ.1)F3 = U1/VELO1
      IF(K.EQ.2)F3 = (U1/VELO1)**2 + P11
      IF(K.EQ.3)F3 = (U1 * V1)/VELO1**2
      IF(K.EQ.4)F3 = E11 * U1/VELO1 + U1*P11/VELO1
      IF(K.EQ.5)F3 = CONH2  * U1/VELO1
      IF(K.EQ.6)F3 = CONO2  * U1/VELO1
      IF(K.EQ.7)F3 = CONH2O * U1/VELO1
      IF(K.EQ.8)F3 = CONN2  * U1/VELO1
      F4           = .5 * (FI(K,I,J-1) + FI(K,I,J))
      G1           = .5 * (GI(K,I+1,J) + GI(K,I,J))
      G2           = 0.0
      IF(K.EQ.3)G2 = PRES(1,NYY)
      IF(K.EQ.1)G3 = V1/VELO1
      IF(K.EQ.2)G3 = V1 * U1/VELO1**2
      IF(K.EQ.3)G3 = V1**2/VELO1**2 + P11
      IF(K.EQ.4)G3 = E11    * V1/VELO1 + V1*P11/VELO1
      IF(K.EQ.5)G3 = CONH2  * V1/VELO1
      IF(K.EQ.6)G3 = CONO2  * V1/VELO1
```

```
      IF(K.EQ.7)G3 = CONH2O * V1/VELO1
      IF(K.EQ.8)G3 = CONN2  * V1/VELO1
      G4           = .5 * (GI(K,I,J-1) + GI(K,I,J))
      FFLUXEN      = (F1 + FV(1,K,I,J))*Y1 + (F2 + FV(2,K,I,J))*Y2
      FFLUXWS      = (F3 + FV(3,K,I,J))*Y3 + (F4 + FV(4,K,I,J))*Y4
      GFLUXEN      = (G1 + GV(1,K,I,J))*X1 + (G2 + GV(2,K,I,J))*X2
      GFLUXWS      = (G3 + GV(3,K,I,J))*X3 + (G4 + GV(4,K,I,J))*X4
      RESID        = (FFLUXEN + FFLUXWS) - (GFLUXEN + GFLUXWS)
      RES(K,I,J)     = RESID / AREA(I,J)
12    CONTINUE
C
C ---   INFLOW BOUNDARY LOWER CORNER CELL
C
      I            = 1
      J            = 1
      X1           = X(I+1,J+1) - X(I+1,J)
      X2           = X(I,J+1)   - X(I+1,J+1)
      X3           = X(I,J)     - X(I,J+1)
      X4           = X(I+1,J)   - X(I,J)
      Y1           = Y(I+1,J+1) - Y(I+1,J)
      Y2           = Y(I,J+1)   - Y(I+1,J+1)
      Y3           = Y(I,J)     - Y(I,J+1)
      Y4           = Y(I+1,J)   - Y(I,J)
      DO   13 K    = 1 , IEQ
      F1           = .5 * (FI(K,I+1,J) + FI(K,I,J))
      F2           = .5 * (FI(K,I,J+1) + FI(K,I,J))
      IF(K.EQ.1)F3 = U1/VELO1
      IF(K.EQ.2)F3 = (U1/VELO1)**2 + P11
      IF(K.EQ.3)F3 = (U1 * V1)/VELO1**2
      IF(K.EQ.4)F3 = E11 * U1/VELO1 + U1*P11/VELO1
      IF(K.EQ.5)F3 = CONH2  * U1/VELO1
      IF(K.EQ.6)F3 = CONO2  * U1/VELO1
      IF(K.EQ.7)F3 = CONH2O * U1/VELO1
      IF(K.EQ.8)F3 = CONN2  * U1/VELO1
      F4           = 0.0
      IF(K.EQ.2)F4 = PRES(1,1)
      G1           = .5 * (GI(K,I+1,J) + GI(K,I,J))
      G2           = .5 * (GI(K,I,J+1) + GI(K,I,J))
      IF(K.EQ.1)G3 = V1/VELO1
      IF(K.EQ.2)G3 = V1 * U1/VELO1**2
      IF(K.EQ.3)G3 = V1**2/VELO1**2 + P11
      IF(K.EQ.4)G3 = E11    * V1/VELO1 + V1*P11/VELO1
      IF(K.EQ.5)G3 = CONH2  * V1/VELO1
      IF(K.EQ.6)G3 = CONO2  * V1/VELO1
      IF(K.EQ.7)G3 = CONH2O * V1/VELO1
      IF(K.EQ.8)G3 = CONN2  * V1/VELO1
      G4           = 0.0
      IF(K.EQ.3)G4 = PRES(1,1)
      FFLUXEN      = (F1 + FV(1,K,I,J))*Y1 + (F2 + FV(2,K,I,J))*Y2
      FFLUXWS      = (F3 + FV(3,K,I,J))*Y3 + (F4 + FV(4,K,I,J))*Y4
```

```
      GFLUXEN        = (G1 + GV(1,K,I,J))*X1 + (G2 + GV(2,K,I,J))*X2
      GFLUXWS        = (G3 + GV(3,K,I,J))*X3 + (G4 + GV(4,K,I,J))*X4
      RESID          = (FFLUXEN + FFLUXWS) - (GFLUXEN + GFLUXWS)
      RES(K,I,J)       = RESID / AREA(I,J)
13    CONTINUE
C
C       -----------------------------------
C ---   LOWER BOUNDARY WALL CELL EVALUATION
C       -----------------------------------
C
      J  = 1
      DO   3  I        = 2 , NXXX
      PW = PRES(I,1)
      X1             = X(I+1,J+1) - X(I+1,J)
      X2             = X(I,J+1)   - X(I+1,J+1)
      X3             = X(I,J)     - X(I,J+1)
      X4             = X(I+1,J)   - X(I,J)
      Y1             = Y(I+1,J+1) - Y(I+1,J)
      Y2             = Y(I,J+1)   - Y(I+1,J+1)
      Y3             = Y(I,J)     - Y(I,J+1)
      Y4             = Y(I+1,J)   - Y(I,J)
      DO   4   K     = 1 , IEQ
      F1             = .5 * (FI(K,I+1,J) + FI(K,I,J))
      F2             = .5 * (FI(K,I,J+1) + FI(K,I,J))
      F3             = .5 * (FI(K,I-1,J) + FI(K,I,J))
      F4             = 0.0
      IF(K.EQ.2)F4 = PW
      G1             = .5 * (GI(K,I+1,J) + GI(K,I,J))
      G2             = .5 * (GI(K,I,J+1) + GI(K,I,J))
      G3             = .5 * (GI(K,I-1,J) + GI(K,I,J))
      G4             = 0.0
      IF(K.EQ.3)G4 = PW
      FFLUXEN        = (F1 + FV(1,K,I,J))*Y1 + (F2 + FV(2,K,I,J))*Y2
      FFLUXWS        = (F3 + FV(3,K,I,J))*Y3 + (F4 + FV(4,K,I,J))*Y4
      GFLUXEN        = (G1 + GV(1,K,I,J))*X1 + (G2 + GV(2,K,I,J))*X2
      GFLUXWS        = (G3 + GV(3,K,I,J))*X3 + (G4 + GV(4,K,I,J))*X4
      RESID          = (FFLUXEN + FFLUXWS) - (GFLUXEN + GFLUXWS)
      RES(K,I,J)       = RESID / AREA(I,J)
4     CONTINUE
3     CONTINUE
C
C       -----------------------------------
C ---   UPPER BOUNDARY WALL CELL EVALUATION
C       -----------------------------------
C
      J              = NYY
      DO    5   I    = 2 , NXXX
      X1             = X(I+1,J+1) - X(I+1,J)
      X2             = X(I,J+1)   - X(I+1,J+1)
      X3             = X(I,J)     - X(I,J+1)
```

```
      X4             = X(I+1,J)   - X(I,J)
      Y1             = Y(I+1,J+1) - Y(I+1,J)
      Y2             = Y(I,J+1)   - Y(I+1,J+1)
      Y3             = Y(I,J)     - Y(I,J+1)
      Y4             = Y(I+1,J)   - Y(I,J)
      DO   6  K      = 1 , IEQ
      F1             = .5 * (FI(K,I+1,J) + FI(K,I,J))
      F2             = 0.0
      IF(K.EQ.2)F2 = PRES(I,NYY)
      F3             = .5 * (FI(K,I-1,J) + FI(K,I,J))
      F4             = .5 * (FI(K,I,J-1) + FI(K,I,J))
      G1             = .5 * (GI(K,I+1,J) + GI(K,I,J))
      G2             = 0.0
      IF(K.EQ.3)G2 = PRES(I,NYY)
      G3             = .5 * (GI(K,I-1,J) + GI(K,I,J))
      G4             = .5 * (GI(K,I,J-1) + GI(K,I,J))
      FFLUXEN        = (F1 + FV(1,K,I,J))*Y1 + (F2 + FV(2,K,I,J))*Y2
      FFLUXWS        = (F3 + FV(3,K,I,J))*Y3 + (F4 + FV(4,K,I,J))*Y4
      GFLUXEN        = (G1 + GV(1,K,I,J))*X1 + (G2 + GV(2,K,I,J))*X2
      GFLUXWS        = (G3 + GV(3,K,I,J))*X3 + (G4 + GV(4,K,I,J))*X4
      RESID          = (FFLUXEN + FFLUXWS) - (GFLUXEN + GFLUXWS)
      RES(K,I,J)      = RESID / AREA(I,J)
6     CONTINUE
5     CONTINUE
C
C       ------------------------
C ---   EXIT BOUNDARY EVALUATION
C       ------------------------
C
C
C ---   UPPER CORNER CELL
C
C
      I              = NXX
      J              = NYY
      X1             = X(I+1,J+1) - X(I+1,J)
      X2             = X(I,J+1)   - X(I+1,J+1)
      X3             = X(I,J)     - X(I,J+1)
      X4             = X(I+1,J)   - X(I,J)
      Y1             = Y(I+1,J+1) - Y(I+1,J)
      Y2             = Y(I,J+1)   - Y(I+1,J+1)
      Y3             = Y(I,J)     - Y(I,J+1)
      Y4             = Y(I+1,J)   - Y(I,J)
      DO   20  K     = 1 , IEQ
      F1             = FI(K,I,J)
      F2             = 0.0
      IF(K.EQ.2)F2 = PRES(I,J)
      F3             = .5 * (FI(K,I-1,J) + FI(K,I,J))
      F4             = .5 * (FI(K,I,J-1) + FI(K,I,J))
      G1             = GI(K,I,J)
```

```
      G2              = 0.0
      IF(K.EQ.3)G2 = PRES(I,J)
      G3              = .5 * (GI(K,I-1,J) + GI(K,I,J))
      G4              = .5 * (GI(K,I,J-1) + GI(K,I,J))
      FFLUXEN         = (F1 + FV(1,K,I,J))*Y1 + (F2 + FV(2,K,I,J))*Y2
      FFLUXWS         = (F3 + FV(3,K,I,J))*Y3 + (F4 + FV(4,K,I,J))*Y4
      GFLUXEN         = (G1 + GV(1,K,I,J))*X1 + (G2 + GV(2,K,I,J))*X2
      GFLUXWS         = (G3 + GV(3,K,I,J))*X3 + (G4 + GV(4,K,I,J))*X4
      RESID           = (FFLUXEN + FFLUXWS) - (GFLUXEN + GFLUXWS)
      RES(K,I,J)      = RESID / AREA(I,J)
20    CONTINUE
C
C ---  LOWER EXIT CORNER ( X = NXX , Y = 1 )
C
      I               = NXX
      J               = 1
      X1              = X(I+1,J+1) - X(I+1,J)
      X2              = X(I,J+1)   - X(I+1,J+1)
      X3              = X(I,J)     - X(I,J+1)
      X4              = X(I+1,J)   - X(I,J)
      Y1              = Y(I+1,J+1) - Y(I+1,J)
      Y2              = Y(I,J+1)   - Y(I+1,J+1)
      Y3              = Y(I,J)     - Y(I,J+1)
      Y4              = Y(I+1,J)   - Y(I,J)
      DO   21   K     = 1 , IEQ
      F1              = FI(K,I,J)
      F2              = .5 * (FI(K,I,J+1) + FI(K,I,J))
      F3              = .5 * (FI(K,I-1,J) + FI(K,I,J))
      F4              = 0.0
      IF(K.EQ.2)F4 = PRES(NXX,J)
      G1              = GI(K,I,J)
      G2              = .5 * (GI(K,I,J+1) + GI(K,I,J))
      G3              = .5 * (GI(K,I-1,J) + GI(K,I,J))
      G4              = 0.0
      IF(K.EQ.3)G4 = PRES(NXX,J)
      FFLUXEN         = (F1 + FV(1,K,I,J))*Y1 + (F2 + FV(2,K,I,J))*Y2
      FFLUXWS         = (F3 + FV(3,K,I,J))*Y3 + (F4 + FV(4,K,I,J))*Y4
      GFLUXEN         = (G1 + GV(1,K,I,J))*X1 + (G2 + GV(2,K,I,J))*X2
      GFLUXWS         = (G3 + GV(3,K,I,J))*X3 + (G4 + GV(4,K,I,J))*X4
      RESID           = (FFLUXEN + FFLUXWS) - (GFLUXEN + GFLUXWS)
      RES(K,I,J)      = RESID / AREA(I,J)
21    CONTINUE
C
C ---    VERTICAL EXIT CELL EVALUATION ( X = NXX , Y = 2 , NYYY )
C
      I               = NXX
      DO   22   J     = 2 , NYYY
      X1              = X(I+1,J+1) - X(I+1,J)
      X2              = X(I,J+1)   - X(I+1,J+1)
      X3              = X(I,J)     - X(I,J+1)
```

```
      X4            = X(I+1,J)    - X(I,J)
      Y1            = Y(I+1,J+1) - Y(I+1,J)
      Y2            = Y(I,J+1)    - Y(I+1,J+1)
      Y3            = Y(I,J)      - Y(I,J+1)
      Y4            = Y(I+1,J)    - Y(I,J)
      DO   22  K    = 1 , IEQ
      F1            =  FI(K,I,J)
      F2            = .5 * (FI(K,I,J+1) + FI(K,I,J))
      F3            = .5 * (FI(K,I-1,J) + FI(K,I,J))
      F4            = .5 * (FI(K,I,J-1) + FI(K,I,J))
      G1            =  GI(K,I,J)
      G2            = .5 * (GI(K,I,J+1) + GI(K,I,J))
      G3            = .5 * (GI(K,I-1,J) + GI(K,I,J))
      G4            = .5 * (GI(K,I,J-1) + GI(K,I,J))
      FFLUXEN       = (F1 + FV(1,K,I,J))*Y1 + (F2 + FV(2,K,I,J))*Y2
      FFLUXWS       = (F3 + FV(3,K,I,J))*Y3 + (F4 + FV(4,K,I,J))*Y4
      GFLUXEN       = (G1 + GV(1,K,I,J))*X1 + (G2 + GV(2,K,I,J))*X2
      GFLUXWS       = (G3 + GV(3,K,I,J))*X3 + (G4 + GV(4,K,I,J))*X4
      RESID         = (FFLUXEN + FFLUXWS) - (GFLUXEN + GFLUXWS)
      RES(K,I,J)    = RESID / AREA(I,J)
22    CONTINUE
      RETURN
      END
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
      SUBROUTINE DAMPX
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
C
C ---   X COMPONENT OF POST SPLIT SMOOTHING OPERATOR
C
      COMMON/VAR0/U(8,60,60,2),FI(8,60,60),GI(8,60,60),AH(8,60,60)
      COMMON/VAR00/FV(4,8,60,60),GV(4,8,60,60)
      COMMON/VAR1/UVEL(60,60),VVEL(60,60),PRES(60,60),TEMP(60,60)
      COMMON/VAR111/YH2(60,60),YO2(60,60),YH2O(60,60),YOH(60,60)
      COMMON/VAR2/DEN(60,60),SOUND(60,60),AINTE(60,60),AMACH(60,60)
      COMMON/VAR3/ENTHP(60,60),VIS(60,60),YYN2(60,60),CPND(60,60)
      COMMON/VAR4/DXX,DYY,X(62,62),Y(62,62),AREA(60,60)
      COMMON/VAR5/DT(60,60)
      COMMON/VAR6/NX,NXX,NXXX,NY,NYY,NYYY,IRES,IEQ
      COMMON/VAR7/IVIS,NITER,NOITER
      COMMON/VAR8/P1,T1,AM1,VISL,U1,V1,AK1,CV,CP,R,GAMA,DEN1,E1
      COMMON/VAR9/P11,C11,E11
      COMMON/VAR10/DFH2,DFO2,DFH2O,DFOH,DFN2
      COMMON/VAR11/CPH2,CPO2,CPH2O,CPOH,CPN2,CVH2,CVO2,CVH2O,CVOH,CVN2
      COMMON/VAR12/CONH2,CONO2,CONH2O,CONOH,CONN2
      COMMON/VAR13/COND,CFL,DCOFF,AL,VELO1
      COMMON/VAR14/RES(8,60,60)
      COMMON/VAR15/ALPHA(4)
      COMMON/VAR16/DXE(4,60,60),DXN(4,60,60),DXW(4,60,60),DXS(4,60,60)
      COMMON/VAR17/DYE(4,60,60),DYN(4,60,60),DYW(4,60,60),DYS(4,60,60)
      COMMON/VAR18/DD(2,8,60,60)
```

```
      COMMON/VAR20/NSX,NSXB,NSXA,NSY,NSYB,NSYA
      COMMON/VAR21/ACOM
      COMMON/VAR22/TRIGTEMP,PHI
      COMMON/VAR23/REN,PR,FACT1,LAMB,SDIFF
      DO 1  J    = 1 , NYY
      DO 1  I    = 2 , NXXX
      DD(1,1,I,J)  = U(1,I+1,J,2) + U(1,I-1,J,2) -2.0*U(1,I,J,2)
      DD(1,2,I,J)  = U(2,I+1,J,2) + U(2,I-1,J,2) -2.0*U(2,I,J,2)
      DD(1,3,I,J)  = U(3,I+1,J,2) + U(3,I-1,J,2) -2.0*U(3,I,J,2)
      DD(1,4,I,J)  = U(4,I+1,J,2) + U(4,I-1,J,2) -2.0*U(4,I,J,2)
      DD(1,5,I,J)  = U(5,I+1,J,2) + U(5,I-1,J,2) -2.0*U(5,I,J,2)
      DD(1,6,I,J)  = U(6,I+1,J,2) + U(6,I-1,J,2) -2.0*U(6,I,J,2)
      DD(1,7,I,J)  = U(7,I+1,J,2) + U(7,I-1,J,2) -2.0*U(7,I,J,2)
      DD(1,8,I,J)  = U(8,I+1,J,2) + U(8,I-1,J,2) -2.0*U(8,I,J,2)
1     CONTINUE
      DO 3 J     = 1 , NYY
C
C ---   GHOST POINT EVALUATION OF U'S - BASED UPON FREESTREAM
C ---                                  BOUNDARY CONDITIONS
C
C
C ---   X = 1
C
      DD(1,1,1,J)     = U(1,2,J,2)  - U(1,1,J,2)
      DD(1,2,1,J)     = U(2,2,J,2)  - U(2,1,J,2)
      DD(1,3,1,J)     = U(3,2,J,2)  - U(3,1,J,2)
      DD(1,4,1,J)     = U(4,2,J,2)  - U(4,1,J,2)
      DD(1,5,1,J)     = U(5,2,J,2)  - U(5,1,J,2)
      DD(1,6,1,J)     = U(6,2,J,2)  - U(6,1,J,2)
      DD(1,7,1,J)     = U(7,2,J,2)  - U(7,1,J,2)
      DD(1,8,1,J)     = U(8,2,J,2)  - U(8,1,J,2)
C
C ---   X = NXX
C
      DD(1,1,NXX,J)  = U(1,NXXX,J,2)  - U(1,NXX,J,2)
      DD(1,2,NXX,J)  = U(2,NXXX,J,2)  - U(2,NXX,J,2)
      DD(1,3,NXX,J)  = U(3,NXXX,J,2)  - U(3,NXX,J,2)
      DD(1,4,NXX,J)  = U(4,NXXX,J,2)  - U(4,NXX,J,2)
      DD(1,5,NXX,J)  = U(5,NXXX,J,2)  - U(5,NXX,J,2)
      DD(1,6,NXX,J)  = U(6,NXXX,J,2)  - U(6,NXX,J,2)
      DD(1,7,NXX,J)  = U(7,NXXX,J,2)  - U(7,NXX,J,2)
      DD(1,8,NXX,J)  = U(8,NXXX,J,2)  - U(8,NXX,J,2)
3     CONTINUE
      RETURN
      END
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      SUBROUTINE DAMPY
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C --- Y COMPONENT OF POST SPLIT SMOOTHING OPERATOR
```

```
C
      COMMON/VAR0/U(8,60,60,2),FI(8,60,60),GI(8,60,60),AH(8,60,60)
      COMMON/VAR00/FV(4,8,60,60),GV(4,8,60,60)
      COMMON/VAR1/UVEL(60,60),VVEL(60,60),PRES(60,60),TEMP(60,60)
      COMMON/VAR111/YH2(60,60),YO2(60,60),YH2O(60,60),YOH(60,60)
      COMMON/VAR2/DEN(60,60),SOUND(60,60),AINTE(60,60),AMACH(60,60)
      COMMON/VAR3/ENTHP(60,60),VIS(60,60),YYN2(60,60),CPND(60,60)
      COMMON/VAR4/DXX,DYY,X(62,62),Y(62,62),AREA(60,60)
      COMMON/VAR5/DT(60,60)
      COMMON/VAR6/NX,NXX,NXXX,NY,NYY,NYYY,IRES,IEQ
      COMMON/VAR7/IVIS,NITER,NOITER
      COMMON/VAR8/P1,T1,AM1,VISL,U1,V1,AK1,CV,CP,R,GAMA,DEN1,E1
      COMMON/VAR9/P11,C11,E11
      COMMON/VAR10/DFH2,DFO2,DFH2O,DFOH,DFN2
      COMMON/VAR11/CPH2,CPO2,CPH2O,CPOH,CPN2,CVH2,CVO2,CVH2O,CVOH,CVN2
      COMMON/VAR12/CONH2,CONO2,CONH2O,CONOH,CONN2
      COMMON/VAR13/COND,CFL,DCOFF,AL,VELO1
      COMMON/VAR14/RES(8,60,60)
      COMMON/VAR15/ALPHA(4)
      COMMON/VAR16/DXE(4,60,60),DXN(4,60,60),DXW(4,60,60),DXS(4,60,60)
      COMMON/VAR17/DYE(4,60,60),DYN(4,60,60),DYW(4,60,60),DYS(4,60,60)
      COMMON/VAR18/DD(2,8,60,60)
      COMMON/VAR20/NSX,NSXB,NSXA,NSY,NSYB,NSYA
      COMMON/VAR21/ACOM
      COMMON/VAR22/TRIGTEMP,PHI
      COMMON/VAR23/REN,PR,FACT1,LAMB,SDIFF
      DO 1 J    = 2 , NYYY
      DO 1 I    = 1 , NXX
      DD(2,1,I,J)  = U(1,I,J+1,2) + U(1,I,J-1,2) -2.0*U(1,I,J,2)
      DD(2,2,I,J)  = U(2,I,J+1,2) + U(2,I,J-1,2) -2.0*U(2,I,J,2)
      DD(2,3,I,J)  = U(3,I,J+1,2) + U(3,I,J-1,2) -2.0*U(3,I,J,2)
      DD(2,4,I,J)  = U(4,I,J+1,2) + U(4,I,J-1,2) -2.0*U(4,I,J,2)
      DD(2,5,I,J)  = U(5,I,J+1,2) + U(5,I,J-1,2) -2.0*U(5,I,J,2)
      DD(2,6,I,J)  = U(6,I,J+1,2) + U(6,I,J-1,2) -2.0*U(6,I,J,2)
      DD(2,7,I,J)  = U(7,I,J+1,2) + U(7,I,J-1,2) -2.0*U(7,I,J,2)
      DD(2,8,I,J)  = U(8,I,J+1,2) + U(8,I,J-1,2) -2.0*U(8,I,J,2)
1     CONTINUE
      DO 3 I     = 1 , NXX
C
C ---   GHOST POINT EVALUATION OF U'S - BASED UPON REFLECTION
C ---                                   BOUNDARY CONDITIONS
C
C
C ---   Y = 1
C
      U1GHOST    =   DEN(I,1)
C
CCCCCCCCCCCCCCCCC  NOTE + SIGN HERE SHOULD BE - VE FOR VISCOUS CAL
C
      U2GHOST    =   DEN(I,1)*UVEL(I,1)
```

```
      U3GHOST     = + DEN(I,1)*VVEL(I,1)
      U4GHOST     =   DEN(I,1)*AINTE(I,1)
      U5GHOST     =   DEN(I,1)*YH2(I,1)
      U6GHOST     =   DEN(I,1)*YO2(I,1)
      U7GHOST     =   DEN(I,1)*YH2O(I,1)
      U8GHOST     =   DEN(I,1)*YYN2(I,1)
      DD(2,1,I,1)  = U(1,I,2,2) + U1GHOST - 2.0*U(1,I,1,2)
      DD(2,2,I,1)  = U(2,I,2,2) + U2GHOST - 2.0*U(2,I,1,2)
      DD(2,3,I,1)  = U(3,I,2,2) + U3GHOST - 2.0*U(3,I,1,2)
      DD(2,4,I,1)  = U(4,I,2,2) + U4GHOST - 2.0*U(4,I,1,2)
      DD(2,5,I,1)  = U(5,I,2,2) + U5GHOST - 2.0*U(5,I,1,2)
      DD(2,6,I,1)  = U(6,I,2,2) + U6GHOST - 2.0*U(6,I,1,2)
      DD(2,7,I,1)  = U(7,I,2,2) + U7GHOST - 2.0*U(7,I,1,2)
      DD(2,8,I,1)  = U(8,I,2,2) + U8GHOST - 2.0*U(8,I,1,2)
C
C ---   Y = NYY
C
      U1GHOST     =   DEN(I,NYY)
      U2GHOST     =   DEN(I,NYY)*UVEL(I,NYY)
      U3GHOST     = + DEN(I,NYY)*VVEL(I,NYY)
      U4GHOST     =   DEN(I,NYY)*AINTE(I,NYY)
      U5GHOST     =   DEN(I,NYY)*YH2(I,NYY)
      U6GHOST     =   DEN(I,NYY)*YO2(I,NYY)
      U7GHOST     =   DEN(I,NYY)*YH2O(I,NYY)
      U8GHOST     =   DEN(I,NYY)*YYN2(I,NYY)
      DD(2,1,I,NYY)  = U(1,I,NYYY,2) + U1GHOST - 2.0*U(1,I,NYY,2)
      DD(2,2,I,NYY)  = U(2,I,NYYY,2) + U2GHOST - 2.0*U(2,I,NYY,2)
      DD(2,3,I,NYY)  = U(3,I,NYYY,2) + U3GHOST - 2.0*U(3,I,NYY,2)
      DD(2,4,I,NYY)  = U(4,I,NYYY,2) + U4GHOST - 2.0*U(4,I,NYY,2)
      DD(2,5,I,NYY)  = U(5,I,NYYY,2) + U5GHOST - 2.0*U(5,I,NYY,2)
      DD(2,6,I,NYY)  = U(6,I,NYYY,2) + U6GHOST - 2.0*U(6,I,NYY,2)
      DD(2,7,I,NYY)  = U(7,I,NYYY,2) + U7GHOST - 2.0*U(7,I,NYY,2)
      DD(2,8,I,NYY)  = U(8,I,NYYY,2) + U8GHOST - 2.0*U(8,I,NYY,2)
3     CONTINUE
C
C ---    ADD SECOND ORDER DAMPING CORRECTIONS( X AND Y) TO  U
C
      DO 2 J      = 1 , NYY
      DO 2 I      = 1 , NXX
      U(1,I,J,2) = U(1,I,J,2)   + DCOFF * (DD(1,1,I,J) + DD(2,1,I,J))
      U(2,I,J,2) = U(2,I,J,2)   + DCOFF * (DD(1,2,I,J) + DD(2,2,I,J))
      U(3,I,J,2) = U(3,I,J,2)   + DCOFF * (DD(1,3,I,J) + DD(2,3,I,J))
      U(4,I,J,2) = U(4,I,J,2)   + DCOFF * (DD(1,4,I,J) + DD(2,4,I,J))
      U(5,I,J,2) = U(5,I,J,2)   + DCOFF * (DD(1,5,I,J) + DD(2,5,I,J))
      U(6,I,J,2) = U(6,I,J,2)   + DCOFF * (DD(1,6,I,J) + DD(2,6,I,J))
      U(7,I,J,2) = U(7,I,J,2)   + DCOFF * (DD(1,7,I,J) + DD(2,7,I,J))
      U(8,I,J,2) = U(8,I,J,2)   + DCOFF * (DD(1,8,I,J) + DD(2,8,I,J))
C
C ---     COMPUTE NEW PRIMATIVE QUANTITIES
C
```

```
      DEN(I,J)    = U(1,I,J,2)
      ODEN        = 1.0 / DEN(I,J)
      UVEL(I,J)   = U(2,I,J,2) * ODEN
      VVEL(I,J)   = U(3,I,J,2) * ODEN
      AINTE(I,J)  = U(4,I,J,2) * ODEN
      YH2(I,J)    = U(5,I,J,2) * ODEN
      YO2(I,J)    = U(6,I,J,2) * ODEN
      YH2O(I,J)   = U(7,I,J,2) * ODEN
      YYN2(I,J)   = U(8,I,J,2) * ODEN
C
      UOH         = 1.0 - YH2(I,J) - YO2(I,J) - YH2O(I,J) - YYN2(I,J)
      CP          = ( YH2(I,J)*CPH2 + YO2(I,J)* CPO2 +YH2O(I,J)*CPH2O
     1              + UOH*CPOH      + YYN2(I,J)* CPN2)
      CV          = ( YH2(I,J)*CVH2 + YO2(I,J)* CVO2 +YH2O(I,J)*CVH2O
     1              + UOH*CVOH      + YYN2(I,J)* CVN2)
      R           = CP - CV
      GAMA        = CP / CV
      DHEATF      = YH2(I,J)*DFH2 + YO2(I,J)*DFO2 + YH2O(I,J)*DFH2O
     1              + UOH*DFOH    + YYN2(I,J)   *DFN2
      VELO        = UVEL(I,J)**2 + VVEL(I,J)**2
      TEMP(I,J)   = VELO1**2/(CV*T1)*(AINTE(I,J) - .5*VELO
     1              - DHEATF / VELO1**2 )
      TEMP(I,J)   = ABS(TEMP(I,J))
      SOUND(I,J)  = SQRT(T1*R*GAMA*TEMP(I,J))/VELO1
      AMACH(I,J)  = SQRT(VELO)/SOUND(I,J)
      PRES(I,J)   = (T1*R/VELO1**2)*DEN(I,J)*TEMP(I,J)
      ENTHP(I,J)  = CP*T1/VELO1**2*TEMP(I,J)+.5*VELO
     1              + DHEATF / VELO1**2
2     CONTINUE
      RETURN
      END
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      SUBROUTINE STAB
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C ---  DETERMINE THE BIGGEST TIME STEPS THE SOLUTION CAN BE ADVANCED
C ---  AND STABILITY BE MAINTAINED
C
      COMMON/VAR0/U(8,60,60,2),FI(8,60,60),GI(8,60,60),AH(8,60,60)
      COMMON/VAR00/FV(4,8,60,60),GV(4,8,60,60)
      COMMON/VAR1/UVEL(60,60),VVEL(60,60),PRES(60,60),TEMP(60,60)
      COMMON/VAR111/YH2(60,60),YO2(60,60),YH2O(60,60),YOH(60,60)
      COMMON/VAR2/DEN(60,60),SOUND(60,60),AINTE(60,60),AMACH(60,60)
      COMMON/VAR3/ENTHP(60,60),VIS(60,60),YYN2(60,60),CPND(60,60)
      COMMON/VAR4/DXX,DYY,X(62,62),Y(62,62),AREA(60,60)
      COMMON/VAR5/DT(60,60)
      COMMON/VAR6/NX,NXX,NXXX,NY,NYY,NYYY,IRES,IEQ
      COMMON/VAR7/IVIS,NITER,NOITER
      COMMON/VAR8/P1,T1,AM1,VISL,U1,V1,AK1,CV,CP,R,GAMA,DEN1,E1
      COMMON/VAR9/P11,C11,E11
```

```
      COMMON/VAR10/DFH2,DFO2,DFH2O,DFOH,DFN2
      COMMON/VAR11/CPH2,CPO2,CPH2O,CPOH,CPN2,CVH2,CVO2,CVH2O,CVOH,CVN2
      COMMON/VAR12/CONH2,CONO2,CONH2O,CONOH,CONN2
      COMMON/VAR13/COND,CFL,DCOFF,AL,VELO1
      COMMON/VAR14/RES(8,60,60)
      COMMON/VAR15/ALPHA(4)
      COMMON/VAR16/DXE(4,60,60),DXN(4,60,60),DXW(4,60,60),DXS(4,60,60)
      COMMON/VAR17/DYE(4,60,60),DYN(4,60,60),DYW(4,60,60),DYS(4,60,60)
      COMMON/VAR18/DD(2,8,60,60)
      COMMON/VAR20/NSX,NSXB,NSXA,NSY,NSYB,NSYA
      COMMON/VAR21/ACOM
      COMMON/VAR22/TRIGTEMP,PHI
      COMMON/VAR23/REN,PR,FACT1,LAMB,SDIFF
      CVIS    = .75
      DO 1 IY = 1 , NYY
      DO 1 IX = 1 , NXX
C
      DXXX = ABS(X(IX+1,IY)-X(IX,IY))
      DYYY = ABS(Y(IX,IY+1)-Y(IX,IY))
      DMX       = ABS(UVEL(IX,IY)) + SOUND(IX,IY)
      DMY       = ABS(VVEL(IX,IY)) + SOUND(IX,IY)
C
      A1        = (DMX/DXXX + DMY/DYYY) / CFL
      DT(IX,IY) = 1.0 / A1
1     CONTINUE
      RETURN
      END
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      SUBROUTINE FROPINV
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C ---  COMPUTE THE INVISCID CONTRIBUTIONS TO THE
C ---  " F "  AND " G " FLUX VECTORS
C
      COMMON/VAR0/U(8,60,60,2),FI(8,60,60),GI(8,60,60),AH(8,60,60)
      COMMON/VAR00/FV(4,8,60,60),GV(4,8,60,60)
      COMMON/VAR1/UVEL(60,60),VVEL(60,60),PRES(60,60),TEMP(60,60)
      COMMON/VAR111/YH2(60,60),YO2(60,60),YH2O(60,60),YOH(60,60)
      COMMON/VAR2/DEN(60,60),SOUND(60,60),AINTE(60,60),AMACH(60,60)
      COMMON/VAR3/ENTHP(60,60),VIS(60,60),YYN2(60,60),CPND(60,60)
      COMMON/VAR4/DXX,DYY,X(62,62),Y(62,62),AREA(60,60)
      COMMON/VAR5/DT(60,60)
      COMMON/VAR6/NX,NXX,NXXX,NY,NYY,NYYY,IRES,IEQ
      COMMON/VAR7/IVIS,NITER,NOITER
      COMMON/VAR8/P1,T1,AM1,VISL,U1,V1,AK1,CV,CP,R,GAMA,DEN1,E1
      COMMON/VAR9/P11,C11,E11
      COMMON/VAR10/DFH2,DFO2,DFH2O,DFOH,DFN2
      COMMON/VAR11/CPH2,CPO2,CPH2O,CPOH,CPN2,CVH2,CVO2,CVH2O,CVOH,CVN2
      COMMON/VAR12/CONH2,CONO2,CONH2O,CONOH,CONN2
      COMMON/VAR13/COND,CFL,DCOFF,AL,VELO1
```

```
      COMMON/VAR14/RES(8,60,60)
      COMMON/VAR15/ALPHA(4)
      COMMON/VAR16/DXE(4,60,60),DXN(4,60,60),DXW(4,60,60),DXS(4,60,60)
      COMMON/VAR17/DYE(4,60,60),DYN(4,60,60),DYW(4,60,60),DYS(4,60,60)
      COMMON/VAR18/DD(2,8,60,60)
      COMMON/VAR20/NSX,NSXB,NSXA,NSY,NSYB,NSYA
      COMMON/VAR21/ACOM
      COMMON/VAR22/TRIGTEMP,PHI
      COMMON/VAR23/REN,PR,FACT1,LAMB,SDIFF
      DO 1  JJ  = 1 , NYY
      DO 1  II  = 1 , NXX
C
C ---   EVALUATE THE INVISCID PARTS OF "F" AND "G"
C
C ---   EVALUATE THE F TERMS
C
      FI(1,II,JJ) = DEN(II,JJ)*UVEL(II,JJ)
      FI(2,II,JJ) = DEN(II,JJ)*UVEL(II,JJ)**2 + PRES(II,JJ)
      FI(3,II,JJ) = DEN(II,JJ)*UVEL(II,JJ)*VVEL(II,JJ)
      FI(4,II,JJ) = DEN(II,JJ)*UVEL(II,JJ)*AINTE(II,JJ)
     1              + UVEL(II,JJ)*PRES(II,JJ)
      FI(5,II,JJ) = DEN(II,JJ)*UVEL(II,JJ)*YH2(II,JJ)
      FI(6,II,JJ) = DEN(II,JJ)*UVEL(II,JJ)*YO2(II,JJ)
      FI(7,II,JJ) = DEN(II,JJ)*UVEL(II,JJ)*YH2O(II,JJ)
      FI(8,II,JJ) = DEN(II,JJ)*UVEL(II,JJ)*YYN2(II,JJ)
C
C       EVALUATE THE G TERMS
C
      GI(1,II,JJ) = DEN(II,JJ)*VVEL(II,JJ)
      GI(2,II,JJ) = DEN(II,JJ)*UVEL(II,JJ)*VVEL(II,JJ)
      GI(3,II,JJ) = DEN(II,JJ)*VVEL(II,JJ)**2 + PRES(II,JJ)
      GI(4,II,JJ) = DEN(II,JJ)*VVEL(II,JJ)*AINTE(II,JJ)
     1              + VVEL(II,JJ)*PRES(II,JJ)
      GI(5,II,JJ) = DEN(II,JJ)*VVEL(II,JJ)*YH2(II,JJ)
      GI(6,II,JJ) = DEN(II,JJ)*VVEL(II,JJ)*YO2(II,JJ)
      GI(7,II,JJ) = DEN(II,JJ)*VVEL(II,JJ)*YH2O(II,JJ)
      GI(8,II,JJ) = DEN(II,JJ)*VVEL(II,JJ)*YYN2(II,JJ)
1     CONTINUE
      RETURN
      END
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      SUBROUTINE CONV
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C --- DETERMINE THE CONVERGENCE HISTORY OF THE SOLUTION METHOD
C
      COMMON/VAR0/U(8,60,60,2),FI(8,60,60),GI(8,60,60),AH(8,60,60)
      COMMON/VAR00/FV(4,8,60,60),GV(4,8,60,60)
      COMMON/VAR1/UVEL(60,60),VVEL(60,60),PRES(60,60),TEMP(60,60)
      COMMON/VAR111/YH2(60,60),YO2(60,60),YH2O(60,60),YOH(60,60)
```

```
      COMMON/VAR2/DEN(60,60),SOUND(60,60),AINTE(60,60),AMACH(60,60)
      COMMON/VAR3/ENTHP(60,60),VIS(60,60),YYN2(60,60),CPND(60,60)
      COMMON/VAR4/DXX,DYY,X(62,62),Y(62,62),AREA(60,60)
      COMMON/VAR5/DT(60,60)
      COMMON/VAR6/NX,NXX,NXXX,NY,NYY,NYYY,IRES,IEQ
      COMMON/VAR7/IVIS,NITER,NOITER
      COMMON/VAR8/P1,T1,AM1,VISL,U1,V1,AK1,CV,CP,R,GAMA,DEN1,E1
      COMMON/VAR9/P11,C11,E11
      COMMON/VAR10/DFH2,DFO2,DFH2O,DFOH,DFN2
      COMMON/VAR11/CPH2,CPO2,CPH2O,CPOH,CPN2,CVH2,CVO2,CVH2O,CVOH,CVN2
      COMMON/VAR12/CONH2,CONO2,CONH2O,CONOH,CONN2
      COMMON/VAR13/COND,CFL,DCOFF,AL,VELO1
      COMMON/VAR14/RES(8,60,60)
      COMMON/VAR15/ALPHA(4)
      COMMON/VAR16/DXE(4,60,60),DXN(4,60,60),DXW(4,60,60),DXS(4,60,60)
      COMMON/VAR17/DYE(4,60,60),DYN(4,60,60),DYW(4,60,60),DYS(4,60,60)
      COMMON/VAR18/DD(2,8,60,60)
      COMMON/VAR20/NSX,NSXB,NSXA,NSY,NSYB,NSYA
      COMMON/VAR21/ACOM
      COMMON/VAR22/TRIGTEMP,PHI
      COMMON/VAR23/REN,PR,FACT1,LAMB,SDIFF
      NXX       = NX - 1
      NYY       = NY - 1
      IRES      = 0
      TRES1     = 0.0
      TRES2     = 0.0
      TRES3     = 0.0
      TRES4     = 0.0
      TRES5     = 0.0
      TRES6     = 0.0
      TRES7     = 0.0
      NPT       = NXX * NYY  +  NSX * NSY
      DO 1 I    = 1 , NXX
      DO 1 J    = 1 , NYY
      RES1      = (U(1,I,J,2) - U(1,I,J,1)) / DT(I,J)
      RES2      = (U(2,I,J,2) - U(2,I,J,1)) / DT(I,J)
      RES3      = (U(3,I,J,2) - U(3,I,J,1)) / DT(I,J)
      RES4      = (U(4,I,J,2) - U(4,I,J,1)) / DT(I,J)
      RES5      = (U(5,I,J,2) - U(5,I,J,1)) / DT(I,J)
      RES6      = (U(6,I,J,2) - U(6,I,J,1)) / DT(I,J)
      RES7      = (U(7,I,J,2) - U(7,I,J,1)) / DT(I,J)
      TRES1     = TRES1 + ABS(RES1)
      TRES2     = TRES2 + ABS(RES2)
      TRES3     = TRES3 + ABS(RES3)
      TRES4     = TRES4 + ABS(RES4)
      TRES5     = TRES5 + ABS(RES5)
      TRES6     = TRES6 + ABS(RES6)
      TRES7     = TRES7 + ABS(RES7)
C
1     CONTINUE
```

```
C
      TRES1    = TRES1 / NPT
      TRES2    = TRES2 / NPT
      TRES3    = TRES3 / NPT
      TRES4    = TRES4 / NPT
      TRES5    = TRES5 / NPT
      TRES6    = TRES6 / NPT
      TRES7    = TRES7 / NPT
      TRES8    = TRES8 / NPT
C
C ---  PRINT OUT THE RESIDUAL HISTORIES FOR EACH ITERATION
C
      WRITE(6,10)TRES1,TRES2,TRES3,TRES4,TRES5,TRES6,TRES7,
     .           TRES8
10    FORMAT(2X,8(E10.4))
C
C?????????????????????????????????????????????????????????
C
C ---   CONVERGENCE TEST
C
C?????????????????????????????????????????????????????????
      IRES   = 0
      IF(TRES1.GT.RESCONV) IRES = 1
C?????????????????????????????????????????????????????????
      RETURN
      END
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      SUBROUTINE OUT
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C ---  LOAD OUTPUT DATA FILES
C
      COMMON/VAR0/U(8,60,60,2),FI(8,60,60),GI(8,60,60),AH(8,60,60)
      COMMON/VAR00/FV(4,8,60,60),GV(4,8,60,60)
      COMMON/VAR1/UVEL(60,60),VVEL(60,60),PRES(60,60),TEMP(60,60)
      COMMON/VAR111/YH2(60,60),YO2(60,60),YH2O(60,60),YOH(60,60)
      COMMON/VAR2/DEN(60,60),SOUND(60,60),AINTE(60,60),AMACH(60,60)
      COMMON/VAR3/ENTHP(60,60),VIS(60,60),YYN2(60,60),CPND(60,60)
      COMMON/VAR4/DXX,DYY,X(62,62),Y(62,62),AREA(60,60)
      COMMON/VAR5/DT(60,60)
      COMMON/VAR6/NX,NXX,NXXX,NY,NYY,NYYY,IRES,IEQ
      COMMON/VAR7/IVIS,NITER,NOITER
      COMMON/VAR8/P1,T1,AM1,VISL,U1,V1,AK1,CV,CP,R,GAMA,DEN1,E1
      COMMON/VAR9/P11,C11,E11
      COMMON/VAR10/DFH2,DFO2,DFH2O,DFOH,DFN2
      COMMON/VAR11/CPH2,CPO2,CPH2O,CPOH,CPN2,CVH2,CVO2,CVH2O,CVOH,CVN2
      COMMON/VAR12/CONH2,CONO2,CONH2O,CONOH,CONN2
      COMMON/VAR13/COND,CFL,DCOFF,AL,VELO1
      COMMON/VAR14/RES(8,60,60)
      COMMON/VAR15/ALPHA(4)
```

```
      COMMON/VAR16/DXE(4,60,60),DXN(4,60,60),DXW(4,60,60),DXS(4,60,60)
      COMMON/VAR17/DYE(4,60,60),DYN(4,60,60),DYW(4,60,60),DYS(4,60,60)
      COMMON/VAR18/DD(2,8,60,60)
      COMMON/VAR20/NSX,NSXB,NSXA,NSY,NSYB,NSYA
      COMMON/VAR21/ACOM
      COMMON/VAR22/TRIGTEMP,PHI
      COMMON/VAR23/REN,PR,FACT1,LAMB,SDIFF
C
C ---  NUMBER OF ITERATIONS
C
     WRITE(6,8)NOITER
8    FORMAT(2X,I4)
C
C     GRID METRIX AND REACTION RATES
C
       DO 6 J = 1 , NY
       DO 6 I = 1 , NX
6      WRITE(6,7)X(I,J),Y(I,J),AH(5,I,J),AH(6,I,J),AH(7,I,J)
7      FORMAT(2X,5(E10.4))
C
C ---   LOAD RESTART FILE " DSTEP.DAT "
C
      DO 5 J = 1 , NYY
      DO 5 I = 1 , NXX
      WRITE(6,14)U(1,I,J,2),U(2,I,J,2),U(3,I,J,2),U(4,I,J,2),
     1           U(5,I,J,2),U(6,I,J,2),U(7,I,J,2),U(8,I,J,2)
5     CONTINUE
14    FORMAT(2X,8(E10.4))
C
      DO 3000 I = 1 , NXX
      DO 3000 J = 1 , NYY
      RESN1 = ABS(U(1,I,J,2) - U(1,I,J,1)) / DT(I,J)
      RESN5 = ABS(U(5,I,J,2) - U(5,I,J,1)) / DT(I,J)
      WRITE(6,3001)RESN1,RESN5
3000  CONTINUE
3001  FORMAT(2X,2(E10.4))
      RETURN
      END
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      SUBROUTINE SOURCE
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C ---          COMPUTE THE SOURCE TERMS
C ---    THE H2 - O2 CHEMISTRY SOURCE TERMS ARE AS
C ---    DESCRIBED BY ROGERS/CHINITZ (AIAA-82-0112)
C
      COMMON/VAR0/U(8,60,60,2),FI(8,60,60),GI(8,60,60),AH(8,60,60)
      COMMON/VAR00/FV(4,8,60,60),GV(4,8,60,60)
      COMMON/VAR1/UVEL(60,60),VVEL(60,60),PRES(60,60),TEMP(60,60)
      COMMON/VAR111/YH2(60,60),YO2(60,60),YH2O(60,60),YOH(60,60)
```

```
      COMMON/VAR2/DEN(60,60),SOUND(60,60),AINTE(60,60),AMACH(60,60)
      COMMON/VAR3/ENTHP(60,60),VIS(60,60),YYN2(60,60),CPND(60,60)
      COMMON/VAR4/DXX,DYY,X(62,62),Y(62,62),AREA(60,60)
      COMMON/VAR5/DT(60,60)
      COMMON/VAR6/NX,NXX,NXXX,NY,NYY,NYYY,IRES,IEQ
      COMMON/VAR7/IVIS,NITER,NOITER
      COMMON/VAR8/P1,T1,AM1,VISL,U1,V1,AK1,CV,CP,R,GAMA,DEN1,E1
      COMMON/VAR9/P11,C11,E11
      COMMON/VAR10/DFH2,DFO2,DFH2O,DFOH,DFN2
      COMMON/VAR11/CPH2,CPO2,CPH2O,CPOH,CPN2,CVH2,CVO2,CVH2O,CVOH,CVN2
      COMMON/VAR12/CONH2,CONO2,CONH2O,CONOH,CONN2
      COMMON/VAR13/COND,CFL,DCOFF,AL,VELO1
      COMMON/VAR14/RES(8,60,60)
      COMMON/VAR15/ALPHA(4)
      COMMON/VAR16/DXE(4,60,60),DXN(4,60,60),DXW(4,60,60),DXS(4,60,60)
      COMMON/VAR17/DYE(4,60,60),DYN(4,60,60),DYW(4,60,60),DYS(4,60,60)
      COMMON/VAR18/DD(2,8,60,60)
      COMMON/VAR20/NSX,NSXB,NSXA,NSY,NSYB,NSYA
      COMMON/VAR21/ACOM
      COMMON/VAR22/TRIGTEMP,PHI
      COMMON/VAR23/REN,PR,FACT1,LAMB,SDIFF
C
C ---    GAS PROPERITY INPUT
C
      AWH2    =  2.0E-3
      AWO2    =  32.0E-3
      AWH2O   =  18.0E-3
      AWOH    =  17.0E-3
      AWN2    =  28.0E-3
      RUCGS   = 1.987
      WSCALE  = AL / (VELO1 * DEN1)
C
C ---    SWEEP THROUGH ALL GRID POINTS TO COMPUTE "WDOT"
C
      FACT4   = ACOM * (1.0E+21/(T1**5)) * (1.0E+20/(T1**5))
      FACT5   = ACOM * (1.0E+20/(T1**5)) * (1.0E+20/(T1**5))
     1                * (1.0E+12/(T1**3))
      S5      = 1.0
C
C
C ---    INITIALIZE CHEMICAL SOURCE TERM ARRAYS
C
      DO 100 K  = 1 , IEQ
      DO 100 J  = 1 , NYY
      DO 100 I  = 1 , NXX
      AH(K,I,J) = 0.0
100   CONTINUE
C
C ---    TEST IF CASE IS REACTING OR NONREACTING- DO ONLY IF
C ---    REACTING CASE
```

```
C
       IF(ACOM.EQ.0.0) GO TO 500
C
       DO 1 J  = 1 , NYY
       DO 1 I  = 1 , NXX
       ATEMP = TEMP(I,J) * T1
       IF(ATEMP.LE.TRIGTEMP) GO TO 1
C
C ---    EXPONENTIAL FACTOR
C
       EQUIL4   = 26.164 * EXP( -8992. / (TEMP(I,J) * T1))
       EQUIL5   = 3.269E-8 * EXP( 69415. / (TEMP(I,J) * T1))
       CONST4   = - 4865. /(1.987 * T1 * TEMP(I,J))
       CONST5   = - 42500./(1.987 * T1 * TEMP(I,J))
       APHI4    = (8.917 * PHI + 31.433 / PHI - 28.950     )
       APHI5    = (2.000       + 1.333  / PHI - .8333 * PHI)
C
C ---   RATE CONSTANTS
C
C       WRITE(5,*)TEMP(I,J),T1
       AF4      = APHI4 / (TEMP(I,J)**5) *  FACT4
     1             *  EXP(CONST4) / (TEMP(I,J)**5)
       AF5      = APHI5 / (TEMP(I,J)**6) * FACT5
     1             *  EXP(CONST5) / (TEMP(I,J)**7)
       AB4      = AF4 /  EQUIL4
       AB5      = AF5 / (EQUIL5 * RUCGS * T1 * TEMP(I,J))
C       WRITE(5,*)'AF4,AF5,AB4,AB5'
C       WRITE(5,*)I,J, AF4,AF5,AB4,AB5
C
C ---   SPECIES CONCENTRATIONS
C
       UDEN     = U(1,I,J,2)
       UH2      = U(5,I,J,2)
       UO2      = U(6,I,J,2)
       UH2O     = U(7,I,J,2)
       UN2      = U(8,I,J,2)
       UOH      = UDEN - UH2 - UO2 - UH2O - UN2
       UOH      = ABS(UOH)
       UH2      = UH2 * DEN1             / AWH2
       UO2      = UO2 * DEN1             / AWO2
       UH2O     = UH2O* DEN1             / AWH2O
       UOH      = UOH * DEN1             / AWOH
       UN2      = UN2 * DEN1             / AWN2
C
C ---   PRODUCTION RATES - 'H'  TERM
C
       S5          = 1.0
       AH(5,I,J)  = AWH2  * (- AF4 * UH2 * UO2    +  AB4 * UOH**2
     1                      - AF5 * UOH**2 *UH2 * S5
     2                      + AB5 * UH2O**2 * S5 )
```

```
      S5          = 1.0
      AH(6,I,J)   = AWO2   * (- AF4 * UH2 * UO2    + AB4 * UOH**2  )
      S5          = 1.0
      AH(7,I,J)   = AWH2O * 2.0 * (  AF5 * UOH**2 * UH2 * S5
     1                               - AB5 * UH2O**2 * S5           )
C       WRITE(5,*)I ,J,AH(5,I,J),AH(6,I,J),AH(7,I,J)
C
C ---    ADD THE SOURCE TERMS TO THE RESIDUALS COMPUTED IN THE
C ---    FLUX BALANCE ROUTINE " SUBROUTINE FLUX"
C ---    NOTE 'WSCALE' IS A PARAMETER USED TO NON-DIMENSIONALIZE WDOT
C
      RES(5,I,J)  = RES(5,I,J) - AH(5,I,J) * WSCALE
      RES(6,I,J)  = RES(6,I,J) - AH(6,I,J) * WSCALE
      RES(7,I,J)  = RES(7,I,J) - AH(7,I,J) * WSCALE
1     CONTINUE
500   CONTINUE
      RETURN
      END
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      SUBROUTINE NSSOLVE(IA)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C=================================================================C
C                                                                 C
C  M A T R I X   E Q U A T I O N   T O   B E   S O L V E D        C
C                                                                 C
C                                                                 C
C      [                     ] [         ]      [         ]       C
C      [    A    B     C     ] [   X1    ]      [   R1    ]       C
C      [                     ] [         ]      [         ]       C
C      [    D    E     F     ] [   X2    ]  =   [   R2    ]       C
C      [                     ] [         ]      [         ]       C
C      [    G    H     I     ] [   X3    ]      [   R3    ]       C
C      [                     ] [         ]      [         ]       C
C                                                                 C
C=================================================================C
C
C ---
C ---   SYSTEM OF LINEAR EQUATIONS SOLVED BY
C ---   GAUSSIAN ELIMINATION - GLOBAL CHEMISTRY MODEL
C ---
C
C ---   COMPUTE THE  TIME SCALING DERIVATIVES OF THE
C ---   \ S \ MATRIX
C
      COMMON/VAR0/U(8,60,60,2),FI(8,60,60),GI(8,60,60),AH(8,60,60)
      COMMON/VAR00/FV(4,8,60,60),GV(4,8,60,60)
      COMMON/VAR1/UVEL(60,60),VVEL(60,60),PRES(60,60),TEMP(60,60)
      COMMON/VAR111/YH2(60,60),YO2(60,60),YH2O(60,60),YOH(60,60)
      COMMON/VAR2/DEN(60,60),SOUND(60,60),AINTE(60,60),AMACH(60,60)
      COMMON/VAR3/ENTHP(60,60),VIS(60,60),YYN2(60,60),CPND(60,60)
```

```
      COMMON/VAR4/DXX,DYY,X(62,62),Y(62,62),AREA(60,60)
      COMMON/VAR5/DT(60,60)
      COMMON/VAR6/NX,NXX,NXXX,NY,NYY,NYYY,IRES,IEQ
      COMMON/VAR7/IVIS,NITER,NOITER
      COMMON/VAR8/P1,T1,AM1,VISL,U1,V1,AK1,CV,CP,R,GAMA,DEN1,E1
      COMMON/VAR9/P11,C11,E11
      COMMON/VAR10/DFH2,DFO2,DFH2O,DFOH,DFN2
      COMMON/VAR11/CPH2,CPO2,CPH2O,CPOH,CPN2,CVH2,CVO2,CVH2O,CVOH,CVN2
      COMMON/VAR12/CONH2,CONO2,CONH2O,CONOH,CONN2
      COMMON/VAR13/COND,CFL,DCOFF,AL,VELO1
      COMMON/VAR14/RES(8,60,60)
      COMMON/VAR15/ALPHA(4)
      COMMON/VAR16/DXE(4,60,60),DXN(4,60,60),DXW(4,60,60),DXS(4,60,60)
      COMMON/VAR17/DYE(4,60,60),DYN(4,60,60),DYW(4,60,60),DYS(4,60,60)
      COMMON/VAR18/DD(2,8,60,60)
      COMMON/VAR20/NSX,NSXB,NSXA,NSY,NSYB,NSYA
      COMMON/VAR21/ACOM
      COMMON/VAR22/TRIGTEMP,PHI
      COMMON/VAR23/REN,PR,FACT1,LAMB,SDIFF
C
C ---    GAS PROPERITY INPUT
C
      AWH2    =  2.0E-3
      AWO2    =  32.0E-3
      AWH2O   =  18.0E-3
      AWOH    =  17.0E-3
      AWN2    =  28.0E-3
      RUCGS   = 1.987
      EQUIL4  = 5.76E-1
      EQUIL5  = 3.47E+3
      WSCALE  = AL / (VELO1 * DEN1)
C
C ---    SWEEP THROUGH ALL GRID POINTS TO COMPUTE "WDOT"
C
      FACT4   = ACOM * (1.0E+21/(T1**5)) * (1.0E+20/(T1**5))
      FACT5   = ACOM * (1.0E+20/(T1**5)) * (1.0E+20/(T1**5))
     1                * (1.0E+12/(T1**3))
      S5      = 1.0
C
C ---    DETERMINE OPTIMUM TIME STEP FOR STABILITY
C
      CALL STAB
C
C --- FREE STREAM SPECIFIC HEAT
C
      CPFS   = CPN2 * CONN2 + CPO2 * CONO2 + CPH2 * CONH2
C
      DO 1 J = 1 , NYY
      DO 1 I = 1 , NXX
C
```

```
      DDT     =  DT(I,J)
C
C ---    IGNITION TEMPERATURE TEST
C
      ATEMP = TEMP(I,J) * T1
      IF(ATEMP.LE.TRIGTEMP.OR.ACOM.EQ.0.0) GO TO 500
C
C ---    EXPONENTIAL FACTOR
C
      EQUIL4  = 26.164 * EXP( -8992. / (TEMP(I,J) * T1))
      EQUIL5  = 3.289E-8 * EXP( 69415. / (TEMP(I,J) * T1))
      CONST4  = - 4865. /(1.987 * T1 * TEMP(I,J))
      CONST5  = - 42500./(1.987 * T1 * TEMP(I,J))
      APHI4   = (8.917 * PHI + 31.433 / PHI - 28.950     )
      APHI5   = (2.000       + 1.333  / PHI - .8333 * PHI)
C
C ---   RATE CONSTANTS
C
      AF4     = APHI4 / (TEMP(I,J)**5) *  FACT4
     1          *  EXP(CONST4) / (TEMP(I,J)**5)
      AF5     = APHI5 / (TEMP(I,J)**6) * FACT5
     1          * EXP(CONST5) / (TEMP(I,J)**7)
      AB4     = AF4 /  EQUIL4
      AB5     = AF5 / (EQUIL5 * RUCGS * T1 * TEMP(I,J))
C
C ---   DK / DT
C
C      WRITE(5,*)I,J,FACT4,FACT5,CONST4,CONST5
      PAF4    = APHI4 / (TEMP(I,J)**5) *  FACT4
     1          * EXP(CONST4) / (TEMP(I,J)**6)
     2          * ( -10. - CONST4 )
      PAF5    = APHI5 / (TEMP(I,J)**6) * FACT5
     1          * EXP(CONST5) / (TEMP(I,J)**8)
     2          * ( -13. - CONST5 )
C
C ---   SPECIES CONCENTRATIONS
C
      UDEN    = U(1,I,J,2)
      UH2     = U(5,I,J,2)
      UO2     = U(6,I,J,2)
      UH2O    = U(7,I,J,2)
      UN2     = U(8,I,J,2)
      UOH     = UDEN - UH2 - UO2 - UH2O - UN2
      UOH     = ABS(UOH)
      UH2     = UH2 * DEN1             / AWH2
      UO2     = UO2 * DEN1             / AWO2
      UH2O    = UH2O* DEN1             / AWH2O
      UOH     = UOH * DEN1             / AWOH
      UN2     = UN2 * DEN1             / AWN2
      CV          = (  YH2(I,J)  * CVH2  + YO2(I,J)  * CVO2
```

```
     1              + YH2O(I,J) * CVH2O + YOH(I,J)  * CVOH
     2              + YYN2(I,J)  * CVN2 )
C
C ---   COMPUTE \ S \ MATRIX ELEMENTS     DH / DU
C ---   IE THE CHARACTERISTIC CHEMICAL TIMES
C
C
C
C       -------------------------
C ---   CHARACTERISTIC TIME SCALES
C       -------------------------
C
      S1       =  1.0 / AWH2
      S2       =  1.0 / AWO2
      S3       =  1.0 / AWH2O
      S4       =  1.0 / AWOH
C
C        --------------
C ---    H2 TIME SCALES
C        --------------
C
      DH2UH2   =  AWH2 *     (- AF4*UO2*S1 - 2.0*AB4*UOH*S4
     1                        + 2.0*AF5*UOH*UH2*S4
     2                        - AF5*UOH**2*S1                   )
      DH2UO2   =  AWH2 *     (- AF4*UH2*S2 - 2.0*AB4*UOH*S4
     1                        + 2.0*AF5*UOH*UH2*S4              )
      DH2UW    =  AWH2 * 2.0*(- AB4*UOH*S4 + AF5*UOH*UH2*S4
     1                        + AB5*UH2O*S3                     )
      DH2UD    =  AWH2 * 2.0*(+ AB4*UOH*S4 - AF5*UOH*UH2*S4     )
      DH2UN2   =  - DH2UD
      DH2UE1   =  (-UH2*UO2    + UOH**2/EQUIL4 ) * PAF4
      EQUILS   =  EQUIL5*RUCGS*T1*TEMP(I,J)
      DH2UE2   =  (-UOH**2*UH2 + UH2O**2/EQUILS) * PAF5
      DH2UE    =  AWH2*VELO1**2/(T1*CV*DEN(I,J))*(DH2UE1+DH2UE2)
C
C        --------------
C ---    O2 TIME SCALES
C        --------------
C
      DO2UH2   =  AWO2  *     (- AF4*UO2*S1 - 2.0*AB4*UOH*S4    )
      DO2UO2   =  AWO2  *     (- AF4*UH2*S2 - 2.0*AB4*UOH*S4    )
      DO2UW    =  AWO2  *     (- 2.0*AB4*UOH*S4                 )
      DO2UD    =  AWO2  *     (  2.0*AB4*UOH*S4                 )
      DO2UN2   = - DO2UD
      DO2UE1   = (-UH2*UO2 + UOH**2/EQUIL4) * PAF4
      DO2UE    = AWO2*VELO1**2/(T1*CV*DEN(I,J))*DO2UE1
C
C        ---------------
C ---    H2O TIME SCALES
C        ---------------
```

```
C
      DH2OUH2   =  AWH2O *2.0*(- 2.0*AF5*UOH*UH2*S4
     1                                     + AF5*UOH**2*S1    )
      DH2OUO2   =  AWH2O  *2.0*(- 2.0*AF5*UOH*UH2*S4           )
      DH2OUW    =  AWH2O*2.0*(- 2.0*AF5*UOH*UH2*S4
     1                                     - 2.0*AB5*UH2O*S3 )
      DH2OUD    =  AWH2O *2.0*(  2.0*AF5*UOH*UH2*S4           )
      DH2OUN2   = - DH2OUD
      EQUILS    =  EQUIL5*RUCGS*T1*TEMP(I,J)
      DH2OUE1   =  2.0*(UOH**2*UH2 - UH2O**2/EQUILS) * PAF5
      DH2OUE    =  AWH2O*VELO1**2/(T1*CV*DEN(I,J))*DH2OUE1
C
C
C ---   DEFINE THE COEFFICIENTS OF THE COEFFICIENT MATRIX \\ S \\
C
      A    = 1.0 - DH2UH2    * DDT * ALPHA(IA) * WSCALE
      B    =     - DH2UO2    * DDT * ALPHA(IA) * WSCALE
      C    =     - DH2UW     * DDT * ALPHA(IA) * WSCALE
      D    =     - DO2UH2    * DDT * ALPHA(IA) * WSCALE
      E    = 1.0 - DO2UO2    * DDT * ALPHA(IA) * WSCALE
      F    =     - DO2UW     * DDT * ALPHA(IA) * WSCALE
      AAG  =     - DH2OUH2   * DDT * ALPHA(IA) * WSCALE
      AAH  =     - DH2OUO2   * DDT * ALPHA(IA) * WSCALE
      AAI  = 1.0 - DH2OUW    * DDT * ALPHA(IA) * WSCALE
C
C ---    NEXT DEFINE THE RESIDUAL VECTOR
C
      DU1   = - ALPHA(IA) * DDT * RES(1,I,J)
      DU1   = DU1 * WSCALE
      DU4   = - ALPHA(IA) * DDT * RES(4,I,J)
      DU4   = DU4 * WSCALE
      DU8   = - ALPHA(IA) * DDT * RES(8,I,J)
      DU8   = DU8 * WSCALE
C
C ---    DU1 * DH/DU  ACCOUNTS THE DEPENDANCE OF WDOT ON
C ---                 DENSITY
C
      R1    = - ALPHA(IA) * DDT * (RES(5,I,J) - DH2UD   * DU1
     1          - DH2UE    * DU4 - DH2UN2 * DU8)
      R2    = - ALPHA(IA) * DDT * (RES(6,I,J) - DO2UD   * DU1
     1          - DO2UE    * DU4 - DO2UN2 * DU8)
      R3    = - ALPHA(IA) * DDT * (RES(7,I,J) - DH2OUD * DU1
     1          - DH2OUE   * DU4 - DH2OUN2 * DU8)
C
C ---    NORMALIZE THE MATRIX ELEMENTS SUCH THAT NONE IS GREATER
C ---    THAN ONE
C
C ---    FIND NORMALIZING VALUES
C
       SC1    = ABS(A)
```

```
          IF(ABS(B).GT.SC1) SC1 = ABS(B)
          IF(ABS(C).GT.SC1) SC1 = ABS(C)
          SC2   = ABS(D)
          IF(ABS(E).GT.SC2) SC2 = ABS(E)
          IF(ABS(F).GT.SC2) SC2 = ABS(F)
          SC3   = ABS(AAG)
          IF(ABS(AAH).GT.SC3) SC3 = ABS(AAH)
          IF(ABS(AAI).GT.SC3) SC3 = ABS(AAI)
C
C ---       NORMALIZE
C
          A   = A / SC1
          B   = B / SC1
          C   = C / SC1
          D   = D / SC2
          E   = E / SC2
          F   = F / SC2
          AAG = AAG / SC3
          AAH = AAH / SC3
          AAI = AAI / SC3
C
          R1  = R1  / SC1
          R2  = R2  / SC2
          R3  = R3  / SC3
C
C          --------------------------------
C ---      SOLVE MATRIX SYSTEM OF EQUATIONS
C          --------------------------------
C
C ---      REDUCE TO DIAGONAL FORM
C
        EP     = 1.0E-15
        TEST   = ABS(D)
        IF(TEST.LT.EP) GO TO 100
        DIV    = A / D
        E      = B  - DIV * E
        F      = C  - DIV * F
        R2     = R1 - DIV * R2
100     CONTINUE
        TEST   = ABS(AAG)
        IF(TEST.LT.EP) GO TO 101
        DIV    = A / AAG
        AAH    = B  - DIV * AAH
        AAI    = C  - DIV * AAI
        R3     = R1 - DIV * R3
101     CONTINUE
        TEST   = ABS(AAH)
        IF(TEST.LT.EP) GO TO 102
        DIV    = E / AAH
        AAI    = F  - DIV * AAI
```

```
      R3     = R2 - DIV * R3
102   CONTINUE
C
C ---  NOW COMPUTE THE X'S VIA BACK SUBSTITUTION
C
      X3     = R3 / AAI
      X2     = (R2 - F * X3) / E
      X1     = (R1 - B * X2 - C * X3) / A
500   CONTINUE
C
C      ----------------------------
C ---  COMPUTE NEW FLUID    " U 'S "
C      ----------------------------
C
      U(1,I,J,2) = U(1,I,J,1) - ALPHA(IA) * DDT * RES(1,I,J)
      U(2,I,J,2) = U(2,I,J,1) - ALPHA(IA) * DDT * RES(2,I,J)
      U(3,I,J,2) = U(3,I,J,1) - ALPHA(IA) * DDT * RES(3,I,J)
      U(4,I,J,2) = U(4,I,J,1) - ALPHA(IA) * DDT * PES(4,I,J)
      U(8,I,J,2) = U(8,I,J,1) - ALPHA(IA) * DDT * RES(8,I,J)
C
C      -----------------------------------------------------
C ---  COMPUTE THE NEW SPECIES STATE QUANTITIES IE "  U'S "
C      -----------------------------------------------------
C
      IF(ATEMP.LE.TRIGTEMP.OR.ACOM.EQ.0.0) GO TO 501
      U(5,I,J,2) = U(5,I,J,1) + X1
      U(6,I,J,2) = U(6,I,J,1) + X2
      U(7,I,J,2) = U(7,I,J,1) + X3
      GO TO 502
501   CONTINUE
      U(5,I,J,2) = U(5,I,J,1) - ALPHA(IA) * DDT * RES(5,I,J)
      U(6,I,J,2) = U(6,I,J,1) - ALPHA(IA) * DDT * RES(6,I,J)
      U(7,I,J,2) = U(7,I,J,1) - ALPHA(IA) * DDT * RES(7,I,J)
502   CONTINUE
C
C       ----------------------
C ---   UPDATE FLUID PROPERTIES
C       ----------------------
C
      DEN(I,J)    = U(1,I,J,2)
      ODEN        = 1.0 / DEN(I,J)
      UVEL(I,J)   = U(2,I,J,2) * ODEN
      VVEL(I,J)   = U(3,I,J,2) * ODEN
      AINTE(I,J)  = U(4,I,J,2) * ODEN
C
C      ----------------------------
C ---  UPDATE SPECIES MASS FRACTIONS
C      ----------------------------
C
      YH2(I,J)    = U(5,I,J,2) * ODEN
```

```
      YO2(I,J)    = U(6,I,J,2) * ODEN
      YH2O(I,J)   = U(7,I,J,2) * ODEN
      YYN2(I,J)   = U(8,I,J,2) * ODEN
C
C       -------------------------
C ---   COMPUTE REMAINING UNKNOWNS
C       -------------------------
C
      UOH         = 1.0 - YH2(I,J) - YO2(I,J) - YH2O(I,J) - YYN2(I,J)
      YOH(I,J)    = UOH
C      WRITE(5,*)I,J,YH2(I,J),YO2(I,J),YH2O(I,J),YOH(I,J),CONN2,
C     1   TEMP(I,J)
      CP          = (  YH2(I,J)  * CPH2  + YO2(I,J)  * CPO2
     1                + YH2O(I,J) * CPH2O + UOH       * CPOH
     2                + YYN2(I,J)  * CPN2 )
C
C --- NON-DIMENSIONAL CP
C
      CPND(I,J)   = CP / CPFS
C
      CV          = (  YH2(I,J)  * CVH2  + YO2(I,J)  * CVO2
     1                + YH2O(I,J) * CVH2O + UOH       * CVOH
     2                + YYN2(I,J)  * CVN2 )
C      WRITE(5,*)I,J,CP,CV,YH2O(I,J),YO2(I,J),YH2O(I,J),YOH(I,J)
      R           = CP - CV
      GAMA        = CP / CV
      DHEATF      = YH2(I,J)*DFH2 + YO2(I,J)* DFO2 + YH2O(I,J)*DFH2O
     1              + UOH*DFOH    + YYN2(I,J)* DFN2
      VELO        = UVEL(I,J)**2 + VVEL(I,J)**2
      TEMP(I,J)   = (VELO1**2/(CV*T1))*(AINTE(I,J) - .5*VELO
     1                - DHEATF/(VELO1**2))
      TEMP(I,J)   = APS(TEMP(I,J))
      SOUND(I,J)  = SQRT(GAMA * R * T1 * TEMP(I,J))/VELO1
      AMACH(I,J)  = SQRT(VELO)/SOUND(I,J)
      PRES(I,J)   = (T1 /VELO1**2) * R * DEN(I,J) * TEMP(I,J)
      ENTHP(I,J)  = CP * T1 / VELO1**2 *TEMP(I,J) + .5 * VELO
     1              + DHEATF / VELO1**2
1     CONTINUE
      RETURN
      END
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      SUBROUTINE PARAM
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                      C
C --- COMPUTE AND PRINT OUT THE GROSS FEATURES OF THE PROBLEM          C
C --- FOR EXAMPLE CALCULATE THE PERCENT OF H2 AND O2CONSUMED OR THE    C
C --- RATIO OF THE HEAT RELEASED TO THE TOTAL HEAT AVAILABLE           C
C                                                                      C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      COMMON/VARO/U(8,60,60,2),FI(8,60,60),GI(8,60,60),AH(8,60,60)
```

```
      COMMON/VAR00/FV(4,8,60,60),GV(4,8,60,60)
      COMMON/VAR1/UVEL(60,60),VVEL(60,60),PRES(60,60),TEMP(60,60)
      COMMON/VAR111/YH2(60,60),YO2(60,60),YH2O(60,60),YOH(60,60)
      COMMON/VAR2/DEN(60,60),SOUND(60,60),AINTE(60,60),AMACH(60,60)
      COMMON/VAR3/ENTHP(60,60),VIS(60,60),YYN2(60,60),CPND(60,60)
      COMMON/VAR4/DXX,DYY,X(62,62),Y(62,62),AREA(60,60)
      COMMON/VAR5/DT(60,60)
      COMMON/VAR6/NX,NXX,NXXX,NY,NYY,NYYY,IRES,IEQ
      COMMON/VAR7/IVIS,NITER,NOITER
      COMMON/VAR8/P1,T1,AM1,VISL,U1,V1,AK1,CV,CP,R,GAMA,DEN1,E1
      COMMON/VAR9/P11,C11,E11
      COMMON/VAR10/DFH2,DFO2,DFH2O,DFOH,DFN2
      COMMON/VAR11/CPH2,CPO2,CPH2O,CPOH,CPN2,CVH2,CVO2,CVH2O,CVOH,CVN2
      COMMON/VAR12/CONH2,CONO2,CONH2O,CONOH,CONN2
      COMMON/VAR13/COND,CFL,DCOFF,AL,VELO1
      COMMON/VAR14/RES(8,60,60)
      COMMON/VAR15/ALPHA(4)
      COMMON/VAR16/DXE(4,60,60),DXN(4,60,60),DXW(4,60,60),DXS(4,60,60)
      COMMON/VAR17/DYE(4,60,60),DYN(4,60,60),DYW(4,60,60),DYS(4,60,60)
      COMMON/VAR18/DD(2,8,60,60)
      COMMON/VAR20/NSX,NSXB,NSXA,NSY,NSYB,NSYA
      COMMON/VAR21/ACOM
      COMMON/VAR22/TRIGTEMP,PHI
      COMMON/VAR23/REN,PR,FACT1,LAMB,SDIFF
C*******************************************************************
C
C ---    COMPUTE THE PERCENT OF H2 CONSUMED
C
C*******************************************************************
C ---     AMASS1  =  MASS OF H2 IN
C ---     AMASS2  =  MASS OF H2 OUT
C
      AMASS1     =  Y(NSY,NYY) * DEN1 * CONH2 * 1.0
C
      AMASS2     = 0.0
      DO 1 J     = 1 , NYY
      DELTAY     = Y(NXX,J+1) - Y(NXX,J)
      AVEL       = SQRT(UVEL(NXX,J)**2 + VVEL(NXX,J)**2)
      DM         = DELTAY * DEN1 * DEN(NXX,J) * YH2(NXX,J) * AVEL
      AMASS2     = AMASS2 + DM
1     CONTINUE
C
C ---   PERCENT OF H2 CONSUMED   PH2
C
      PH2        =  100. * ( 1.0 - AMASS2  / AMASS1 )
C
C*****************************************************************
C
C ---    COMPUTE THE PERCENT OF O2 CONSUMED
C
```

```
C*************************************************************
C ---      AMASS1  =  MASS OF O2 IN
C ---      AMASS2  =  MASS OF O? OUT
C
      AMASS1      =  Y(NSY,NYY) * DEN1 * CONO2 * 1.0
C
      AMASS2      = 0.0
      DO 2 J      = 1 , NYY
      DELTAY      = Y(NXX,J+1) - Y(NXX,J)
      AVEL        = SQRT(UVEL(NXX,J)**2 + VVEL(NXX,J)**2)
      DM          = DELTAY * DEN1 * DEN(NXX,J) * YO2(NXX,J) * AVEL
      AMASS2      = AMASS2 + DM
2     CONTINUE
C
C ---   PERCENT OF O2 COMSUMED  PO2
C
      PO2         = 100. * ( 1.0 - AMASS2  / AMASS1 )
C
C
C*************************************************************
C
C ---    COMPUTE THE HEAT RELEASE PARAMETER   PH = DHF / HTO
C
C*************************************************************
C ---      HTO     =  ENTERING TOTAL ENTHALPY
C ---      DHF     =  HEAT RELEASED DUE TO THE FORMATION OF H2O
C
      HTO         = (CPN2 * CONN2 + CPH2 * CONH2 + CPO2 * CONO2) * T1
     1                +  .5
      DO 3 J      = 1 , NYY
      DM          = YH2O(NXX,J) * DFH2O
      DHF         = DHF  + DM
3     CONTINUE
      DHF         = DHF / (Y(NXX,NYY) - Y(NXX,1))
      PH          = DHF / HTO
C
C ---      PRINT OUTPUT
C
      WRITE(6,11) PH2 , PO2 , PH
11    FORMAT(2X,3(E10.4))
      RETURN
      END
```

Appendix 3 2-D Viscous Code

```
      PROGRAM NSFRC(UNIT6=NSFRCD)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                      C
C                                                                      C
C                  ------------------------                            C
C                  \                      \                            C
C                  \      N S F R C       \                            C
C                  \                      \                            C
C                  ------------------------                            C
C                                                                      C
C                                                                      C
C                                                                      C
C            (  NAVIER - STOKES FINITE RATE CHEMISTRY   )              C
C                                                                      C
C                                                                      C
C                                                                      C
C            2-D VISCOUS NON-EQUILIBRIUM CHEMISTRY SOLVER              C
C            --------------------------------------------              C
C                                                                      C
C            THE PROGRAM IS PRESENTLY SET UP TO SOLVE                  C
C            FOR SUPERSONIC FLOW IN CHANNELS AND OVER                  C
C            REARWARD FACING STEPS. THE PROGRAM CAN ALSO               C
C            HANDLE FLOWS WITH EMBEDDED SUBSONIC ZONES.                C
C            THE SCHEME IS BASED ON AN EXTENED FINITE VOLUME           C
C            MULTISTEP EULER EQUATION METHOD PROPOSED BY               C
C            JAMESON( JUNE 1983).                                      C
C                                                                      C
C            THE EQUATIONS SOLVED ARE THE 2 - D NAVIER                 C
C            STOKES COUPLED WITH FINITE RATE CHEMISTRY.                C
C            THE ROGERS AND CHINITZ H2-AIR CHEMISTRY                   C
C            MODEL IS MODEL THE COMBUSTION PROCESS                     C
C                                                                      C
C            THE SOURCE TERMS ARE INTEGRATED IMPLICITLY                C
C            WHILE THE SPATIAL DERIVATIES ARE INTEGRATED               C
C            EXPLICITLY (POINT IMPLICIT METHOD).                       C
C                                                                      C
C                                                                      C
C            THE CODE IS CURRENTLY SET UP TO TREAT THE                 C
C            WALLS ADABATICALLY. K, THE THERMAL DIFFUSION              C
C            COEFFICIENT IS ASSUMED CONSTANT AND CP AND CV             C
```

```
C            ARE CURRENTLY DEPENDENT ONLY ON THE SPECIES                  C
C            DENSITY FRACTIONS.                                           C
C                                                                         C
C                                                                         C
C                     WRITTEN AT THE                                      C
C            MASSACHUSETTS INSTITUTE OF TECHNOLOGY                        C
C                      DATE STARTED                                       C
C                       MARCH 1984                                        C
C                           BY                                            C
C                                                                         C
C                    THOMAS   BUSSING                                     C
C                                                                         C
C            THE DETAILS OF THE EQUATIONS AND NUMERICAL                   C
C            METHOD USED ARE GIVEN IN THOMAS BUSSING                      C
C            MIT PH-D THESIS (AUGUST 1985)                                C
C                                                                         C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      COMMON/VAR0/U(8,53,33,2),FI(8,53,33),GI(8,53,33),AH(8,53,33)
      COMMON/VAR00/FV(4,8,53,33),GV(4,8,53,33)
      COMMON/VAR1/UVEL(53,33),VVEL(53,33),PRES(53,33),TEMP(53,33)
      COMMON/VAR111/YH2(53,33),YO2(53,33),YH2O(53,33),YOH(53,33)
      COMMON/VAR2/DEN(53,33),SOUND(53,33),AINTE(53,33),AMACH(53,33)
      COMMON/VAR3/ENTHP(53,33),VIS(53,33),YYN2(53,33),CPND(53,33)
      COMMON/VAR4/DXX,DYY,X(55,35),Y(55,35),AREA(53,33)
      COMMON/VAR5/DT(53,33)
      COMMON/VAR6/NX,NXX,NXXX,NY,NYY,NYYY,IRES,IEQ
      COMMON/VAR7/IVIS,NITER,NOITER
      COMMON/VAR8/P1,T1,AM1,VISL,U1,V1,AK1,CV,CP,R,GAMA,DEN1,E1
      COMMON/VAR9/P11,C11,E11
      COMMON/VAR10/DFH2,DFO2,DFH2O,DFOH,DFN2
      COMMON/VAR11/CPH2,CPO2,CPH2O,CPOH,CPN2,CVH2,CVO2,CVH2O,CVOH,CVN2
      COMMON/VAR12/CONH2,CONO2,CONH2O,CONOH,CONN2
      COMMON/VAR13/COND,CFL,DCOFF,AL,VELO1
      COMMON/VAR14/RES(8,53,33)
      COMMON/VAR15/ALPHA(4)
      COMMON/VAR16/DXE(4,53,33),DXN(4,53,33),DXW(4,53,33),DXS(4,53,33)
      COMMON/VAR17/DYE(4,53,33),DYN(4,53,33),DYW(4,53,33),DYS(4,53,33)
      COMMON/VAR18/DD(2,8,53,33)
      COMMON/VAR20/NSX,NSXB,NSXA,NSY,NSYB,NSYA
      COMMON/VAR21/ACOM
      COMMON/VAR22/TRIGTEMP,PHI
      COMMON/VAR23/REN,PR,FACT1,LAMB,SDIFF
      OPEN(UNIT=6,NAME='NSFCRD.DAT',TYPE='UNKNOWN',FORM='FORMATTED')
C
C ---     READ INPUT DATA("NSINPUT.DAT")
C
      CALL INPUT
C
C ---    STEP UP-INITALIZE
C
```

```
      CALL INIT
      CALL GRID
C
      DO 1 I        = 1 , NITER
      NOITER        = I
C
C --- MULTISTEP INTEGRATION - FOUR STEPS
C
      IA  = 1
C
      IF(IVIS.EQ.0) GO TO 10
      CALL VISS
      CALL PROPV
      CALL LOWERST
10    CONTINUE
C      CALL UPPERBD
      CALL PROPINV
      CALL STAB
      CALL FLUXST
      CALL SOURCE
      CALL NSSOLVE(IA)
C
      IA  = 2
C
      IF(IVIS.EQ.0) GO TO 11
      CALL PROPV
      CALL LOWERST
11    CONTINUE
C      CALL UPPERBD
      CALL PROPINV
      CALL STAB
      CALL FLUXST
      CALL SOURCE
      CALL NSSOLVE(IA)
C
      IA  = 3
C
      IF(IVIS.EQ.0) GO TO 12
      CALL PROPV
      CALL LOWERST
12    CONTINUE
C      CALL UPPERBD
      CALL PROPINV
      CALL STAB
      CALL FLUXST
      CALL SOURCE
      CALL NSSOLVE(IA)
C
      IA  = 4
C
```

```
      IF(IVIS.EQ.0) GO TO 13
      CALL PROPV
      CALL LOWERST
13    CONTINUE
C      CALL UPPERBD
      CALL PROPINV
      CALL STAB
      CALL FLUXST
      CALL SOURCE
      CALL NSSOLVE(IA)
C
C ---  POST SPLIT SMOOTHING
C
      CALL DAMPXST
      CALL DAMPYST
C
C ---  DETERMINE IF STEADY STATE HAS BEEN ACHIEVED
C
      CALL CONV
      IF(IRES.EQ.0.AND.I.GT.50) GO TO 100
      DO 2 IY        = 1 , NYY
      DO 2 IX        = 1 , NXX
      DO 2 IZ        = 1 , IEQ
      U(IZ,IX,IY,1) = U(IZ,IX,IY,2)
2     CONTINUE
1     CONTINUE
100   CONTINUE
C
C ---   DUMP OUTPUT DATA TO DATA FILES
C
      CALL OUT
C
C ---   COMPUTE AND PRINT OUT GROSS COMBUSTION PARAMETERS
C
      CALL PARAM
      STOP
      END
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      SUBROUTINE INPUT
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C --- LOAD INPUT DATA
C
      COMMON/VAR0/U(8,53,33,2),FI(8,53,33),GI(8,53,33),AH(8,53,33)
      COMMON/VAR00/FV(4,8,53,33),GV(4,8,53,33)
      COMMON/VAR1/UVEL(53,33),VVEL(53,33),PRES(53,33),TEMP(53,33)
      COMMON/VAR111/YH2(53,33),YO2(53,33),YH2O(53,33),YOH(53,33)
      COMMON/VAR2/DEN(53,33),SOUND(53,33),AINTE(53,33),AMACH(53,33)
      COMMON/VAR3/ENTHP(53,33),VIS(53,33),YYN2(53,33),CPND(53,33)
      COMMON/VAR4/DXX,DYY,X(55,35),Y(55,35),AREA(53,33)
```

```
      COMMON/VAR5/DT(53,33)
      COMMON/VAR6/NX,NXX,NXXX,NY,NYY,NYYY,IRES,IEQ
      COMMON/VAR7/IVIS,NITER,NOITER
      COMMON/VAR8/P1,T1,AM1,VISL,U1,V1,AK1,CV,CP,R,GAMA,DEN1,E1
      COMMON/VAR9/P11,C11,E11
      COMMON/VAR10/DFH2,DFO2,DFH2O,DFOH,DFN2
      COMMON/VAR11/CPH2,CPO2,CPH2O,CPOH,CPN2,CVH2,CVO2,CVH2O,CVOH,CVN2
      COMMON/VAR12/CONH2,CONO2,CONH2O,CONOH,CONN2
      COMMON/VAR13/COND,CFL,DCOFF,AL,VELO1
      COMMON/VAR14/RES(8,53,33)
      COMMON/VAR15/ALPHA(4)
      COMMON/VAR16/DXE(4,53,33),DXN(4,53,33),DXW(4,53,33),DXS(4,53,33)
      COMMON/VAR17/DYE(4,53,33),DYN(4,53,33),DYW(4,53,33),DYS(4,53,33)
      COMMON/VAR18/DD(2,8,53,33)
      COMMON/VAR20/NSX,NSXB,NSXA,NSY,NSYB,NSYA
      COMMON/VAR21/ACOM
      COMMON/VAR22/TRIGTEMP,PHI
      COMMON/VAR23/REN,PR,FACT1,LAMB,SDIFF
C
C ---   READ INPUT DATA("INPUT.DAT")
C
C --- NITER = MAXIMUN NUMBER OF ITERATIONS
C
      NITER   = 600
C
C --- NX, NY NUMBER OF POINTS IN THE X AND Y DIRECTIONS
C
      NX      = 51
      NY      = 31
C
C --- NSX , NSY POINTS WHERE STEP ENDS
C
      NSX     = 11
      NSY     = 11
C
C --- IEQ - NUMBER OF TRANSPORT EQUATIONS SOLVED
C
      IEQ     = 8
C
C --- FOR PURELY INVISCID CALCULATION   CFL = 2. (APPROXIMATELY)
C --- FOR A VISCOUS CALCULATION         CFL = .5 (APPROXIMATELY)
C
      CFL     = .5
C
C --- ARTIFICIAL VISCOSITY COEFFICIENT  CALCULATION  DCOFF = .1
C --- ARTIFICIAL VISCOSITY COEFFICIENT  CALCULATION  DCOFF = .05
C
      DCOFF   = .05
C
C --- RESCONV  CONVERGENCE CRITERIA
```

```
C
      RESCONV = .001
C
C ---  ALPHA(1,2,3,4) CONSTANTS USED BY THE TIME INTEGRATOR
C
      ALPHA(1) = .25
      ALPHA(2) = .33
      ALPHA(3) = .5
      ALPHA(4) = 1.
C
C ---   FOR INVISCID CALCULATION    IVIS = 0
C ---   FOR VISCOUS  CALCULATION    IVIS = 1
C
      IVIS     = 1
C
C ---  P1 FREE STREAM PRESSURE (N/M**2)
      P1       = 100000.
C
C ---  T1 FREE STREAM TEMPERATURE (K)
C
      T1       = 900.
C
C ---  U1 FREE STREAM U VELOCITY (M/S)
C
      U1       = 1200.
C
C ---  V1 FREE STREAM V VELOCITY (M/S)
C
      V1       = 0.0
C
C ---  AL TEST SECTION LENGTH
C
      AL       = .025
C
C ---  VISL   VISCOSITY AT FREE STREAM TEMPERATURE (N/M**2S)
C
      VISL     = 5.0E-5
C
C ---  COND   THERMAL DIFFUSIVITY AT FREE STREAM TEMPERATURE (W/MK)
C
      COND     = 2.4E-2
C
C ---  CP, CV    SPECIFIC HEATS AT FREE STREAM TEMPERATURE (J/KGK)
C
      CP       = 1000.
      CV       = 718.
C
C ---  GAS HEATS OF FORMATION AT ZERO DEGREES KELVIN (J/KG)
C
      DFH2     = 0.0
```

```
      DFO2     = 0.0
      DFH2O    = -1.44E+7
      DFOH     = 2.3E+6
      DFN2     = 0.0
C
C --- GAS SPECIFIC HEATS AT CONSTANT PRESSURE (J/KG*K)
C
      CPH2     = 17160.
      CPO2     = 1181.
      CPH2O    = 2854.
      CPOH     = 2041
      CPN2     = 1285.
C
C --- GAS SPECIFIC HEATS AT CONSTANT VOLUME (J/KG*K)
C
      CVH2     = 13000.
      CVO2     = 921.2
      CVH2O    = 2390.
      CVOH     = 1552.
      CVN2     = 988.
C
C --- INFLOW SPECIES DENSITY FRACTIONS
C
      CONH2    = .02619
      CONO2    = .2095
      CONH2O   = .0
      CONOH    = .0
      CONN2    = .76431
C
C --- IF NO CHEMICAL REACTION IS DESIRED ACOM = 0
C --- IF CHEMICAL REACTION IS DESIRED    ACOM = 1
C
      ACOM     = 1.
C
C --- TRIGGER TEMPERATURE - ABOVE THIS TEMPERATURE CHEMICAL REACTION
C                           OCCURS
C                         - BELOW THIS TEMPERATURE THE CHEMISTRY IS
C                           FROZEN
      TRIGTEMP = 1450.
C
C
C**********************************************************************
C
C
C --- WRITE OUT THE INPUT QUANTITIES TO THE OUTPUT DATA FILE  "NSFRCD"
C
      WRITE(6,10) NX , NY , NITER , IFQ
      WRITE(6,11) NSX , NSY
      WRITE(6,12) CFL , DCOFF
      WRITE(6,13) RESCONV
```

```
      WRITE(6,14) ALPHA(1),ALPHA(2),ALPHA(3),ALPHA(4)
      WRITE(6,15) IVIS
      WRITE(6,16) P1 , T1 , U1 , V1 ,AL
      WRITE(6,14) VISL , COND , CP , CV
      WRITE(6,16) DFH2 , DFO2 , DFH2O , DFOH , DFN2
      WRITE(6,16) CPH2 , CPO2 , CPH2O , CPOH , CPN2
      WRITE(6,16) CVH2 , CVO2 , CVH2O , CVOH , CVN2
      WRITE(6,16) CONH2 , CONO2 , CONH2O , CONOH , CONN2
      WRITE(6,17) ACOM , TRIGTEMP , PHI
10    FORMAT(1X,4(I4))
11    FORMAT(1X,2(I4))
12    FORMAT(1X,2(E10.4))
13    FORMAT(1X,E10.4)
14    FORMAT(1X,4(E10.4))
15    FORMAT(1X,I4)
16    FORMAT(1X,5(E10.4))
17    FORMAT(1X,3(E10.4))
      RETURN
      END
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      SUBROUTINE INIT
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C ---  GENERATE INITIAL CONDITIONS
C
      COMMON/VAR0/U(8,53,33,2),FI(8,53,33),GI(8,53,33),AH(8,53,33)
      COMMON/VAR00/FV(4,8,53,33),GV(4,8,53,33)
      COMMON/VAR1/UVEL(53,33),VVEL(53,33),PRES(53,33),TEMP(53,33)
      COMMON/VAR111/YH2(53,33),YO2(53,33),YH2O(53,33),YOH(53,33)
      COMMON/VAR2/DEN(53,33),SOUND(53,33),AINTE(53,33),AMACH(53,33)
      COMMON/VAR3/ENTHP(53,33),VIS(53,33),YYN2(53,33),CPND(53,33)
      COMMON/VAR4/DXX,DYY,X(55,35),Y(55,35),AREA(53,33)
      COMMON/VAR5/DT(53,33)
      COMMON/VAR6/NX,NXX,NXXX,NY,NYY,NYYY,IRES,IEQ
      COMMON/VAR7/IVIS,NITER,NOITER
      COMMON/VAR8/P1,T1,AM1,VISL,U1,V1,AK1,CV,CP,R,GAMA,DEN1,E1
      COMMON/VAR9/P11,C11,E11
      COMMON/VAR10/DFH2,DFO2,DFH2O,DFOH,DFN2
      COMMON/VAR11/CPH2,CPO2,CPH2O,CPOH,CPN2,CVH2,CVO2,CVH2O,CVOH,CVN2
      COMMON/VAR12/CONH2,CONO2,CONH2O,CONOH,CONN2
      COMMON/VAR13/COND,CFL,DCOFF,AL,VELO1
      COMMON/VAR14/RES(8,53,33)
      COMMON/VAR15/ALPHA(4)
      COMMON/VAR16/DXE(4,53,33),DXN(4,53,33),DXW(4,53,33),DXS(4,53,33)
      COMMON/VAR17/DYE(4,53,33),DYN(4,53,33),DYW(4,53,33),DYS(4,53,33)
      COMMON/VAR18/DD(2,8,53,33)
      COMMON/VAR20/NSX,NSXB,NSXA,NSY,NSYB,NSYA
      COMMON/VAR21/ACOM
      COMMON/VAR22/TRIGTEMP,PHI
      COMMON/VAR23/REN,PR,FACT1,LAMB,SDIFF
```

```
C
C ---    DETERMINE REMAINING INPUT UNKNOWNS
C
      CP     = CONH2 * CPH2  + CONO2 * CPO2 + CONH2O * CPH2O
     1                       + CONOH * CPOH + CONN2  * CPN2
      CV     = CONH2 * CVH2  + CONO2 * CVO2 + CONH2O * CVH2O
     1                       + CONOH * CVOH + CONN2  * CVN2
      R      = CP - CV
      GAMA   = CP / CV
      DHEATF= CONH2 * DFH2 + CONO2 * DFO2 + CONH2O * DFH2O
     1                     + CONOH * DFOH + CONN2  * DFN2
C
C ---  PHI IS THE FUEL EQUIVALENCE RATIO
C
      PHI    = (CONH2 / CONO2) * 8.
C
      DEN1   = P1/(R*T1)
      VELO1  = SQRT(U1**2 + V1**2)
      E1     = CV*T1+.5*VELO1**2 + DHEATF / VELO1**2
C
C ---    DETERMINE  NON-DIMENSIONAL VARIABLES
C
      P11          = P1/(DEN1*VELO1**2)
      E11          = CV*T1/VELO1**2 + DHEATF /VELO1**2 + .5
      C11          = (SQRT(GAMA*R*T1))/VELO1
C
C ---    COMPUTE THE FREE STREAM NON-DIMENSIONAL VARIABLES
C
      REN          = DEN1 * VELO1 * AL / VISL
      PR           = VISL * CP / COND
      AM1          = VELO1/SQRT(GAMA*R*T1)
      FACT         = 1.0 / ((GAMA - 1.0)*AM1**2)
      LAMB         = -.6666 / REN
      SDIFF        = 1.0 / (REN * PR)
C
      DO 1 J       = 1 , NY
      DO 1 I       = 1 , NX
      PRES(I,J)  = P11
      TEMP(I,J)  = 1.0
      UVEL(I,J)  = U1 / VELO1
      VVEL(I,J)  = V1 / VELO1
      YH2(I,J)   = CONH2
      YO2(I,J)   = CONO2
      YH2O(I,J)  = CONH2O
      YOH(I,J)   = CONOH
      YYN2(I,J)   = CONN2
      VELO        = UVEL(I,J)**2 + VVEL(I,J)**2
      AINTE(I,J) = E11
      ENTHP(I,J) = CP*T1/(VELO1**2)+.5*VELO + DHEATF / VELO1**2
      SOUND(I,J) = C11
```

```
AMACH(I,J) = AM1
CPND(I,J)  = 1.0
DEN(I,J)   = 1.0
U(1,I,J,1) = 1.0
U(2,I,J,1) = UVEL(1,J)
U(3,I,J,1) = VVEL(I,J)
U(4,I,J,1) = E11
U(5,I,J,1) = CONH2
U(6,I,J,1) = CONO2
U(7,I,J,1) = CONH2O
U(8,I,J,1) = CONN2
U(1,I,J,2) = 1.0
U(2,I,J,2) = UVEL(1,J)
U(3,I,J,2) = VVEL(I,J)
U(4,I,J,2) = E11
U(5,I,J,2) = CONH2
U(6,I,J,2) = CONO2
U(7,I,J,2) = CONH2O
U(8,I,J,2) = CONN2
FI(1,I,J)  = UVEL(1,J)
FI(2,I,J)  = UVEL(1,J)**2 + P11
FI(3,I,J)  = UVEL(I,J)*VVEL(I,J)
FI(4,I,J)  = (P11 + E11)*UVEL(1,J)
FI(5,I,J)  = UVEL(1,J) * CONH2
FI(6,I,J)  = UVEL(1,J) * CONO2
FI(7,I,J)  = UVEL(1,J) * CONH2O
FI(8,I,J)  = UVEL(1,J) * CONN2
GI(1,I,J)  = VVEL(I,J)
GI(2,I,J)  = UVEL(I,J)*VVEL(I,J)
GI(3,I,J)  = VVEL(I,J)**2 + P11
GI(4,I,J)  = (P11 + E11 )*VVEL(I,J)
GI(5,I,J)  = VVEL(1,J) * CONH2
GI(6,I,J)  = VVEL(1,J) * CONO2
GI(7,I,J)  = VVEL(1,J) * CONH2O
GI(8,I,J)  = VVEL(1,J) * CONN2
AH(1,I,J)  = 0.0
AH(2,I,J)  = 0.0
AH(3,I,J)  = 0.0
AH(4,I,J)  = 0.0
AH(5,I,J)  = 0.0
AH(6,I,J)  = 0.0
AH(7,I,J)  = 0.0
AH(8,I,J)  = 0.0
DO 1 K     = 1 , IEQ
FV(1,K,I,J) = 0.0
FV(2,K,I,J) = 0.0
FV(3,K,I,J) = 0.0
FV(4,K,I,J) = 0.0
GV(1,K,I,J) = 0.0
GV(2,K,I,J) = 0.0
```

```
      GV(3,K,I,J) = 0.0
      GV(4,K,I,J) = 0.0
1     CONTINUE
C
C ---   ZERO OUT ALL VISCOUS LENGTH ARRAYS
C
      DO 2 K = 1 , 4
      DO 2 I = 1 , NX
      DO 2 J = 1 , NY
      DXE(K,I,J) = 0.0
      DXN(K,I,J) = 0.0
      DXW(K,I,J) = 0.0
      DXS(K,I,J) = 0.0
      DYE(K,I,J) = 0.0
      DYN(K,I,J) = 0.0
      DYW(K,I,J) = 0.0
      DYS(K,I,J) = 0.0
2     CONTINUE
      RETURN
      END
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      SUBROUTINE GRID
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C ---  GENERATE THE PHYSICAL GRID
C
      COMMON/VAR0/U(8,53,33,2),FI(8,53,33),GI(8,53,33),AH(8,53,33)
      COMMON/VAR00/FV(4,8,53,33),GV(4,8,53,33)
      COMMON/VAR1/UVEL(53,33),VVEL(53,33),PRES(53,33),TEMP(53,33)
      COMMON/VAR111/YH2(53,33),YO2(53,33),YH2O(53,33),YOH(53,33)
      COMMON/VAR2/DEN(53,33),SOUND(53,33),AINTE(53,33),AMACH(53,33)
      COMMON/VAR3/ENTHP(53,33),VIS(53,33),YYN2(53,33),CPND(53,33)
      COMMON/VAR4/DXX,DYY,X(55,35),Y(55,35),AREA(53,33)
      COMMON/VAR5/DT(53,33)
      COMMON/VAR6/NX,NXX,NXXX,NY,NYY,NYYY,IRES,IEQ
      COMMON/VAR7/IVIS,NITER,NOITER
      COMMON/VAR8/P1,T1,AM1,VISL,U1,V1,AK1,CV,CP,R,GAMA,DEN1,E1
      COMMON/VAR9/P11,C11,E11
      COMMON/VAR10/DFH2,DFO2,DFH2O,DFOH,DFN2
      COMMON/VAR11/CPH2,CPO2,CPH2O,CPOH,CPN2,CVH2,CVO2,CVH2O,CVOH,CVN2
      COMMON/VAR12/CONH2,CONO2,CONH2O,CONOH,CONN2
      COMMON/VAR13/COND,CFL,DCOFF,AL,VELO1
      COMMON/VAR14/RES(8,53,33)
      COMMON/VAR15/ALPHA(4)
      COMMON/VAR16/DXE(4,53,33),DXN(4,53,33),DXW(4,53,33),DXS(4,53,33)
      COMMON/VAR17/DYE(4,53,33),DYN(4,53,33),DYW(4,53,33),DYS(4,53,33)
      COMMON/VAR18/DD(2,8,53,33)
      COMMON/VAR20/NSX,NSXB,NSXA,NSY,NSYB,NSYA
      COMMON/VAR21/ACOM
      COMMON/VAR22/TRIGTEMP,PHI
```

```
      COMMON/VAR23/REN,PR,FACT1,LAMB,SDIFF
C
      NXMAX= NX + 1
      NYMAX= NY + 1
      NXX  = NX-1
      NXXX = NX - 2
      NYY  = NY-1
      NYYY = NY - 2
      NSXB = NSX - 1
      NSXA = NSX + 1
      NSYB = NSY - 1
      NSYA = NSY + 1
C
C      ********************************
C ---  INPUT THE MESH DISTRIBUTION HERE
C      ********************************
C
C ---  X COORDINATE FORMULATION
C
C      DXX      = 1.0 / NXX
C      DYY      = .3  / NYY
C      X(1,1)   = 0.0
C      DO 1 I   = 2 , NXMAX
C1     X(I,1)   = X(I-1,1)+DXX
C      DO 2 J   = 2 , NYMAX
C      DO 2 I   = 1 ,NXMAX
C2     X(I,J)   = X(I,1)
      X(1,1)   = 0.0
      DO 1 I   = 2 , NXMAX
C
      AI  = I
C
      DXX = .025 / 50. * 1.0/.025
C      IF(I.LT.21) DXX = .5/1. * 1.0/20.
C                               *(21. - AI)/10.0
C      IF(I.GE.21) DXX = .003/1. * 1./15.
C      IF(I.GE.36) DXX = .5 * ((AI - 35.)/16.)**2.14
C      IF(I.GE.21) DXX = .5/1. * 1.0/20. * 1.0/3.
C    *                    * (1.0 + (AI - 20.)/5.)
C      IF(I.GE.41) DXX = .25/1. * 1.0/10.
C
1     X(I,1)   = X(I-1,1) + DXX
      DO 2 J   = 2 , NYMAX
      DO 2 I   = 1 ,NXMAX
2     X(I,J)   = X(I,1)
C
C ---  Y COORDINATE FORMULATION
C
      DO 4 I   = 1 , NXMAX
      Y(I,1)   = 0.0
```

```
      DO 4 J    = 2 , NYMAX
      AJ = J
C       IF(J.LT.11) DYY = .0015/1. * 1.0/10.
C       IF(J.GE.11) DYY = .006/1. * 1.0/20. * 1.0/3.0
C    $                            * (1.0 + (AJ - 11.)/5.)
      DYY = .015/.025 * 1./30.
      Y(I,J)           = Y(I,J-1) + DYY
4     CONTINUE

C       ********************************************
C ---   REMAINDER OF GRID GENERATION PROCESS AUTOMATIC
C       ********************************************
C
C ---   DETERMINE THE AREA OF EACH CELL
C
      DO 10 J = 1 , NY
      DO 10 I = 1 , NX
      A1        = (X(I+1,J+1) - X(I,J))   * (Y(I,J+1)   - Y(I+1,J))
      A2        = (X(I,J+1)   - X(I+1,J)) * (Y(I+1,J+1) - Y(I,J))
      AREA(I,J) = (ABS(A1) + ABS(A2)) / 2.0
10    CONTINUE
      DO 20 J     = 1 , NYY
      DO 20 I     = 1 , NXX
C
C ---  PROJECTIONS OF VISCOUS CELL EAST SIDE
C
      XE1          = X(I+2,J)   - X(I+2,J+1)
      XE2          = X(I+1,J)   - X(I+1,J+1)
      XN1          = X(I+2,J+1) - X(I+1,J+1)
      XN2          = X(I+1,J+1) - X(I,J+1)
      XW1          = X(I+1,J+1) - X(I+1,J)
      XW2          = X(I,J+1)   - X(I,J)
      XS1          = X(I,J)     - X(I+1,J)
      XS2          = X(I+1,J)   - X(I+2,J)
      YE1          = Y(I+2,J)   - Y(I+2,J+1)
      YE2          = Y(I+1,J)   - Y(I+1,J+1)
      YN1          = Y(I+2,J+1) - Y(I+1,J+1)
      YN2          = Y(I+1,J+1) - Y(I,J+1)
      YW1          = Y(I+1,J+1) - Y(I+1,J)
      YW2          = Y(I,J+1)   - Y(I,J)
      YS1          = Y(I,J)     - Y(I+1,J)
      YS2          = Y(I+1,J)   - Y(I+2,J)
      DXE(1,I,J)        = - .5 * (XE1 + XE2)
      DXN(1,I,J)        = - .5 * (XN1 + XN2)
      DXW(1,I,J)        = - .5 * (XW1 + XW2)
      DXS(1,I,J)        = - .5 * (XS1 + XS2)
      DYE(1,I,J)        = - .5 * (YE1 + YE2)
      DYN(1,I,J)        = - .5 * (YN1 + YN2)
      DYW(1,I,J)        = - .5 * (YW1 + YW2)
      DYS(1,I,J)        = - .5 * (YS1 + YS2)
```

```
      DYS(1,I,J)         = - .5 * (YS1 + YS2)
20    CONTINUE
C
C ---  PROJECTIONS OF VISCOUS CELL WEST SIDE
C
      DO 21 J      = 1 , NYY
      DO 21 I      = 2 , NXX
      XE1         = X(I+1,J)   - X(I+1,J+1)
      XE2         = X(I,J)     - X(I,J+1)
      XN1         = X(I+1,J+1) - X(I,J+1)
      XN2         = X(I,J+1)   - X(I-1,J+1)
      XW1         = X(I,J+1)   - X(I,J)
      XW2         = X(I-1,J+1) - X(I-1,J)
      XS1         = X(I-1,J)   - X(I,J)
      XS2         = X(I,J)     - X(I+1,J)
      YE1         = Y(I+1,J)   - Y(I+1,J+1)
      YE2         = Y(I,J)     - Y(I,J+1)
      YN1         = Y(I+1,J+1) - Y(I,J+1)
      YN2         = Y(I,J+1)   - Y(I-1,J+1)
      YW1         = Y(I,J+1)   - Y(I,J)
      YW2         = Y(I-1,J+1) - Y(I-1,J)
      YS1         = Y(I-1,J)   - Y(I,J)
      YS2         = Y(I,J)     - Y(I+1,J)
      DXE(3,I,J)         = - .5 * (XE1 + XE2)
      DXN(3,I,J)         = - .5 * (XN1 + XN2)
      DXW(3,I,J)         = - .5 * (XW1 + XW2)
      DXS(3,I,J)         = - .5 * (XS1 + XS2)
      DYE(3,I,J)         = - .5 * (YE1 + YE2)
      DYN(3,I,J)         = - .5 * (YN1 + YN2)
      DYW(3,I,J)         = - .5 * (YW1 + YW2)
      DYS(3,I,J)         = - .5 * (YS1 + YS2)
21    CONTINUE
C
C ---  PROJECTIONS OF VISCOUS CELL WEST SIDE FIRST CELL
C
      I           = 1
      DO 22  J    = 1 , NYY
      XE1         = X(I+1,J)   - X(I+1,J+1)
      XE2         = X(I,J)     - X(I,J+1)
      XN1         = X(I+1,J+1) - X(I,J+1)
      XN2         = XN1
      XW1         = X(I,J+1)   - X(I,J)
      XW2         = XW1
      XS2         = X(I,J)     - X(I+1,J)
      XS1         = XS2
      YE1         = Y(I+1,J)   - Y(I+1,J+1)
      YE2         = Y(I,J)     - Y(I,J+1)
      YN1         = Y(I+1,J+1) - Y(I,J+1)
      YN2         = YN1
      YW1         = Y(I,J+1)   - Y(I,J)
```

```
      YW2         = YW1
      YS2         = Y(I,J)       - Y(I+1,J)
      YS1         = YS2
      DXE(3,I,J)        = - .5 * (XE1 + XE2)
      DXN(3,I,J)        = - .5 * (XN1 + XN2)
      DXW(3,I,J)        = - .5 * (XW1 + XW2)
      DXS(3,I,J)        = - .5 * (XS1 + XS2)
      DYE(3,I,J)        = - .5 * (YE1 + YE2)
      DYN(3,I,J)        = - .5 * (YN1 + YN2)
      DYW(3,I,J)        = - .5 * (YW1 + YW2)
      DYS(3,I,J)        = - .5 * (YS1 + YS2)
22    CONTINUE
C
C ---  PROJECTIONS OF VISCOUS CELL NORTH FACE
C
      DO 30 J     = 1 , NYY
      DO 30 I     = 1 , NXX
      XE1         = X(I+1,J)       - X(I+1,J+1)
      XE2         = X(I+1,J+1)     - X(I+1,J+2)
      XN1         = X(I+1,J+1)     - X(I,J+1)
      XN2         = X(I+1,J+2)     - X(I,J+2)
      XW1         = X(I,J+2)       - X(I,J+1)
      XW2         = X(I,J+1)       - X(I,J)
      XS1         = X(I,J+1)       - X(I+1,J+1)
      XS2         = X(I,J)         - X(I+1,J)
      YE1         = Y(I+1,J)       - Y(I+1,J+1)
      YE2         = Y(I+1,J+1)     - Y(I+1,J+2)
      YN1         = Y(I+1,J+1)     - Y(I,J+1)
      YN2         = Y(I+1,J+2)     - Y(I,J+2)
      YW1         = Y(I,J+2)       - Y(I,J+1)
      YW2         = Y(I,J+1)       - Y(I,J)
      YS1         = Y(I,J+1)       - Y(I+1,J+1)
      YS2         = Y(I,J)         - Y(I+1,J)
      DXE(2,I,J)       = - .5 * (XE1 + XE2)
      DXN(2,I,J)       = - .5 * (XN1 + XN2)
      DXW(2,I,J)       = - .5 * (XW1 + XW2)
      DXS(2,I,J)       = - .5 * (XS1 + XS2)
      DYE(2,I,J)       = - .5 * (YE1 + YE2)
      DYN(2,I,J)       = - .5 * (YN1 + YN2)
      DYW(2,I,J)       = - .5 * (YW1 + YW2)
      DYS(2,I,J)       = - .5 * (YS1 + YS2)
30    CONTINUE
C
C ---  PROJECTIONS OF VISCOUS CELL SOUTH SIDE
C
      DO 50 J      = 2 , NYY
      DO 50 I      = 1 , NXX
      XE1         = X(I+1,J-1)     - X(I+1,J)
      XE2         = X(I+1,J)       - X(I+1,J+1)
      XN1         = X(I+1,J)       - X(I,J)
```

```
      XN2         = X(I+1,J+1)    - X(I,J+1)
      XW1         = X(I,J+1)      - X(I,J)
      XW2         = X(I,J)        - X(I,J-1)
      XS1         = X(I,J)        - X(I+1,J)
      XS2         = X(I,J-1)      - X(I+1,J-1)
      YE1         = Y(I+1,J-1)    - Y(I+1,J)
      YE2         = Y(I+1,J)      - Y(I+1,J+1)
      YN1         = Y(I+1,J)      - Y(I,J)
      YN2         = Y(I+1,J+1)    - Y(I,J+1)
      YW1         = Y(I,J+1)      - Y(I,J)
      YW2         = Y(I,J)        - Y(I,J-1)
      YS1         = Y(I,J)        - Y(I+1,J)
      YS2         = Y(I,J-1)      - Y(I+1,J-1)
      DXE(4,I,J)        = - .5 * (XE1 + XE2)
      DXN(4,I,J)        = - .5 * (XN1 + XN2)
      DXW(4,I,J)        = - .5 * (XW1 + XW2)
      DXS(4,I,J)        = - .5 * (XS1 + XS2)
      DYE(4,I,J)        = - .5 * (YE1 + YE2)
      DYN(4,I,J)        = - .5 * (YN1 + YN2)
      DYW(4,I,J)        = - .5 * (YW1 + YW2)
      DYS(4,I,J)        = - .5 * (YS1 + YS2)
50    CONTINUE
C
C ---   LOWER WALL BOUNDARY CELL FLUX
C
      J           = 1
      DO 60 I     = 1 , NXX
      XE2         = X(I+1,J)      - X(I+1,J+1)
      XE1         = XE2
      XN1         = X(I+1,J)      - X(I,J)
      XN2         = X(I+1,J+1)    - X(I,J+1)
      XW1         = X(I,J+1)      - X(I,J)
      XW2         = XW1
      XS1         = -XN1
      XS2         = -XN2
      YE2         = Y(I+1,J)      - Y(I+1,J+1)
      YE1         = YE2
      YN1         = Y(I+1,J)      - Y(I,J)
      YN2         = Y(I+1,J+1)    - Y(I,J+1)
      YW1         = Y(I,J+1)      - Y(I,J)
      YW2         = YW1
      YS1         = -YN1
      YS2         = -YN2
      DXE(4,I,J)        = - .5 * (XE1 + XE2)
      DXN(4,I,J)        = - .5 * (XN1 + XN2)
      DXW(4,I,J)        = - .5 * (XW1 + XW2)
      DXS(4,I,J)        = - .5 * (XS1 + XS2)
      DYE(4,I,J)        = - .5 * (YE1 + YE2)
      DYN(4,I,J)        = - .5 * (YN1 + YN2)
      DYW(4,I,J)        = - .5 * (YW1 + YW2)
```

```
      DYS(4,I,J)        = - .5 * (YS1 + YS2)
60    CONTINUE
      RETURN
      END
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      SUBROUTINE FLUX
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C --- INDIVIDUAL STEP OF MULTISTEP INTEGRATOR
C
      COMMON/VAR0/U(8,53,33,2),FI(8,53,33),GI(8,53,33),AH(8,53,33)
      COMMON/VAR00/FV(4,8,53,33),GV(4,8,53,33)
      COMMON/VAR1/UVEL(53,33),VVEL(53,33),PRES(53,33),TEMP(53,33)
      COMMON/VAR111/YH2(53,33),YO2(53,33),YH2O(53,33),YOH(53,33)
      COMMON/VAR2/DEN(53,33),SOUND(53,33),AINTE(53,33),AMACH(53,33)
      COMMON/VAR3/ENTHP(53,33),VIS(53,33),YYN2(53,33),CPND(53,33)
      COMMON/VAR4/DXX,DYY,X(55,35),Y(55,35),AREA(53,33)
      COMMON/VAR5/DT(53,33)
      COMMON/VAR6/NX,NXX,NXXX,NY,NYY,NYYY,IRES,IEQ
      COMMON/VAR7/IVIS,NITER,NOITER
      COMMON/VAR8/P1,T1,AM1,VISL,U1,V1,AK1,CV,CP,R,GAMA,DEN1,E1
      COMMON/VAR9/P11,C11,E11
      COMMON/VAR10/DFH2,DFO2,DFH2O,DFOH,DFN2
      COMMON/VAR11/CPH2,CPO2,CPH2O,CPOH,CPN2,CVH2,CVO2,CVH2O,CVOH,CVN2
      COMMON/VAR12/CONH2,CONO2,CONH2O,CONOH,CONN2
      COMMON/VAR13/COND,CFL,DCOFF,AL,VELO1
      COMMON/VAR14/RES(8,53,33)
      COMMON/VAR15/ALPHA(4)
      COMMON/VAR16/DXE(4,53,33),DXN(4,53,33),DXW(4,53,33),DXS(4,53,33)
      COMMON/VAR17/DYE(4,53,33),DYN(4,53,33),DYW(4,53,33),DYS(4,53,33)
      COMMON/VAR18/DD(2,8,53,33)
      COMMON/VAR20/NSX,NSXB,NSXA,NSY,NSYB,NSYA
      COMMON/VAR21/ACOM
      COMMON/VAR22/TRIGTEMP,PHI
      COMMON/VAR23/REN,PR,FACT1,LAMB,SDIFF
      DO   1  J    = 2 , NYYY
      DO   1  I    = 2 , NXXX
      X1           = X(I+1,J+1) - X(I+1,J)
      X2           = X(I,J+1)   - X(I+1,J+1)
      X3           = X(I,J)     - X(I,J+1)
      X4           = X(I+1,J)   - X(I,J)
      Y1           = Y(I+1,J+1) - Y(I+1,J)
      Y2           = Y(I,J+1)   - Y(I+1,J+1)
      Y3           = Y(I,J)     - Y(I,J+1)
      Y4           = Y(I+1,J)   - Y(I,J)
      DO   2  K    = 1 , IEQ
      F1           = .5 * (FI(K,I+1,J) + FI(K,I,J))
      F2           = .5 * (FI(K,I,J+1) + FI(K,I,J))
      F3           = .5 * (FI(K,I-1,J) + FI(K,I,J))
      F4           = .5 * (FI(K,I,J-1) + FI(K,I,J))
```

```
      G1            = .5 * (GI(K,I+1,J) + GI(K,I,J))
      G2            = .5 * (GI(K,I,J+1) + GI(K,I,J))
      G3            = .5 * (GI(K,I-1,J) + GI(K,I,J))
      G4            = .5 * (GI(K,I,J-1) + GI(K,I,J))
      FFLUXEN       = (F1 + FV(1,K,I,J))*Y1 + (F2 + FV(2,K,I,J))*Y2
      FFLUXWS       = (F3 + FV(3,K,I,J))*Y3 + (F4 + FV(4,K,I,J))*Y4
      GFLUXEN       = (G1 + GV(1,K,I,J))*X1 + (G2 + GV(2,K,I,J))*X2
      GFLUXWS       = (G3 + GV(3,K,I,J))*X3 + (G4 + GV(4,K,I,J))*X4
      RESID         = (FFLUXEN + FFLUXWS) - (GFLUXEN + GFLUXWS)
      RES(K,I,J)    = RESID / AREA(I,J)
2     CONTINUE
1     CONTINUE
C
C       ---------------------------
C ---   INFLOW  BOUNDARY  EVALUATION
C       ---------------------------
C
      I             = 1
      DO   10 J     = 2 , NYYY
      X1            = X(I+1,J+1) - X(I+1,J)
      X2            = X(I,J+1)   - X(I+1,J+1)
      X3            = X(I,J)     - X(I,J+1)
      X4            = X(I+1,J)   - X(I,J)
      Y1            = Y(I+1,J+1) - Y(I+1,J)
      Y2            = Y(I,J+1)   - Y(I+1,J+1)
      Y3            = Y(I,J)     - Y(I,J+1)
      Y4            = Y(I+1,J)   - Y(I,J)
      DO   11 K     = 1 , IEQ
      F1            = .5 * (FI(K,I+1,J) + FI(K,I,J))
      F2            = .5 * (FI(K,I,J+1) + FI(K,I,J))
      IF(K.EQ.1)F3 = U1/VELO1
      IF(K.EQ.2)F3 = (U1/VELO1)**2 + P11
      IF(K.EQ.3)F3 = (U1 * V1)/VELO1**2
      IF(K.EQ.4)F3 = E11 * U1/VELO1 + U1*P11/VELO1
      IF(K.EQ.5)F3 = CONH2  * U1/VELO1
      IF(K.EQ.6)F3 = CONO2  * U1/VELO1
      IF(K.EQ.7)F3 = CONH2O * U1/VELO1
      IF(K.EQ.8)F3 = CONN2  * U1/VELO1
      F4            = .5 * (FI(K,I,J-1) + FI(K,I,J))
      G1            = .5 * (GI(K,I+1,J) + GI(K,I,J))
      G2            = .5 * (GI(K,I,J+1) + GI(K,I,J))
      IF(K.EQ.1)G3 = V1/VELO1
      IF(K.EQ.2)G3 = V1 * U1/VELO1**2
      IF(K.EQ.3)G3 = V1**2/VELO1**2 + P11
      IF(K.EQ.4)G3 = E11    * V1/VELO1 + V1*P11/VELO1
      IF(K.EQ.5)G3 = CONH2  * V1/VELO1
      IF(K.EQ.6)G3 = CONO2  * V1/VELO1
      IF(K.EQ.7)G3 = CONH2O * V1/VELO1
      IF(K.EQ.8)G3 = CONN2  * V1/VELO1
      G4            = .5 * (GI(K,I,J-1) + GI(K,I,J))
```

```
      FFLUXEN        = (F1 + FV(1,K,I,J))*Y1 + (F2 + FV(2,K,I,J))*Y2
      FFLUXWS        = (F3 + FV(3,K,I,J))*Y3 + (F4 + FV(4,K,I,J))*Y4
      GFLUXEN        = (G1 + GV(1,K,I,J))*X1 + (G2 + GV(2,K,I,J))*X2
      GFLUXWS        = (G3 + GV(3,K,I,J))*X3 + (G4 + GV(4,K,I,J))*X4
      RESID          = (FFLUXEN + FFLUXWS) - (GFLUXEN + GFLUXWS)
      RES(K,I,J)      = RESID / AREA(I,J)
11    CONTINUE
10    CONTINUE
C
C ---  INFLOW BOUNDARY UPPER CORNER CELL
C
      I              = 1
      J              = NYY
      X1             = X(I+1,J+1) - X(I+1,J)
      X2             = X(I,J+1)   - X(I+1,J+1)
      X3             = X(I,J)     - X(I,J+1)
      X4             = X(I+1,J)   - X(I,J)
      Y1             = Y(I+1,J+1) - Y(I+1,J)
      Y2             = Y(I,J+1)   - Y(I+1,J+1)
      Y3             = Y(I,J)     - Y(I,J+1)
      Y4             = Y(I+1,J)   - Y(I,J)
      DO   12 K      = 1 , IEQ
      F1             = .5 * (FI(K,I+1,J) + FI(K,I,J))
      F2             = 0.0
      IF(K.EQ.2)F2 = PRES(1,NYY)
      IF(K.EQ.1)F3 = U1/VELO1
      IF(K.EQ.2)F3 = (U1/VELO1)**2 + P11
      IF(K.EQ.3)F3 = (U1 * V1)/VELO1**2
      IF(K.EQ.4)F3 = E11 * U1/VELO1 + U1*P11/VELO1
      IF(K.EQ.5)F3 = CONH2  * U1/VELO1
      IF(K.EQ.6)F3 = CONO2  * U1/VELO1
      IF(K.EQ.7)F3 = CONH2O * U1/VELO1
      IF(K.EQ.8)F3 = CONN2  * U1/VELO1
      F4             = .5 * (FI(K,I,J-1) + FI(K,I,J))
      G1             = .5 * (GI(K,I+1,J) + GI(K,I,J))
      G2             = 0.0
      IF(K.EQ.3)G2 = PRES(1,NYY)
      IF(K.EQ.1)G3 = V1/VELO1
      IF(K.EQ.2)G3 = V1 * U1/VELO1**2
      IF(K.EQ.3)G3 = V1**2/VELO1**2 + P11
      IF(K.EQ.4)G3 = E11    * V1/VELO1 + V1*P11/VELO1
      IF(K.EQ.5)G3 = CONH2  * V1/VELO1
      IF(K.EQ.6)G3 = CONO2  * V1/VELO1
      IF(K.EQ.7)G3 = CONH2O * V1/VELO1
      IF(K.EQ.8)G3 = CONN2  * V1/VELO1
      G4             = .5 * (GI(K,I,J-1) + GI(K,I,J))
      FFLUXEN        = (F1 + FV(1,K,I,J))*Y1 + (F2 + FV(2,K,I,J))*Y2
      FFLUXWS        = (F3 + FV(3,K,I,J))*Y3 + (F4 + FV(4,K,I,J))*Y4
      GFLUXEN        = (G1 + GV(1,K,I,J))*X1 + (G2 + GV(2,K,I,J))*X2
      GFLUXWS        = (G3 + GV(3,K,I,J))*X3 + (G4 + GV(4,K,I,J))*X4
```

```
      RESID          = (FFLUXEN + FFLUXWS) - (GFLUXEN + GFLUXWS)
      RES(K,I,J)      = RESID / AREA(I,J)
12    CONTINUE
C
C ---   INFLOW BOUNDARY LOWER CORNER CELL
C
      I              = 1
      J              = 1
      X1             = X(I+1,J+1) - X(I+1,J)
      X2             = X(I,J+1)   - X(I+1,J+1)
      X3             = X(I,J)     - X(I,J+1)
      X4             = X(I+1,J)   - X(I,J)
      Y1             = Y(I+1,J+1) - Y(I+1,J)
      Y2             = Y(I,J+1)   - Y(I+1,J+1)
      Y3             = Y(I,J)     - Y(I,J+1)
      Y4             = Y(I+1,J)   - Y(I,J)
      DO   13 K      = 1 , IEQ
      F1             = .5 * (FI(K,I+1,J) + FI(K,I,J))
      F2             = .5 * (FI(K,I,J+1) + FI(K,I,J))
      IF(K.EQ.1)F3 = U1/VELO1
      IF(K.EQ.2)F3 = (U1/VELO1)**2 + P11
      IF(K.EQ.3)F3 = (U1 * V1)/VELO1**2
      IF(K.EQ.4)F3 = E11 * U1/VELO1 + U1*P11/VELO1
      IF(K.EQ.5)F3 = CONH2  * U1/VELO1
      IF(K.EQ.6)F3 = CONO2  * U1/VELO1
      IF(K.EQ.7)F3 = CONH2O * U1/VELO1
      IF(K.EQ.8)F3 = CONN2  * U1/VELO1
      F4             = 0.0
      IF(K.EQ.2)F4 = PRES(1,1)
      G1             = .5 * (GI(K,I+1,J) + GI(K,I,J))
      G2             = .5 * (GI(K,I,J+1) + GI(K,I,J))
      IF(K.EQ.1)G3 = V1/VELO1
      IF(K.EQ.2)G3 = V1 * U1/VELO1**2
      IF(K.EQ.3)G3 = V1**2/VELO1**2 + P11
      IF(K.EQ.4)G3 = E11     * V1/VELO1 + V1*P11/VELO1
      IF(K.EQ.5)G3 = CONH2  * V1/VELO1
      IF(K.EQ.6)G3 = CONO2  * V1/VELO1
      IF(K.EQ.7)G3 = CONH2O * V1/VELO1
      IF(K.EQ.8)G3 = CONN2  * V1/VELO1
      G4             = 0.0
      IF(K.EQ.3)G4 = PRES(1,1)
      FFLUXEN        = (F1 + FV(1,K,I,J))*Y1 + (F2 + FV(2,K,I,J))*Y2
      FFLUXWS        = (F3 + FV(3,K,I,J))*Y3 + (F4 + FV(4,K,I,J))*Y4
      GFLUXEN        = (G1 + GV(1,K,I,J))*X1 + (G2 + GV(2,K,I,J))*X2
      GFLUXWS        = (G3 + GV(3,K,I,J))*X3 + (G4 + GV(4,K,I,J))*X4
      RESID          = (FFLUXEN + FFLUXWS) - (GFLUXEN + GFLUXWS)
      RES(K,I,J)      = RESID / AREA(I,J)
13    CONTINUE
C
C       ----------------------------------
```

```
C ---   LOWER BOUNDARY WALL CELL EVALUATION
C       -----------------------------------
C
      J  = 1
      DO   3  I       = 2 , NXXX
      PW = PRES(I,1)
      X1            = X(I+1,J+1) - X(I+1,J)
      X2            = X(I,J+1)   - X(I+1,J+1)
      X3            = X(I,J)     - X(I,J+1)
      X4            = X(I+1,J)   - X(I,J)
      Y1            = Y(I+1,J+1) - Y(I+1,J)
      Y2            = Y(I,J+1)   - Y(I+1,J+1)
      Y3            = Y(I,J)     - Y(I,J+1)
      Y4            = Y(I+1,J)   - Y(I,J)
      DO   4   K    = 1 , IEQ
      F1            = .5 * (FI(K,I+1,J) + FI(K,I,J))
      F2            = .5 * (FI(K,I,J+1) + FI(K,I,J))
      F3            = .5 * (FI(K,I-1,J) + FI(K,I,J))
      F4            = 0.0
      IF(K.EQ.2)F4 = PW
      G1            = .5 * (GI(K,I+1,J) + GI(K,I,J))
      G2            = .5 * (GI(K,I,J+1) + GI(K,I,J))
      G3            = .5 * (GI(K,I-1,J) + GI(K,I,J))
      G4            = 0.0
      IF(K.EQ.3)G4 = PW
      FFLUXEN       = (F1 + FV(1,K,I,J))*Y1 + (F2 + FV(2,K,I,J))*Y2
      FFLUXWS       = (F3 + FV(3,K,I,J))*Y3 + (F4 + FV(4,K,I,J))*Y4
      GFLUXEN       = (G1 + GV(1,K,I,J))*X1 + (G2 + GV(2,K,I,J))*X2
      GFLUXWS       = (G3 + GV(3,K,I,J))*X3 + (G4 + GV(4,K,I,J))*X4
      RESID         = (FFLUXEN + FFLUXWS) - (GFLUXEN + GFLUXWS)
      RES(K,I,J)     = RESID / AREA(I,J)
4     CONTINUE
3     CONTINUE
C
C       -----------------------------------
C ---   UPPER BOUNDARY WALL CELL EVALUATION
C       -----------------------------------
C
      J             = NYY
      DO   5  I     = 2 , NXXX
      X1            = X(I+1,J+1) - X(I+1,J)
      X2            = X(I,J+1)   - X(I+1,J+1)
      X3            = X(I,J)     - X(I,J+1)
      X4            = X(I+1,J)   - X(I,J)
      Y1            = Y(I+1,J+1) - Y(I+1,J)
      Y2            = Y(I,J+1)   - Y(I+1,J+1)
      Y3            = Y(I,J)     - Y(I,J+1)
      Y4            = Y(I+1,J)   - Y(I,J)
      DO   6  K     = 1 , IEQ
      F1            = .5 * (FI(K,I+1,J) + FI(K,I,J))
```

```
      F2             = 0.0
      IF(K.EQ.2)F2 = PRES(I,NYY)
      F3             = .5 * (FI(K,I-1,J) + FI(K,I,J))
      F4             = .5 * (FI(K,I,J-1) + FI(K,I,J))
      G1             = .5 * (GI(K,I+1,J) + GI(K,I,J))
      G2             = 0.0
      IF(K.EQ.3)G2 = PRES(I,NYY)
      G3             = .5 * (GI(K,I-1,J) + GI(K,I,J))
      G4             = .5 * (GI(K,I,J-1) + GI(K,I,J))
      FFLUXEN        = (F1 + FV(1,K,I,J))*Y1 + (F2 + FV(2,K,I,J))*Y2
      FFLUXWS        = (F3 + FV(3,K,I,J))*Y3 + (F4 + FV(4,K,I,J))*Y4
      GFLUXEN        = (G1 + GV(1,K,I,J))*X1 + (G2 + GV(2,K,I,J))*X2
      GFLUXWS        = (G3 + GV(3,K,I,J))*X3 + (G4 + GV(4,K,I,J))*X4
      RESID          = (FFLUXEN + FFLUXWS) - (GFLUXEN + GFLUXWS)
      RES(K,I,J)       = RESID / AREA(I,J)
6     CONTINUE
5     CONTINUE
C
C       -----------------------
C ---   EXIT BOUNDARY EVALUATION
C       -----------------------
C
C
C ---   UPPER CORNER CELL
C
C
      I              = NXX
      J              = NYY
      X1             = X(I+1,J+1) - X(I+1,J)
      X2             = X(I,J+1)   - X(I+1,J+1)
      X3             = X(I,J)     - X(I,J+1)
      X4             = X(I+1,J)   - X(I,J)
      Y1             = Y(I+1,J+1) - Y(I+1,J)
      Y2             = Y(I,J+1)   - Y(I+1,J+1)
      Y3             = Y(I,J)     - Y(I,J+1)
      Y4             = Y(I+1,J)   - Y(I,J)
      DO   20  K     = 1 , IEQ
      F1             = FI(K,I,J)
      F2             = 0.0
      IF(K.EQ.2)F2 = PRES(I,J)
      F3             = .5 * (FI(K,I-1,J) + FI(K,I,J))
      F4             = .5 * (FI(K,I,J-1) + FI(K,I,J))
      G1             = GI(K,I,J)
      G2             = 0.0
      IF(K.EQ.3)G2 = PRES(I,J)
      G3             = .5 * (GI(K,I-1,J) + GI(K,I,J))
      G4             = .5 * (GI(K,I,J-1) + GI(K,I,J))
      FFLUXEN        = (F1 + FV(1,K,I,J))*Y1 + (F2 + FV(2,K,I,J))*Y2
      FFLUXWS        = (F3 + FV(3,K,I,J))*Y3 + (F4 + FV(4,K,I,J))*Y4
      GFLUXEN        = (G1 + GV(1,K,I,J))*X1 + (G2 + GV(2,K,I,J))*X2
```

```
      GFLUXWS        = (G3 + GV(3,K,I,J))*X3 + (G4 + GV(4,K,I,J))*X4
      RESID          = (FFLUXEN + FFLUXWS) - (GFLUXEN + GFLUXWS)
      RES(K,I,J)     = RESID / AREA(I,J)
 20   CONTINUE
C
C ---  LOWER EXIT CORNER ( X = NXX , Y = 1 )
C
      I              = NXX
      J              = 1
      X1             = X(I+1,J+1) - X(I+1,J)
      X2             = X(I,J+1)   - X(I+1,J+1)
      X3             = X(I,J)     - X(I,J+1)
      X4             = X(I+1,J)   - X(I,J)
      Y1             = Y(I+1,J+1) - Y(I+1,J)
      Y2             = Y(I,J+1)   - Y(I+1,J+1)
      Y3             = Y(I,J)     - Y(I,J+1)
      Y4             = Y(I+1,J)   - Y(I,J)
      DO   21  K     = 1 , IEQ
      F1             = FI(K,I,J)
      F2             = .5 * (FI(K,I,J+1) + FI(K,I,J))
      F3             = .5 * (FI(K,I-1,J) + FI(K,I,J))
      F4             = 0.0
      IF(K.EQ.2)F4   = PRES(NXX,J)
      G1             = GI(K,I,J)
      G2             = .5 * (GI(K,I,J+1) + GI(K,I,J))
      G3             = .5 * (GI(K,I-1,J) + GI(K,I,J))
      G4             = 0.0
      IF(K.EQ.3)G4   = PRES(NXX,J)
      FFLUXEN        = (F1 + FV(1,K,I,J))*Y1 + (F2 + FV(2,K,I,J))*Y2
      FFLUXWS        = (F3 + FV(3,K,I,J))*Y3 + (F4 + FV(4,K,I,J))*Y4
      GFLUXEN        = (G1 + GV(1,K,I,J))*X1 + (G2 + GV(2,K,I,J))*X2
      GFLUXWS        = (G3 + GV(3,K,I,J))*X3 + (G4 + GV(4,K,I,J))*X4
      RESID          = (FFLUXEN + FFLUXWS) - (GFLUXEN + GFLUXWS)
      RES(K,I,J)     = RESID / AREA(I,J)
 21   CONTINUE
C
C ---    VERTICAL EXIT CELL EVALUATION ( X = NXX , Y = 2 , NYYY )
C
      I              = NXX
      DO   22  J     = 2 , NYYY
      X1             = X(I+1,J+1) - X(I+1,J)
      X2             = X(I,J+1)   - X(I+1,J+1)
      X3             = X(I,J)     - X(I,J+1)
      X4             = X(I+1,J)   - X(I,J)
      Y1             = Y(I+1,J+1) - Y(I+1,J)
      Y2             = Y(I,J+1)   - Y(I+1,J+1)
      Y3             = Y(I,J)     - Y(I,J+1)
      Y4             = Y(I+1,J)   - Y(I,J)
      DO   22  K     = 1 , IEQ
      F1             =  FI(K,I,J)
```

```
      F2           = .5 * (FI(K,I,J+1) + FI(K,I,J))
      F3           = .5 * (FI(K,I-1,J) + FI(K,I,J))
      F4           = .5 * (FI(K,I,J-1) + FI(K,I,J))
      G1           =  GI(K,I,J)
      G2           = .5 * (GI(K,I,J+1) + GI(K,I,J))
      G3           = .5 * (GI(K,I-1,J) + GI(K,I,J))
      G4           = .5 * (GI(K,I,J-1) + GI(K,I,J))
      FFLUXEN      = (F1 + FV(1,K,I,J))*Y1 + (F2 + FV(2,K,I,J))*Y2
      FFLUXWS      = (F3 + FV(3,K,I,J))*Y3 + (F4 + FV(4,K,I,J))*Y4
      GFLUXEN      = (G1 + GV(1,K,I,J))*X1 + (G2 + GV(2,K,I,J))*X2
      GFLUXWS      = (G3 + GV(3,K,I,J))*X3 + (G4 + GV(4,K,I,J))*X4
      RESID        = (FFLUXEN + FFLUXWS) - (GFLUXEN + GFLUXWS)
      RES(K,I,J)   = RESID / AREA(I,J)
22    CONTINUE
      RETURN
      END
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      SUBROUTINE FLUXST
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C ---  INDIVIDUAL STEP OF MULTISTEP INTEGRATOR
C
      COMMON/VAR0/U(8,53,33,2),FI(8,53,33),GI(8,53,33),AH(8,53,33)
      COMMON/VAR00/FV(4,8,53,33),GV(4,8,53,33)
      COMMON/VAR1/UVEL(53,33),VVEL(53,33),PRES(53,33),TEMP(53,33)
      COMMON/VAR111/YH2(53,33),YO2(53,33),YH2O(53,33),YOH(53,33)
      COMMON/VAR2/DEN(53,33),SOUND(53,33),AINTE(53,33),AMACH(53,33)
      COMMON/VAR3/ENTHP(53,33),VIS(53,33),YYN2(53,33),CPND(53,33)
      COMMON/VAR4/DXX,DYY,X(55,35),Y(55,35),AREA(53,33)
      COMMON/VAR5/DT(53,33)
      COMMON/VAR6/NX,NXX,NXXX,NY,NYY,NYYY,IRES,IEQ
      COMMON/VAR7/IVIS,NITER,NOITER
      COMMON/VAR8/P1,T1,AM1,VISL,U1,V1,AK1,CV,CP,R,GAMA,DEN1,E1
      COMMON/VAR9/P11,C11,E11
      COMMON/VAR10/DFH2,DFO2,DFH2O,DFOH,DFN2
      COMMON/VAR11/CPH2,CPO2,CPH2O,CPOH,CPN2,CVH2,CVO2,CVH2O,CVOH,CVN2
      COMMON/VAR12/CONH2,CONO2,CONH2O,CONOH,CONN2
      COMMON/VAR13/COND,CFL,DCOFF,AL,VELO1
      COMMON/VAR14/RES(8,53,33)
      COMMON/VAR15/ALPHA(4)
      COMMON/VAR16/DXE(4,53,33),DXN(4,53,33),DXW(4,53,33),DXS(4,53,33)
      COMMON/VAR17/DYE(4,53,33),DYN(4,53,33),DYW(4,53,33),DYS(4,53,33)
      COMMON/VAR18/DD(2,8,53,33)
      COMMON/VAR20/NSX,NSXB,NSXA,NSY,NSYB,NSYA
      COMMON/VAR21/ACOM
      COMMON/VAR22/TRIGTEMP,PHI
      COMMON/VAR23/REN,PR,FACT1,LAMB,SDIFF
      DIMENSION UIN(33),TIN(33),DENIN(33),EIN(33)
C
C ---    INPUT STARTING BOUNDARY LAYER PROFILE
```

```
C
      DFACT          = 1.1
      DO 779 J       = 11 , NYY
      AHIGHT         = 1.0/.2 * (Y(1,J) - Y(1,10))
      UIN(J)         = VELO1 * (1.5 * AHIGHT - .5 * AHIGHT**3)
      TIN11          = 1.5 * AHIGHT/DFACT - .5 *(AHIGHT/DFACT)**3
      TIN(J)         = 1.64 + (1.0 -1.64) * TIN11
      DENIN1         = P1/(R*T1)
      DENIN(J)       = P1/(R * TIN(J) *T1) * 1./DENIN1
      EIN(J)         = CV * TIN(J)* T1/VELO1**2
     *                    + .5 * (UIN(J)/VELO1)**2
      IF(J.GT.20)  UIN(J) = U1
      IF(J.GT.20)  TIN(J) = 1.0
      IF(J.GT.20)  DENIN(J)= 1.0
      IF(J.GT.20)  EIN(J) = CV * T1/VELO1**2 + .5
779   CONTINUE
C
C ---
C
      DO    1  J     = 2 , NYYY
      DO    1  I     = 2 , NXXX
      IF(I.LT.NSX.AND.J.LT.NSY) GO TO 1
      X1             = X(I+1,J+1) - X(I+1,J)
      X2             = X(I,J+1)   - X(I+1,J+1)
      X3             = X(I,J)     - X(I,J+1)
      X4             = X(I+1,J)   - X(I,J)
      Y1             = Y(I+1,J+1) - Y(I+1,J)
      Y2             = Y(I,J+1)   - Y(I+1,J+1)
      Y3             = Y(I,J)     - Y(I,J+1)
      Y4             = Y(I+1,J)   - Y(I,J)
      DO    2  K     = 1 , IEQ
      F1             = .5 * (FI(K,I+1,J) + FI(K,I,J))
      F2             = .5 * (FI(K,I,J+1) + FI(K,I,J))
      F3             = .5 * (FI(K,I-1,J) + FI(K,I,J))
      F4             = .5 * (FI(K,I,J-1) + FI(K,I,J))
      G1             = .5 * (GI(K,I+1,J) + GI(K,I,J))
      G2             = .5 * (GI(K,I,J+1) + GI(K,I,J))
      G3             = .5 * (GI(K,I-1,J) + GI(K,I,J))
      G4             = .5 * (GI(K,I,J-1) + GI(K,I,J))
      FFLUXEN        = (F1 + FV(1,K,I,J))*Y1 + (F2 + FV(2,K,I,J))*Y2
      FFLUXWS        = (F3 + FV(3,K,I,J))*Y3 + (F4 + FV(4,K,I,J))*Y4
      GFLUXEN        = (G1 + GV(1,K,I,J))*X1 + (G2 + GV(2,K,I,J))*X2
      GFLUXWS        = (G3 + GV(3,K,I,J))*X3 + (G4 + GV(4,K,I,J))*X4
      RESID          = (FFLUXEN + FFLUXWS) - (GFLUXEN + GFLUXWS)
      RES(K,I,J)     = RESID / AREA(I,J)
2     CONTINUE
1     CONTINUE
C
C     ----------------------------
C --- INFLOW  BOUNDARY  EVALUATION
```

```
C       --------------------------
C
        I               = 1
        DO    10 J      = NSYA , NYYY
        X1              = X(I+1,J+1) - X(I+1,J)
        X2              = X(I,J+1)   - X(I+1,J+1)
        X3              = X(I,J)     - X(I,J+1)
        X4              = X(I+1,J)   - X(I,J)
        Y1              = Y(I+1,J+1) - Y(I+1,J)
        Y2              = Y(I,J+1)   - Y(I+1,J+1)
        Y3              = Y(I,J)     - Y(I,J+1)
        Y4              = Y(I+1,J)   - Y(I,J)
        DO    11 K      = 1 , IEQ
        F1              = .5 * (FI(K,I+1,J) + FI(K,I,J))
        F2              = .5 * (FI(K,I,J+1) + FI(K,I,J))
        IF(K.EQ.1)F3 = DENIN(J) * UIN(J)/VELO1
        IF(K.EQ.2)F3 = DENIN(J) * (UIN(J)/VELO1)**2 + P11
        IF(K.EQ.3)F3 = DENIN(J) * (UIN(J) * V1)/VELO1**2
        IF(K.EQ.4)F3 = DENIN(J) * EIN(J) * UIN(J)/VELO1
     &                   + UIN(J)*P11/VELO1
        IF(K.EQ.5)F3 = DENIN(J) * CONH2  * UIN(J)/VELO1
        IF(K.EQ.6)F3 = DENIN(J) * CONO2  * UIN(J)/VELO1
        IF(K.EQ.7)F3 = DENIN(J) * CONH2O * UIN(J)/VELO1
        IF(K.EQ.8)F3 = DENIN(J) * CONN2  * UIN(J)/VELO1
        F4              = .5 * (FI(K,I,J-1) + FI(K,I,J))
        G1              = .5 * (GI(K,I+1,J) + GI(K,I,J))
        G2              = .5 * (GI(K,I,J+1) + GI(K,I,J))
        IF(K.EQ.1)G3 = DENIN(J) * V1/VELO1
        IF(K.EQ.2)G3 = DENIN(J) * (V1 * UIN(J))/VELO1**2
        IF(K.EQ.3)G3 = DENIN(J) * V1**2/VELO1**2 + P11
        IF(K.EQ.4)G3 = DENIN(J) * EIN(J)    * V1/VELO1
     &                         + V1*P11/VELO1
        IF(K.EQ.5)G3 = DENIN(J) * CONH2  * V1/VELO1
        IF(K.EQ.6)G3 = DENIN(J) * CONO2  * V1/VELO1
        IF(K.EQ.7)G3 = DENIN(J) * CONH2O * V1/VELO1
        IF(K.EQ.8)G3 = DENIN(J) * CONN2  * V1/VELO1
        G4              = .5 * (GI(K,I,J-1) + GI(K,I,J))
        FFLUXEN         = (F1 + FV(1,K,I,J))*Y1 + (F2 + FV(2,K,I,J))*Y2
        FFLUXWS         = (F3 + FV(3,K,I,J))*Y3 + (F4 + FV(4,K,I,J))*Y4
        GFLUXEN         = (G1 + GV(1,K,I,J))*X1 + (G2 + GV(2,K,I,J))*X2
        GFLUXWS         = (G3 + GV(3,K,I,J))*X3 + (G4 + GV(4,K,I,J))*X4
        RESID           = (FFLUXEN + FFLUXWS) - (GFLUXEN + GFLUXWS)
        RES(K,I,J)        = RESID / AREA(I,J)
11      CONTINUE
10      CONTINUE
C
C ---    INFLOW BOUNDARY UPPER CORNER CELL
C
        I               = 1
        J               = NYY
```

```
      X1             = X(I+1,J+1) - X(I+1,J)
      X2             = X(I,J+1)   - X(I+1,J+1)
      X3             = X(I,J)     - X(I,J+1)
      X4             = X(I+1,J)   - X(I,J)
      Y1             = Y(I+1,J+1) - Y(I+1,J)
      Y2             = Y(I,J+1)   - Y(I+1,J+1)
      Y3             = Y(I,J)     - Y(I,J+1)
      Y4             = Y(I+1,J)   - Y(I,J)
      DO   12 K      = 1 , IEQ
      F1             = .5 * (FI(K,I+1,J) + FI(K,I,J))
      F2             = 0.0
      IF(K.EQ.2)F2 = PRES(1,NYY)
      IF(K.EQ.1)F3 = DENIN(J) * UIN(J)/VELO1
      IF(K.EQ.2)F3 = DENIN(J) * (UIN(J)/VELO1)**2 + P11
      IF(K.EQ.3)F3 = DENIN(J) * (UIN(J) * V1)/VELO1**2
      IF(K.EQ.4)F3 = DENIN(J) * EIN(J) * UIN(J)/VELO1
     &                 + UIN(J)*P11/VELO1
      IF(K.EQ.5)F3 = DENIN(J) * CONH2  * UIN(J)/VELO1
      IF(K.EQ.6)F3 = DENIN(J) * CONO2  * UIN(J)/VELO1
      IF(K.EQ.7)F3 = DENIN(J) * CONH2O * UIN(J)/VELO1
      IF(K.EQ.8)F3 = DENIN(J) * CONN2  * UIN(J)/VELO1
      F4             = .5 * (FI(K,I,J-1) + FI(K,I,J))
      G1             = .5 * (GI(K,I+1,J) + GI(K,I,J))
      G2             = 0.0
      IF(K.EQ.3)G2 = PRES(1,NYY)
      IF(K.EQ.1)G3 = DENIN(J) * V1/VELO1
      IF(K.EQ.2)G3 = DENIN(J) * (V1 * UIN(J))/VELO1**2
      IF(K.EQ.3)G3 = DENIN(J) * V1**2/VELO1**2 + P11
      IF(K.EQ.4)G3 = DENIN(J) * EIN(J)    * V1/VELO1
     &                      + V1*P11/VELO1
      IF(K.EQ.5)G3 = DENIN(J) * CONH2  * V1/VELO1
      IF(K.EQ.6)G3 = DENIN(J) * CONO2  * V1/VELO1
      IF(K.EQ.7)G3 = DENIN(J) * CONH2O * V1/VELO1
      IF(K.EQ.8)G3 = DENIN(J) * CONN2  * V1/VELO1
      G4             = .5 * (GI(K,I,J-1) + GI(K,I,J))
      FFLUXEN        = (F1 + FV(1,K,I,J))*Y1 + (F2 + FV(2,K,I,J))*Y2
      FFLUXWS        = (F3 + FV(3,K,I,J))*Y3 + (F4 + FV(4,K,I,J))*Y4
      GFLUXEN        = (G1 + GV(1,K,I,J))*X1 + (G2 + GV(2,K,I,J))*X2
      GFLUXWS        = (G3 + GV(3,K,I,J))*X3 + (G4 + GV(4,K,I,J))*X4
      RESID          = (FFLUXEN + FFLUXWS) - (GFLUXEN + GFLUXWS)
      RES(K,I,J)       = RESID / AREA(I,J)
12    CONTINUE
C
C ---   INFLOW BOUNDARY LOWER CORNER CELL
C
      I              = 1
      J              = NSY
      X1             = X(I+1,J+1) - X(I+1,J)
      X2             = X(I,J+1)   - X(I+1,J+1)
      X3             = X(I,J)     - X(I,J+1)
```

```
      X4            = X(I+1,J)   - X(I,J)
      Y1            = Y(I+1,J+1) - Y(I+1,J)
      Y2            = Y(I,J+1)   - Y(I+1,J+1)
      Y3            = Y(I,J)     - Y(I,J+1)
      Y4            = Y(I+1,J)   - Y(I,J)
      DO   13 K     = 1 , IEQ
      F1            = .5 * (FI(K,I+1,J) + FI(K,I,J))
      F2            = .5 * (FI(K,I,J+1) + FI(K,I,J))
      IF(K.EQ.1)F3 = DENIN(J) * UIN(J)/VELO1
      IF(K.EQ.2)F3 = DENIN(J) * (UIN(J)/VELO1)**2 + P11
      IF(K.EQ.3)F3 = DENIN(J) * (UIN(J) * V1)/VELO1**2
      IF(K.EQ.4)F3 = DENIN(J) * (EIN(J) * UIN(J))/VELO1
     &                  + UIN(J)*P11/VELO1
      IF(K.EQ.5)F3 = DENIN(J) * CONH2  * UIN(J)/VELO1
      IF(K.EQ.6)F3 = DENIN(J) * CONO2  * UIN(J)/VELO1
      IF(K.EQ.7)F3 = DENIN(J) * CONH2O * UIN(J)/VELO1
      IF(K.EQ.8)F3 = DENIN(J) * CONN2  * UIN(J)/VELO1
      F4            = 0.0
      IF(K.EQ.2)F4 = P11
      G1            = .5 * (GI(K,I+1,J) + GI(K,I,J))
      G2            = .5 * (GI(K,I,J+1) + GI(K,I,J))
      IF(K.EQ.1)G3 = DENIN(J) * V1/VELO1
      IF(K.EQ.2)G3 = DENIN(J) * (V1 * UIN(J))/VELO1**2
      IF(K.EQ.3)G3 = DENIN(J) * V1**2/VELO1**2 + P11
      IF(K.EQ.4)G3 = DENIN(J) * EIN(J)    * V1/VELO1
     &                      + V1*P11/VELO1
      IF(K.EQ.5)G3 = DENIN(J) * CONH2  * V1/VELO1
      IF(K.EQ.6)G3 = DENIN(J) * CONO2  * V1/VELO1
      IF(K.EQ.7)G3 = DENIN(J) * CONH2O * V1/VELO1
      IF(K.EQ.8)G3 = DENIN(J) * CONN2  * V1/VELO1
      G4            = 0.0
      IF(K.EQ.3)G4 = P11
      FFLUXEN       = (F1 + FV(1,K,I,J))*Y1 + (F2 + FV(2,K,I,J))*Y2
      FFLUXWS       = (F3 + FV(3,K,I,J))*Y3 + (F4 + FV(4,K,I,J))*Y4
      GFLUXEN       = (G1 + GV(1,K,I,J))*X1 + (G2 + GV(2,K,I,J))*X2
      GFLUXWS       = (G3 + GV(3,K,I,J))*X3 + (G4 + GV(4,K,I,J))*X4
      RESID         = (FFLUXEN + FFLUXWS) - (GFLUXEN + GFLUXWS)
      RES(K,I,J)     = RESID / AREA(I,J)
13    CONTINUE
C
C      ----------------------------------
C ---  LOWER BOUNDARY WALL CELL EVALUATION
C      ----------------------------------
C
      DO   3  I        = 2 , NXXX
      IF(I.EQ.NSX) GO TO 3
      IF(I.LT.NSX) J  = NSY
      IF(I.LT.NSX) PW = PRES(I,NSY)
      IF(I.GT.NSX) J  = 1
      IF(I.GT.NSX) PW = PRES(I,1)
```

```
      X1             = X(I+1,J+1) - X(I+1,J)
      X2             = X(I,J+1)   - X(I+1,J+1)
      X3             = X(I,J)     - X(I,J+1)
      X4             = X(I+1,J)   - X(I,J)
      Y1             = Y(I+1,J+1) - Y(I+1,J)
      Y2             = Y(I,J+1)   - Y(I+1,J+1)
      Y3             = Y(I,J)     - Y(I,J+1)
      Y4             = Y(I+1,J)   - Y(I,J)
      DO   4   K     = 1 , IEQ
      F1             = .5 * (FI(K,I+1,J) + FI(K,I,J))
      F2             = .5 * (FI(K,I,J+1) + FI(K,I,J))
      F3             = .5 * (FI(K,I-1,J) + FI(K,I,J))
      F4             = 0.0
      IF(K.EQ.2)F4 = PW
      G1             = .5 * (GI(K,I+1,J) + GI(K,I,J))
      G2             = .5 * (GI(K,I,J+1) + GI(K,I,J))
      G3             = .5 * (GI(K,I-1,J) + GI(K,I,J))
      G4             = 0.0
      IF(K.EQ.3)G4 = PW
      FFLUXEN        = (F1 + FV(1,K,I,J))*Y1 + (F2 + FV(2,K,I,J))*Y2
      FFLUXWS        = (F3 + FV(3,K,I,J))*Y3 + (F4 + FV(4,K,I,J))*Y4
      GFLUXEN        = (G1 + GV(1,K,I,J))*X1 + (G2 + GV(2,K,I,J))*X2
      GFLUXWS        = (G3 + GV(3,K,I,J))*X3 + (G4 + GV(4,K,I,J))*X4
      RESID          = (FFLUXEN + FFLUXWS) - (GFLUXEN + GFLUXWS)
      RES(K,I,J)      = RESID / AREA(I,J)
4     CONTINUE
3     CONTINUE

C
C ---   LOWER BOUNDARY STEP EDGE CELL EVALUATION
C
      J              = NSY
      I              = NSX
      X1             = X(I+1,J+1) - X(I+1,J)
      X2             = X(I,J+1)   - X(I+1,J+1)
      X3             = X(I,J)     - X(I,J+1)
      X4             = X(I+1,J)   - X(I,J)
      Y1             = Y(I+1,J+1) - Y(I+1,J)
      Y2             = Y(I,J+1)   - Y(I+1,J+1)
      Y3             = Y(I,J)     - Y(I,J+1)
      Y4             = Y(I+1,J)   - Y(I,J)
      DO   14  K     = 1 , IEQ
      F1             = .5 * (FI(K,I+1,J) + FI(K,I,J))
      F2             = .5 * (FI(K,I,J+1) + FI(K,I,J))
      F3             = .5 * (FI(K,I-1,J) + FI(K,I,J))
      F4             = .5 * (FI(K,I,J-1) + FI(K,I,J))
      G1             = .5 * (GI(K,I+1,J) + GI(K,I,J))
      G2             = .5 * (GI(K,I,J+1) + GI(K,I,J))
      G3             = .5 * (GI(K,I-1,J) + GI(K,I,J))
      G4             = .5 * (GI(K,I,J-1) + GI(K,I,J))
```

```
      FFLUXEN       = (F1 + FV(1,K,I,J))*Y1 + (F2 + FV(2,K,I,J))*Y2
      FFLUXWS       = (F3 + FV(3,K,I,J))*Y3 + (F4 + FV(4,K,I,J))*Y4
      GFLUXEN       = (G1 + GV(1,K,I,J))*X1 + (G2 + GV(2,K,I,J))*X2
      GFLUXWS       = (G3 + GV(3,K,I,J))*X3 + (G4 + GV(4,K,I,J))*X4
      RESID         = (FFLUXEN + FFLUXWS) - (GFLUXEN + GFLUXWS)
      RES(K,I,J)      = RESID / AREA(I,J)
14    CONTINUE
C
C ---   LOWER STEP VERTICAL WALL CELL EVALUATION
C
      I             = NSX
      DO   16 J     = 2 , NSYB
      X1            = X(I+1,J+1) - X(I+1,J)
      X2            = X(I,J+1)   - X(I+1,J+1)
      X3            = X(I,J)     - X(I,J+1)
      X4            = X(I+1,J)   - X(I,J)
      Y1            = Y(I+1,J+1) - Y(I+1,J)
      Y2            = Y(I,J+1)   - Y(I+1,J+1)
      Y3            = Y(I,J)     - Y(I,J+1)
      Y4            = Y(I+1,J)   - Y(I,J)
      DO   17 K     = 1 , IEQ
      F1            = .5 * (FI(K,I+1,J) + FI(K,I,J))
      F2            = .5 * (FI(K,I,J+1) + FI(K,I,J))
      F3            = 0.0
      IF(K.EQ.2)F3  = PRES(NSX,J)
      F4            = .5 * (FI(K,I,J-1) + FI(K,I,J))
      G1            = .5 * (GI(K,I+1,J) + GI(K,I,J))
      G2            = .5 * (GI(K,I,J+1) + GI(K,I,J))
      G3            = 0.0
      IF(K.EQ.3)G3  = PRES(NSX,J)
      G4            = .5 * (GI(K,I,J-1) + GI(K,I,J))
      FFLUXEN       = (F1 + FV(1,K,I,J))*Y1 + (F2 + FV(2,K,I,J))*Y2
      FFLUXWS       = (F3 + FV(3,K,I,J))*Y3 + (F4 + FV(4,K,I,J))*Y4
      GFLUXEN       = (G1 + GV(1,K,I,J))*X1 + (G2 + GV(2,K,I,J))*X2
      GFLUXWS       = (G3 + GV(3,K,I,J))*X3 + (G4 + GV(4,K,I,J))*X4
      RESID         = (FFLUXEN + FFLUXWS) - (GFLUXEN + GFLUXWS)
      RES(K,I,J)      = RESID / AREA(I,J)
17    CONTINUE
16    CONTINUE
C
C ---   LOWER WALL STEP CORNER CELL EVALUATION
C
      I             = NSX
      J             = 1
      X1            = X(I+1,J+1) - X(I+1,J)
      X2            = X(I,J+1)   - X(I+1,J+1)
      X3            = X(I,J)     - X(I,J+1)
      X4            = X(I+1,J)   - X(I,J)
      Y1            = Y(I+1,J+1) - Y(I+1,J)
      Y2            = Y(I,J+1)   - Y(I+1,J+1)
```

```
      Y3            = Y(I,J)      - Y(I,J+1)
      Y4            = Y(I+1,J)    - Y(I,J)
      DO   15 K     = 1 , IEQ
      F1            = .5 * (FI(K,I+1,J) + FI(K,I,J))
      F2            = .5 * (FI(K,I,J+1) + FI(K,I,J))
      F3            = 0.0
      IF(K.EQ.2)F3  = PRES(NSX,1)
      F4            = 0.0
      IF(K.EQ.2)F4  = PRES(NSX,1)
      G1            = .5* (GI(K,I+1,J) + GI(K,I,J))
      G2            = .5* (GI(K,I,J+1) + GI(K,I,J))
      G3            = 0.0
      IF(K.EQ.3)G3  = PRES(NSX,1)
      G4            = 0.0
      IF(K.EQ.3)G4  = PRES(NSX,1)
      FFLUXEN       = (F1 + FV(1,K,I,J))*Y1 + (F2 + FV(2,K,I,J))*Y2
      FFLUXWS       = (F3 + FV(3,K,I,J))*Y3 + (F4 + FV(4,K,I,J))*Y4
      GFLUXEN       = (G1 + GV(1,K,I,J))*X1 + (G2 + GV(2,K,I,J))*X2
      GFLUXWS       = (G3 + GV(3,K,I,J))*X3 + (G4 + GV(4,K,I,J))*X4
      RESID         = (FFLUXEN + FFLUXWS) - (GFLUXEN + GFLUXWS)
      RES(K,I,J)      = RESID / AREA(I,J)
15    CONTINUE
C
C       ----------------------------------
C ---   UPPER BOUNDARY WALL CELL EVALUATION
C       ----------------------------------
C
      J             = NYY
      DO   5  I     = 2 , NXXX
      X1            = X(I+1,J+1) - X(I+1,J)
      X2            = X(I,J+1)    - X(I+1,J+1)
      X3            = X(I,J)      - X(I,J+1)
      X4            = X(I+1,J)    - X(I,J)
      Y1            = Y(I+1,J+1) - Y(I+1,J)
      Y2            = Y(I,J+1)    - Y(I+1,J+1)
      Y3            = Y(I,J)      - Y(I,J+1)
      Y4            = Y(I+1,J)    - Y(I,J)
      DO   6  K     = 1 , IEQ
      F1            = .5 * (FI(K,I+1,J) + FI(K,I,J))
      F2            = 0.0
      IF(K.EQ.2)F2  = PRES(I,NYY)
      F3            = .5 * (FI(K,I-1,J) + FI(K,I,J))
      F4            = .5 * (FI(K,I,J-1) + FI(K,I,J))
      G1            = .5 * (GI(K,I+1,J) + GI(K,I,J))
      G2            = 0.0
      IF(K.EQ.3)G2  = PRES(I,NYY)
      G3            = .5 * (GI(K,I-1,J) + GI(K,I,J))
      G4            = .5 * (GI(K,I,J-1) + GI(K,I,J))
      FFLUXEN       = (F1 + FV(1,K,I,J))*Y1 + (F2 + FV(2,K,I,J))*Y2
      FFLUXWS       = (F3 + FV(3,K,I,J))*Y3 + (F4 + FV(4,K,I,J))*Y4
```

```
      GFLUXEN      = (G1 + GV(1,K,I,J))*X1 + (G2 + GV(2,K,I,J))*X2
      GFLUXWS      = (G3 + GV(3,K,I,J))*X3 + (G4 + GV(4,K,I,J))*X4
      RESID        = (FFLUXEN + FFLUXWS) - (GFLUXEN + GFLUXWS)
      RES(K,I,J)     = RESID / AREA(I,J)
6     CONTINUE
5     CONTINUE
C
C       -----------------------
C ---   EXIT BOUNDARY EVALUATION
C       -----------------------
C
C
C ---   UPPER CORNER CELL
C
C
      I            = NXX
      J            = NYY
      X1           = X(I+1,J+1) - X(I+1,J)
      X2           = X(I,J+1)   - X(I+1,J+1)
      X3           = X(I,J)     - X(I,J+1)
      X4           = X(I+1,J)   - X(I,J)
      Y1           = Y(I+1,J+1) - Y(I+1,J)
      Y2           = Y(I,J+1)   - Y(I+1,J+1)
      Y3           = Y(I,J)     - Y(I,J+1)
      Y4           = Y(I+1,J)   - Y(I,J)
      DO   20  K   = 1 , IEQ
      F1           = FI(K,I,J)
      F2           = 0.0
      IF(K.EQ.2)F2 = PRES(I,J)
      F3           = .5 * (FI(K,I-1,J) + FI(K,I,J))
      F4           = .5 * (FI(K,I,J-1) + FI(K,I,J))
      G1           = GI(K,I,J)
      G2           = 0.0
      IF(K.EQ.3)G2 = PRES(I,J)
      G3           = .5 * (GI(K,I-1,J) + GI(K,I,J))
      G4           = .5 * (GI(K,I,J-1) + GI(K,I,J))
      FFLUXEN      = (F1 + FV(1,K,I,J))*Y1 + (F2 + FV(2,K,I,J))*Y2
      FFLUXWS      = (F3 + FV(3,K,I,J))*Y3 + (F4 + FV(4,K,I,J))*Y4
      GFLUXEN      = (G1 + GV(1,K,I,J))*X1 + (G2 + GV(2,K,I,J))*X2
      GFLUXWS      = (G3 + GV(3,K,I,J))*X3 + (G4 + GV(4,K,I,J))*X4
      RESID        = (FFLUXEN + FFLUXWS) - (GFLUXEN + GFLUXWS)
      RES(K,I,J)   = RESID / AREA(I,J)
20    CONTINUE
C
C ---  LOWER EXIT CORNER ( X = NXX , Y = 1 )
C
      I            = NXX
      J            = 1
      X1           = X(I+1,J+1) - X(I+1,J)
      X2           = X(I,J+1)   - X(I+1,J+1)
```

```
      X3            = X(I,J)     - X(I,J+1)
      X4            = X(I+1,J)   - X(I,J)
      Y1            = Y(I+1,J+1) - Y(I+1,J)
      Y2            = Y(I,J+1)   - Y(I+1,J+1)
      Y3            = Y(I,J)     - Y(I,J+1)
      Y4            = Y(I+1,J)   - Y(I,J)
      DO   21  K    = 1 , IEQ
      F1            = FI(K,I,J)
      F2            = .5 * (FI(K,I,J+1) + FI(K,I,J))
      F3            = .5 * (FI(K,I-1,J) + FI(K,I,J))
      F4            = 0.0
      IF(K.EQ.2)F4  = PRES(NXX,J)
      G1            = GI(K,I,J)
      G2            = .5 * (GI(K,I,J+1) + GI(K,I,J))
      G3            = .5 * (GI(K,I-1,J) + GI(K,I,J))
      G4            = 0.0
      IF(K.EQ.3)G4  = PRES(NXX,J)
      FFLUXEN       = (F1 + FV(1,K,I,J))*Y1 + (F2 + FV(2,K,I,J))*Y2
      FFLUXWS       = (F3 + FV(3,K,I,J))*Y3 + (F4 + FV(4,K,I,J))*Y4
      GFLUXEN       = (G1 + GV(1,K,I,J))*X1 + (G2 + GV(2,K,I,J))*X2
      GFLUXWS       = (G3 + GV(3,K,I,J))*X3 + (G4 + GV(4,K,I,J))*X4
      RESID         = (FFLUXEN + FFLUXWS) - (GFLUXEN + GFLUXWS)
      RES(K,I,J)    = RESID / AREA(I,J)
21    CONTINUE
C
C ---    VERTICAL EXIT CELL EVALUATION ( X = NXX , Y = 2 , NYYY )
C
      I             = NXX
      DO   22  J    = 2 , NYYY
      X1            = X(I+1,J+1) - X(I+1,J)
      X2            = X(I,J+1)   - X(I+1,J+1)
      X3            = X(I,J)     - X(I,J+1)
      X4            = X(I+1,J)   - X(I,J)
      Y1            = Y(I+1,J+1) - Y(I+1,J)
      Y2            = Y(I,J+1)   - Y(I+1,J+1)
      Y3            = Y(I,J)     - Y(I,J+1)
      Y4            = Y(I+1,J)   - Y(I,J)
      DO   22  K    = 1 , IEQ
      F1            =  FI(K,I,J)
      F2            = .5 * (FI(K,I,J+1) + FI(K,I,J))
      F3            = .5 * (FI(K,I-1,J) + FI(K,I,J))
      F4            = .5 * (FI(K,I,J-1) + FI(K,I,J))
      G1            =  GI(K,I,J)
      G2            = .5 * (GI(K,I,J+1) + GI(K,I,J))
      G3            = .5 * (GI(K,I-1,J) + GI(K,I,J))
      G4            = .5 * (GI(K,I,J-1) + GI(K,I,J))
      FFLUXEN       = (F1 + FV(1,K,I,J))*Y1 + (F2 + FV(2,K,I,J))*Y2
      FFLUXWS       = (F3 + FV(3,K,I,J))*Y3 + (F4 + FV(4,K,I,J))*Y4
      GFLUXEN       = (G1 + GV(1,K,I,J))*X1 + (G2 + GV(2,K,I,J))*X2
      GFLUXWS       = (G3 + GV(3,K,I,J))*X3 + (G4 + GV(4,K,I,J))*X4
```

```
      RESID         = (FFLUXEN + FFLUXWS) - (GFLUXEN + GFLUXWS)
      RES(K,I,J)    = RESID / AREA(I,J)
22    CONTINUE
      RETURN
      END
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      SUBROUTINE DAMPX
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C --- X COMPONENT OF POST SPLIT SMOOTHING OPERATOR
C
      COMMON/VAR0/U(8,53,33,2),FI(8,53,33),GI(8,53,33),AH(8,53,33)
      COMMON/VAR00/FV(4,8,53,33),GV(4,8,53,33)
      COMMON/VAR1/UVEL(53,33),VVEL(53,33),PRES(53,33),TEMP(53,33)
      COMMON/VAR111/YH2(53,33),YO2(53,33),YH2O(53,33),YOH(53,33)
      COMMON/VAR2/DEN(53,33),SOUND(53,33),AINTE(53,33),AMACH(53,33)
      COMMON/VAR3/ENTHP(53,33),VIS(53,33),YYN2(53,33),CPND(53,33)
      COMMON/VAR4/DXX,DYY,X(55,35),Y(55,35),AREA(53,33)
      COMMON/VAR5/DT(53,33)
      COMMON/VAR6/NX,NXX,NXXX,NY,NYY,NYYY,IRES,IEQ
      COMMON/VAR7/IVIS,NITER,NOITER
      COMMON/VAR8/P1,T1,AM1,VISL,U1,V1,AK1,CV,CP,R,GAMA,DEN1,E1
      COMMON/VAR9/P11,C11,E11
      COMMON/VAR10/DFH2,DFO2,DFH2O,DFOH,DFN2
      COMMON/VAR11/CPH2,CPO2,CPH2O,CPOH,CPN2,CVH2,CVO2,CVH2O,CVOH,CVN2
      COMMON/VAR12/CONH2,CONO2,CONH2O,CONOH,CONN2
      COMMON/VAR13/COND,CFL,DCOFF,AL,VELO1
      COMMON/VAR14/RES(8,53,33)
      COMMON/VAR15/ALPHA(4)
      COMMON/VAR16/DXE(4,53,33),DXN(4,53,33),DXW(4,53,33),DXS(4,53,33)
      COMMON/VAR17/DYE(4,53,33),DYN(4,53,33),DYW(4,53,33),DYS(4,53,33)
      COMMON/VAR18/DD(2,8,53,33)
      COMMON/VAR20/NSX,NSXB,NSXA,NSY,NSYB,NSYA
      COMMON/VAR21/ACOM
      COMMON/VAR22/TRIGTEMP,PHI
      COMMON/VAR23/REN,PR,FACT1,LAMB,SDIFF
      DO 1  J     = 1 , NYY
      DO 1  I     = 2 , NXXX
      DD(1,1,I,J)  = U(1,I+1,J,2) + U(1,I-1,J,2) -2.0*U(1,I,J,2)
      DD(1,2,I,J)  = U(2,I+1,J,2) + U(2,I-1,J,2) -2.0*U(2,I,J,2)
      DD(1,3,I,J)  = U(3,I+1,J,2) + U(3,I-1,J,2) -2.0*U(3,I,J,2)
      DD(1,4,I,J)  = U(4,I+1,J,2) + U(4,I-1,J,2) -2.0*U(4,I,J,2)
      DD(1,5,I,J)  = U(5,I+1,J,2) + U(5,I-1,J,2) -2.0*U(5,I,J,2)
      DD(1,6,I,J)  = U(6,I+1,J,2) + U(6,I-1,J,2) -2.0*U(6,I,J,2)
      DD(1,7,I,J)  = U(7,I+1,J,2) + U(7,I-1,J,2) -2.0*U(7,I,J,2)
      DD(1,8,I,J)  = U(8,I+1,J,2) + U(8,I-1,J,2) -2.0*U(8,I,J,2)
1     CONTINUE
      DO 3 J      = 1 , NYY
C
C ---  GHOST POINT EVALUATION OF U'S - BASED UPON FREESTREAM
```

```
C ---                                          BOUNDARY CONDITIONS
C
C
C ---    X = 1
C
       DD(1,1,1,J)     = U(1,2,J,2)   - U(1,1,J,2)
       DD(1,2,1,J)     = U(2,2,J,2)   - U(2,1,J,2)
       DD(1,3,1,J)     = U(3,2,J,2)   - U(3,1,J,2)
       DD(1,4,1,J)     = U(4,2,J,2)   - U(4,1,J,2)
       DD(1,5,1,J)     = U(5,2,J,2)   - U(5,1,J,2)
       DD(1,6,1,J)     = U(6,2,J,2)   - U(6,1,J,2)
       DD(1,7,1,J)     = U(7,2,J,2)   - U(7,1,J,2)
       DD(1,8,1,J)     = U(8,2,J,2)   - U(8,1,J,2)
C
C ---    X = NXX
C
       DD(1,1,NXX,J)  = U(1,NXXX,J,2)   - U(1,NXX,J,2)
       DD(1,2,NXX,J)  = U(2,NXXX,J,2)   - U(2,NXX,J,2)
       DD(1,3,NXX,J)  = U(3,NXXX,J,2)   - U(3,NXX,J,2)
       DD(1,4,NXX,J)  = U(4,NXXX,J,2)   - U(4,NXX,J,2)
       DD(1,5,NXX,J)  = U(5,NXXX,J,2)   - U(5,NXX,J,2)
       DD(1,6,NXX,J)  = U(6,NXXX,J,2)   - U(6,NXX,J,2)
       DD(1,7,NXX,J)  = U(7,NXXX,J,2)   - U(7,NXX,J,2)
       DD(1,8,NXX,J)  = U(8,NXXX,J,2)   - U(8,NXX,J,2)
3      CONTINUE
       RETURN
       END
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
       SUBROUTINE DAMPY
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C ---  Y COMPONENT OF POST SPLIT SMOOTHING OPERATOR
C
       COMMON/VAR0/U(8,53,33,2),FI(8,53,33),GI(8,53,33),AH(8,53,33)
       COMMON/VAR00/FV(4,8,53,33),GV(4,8,53,33)
       COMMON/VAR1/UVEL(53,33),VVEL(53,33),PRES(53,33),TEMP(53,33)
       COMMON/VAR111/YH2(53,33),YO2(53,33),YH2O(53,33),YOH(53,33)
       COMMON/VAR2/DEN(53,33),SOUND(53,33),AINTE(53,33),AMACH(53,33)
       COMMON/VAR3/ENTHP(53,33),VIS(53,33),YYN2(53,33),CPND(53,33)
       COMMON/VAR4/DXX,DYY,X(55,35),Y(55,35),AREA(53,33)
       COMMON/VAR5/DT(53,33)
       COMMON/VAR6/NX,NXX,NXXX,NY,NYY,NYYY,IRES,IEQ
       COMMON/VAR7/IVIS,NITER,NOITER
       COMMON/VAR8/P1,T1,AM1,VISL,U1,V1,AK1,CV,CP,R,GAMA,DEN1,E1
       COMMON/VAR9/P11,C11,E11
       COMMON/VAR10/DFH2,DFO2,DFH2O,DFOH,DFN2
       COMMON/VAR11/CPH2,CPO2,CPH2O,CPOH,CPN2,CVH2,CVO2,CVH2O,CVOH,CVN2
       COMMON/VAR12/CONH2,CONO2,CONH2O,CONOH,CONN2
       COMMON/VAR13/COND,CFL,DCOFF,AL,VELO1
       COMMON/VAR14/RES(8,53,33)
```

```
      COMMON/VAR15/ALPHA(4)
      COMMON/VAR16/DXE(4,53,33),DXN(4,53,33),DXW(4,53,33),DXS(4,53,33)
      COMMON/VAR17/DYE(4,53,33),DYN(4,53,33),DYW(4,53,33),DYS(4,53,33)
      COMMON/VAR18/DD(2,8,53,33)
      COMMON/VAR20/NSX,NSXB,NSXA,NSY,NSYB,NSYA
      COMMON/VAR21/ACOM
      COMMON/VAR22/TRIGTEMP,PHI
      COMMON/VAR23/REN,PR,FACT1,LAMB,SDIFF
      DO 1  J     = 2 , NYYY
      DO 1  I     = 1 , NXX
      DD(2,1,I,J)  = U(1,I,J+1,2) + U(1,I,J-1,2) -2.0*U(1,I,J,2)
      DD(2,2,I,J)  = U(2,I,J+1,2) + U(2,I,J-1,2) -2.0*U(2,I,J,2)
      DD(2,3,I,J)  = U(3,I,J+1,2) + U(3,I,J-1,2) -2.0*U(3,I,J,2)
      DD(2,4,I,J)  = U(4,I,J+1,2) + U(4,I,J-1,2) -2.0*U(4,I,J,2)
      DD(2,5,I,J)  = U(5,I,J+1,2) + U(5,I,J-1,2) -2.0*U(5,I,J,2)
      DD(2,6,I,J)  = U(6,I,J+1,2) + U(6,I,J-1,2) -2.0*U(6,I,J,2)
      DD(2,7,I,J)  = U(7,I,J+1,2) + U(7,I,J-1,2) -2.0*U(7,I,J,2)
      DD(2,8,I,J)  = U(8,I,J+1,2) + U(8,I,J-1,2) -2.0*U(8,I,J,2)
1     CONTINUE
      DO 3 I      = 1 , NXX
C
C ---   GHOST POINT EVALUATION OF U'S - BASED UPON REFLECTION
C ---                                 BOUNDARY CONDITIONS
C
C
C ---   Y = 1
C
      U1GHOST    =   DEN(I,1)
C
CCCCCCCCCCCCCCCCC  NOTE + SIGN HERE SHOULD BE - VE FOR VISCOUS CAL
C
      U2GHOST    =   DEN(I,1)*UVEL(I,1)
      U3GHOST    = - DEN(I,1)*VVEL(I,1)
      U4GHOST    =   DEN(I,1)*AINTE(I,1)
      U5GHOST    =   DEN(I,1)*YH2(I,1)
      U6GHOST    =   DEN(I,1)*YO2(I,1)
      U7GHOST    =   DEN(I,1)*YH2O(I,1)
      U8GHOST    =   DEN(I,1)*YYN2(I,1)
      DD(2,1,I,1)  = U(1,I,2,2) + U1GHOST - 2.0*U(1,I,1,2)
      DD(2,2,I,1)  = U(2,I,2,2) + U2GHOST - 2.0*U(2,I,1,2)
      DD(2,3,I,1)  = U(3,I,2,2) + U3GHOST - 2.0*U(3,I,1,2)
      DD(2,4,I,1)  = U(4,I,2,2) + U4GHOST - 2.0*U(4,I,1,2)
      DD(2,5,I,1)  = U(5,I,2,2) + U5GHOST - 2.0*U(5,I,1,2)
      DD(2,6,I,1)  = U(6,I,2,2) + U6GHOST - 2.0*U(6,I,1,2)
      DD(2,7,I,1)  = U(7,I,2,2) + U7GHOST - 2.0*U(7,I,1,2)
      DD(2,8,I,1)  = U(8,I,2,2) + U8GHOST - 2.0*U(8,I,1,2)
C
C ---   Y = NYY
C
      U1GHOST    =   DEN(I,NYY)
```

```
      U2GHOST     =   DEN(I,NYY)*UVEL(I,NYY)
      U3GHOST     = - DEN(I,NYY)*VVEL(I,NYY)
      U4GHOST     =   DEN(I,NYY)*AINTE(I,NYY)
      U5GHOST     =   DEN(I,NYY)*YH2(I,NYY)
      U6GHOST     =   DEN(I,NYY)*YO2(I,NYY)
      U7GHOST     =   DEN(I,NYY)*YH2O(I,NYY)
      U8GHOST     =   DEN(I,NYY)*YYN2(I,NYY)
      DD(2,1,I,NYY)  = U(1,I,NYYY,2) + U1GHOST - 2.0*U(1,I,NYY,2)
      DD(2,2,I,NYY)  = U(2,I,NYYY,2) + U2GHOST - 2.0*U(2,I,NYY,2)
      DD(2,3,I,NYY)  = U(3,I,NYYY,2) + U3GHOST - 2.0*U(3,I,NYY,2)
      DD(2,4,I,NYY)  = U(4,I,NYYY,2) + U4GHOST - 2.0*U(4,I,NYY,2)
      DD(2,5,I,NYY)  = U(5,I,NYYY,2) + U5GHOST - 2.0*U(5,I,NYY,2)
      DD(2,6,I,NYY)  = U(6,I,NYYY,2) + U6GHOST - 2.0*U(6,I,NYY,2)
      DD(2,7,I,NYY)  = U(7,I,NYYY,2) + U7GHOST - 2.0*U(7,I,NYY,2)
      DD(2,8,I,NYY)  = U(8,I,NYYY,2) + U8GHOST - 2.0*U(8,I,NYY,2)
3     CONTINUE
C
C ---    ADD SECOND ORDER DAMPING CORRECTIONS( X AND Y) TO  U
C
      DO 2 J      = 1 , NYY
      DO 2 I      = 1 , NXX
      U(1,I,J,2) = U(1,I,J,2)   + DCOFF * (DD(1,1,I,J) + DD(2,1,I,J))
      U(2,I,J,2) = U(2,I,J,2)   + DCOFF * (DD(1,2,I,J) + DD(2,2,I,J))
      U(3,I,J,2) = U(3,I,J,2)   + DCOFF * (DD(1,3,I,J) + DD(2,3,I,J))
      U(4,I,J,2) = U(4,I,J,2)   + DCOFF * (DD(1,4,I,J) + DD(2,4,I,J))
      U(5,I,J,2) = U(5,I,J,2)   + DCOFF * (DD(1,5,I,J) + DD(2,5,I,J))
      U(6,I,J,2) = U(6,I,J,2)   + DCOFF * (DD(1,6,I,J) + DD(2,6,I,J))
      U(7,I,J,2) = U(7,I,J,2)   + DCOFF * (DD(1,7,I,J) + DD(2,7,I,J))
      U(8,I,J,2) = U(8,I,J,2)   + DCOFF * (DD(1,8,I,J) + DD(2,8,I,J))
C
C ---     COMPUTE NEW PRIMATIVE QUANTITIES
C
      DEN(I,J)    = U(1,I,J,2)
      ODEN        = 1.0 / DEN(I,J)
      UVEL(I,J)   = U(2,I,J,2) * ODEN
      VVEL(I,J)   = U(3,I,J,2) * ODEN
      AINTE(I,J)  = U(4,I,J,2) * ODEN
      YH2(I,J)    = U(5,I,J,2) * ODEN
      YO2(I,J)    = U(6,I,J,2) * ODEN
      YH2O(I,J)   = U(7,I,J,2) * ODEN
      YYN2(I,J)   = U(8,I,J,2) * ODEN
C
      UOH         = 1.0 - YH2(I,J) - YO2(I,J) - YH2O(I,J) - YYN2(I,J)
      CP          = ( YH2(I,J)*CPH2 + YO2(I,J)* CPO2 +YH2O(I,J)*CPH2O
     1              + UOH*CPOH      + YYN2(I,J)* CPN2)
      CV          = ( YH2(I,J)*CVH2 + YO2(I,J)* CVO2 +YH2O(I,J)*CVH2O
     1              + UOH*CVOH      + YYN2(I,J)* CVN2)
      R           = CP - CV
      GAMA        = CP / CV
      DHEATF      = YH2(I,J)*DFH2 + YO2(I,J)*DFO2 + YH2O(I,J)*DFH2O
```

```
     1                + UOH*DFOH    + YYN2(I,J)    *DFN2
       VELO         = UVEL(I,J)**2 + VVEL(I,J)**2
       TEMP(I,J)    = VELO1**2/(CV*T1)*(AINTE(I,J) - .5*VELO
     1                - DHEATF / VELO1**2 )
       TEMP(I,J)    = ABS(TEMP(I,J))
       SOUND(I,J)   = SQRT(T1*R*GAMA*TEMP(I,J))/VELO1
       AMACH(I,J)   = SQRT(VELO)/SOUND(I,J)
       PRES(I,J)    = (T1*R/VELO1**2)*DEN(I,J)*TEMP(I,J)
       ENTHP(I,J)   = CP*T1/VELO1**2*TEMP(I,J)+.5*VELO
     1                + DHEATF / VELO1**2
2      CONTINUE
       RETURN
       END
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
       SUBROUTINE DAMPXST
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
       COMMON/VAR0/U(8,53,33,2),FI(8,53,33),GI(8,53,33),AH(8,53,33)
       COMMON/VAR00/FV(4,8,53,33),GV(4,8,53,33)
       COMMON/VAR1/UVEL(53,33),VVEL(53,33),PRES(53,33),TEMP(53,33)
       COMMON/VAR111/YH2(53,33),YO2(53,33),YH2O(53,33),YOH(53,33)
       COMMON/VAR2/DEN(53,33),SOUND(53,33),AINTE(53,33),AMACH(53,33)
       COMMON/VAR3/ENTHP(53,33),VIS(53,33),YYN2(53,33),CPND(53,33)
       COMMON/VAR4/DXX,DYY,X(55,35),Y(55,35),AREA(53,33)
       COMMON/VAR5/DT(53,33)
       COMMON/VAR6/NX,NXX,NXXX,NY,NYY,NYYY,IRES,IEQ
       COMMON/VAR7/IVIS,NITER,NOITER
       COMMON/VAR8/P1,T1,AM1,VISL,U1,V1,AK1,CV,CP,R,GAMA,DEN1,E1
       COMMON/VAR9/P11,C11,E11
       COMMON/VAR10/DFH2,DFO2,DFH2O,DFOH,DFN2
       COMMON/VAR11/CPH2,CPO2,CPH2O,CPOH,CPN2,CVH2,CVO2,CVH2O,CVOH,CVN2
       COMMON/VAR12/CONH2,CONO2,CONH2O,CONOH,CONN2
       COMMON/VAR13/COND,CFL,DCOFF,AL,VELO1
       COMMON/VAR14/RES(8,53,33)
       COMMON/VAR15/ALPHA(4)
       COMMON/VAR16/DXE(4,53,33),DXN(4,53,33),DXW(4,53,33),DXS(4,53,33)
       COMMON/VAR17/DYE(4,53,33),DYN(4,53,33),DYW(4,53,33),DYS(4,53,33)
       COMMON/VAR18/DD(2,8,53,33)
       COMMON/VAR20/NSX,NSXB,NSXA,NSY,NSYB,NSYA
       COMMON/VAR21/ACOM
       COMMON/VAR22/TRIGTEMP,PHI
       COMMON/VAR23/REN,PR,FACT1,LAMB,SDIFF
C
C ---  X COMPONENT OF POST SPLIT SMOOTHING OPERATOR
C
C       INCLUDE 'COMNS.INC'
       DO 1  J     = 1 , NYY
       DO 1  I     = 2 , NXXX
       IF(I.LT.NSX.AND.J.LT.NSY) GO TO 1
       DD(1,1,I,J)  = U(1,I+1,J,2) + U(1,I-1,J,2) -2.0*U(1,I,J,2)
       DD(1,2,I,J)  = U(2,I+1,J,2) + U(2,I-1,J,2) -2.0*U(2,I,J,2)
```

```
      DD(1,3,I,J)  = U(3,I+1,J,2) + U(3,I-1,J,2) -2.0*U(3,I,J,2)
      DD(1,4,I,J)  = U(4,I+1,J,2) + U(4,I-1,J,2) -2.0*U(4,I,J,2)
      DD(1,5,I,J)  = U(5,I+1,J,2) + U(5,I-1,J,2) -2.0*U(5,I,J,2)
      DD(1,6,I,J)  = U(6,I+1,J,2) + U(6,I-1,J,2) -2.0*U(6,I,J,2)
      DD(1,7,I,J)  = U(7,I+1,J,2) + U(7,I-1,J,2) -2.0*U(7,I,J,2)
      DD(1,8,I,J)  = U(8,I+1,J,2) + U(8,I-1,J,2) -2.0*U(8,I,J,2)
1     CONTINUE
      DO 3 J     = 1 , NYY
C
C ---  GHOST POINT EVALUATION OF U'S - BASED UPON FREESTREAM
C ---                                 BOUNDARY CONDITIONS
C
C
C ---  X = 1
C
      IF(J.LT.NSY) GO TO 4
      DD(1,1,1,J)    = U(1,2,J,2)  - U(1,1,J,2)
      DD(1,2,1,J)    = U(2,2,J,2)  - U(2,1,J,2)
      DD(1,3,1,J)    = U(3,2,J,2)  - U(3,1,J,2)
      DD(1,4,1,J)    = U(4,2,J,2)  - U(4,1,J,2)
      DD(1,5,1,J)    = U(5,2,J,2)  - U(5,1,J,2)
      DD(1,6,1,J)    = U(6,2,J,2)  - U(6,1,J,2)
      DD(1,7,1,J)    = U(7,2,J,2)  - U(7,1,J,2)
      DD(1,8,1,J)    = U(8,2,J,2)  - U(8,1,J,2)
      GO TO 5
4     CONTINUE
      U1GHOST      =   DEN(NSX,J)
      U2GHOST      = - DEN(NSX,J)*UVEL(NSX,J)
      U3GHOST      = - DEN(NSX,J)*VVEL(NSX,J)
      U4GHOST      =   DEN(NSX,J)*AINTE(NSX,J)
      U5GHOST      =   DEN(NSX,J)*YH2(NSX,J)
      U6GHOST      =   DEN(NSX,J)*YO2(NSX,J)
      U7GHOST      =   DEN(NSX,J)*YH2O(NSX,J)
      U8GHOST      =   DEN(NSX,J)*YYN2(NSX,J)
      DD(1,1,NSX,J)  = U(1,NSXA,J,2) +U1GHOST -2.0*U(1,NSX,J,2)
      DD(1,2,NSX,J)  = U(2,NSXA,J,2) +U2GHOST -2.0*U(2,NSX,J,2)
      DD(1,3,NSX,J)  = U(3,NSXA,J,2) +U3GHOST -2.0*U(3,NSX,J,2)
      DD(1,4,NSX,J)  = U(4,NSXA,J,2) +U4GHOST -2.0*U(4,NSX,J,2)
      DD(1,5,NSX,J)  = U(5,NSXA,J,2) +U5GHOST -2.0*U(5,NSX,J,2)
      DD(1,6,NSX,J)  = U(6,NSXA,J,2) +U6GHOST -2.0*U(6,NSX,J,2)
      DD(1,7,NSX,J)  = U(7,NSXA,J,2) +U7GHOST -2.0*U(7,NSX,J,2)
      DD(1,8,NSX,J)  = U(8,NSXA,J,2) +U8GHOST -2.0*U(8,NSX,J,2)
5     CONTINUE
C
C ---  X = NXX
C
      DD(1,1,NXX,J)  = U(1,NXXX,J,2)  - U(1,NXX,J,2)
      DD(1,2,NXX,J)  = U(2,NXXX,J,2)  - U(2,NXX,J,2)
      DD(1,3,NXX,J)  = U(3,NXXX,J,2)  - U(3,NXX,J,2)
      DD(1,4,NXX,J)  = U(4,NXXX,J,2)  - U(4,NXX,J,2)
```

```
      DD(1,5,NXX,J)  = U(5,NXXX,J,2)  - U(5,NXX,J,2)
      DD(1,6,NXX,J)  = U(6,NXXX,J,2)  - U(6,NXX,J,2)
      DD(1,7,NXX,J)  = U(7,NXXX,J,2)  - U(7,NXX,J,2)
      DD(1,8,NXX,J)  = U(8,NXXX,J,2)  - U(8,NXX,J,2)
3     CONTINUE
      DO 2 J     = 1 , NYY
      DO 2 I     = 1 , NXX
      IF(I.LT.NSX.AND.J.LT.NSY) GO TO 2
      IF(I.LT.NSX) GO TO 2
      U(1,I,J,2) = U(1,I,J,2)   + DCOFF * DD(1,1,I,J)
      U(2,I,J,2) = U(2,I,J,2)   + DCOFF * DD(1,2,I,J)
      U(3,I,J,2) = U(3,I,J,2)   + DCOFF * DD(1,3,I,J)
      U(4,I,J,2) = U(4,I,J,2)   + DCOFF * DD(1,4,I,J)
      U(5,I,J,2) = U(5,I,J,2)   + DCOFF * DD(1,5,I,J)
      U(6,I,J,2) = U(6,I,J,2)   + DCOFF * DD(1,6,I,J)
      U(7,I,J,2) = U(7,I,J,2)   + DCOFF * DD(1,7,I,J)
      U(8,I,J,2) = U(8,I,J,2)   + DCOFF * DD(1,8,I,J)
2     CONTINUE
      RETURN
      END
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      SUBROUTINE DAMPYST
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      COMMON/VAR0/U(8,53,33,2),FI(8,53,33),GI(8,53,33),AH(8,53,33)
      COMMON/VAR00/FV(4,8,53,33),GV(4,8,53,33)
      COMMON/VAR1/UVEL(53,33),VVEL(53,33),PRES(53,33),TEMP(53,33)
      COMMON/VAR111/YH2(53,33),YO2(53,33),YH2O(53,33),YOH(53,33)
      COMMON/VAR2/DEN(53,33),SOUND(53,33),AINTE(53,33),AMACH(53,33)
      COMMON/VAR3/ENTHP(53,33),VIS(53,33),YYN2(53,33),CPND(53,33)
      COMMON/VAR4/DXX,DYY,X(55,35),Y(55,35),AREA(53,33)
      COMMON/VAR5/DT(53,33)
      COMMON/VAR6/NX,NXX,NXXX,NY,NYY,NYYY,IRES,IEQ
      COMMON/VAR7/IVIS,NITER,NOITER
      COMMON/VAR8/P1,T1,AM1,VISL,U1,V1,AK1,CV,CP,R,GAMA,DEN1,E1
      COMMON/VAR9/P11,C11,E11
      COMMON/VAR10/DFH2,DFO2,DFH2O,DFOH,DFN2
      COMMON/VAR11/CPH2,CPO2,CPH2O,CPOH,CPN2,CVH2,CVO2,CVH2O,CVOH,CVN2
      COMMON/VAR12/CONH2,CONO2,CONH2O,CONOH,CONN2
      COMMON/VAR13/COND,CFL,DCOFF,AL,VELO1
      COMMON/VAR14/RES(8,53,33)
      COMMON/VAR15/ALPHA(4)
      COMMON/VAR16/DXE(4,53,33),DXN(4,53,33),DXW(4,53,33),DXS(4,53,33)
      COMMON/VAR17/DYE(4,53,33),DYN(4,53,33),DYW(4,53,33),DYS(4,53,33)
      COMMON/VAR18/DD(2,8,33,33)
      COMMON/VAR20/NSX,NSXB,NSXA,NSY,NSYB,NSYA
      COMMON/VAR21/ACOM
      COMMON/VAR22/TRIGTEMP,PHI
      COMMON/VAR23/REN,PR,FACT1,LAMB,SDIFF
C
C ---  Y COMPONENT OF POST SPLIT SMOOTHING OPERATOR
```

```
C
C        INCLUDE 'COMNS.INC'
         DO 1  J     = 2 , NYYY
         DO 1  I     = 1 , NXX
         IF(I.LT.NSX.AND.J.LT.NSY) GO TO 1
         DD(2,1,I,J)  = U(1,I,J+1,2) + U(1,I,J-1,2) -2.0*U(1,I,J,2)
         DD(2,2,I,J)  = U(2,I,J+1,2) + U(2,I,J-1,2) -2.0*U(2,I,J,2)
         DD(2,3,I,J)  = U(3,I,J+1,2) + U(3,I,J-1,2) -2.0*U(3,I,J,2)
         DD(2,4,I,J)  = U(4,I,J+1,2) + U(4,I,J-1,2) -2.0*U(4,I,J,2)
         DD(2,5,I,J)  = U(5,I,J+1,2) + U(5,I,J-1,2) -2.0*U(5,I,J,2)
         DD(2,6,I,J)  = U(6,I,J+1,2) + U(6,I,J-1,2) -2.0*U(6,I,J,2)
         DD(2,7,I,J)  = U(7,I,J+1,2) + U(7,I,J-1,2) -2.0*U(7,I,J,2)
         DD(2,8,I,J)  = U(8,I,J+1,2) + U(8,I,J-1,2) -2.0*U(8,I,J,2)
1        CONTINUE
         DO 3 I      = 1 , NXX
C
C ---     GHOST POINT EVALUATION OF U'S - BASED UPON REFLECTION
C ---                                     BOUNDARY CONDITIONS
C
C
C ---     Y = 1
C
         IF(I.LT.NSX) GO TO 4
         U1GHOST     =   DEN(I,1)
         U2GHOST     = - DEN(I,1)*UVEL(I,1)
         U3GHOST     = - DEN(I,1)*VVEL(I,1)
         U4GHOST     =   DEN(I,1)*AINTE(I,1)
         U5GHOST     =   DEN(I,1)*YH2(I,1)
         U6GHOST     =   DEN(I,1)*YO2(I,1)
         U7GHOST     =   DEN(I,1)*YH2O(I,1)
         U8GHOST     =   DEN(I,1)*YYN2(I,1)
         DD(2,1,I,1)  = U(1,I,2,2) + U1GHOST - 2.0*U(1,I,1,2)
         DD(2,2,I,1)  = U(2,I,2,2) + U2GHOST - 2.0*U(2,I,1,2)
         DD(2,3,I,1)  = U(3,I,2,2) + U3GHOST - 2.0*U(3,I,1,2)
         DD(2,4,I,1)  = U(4,I,2,2) + U4GHOST - 2.0*U(4,I,1,2)
         DD(2,5,I,1)  = U(5,I,2,2) + U5GHOST - 2.0*U(5,I,1,2)
         DD(2,6,I,1)  = U(6,I,2,2) + U6GHOST - 2.0*U(6,I,1,2)
         DD(2,7,I,1)  = U(7,I,2,2) + U7GHOST - 2.0*U(7,I,1,2)
         DD(2,8,I,1)  = U(8,I,2,2) + U8GHOST - 2.0*U(8,I,1,2)
         GO TO 5
4        CONTINUE
         U1GHOST     =   DEN(I,NSY)
         U2GHOST     = - DEN(I,NSY)*UVEL(I,NSY)
         U3GHOST     = - DEN(I,NSY)*VVEL(I,NSY)
         U4GHOST     =   DEN(I,NSY)*AINTE(I,NSY)
         U5GHOST     =   DEN(I,NSY)*YH2(I,NSY)
         U6GHOST     =   DEN(I,NSY)*YO2(I,NSY)
         U7GHOST     =   DEN(I,NSY)*YH2O(I,NSY)
         U8GHOST     =   DEN(I,NSY)*YYN2(I,NSY)
         DD(2,1,I,NSY)  = U(1,I,NSYA,2) + U1GHOST - 2.0*U(1,I,NSY,2)
```

```
      DD(2,2,I,NSY)  = U(2,I,NSYA,2) + U2GHOST - 2.0*U(2,I,NSY,2)
      DD(2,3,I,NSY)  = U(3,I,NSYA,2) + U3GHOST - 2.0*U(3,I,NSY,2)
      DD(2,4,I,NSY)  = U(4,I,NSYA,2) + U4GHOST - 2.0*U(4,I,NSY,2)
      DD(2,5,I,NSY)  = U(5,I,NSYA,2) + U5GHOST - 2.0*U(5,I,NSY,2)
      DD(2,6,I,NSY)  = U(6,I,NSYA,2) + U6GHOST - 2.0*U(6,I,NSY,2)
      DD(2,7,I,NSY)  = U(7,I,NSYA,2) + U7GHOST - 2.0*U(7,I,NSY,2)
      DD(2,8,I,NSY)  = U(8,I,NSYA,2) + U8GHOST - 2.0*U(8,I,NSY,2)
5     CONTINUE
C
C ---   Y = NYY
C
      U1GHOST    =    DEN(I,NYY)
      U2GHOST    =    DEN(I,NYY)*UVEL(I,NYY)
      U3GHOST    = -  DEN(I,NYY)*VVEL(I,NYY)
      U4GHOST    =    DEN(I,NYY)*AINTE(I,NYY)
      U5GHOST    =    DEN(I,NYY)*YH2(I,NYY)
      U6GHOST    =    DEN(I,NYY)*YO2(I,NYY)
      U7GHOST    =    DEN(I,NYY)*YH2O(I,NYY)
      U8GHOST    =    DEN(I,NYY)*YYN2(I,NYY)
      DD(2,1,I,NYY)  = U(1,I,NYYY,2) + U1GHOST - 2.0*U(1,I,NYY,2)
      DD(2,2,I,NYY)  = U(2,I,NYYY,2) + U2GHOST - 2.0*U(2,I,NYY,2)
      DD(2,3,I,NYY)  = U(3,I,NYYY,2) + U3GHOST - 2.0*U(3,I,NYY,2)
      DD(2,4,I,NYY)  = U(4,I,NYYY,2) + U4GHOST - 2.0*U(4,I,NYY,2)
      DD(2,5,I,NYY)  = U(5,I,NYYY,2) + U5GHOST - 2.0*U(5,I,NYY,2)
      DD(2,6,I,NYY)  = U(6,I,NYYY,2) + U6GHOST - 2.0*U(6,I,NYY,2)
      DD(2,7,I,NYY)  = U(7,I,NYYY,2) + U7GHOST - 2.0*U(7,I,NYY,2)
      DD(2,8,I,NYY)  = U(8,I,NYYY,2) + U8GHOST - 2.0*U(8,I,NYY,2)
3     CONTINUE
      DO 2 J      = 1 , NYY
      DO 2 I      = 1 , NXX
      IF(I.LT.NSX.AND.J.LT.NSY) GO TO 2
      IF(I.LT.NSX) GO TO 2
      U(1,I,J,2) = U(1,I,J,2)   + DCOFF * DD(2,1,I,J)
      U(2,I,J,2) = U(2,I,J,2)   + DCOFF * DD(2,2,I,J)
      U(3,I,J,2) = U(3,I,J,2)   + DCOFF * DD(2,3,I,J)
      U(4,I,J,2) = U(4,I,J,2)   + DCOFF * DD(2,4,I,J)
      U(5,I,J,2) = U(5,I,J,2)   + DCOFF * DD(2,5,I,J)
      U(6,I,J,2) = U(6,I,J,2)   + DCOFF * DD(2,6,I,J)
      U(7,I,J,2) = U(7,I,J,2)   + DCOFF * DD(2,7,I,J)
      U(8,I,J,2) = U(8,I,J,2)   + DCOFF * DD(2,8,I,J)
C
      DEN(I,J)     = U(1,I,J,2)
      ODEN         = 1.0 / DEN(I,J)
      UVEL(I,J)    = U(2,I,J,2) * ODEN
      VVEL(I,J)    = U(3,I,J,2) * ODEN
      AINTE(I,J)   = U(4,I,J,2) * ODEN
      YH2(I,J)     = U(5,I,J,2) * ODEN
      YO2(I,J)     = U(6,I,J,2) * ODEN
      YH2O(I,J)    = U(7,I,J,2) * ODEN
      YYN2(I,J)    = U(8,I,J,2) * ODEN
```

```
C
      UOH          = 1.0 - YH2(I,J) - YO2(I,J) - YH2O(I,J) - YYN2(I,J)
      CP           = ( YH2(I,J)*CPH2 + YO2(I,J)* CPO2 +YH2O(I,J)*CPH2O
     1               + UOH*CPOH       + YYN2(I,J)* CPN2)
      CV           = ( YH2(I,J)*CVH2 + YO2(I,J)* CVO2 +YH2O(I,J)*CVH2O
     1               + UOH*CVOH       + YYN2(I,J)* CVN2)
      R            = CP - CV
      GAMA         = CP / CV
      DHEATF       = YH2(I,J)*DFH2 + YO2(I,J)*DFO2 + YH2O(I,J)*DFH2O
     1               + UOH*DFOH     + YYN2(I,J)    *DFN2
      VELO         = UVEL(I,J)**2 + VVEL(I,J)**2
      TEMP(I,J)    = VELO1**2/(CV*T1)*(AINTE(I,J) - .5*VELO
     1               - DHEATF / VELO1**2 )
      TEMP(I,J)    = ABS(TEMP(I,J))
      SOUND(I,J)   = SQRT(T1*R*GAMA*TEMP(I,J))/VELO1
      AMACH(I,J)   = SQRT(VELO)/SOUND(I,J)
      PRES(I,J)    = (T1*R/VELO1**2)*DEN(I,J)*TEMP(I,J)
      ENTHP(I,J)   = CP*T1/VELO1**2*TEMP(I,J)+.5*VELO
     1               + DHEATF / VELO1**2
2     CONTINUE
      RETURN
      END
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      SUBROUTINE STAB
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C ---  DETERMINE THE BIGGEST TIME STEPS THE SOLUTION CAN BE ADVANCED
C ---  AND STABILITY BE MAINTAINED
C
      COMMON/VAR0/U(8,53,33,2),FI(8,53,33),GI(8,53,33),AH(8,53,33)
      COMMON/VAR00/FV(4,8,53,33),GV(4,8,53,33)
      COMMON/VAR1/UVEL(53,33),VVEL(53,33),PRES(53,33),TEMP(53,33)
      COMMON/VAR111/YH2(53,33),YO2(53,33),YH2O(53,33),YOH(53,33)
      COMMON/VAR2/DEN(53,33),SOUND(53,33),AINTE(53,33),AMACH(53,33)
      COMMON/VAR3/ENTHP(53,33),VIS(53,33),YYN2(53,33),CPND(53,33)
      COMMON/VAR4/DXX,DYY,X(55,35),Y(55,35),AREA(53,33)
      COMMON/VAR5/DT(53,33)
      COMMON/VAR6/NX,NXX,NXXX,NY,NYY,NYYY,IRES,IEQ
      COMMON/VAR7/IVIS,NITER,NOITER
      COMMON/VAR8/P1,T1,AM1,VISL,U1,V1,AK1,CV,CP,R,GAMA,DEN1,E1
      COMMON/VAR9/P11,C11,E11
      COMMON/VAR10/DFH2,DFO2,DFH2O,DFOH,DFN2
      COMMON/VAR11/CPH2,CPO2,CPH2O,CPOH,CPN2,CVH2,CVO2,CVH2O,CVOH,CVN2
      COMMON/VAR12/CONH2,CONO2,CONH2O,CONOH,CONN2
      COMMON/VAR13/COND,CFL,DCOFF,AL,VELO1
      COMMON/VAR14/RES(8,53,33)
      COMMON/VAR15/ALPHA(4)
      COMMON/VAR16/DXE(4,53,33),DXN(4,53,33),DXW(4,53,33),DXS(4,53,33)
      COMMON/VAR17/DYE(4,53,33),DYN(4,53,33),DYW(4,53,33),DYS(4,53,33)
      COMMON/VAR18/DD(2,8,53,33)
```

```
      COMMON/VAR20/NSX,NSXB,NSXA,NSY,NSYB,NSYA
      COMMON/VAR21/ACOM
      COMMON/VAR22/TRIGTEMP,PHI
      COMMON/VAR23/REN,PR,FACT1,LAMB,SDIFF
      CVIS    = .75
      DO 1 IY = 1 , NYY
      DO 1 IX = 1 , NXX
C
      DXXX = ABS(X(IX+1,IY)-X(IX,IY))
      DYYY = ABS(Y(IX,IY+1)-Y(IX,IY))
      DMX        = ABS(UVEL(IX,IY)) + SOUND(IX,IY)
      DMY        = ABS(VVEL(IX,IY)) + SOUND(IX,IY)
C
      A1         = (DMX/DXXX + DMY/DYYY) / CFL
      A2         = VIS(IX,IY)/(DXXX**2*DEN(IX,IY))
     &           + VIS(IX,IY)/(DYYY**2*DEN(IX,IY))
      DT(IX,IY) = CVIS / (A1 + A2)
      IF(IVIS.EQ.1) DT(IX,IY) = 1.0 / A1
1     CONTINUE
      RETURN
      END
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      SUBROUTINE PROPINV
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C ---  COMPUTE THE INVISCID CONTRIBUTIONS TO THE
C ---  " F "  AND " G " FLUX VECTORS
C
      COMMON/VAR0/U(8,53,33,2),FI(8,53,33),GI(8,53,33),AH(8,53,33)
      COMMON/VAR00/FV(4,8,53,33),GV(4,8,53,33)
      COMMON/VAR1/UVEL(53,33),VVEL(53,33),PRES(53,33),TEMP(53,33)
      COMMON/VAR111/YH2(53,33),YO2(53,33),YH2O(53,33),YOH(53,33)
      COMMON/VAR2/DEN(53,33),SOUND(53,33),AINTE(53,33),AMACH(53,33)
      COMMON/VAR3/ENTHP(53,33),VIS(53,33),YYN2(53,33),CPND(53,33)
      COMMON/VAR4/DXX,DYY,X(55,35),Y(55,35),AREA(53,33)
      COMMON/VAR5/DT(53,33)
      COMMON/VAR6/NX,NXX,NXXX,NY,NYY,NYYY,IRES,IEQ
      COMMON/VAR7/IVIS,NITER,NOITER
      COMMON/VAR8/P1,T1,AM1,VISL,U1,V1,AK1,CV,CP,R,GAMA,DEN1,E1
      COMMON/VAR9/P11,C11,E11
      COMMON/VAR10/DFH2,DFO2,DFH2O,DFOH,DFN2
      COMMON/VAR11/CPH2,CPO2,CPH2O,CPOH,CPN2,CVH2,CVO2,CVH2O,CVOH,CVN2
      COMMON/VAR12/CONH2,CONO2,CONH2O,CONOH,CONN2
      COMMON/VAR13/COND,CFL,DCOFF,AL,VELO1
      COMMON/VAR14/RES(8,53,33)
      COMMON/VAR15/ALPHA(4)
      COMMON/VAR16/DXE(4,53,33),DXN(4,53,33),DXW(4,53,33),DXS(4,53,33)
      COMMON/VAR17/DYE(4,53,33),DYN(4,53,33),DYW(4,53,33),DYS(4,53,33)
      COMMON/VAR18/DD(2,8,53,33)
      COMMON/VAR20/NSX,NSXB,NSXA,NSY,NSYB,NSYA
```

```
      COMMON/VAR21/ACOM
      COMMON/VAR22/TRIGTEMP,PHI
      COMMON/VAR23/REN,PR,FACT1,LAMB,SDIFF
      DO 1  JJ  = 1 , NYY
      DO 1  II  = 1 , NXX
C
C ---  EVALUATE THE INVISCID PARTS OF "F" AND "G"
C
C ---  EVALUATE THE F TERMS
C
      FI(1,II,JJ) = DEN(II,JJ)*UVEL(II,JJ)
      FI(2,II,JJ) = DEN(II,JJ)*UVEL(II,JJ)**2 + PRES(II,JJ)
      FI(3,II,JJ) = DEN(II,JJ)*UVEL(II,JJ)*VVEL(II,JJ)
      FI(4,II,JJ) = DEN(II,JJ)*UVEL(II,JJ)*AINTE(II,JJ)
     1              + UVEL(II,JJ)*PRES(II,JJ)
      FI(5,II,JJ) = DEN(II,JJ)*UVEL(II,JJ)*YH2(II,JJ)
      FI(6,II,JJ) = DEN(II,JJ)*UVEL(II,JJ)*YO2(II,JJ)
      FI(7,II,JJ) = DEN(II,JJ)*UVEL(II,JJ)*YH2O(II,JJ)
      FI(8,II,JJ) = DEN(II,JJ)*UVEL(II,JJ)*YYN2(II,JJ)
C
C      EVALUATE THE G TERMS
C
      GI(1,II,JJ) = DEN(II,JJ)*VVEL(II,JJ)
      GI(2,II,JJ) = DEN(II,JJ)*UVEL(II,JJ)*VVEL(II,JJ)
      GI(3,II,JJ) = DEN(II,JJ)*VVEL(II,JJ)**2 + PRES(II,JJ)
      GI(4,II,JJ) = DEN(II,JJ)*VVEL(II,JJ)*AINTE(II,JJ)
     1              + VVEL(II,JJ)*PRES(II,JJ)
      GI(5,II,JJ) = DEN(II,JJ)*VVEL(II,JJ)*YH2(II,JJ)
      GI(6,II,JJ) = DEN(II,JJ)*VVEL(II,JJ)*YO2(II,JJ)
      GI(7,II,JJ) = DEN(II,JJ)*VVEL(II,JJ)*YH2O(II,JJ)
      GI(8,II,JJ) = DEN(II,JJ)*VVEL(II,JJ)*YYN2(II,JJ)
1     CONTINUE
      RETURN
      END
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      SUBROUTINE PROPV
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C --- COMPUTE THE VISCOUS CONTRIBUTIONS TO THE
C --- " F "  AND " G " FLUX VECTORS
C
      COMMON/VAR0/U(8,53,33,2),FI(8,53,33),GI(8,53,33),AH(8,53,33)
      COMMON/VAR00/FV(4,8,53,33),GV(4,8,53,33)
      COMMON/VAR1/UVEL(53,33),VVEL(53,33),PRES(53,33),TEMP(53,33)
      COMMON/VAR111/YH2(53,33),YO2(53,33),YH2O(53,33),YOH(53,33)
      COMMON/VAR2/DEN(53,33),SOUND(53,33),AINTE(53,33),AMACH(53,33)
      COMMON/VAR3/ENTHP(53,33),VIS(53,33),YYN2(53,33),CPND(53,33)
      COMMON/VAR4/DXX,DYY,X(55,35),Y(55,35),AREA(53,33)
      COMMON/VAR5/DT(53,33)
      COMMON/VAR6/NX,NXX,NXXX,NY,NYY,NYYY,IRES,IEQ
```

```
      COMMON/VAR7/IVIS,NITER,NOITER
      COMMON/VAR8/P1,T1,AM1,VISL,U1,V1,AK1,CV,CP,R,GAMA,DEN1,E1
      COMMON/VAR9/P11,C11,E11
      COMMON/VAR10/DFH2,DFO2,DFH2O,DFOH,DFN2
      COMMON/VAR11/CPH2,CPO2,CPH2O,CPOH,CPN2,CVH2,CVO2,CVH2O,CVOH,CVN2
      COMMON/VAR12/CONH2,CONO2,CONH2O,CONOH,CONN2
      COMMON/VAR13/COND,CFL,DCOFF,AL,VELO1
      COMMON/VAR14/RES(8,53,33)
      COMMON/VAR15/ALPHA(4)
      COMMON/VAR16/DXE(4,53,33),DXN(4,53,33),DXW(4,53,33),DXS(4,53,33)
      COMMON/VAR17/DYE(4,53,33),DYN(4,53,33),DYW(4,53,33),DYS(4,53,33)
      COMMON/VAR18/DD(2,8,53,33)
      COMMON/VAR20/NSX,NSXB,NSXA,NSY,NSYB,NSYA
      COMMON/VAR21/ACOM
      COMMON/VAR22/TRIGTEMP,PHI
      COMMON/VAR23/REN,PR,FACT1,LAMB,SDIFF
C
C ---  THE REST OF THIS SUBROUTINE WILL BE DEVOTED TO DETERMINING THE
C ---
C ---  VISCOUS GRADIENT TERMS IE:      \ D Z        D Z \
C ---                                  \ ---   ,    --- \
C ---                                  \ D X        D Y \
C ---
C ---  WHERE Z COULD BE ANY QUANTITY IE: U , V , T , YI
C ---  NOTE Y1 = YH2 , Y2 = YO2 AND Y3 = YH2O
C
C
C                 ---------------
C ---             \  EAST FACE  \
C                 ---------------
C
      DO 20  J    = 2 , NYYY
      DO 20  I    = 2 , NXXX
      AV          = .5 * (AREA(I,J) + AREA(I+1,J))
C
C ---  U-VELOCITY
C
      UE          = UVEL(I+1,J)
      UW          = UVEL(I,J)
      UN          = .25 * (UVEL(I+1,J) + UVEL(I+1,J+1)
     1                +  UVEL(I,J+1) + UVEL(I,J))
      US          = .25 * (UVEL(I+1,J) + UVEL(I,J)
     1                +  UVEL(I,J-1) + UVEL(I+1,J-1))
C
C ---  V-VELOCITY
C
      VE          = VVEL(I+1,J)
      VW          = VVEL(I,J)
      VN          = .25 * (VVEL(I+1,J) + VVEL(I+1,J+1)
     1                +  VVEL(I,J+1) + VVEL(I+1,J))
```

```
      VS          = .25 * (VVEL(I+1,J) + VVEL(I,J)
     1                  +  VVEL(I,J-1) + VVEL(I+1,J-1))
C
C ---   TEMPERATURE
C
      TE          = TEMP(I+1,J)
      TW          = TEMP(I,J)
      TN          = .25 * (TEMP(I+1,J) + TEMP(I+1,J+1)
     1                  +  TEMP(I,J+1) + TEMP(I+1,J))
      TS          = .25 * (TEMP(I+1,J) + TEMP(I,J)
     1                  +  TEMP(I,J-1) + TEMP(I+1,J-1))
C
C ---   YH2
C
      Y1E          = YH2(I+1,J)
      Y1W          = YH2(I,J)
      Y1N          = .25 * (YH2(I+1,J) + YH2(I+1,J+1)
     1                   +   YH2(I,J+1) + YH2(I+1,J))
      Y1S          = .25 * (YH2(I+1,J) + YH2(I,J)
     1                   +   YH2(I,J-1) + YH2(I+1,J-1))
C
C ---   YO2
C
      Y2E          = YO2(I+1,J)
      Y2W          = YO2(I,J)
      Y2N          = .25 * (YO2(I+1,J) + YO2(I+1,J+1)
     1                   +   YO2(I,J+1) + YO2(I+1,J))
      Y2S          = .25 * (YO2(I+1,J) + YO2(I,J)
     1                   +   YH2(I,J-1) + YO2(I+1,J-1))
C
C ---   YH2O
C
      Y3E          = YH2O(I+1,J)
      Y3W          = YH2O(I,J)
      Y3N          = .25 * (YH2O(I+1,J) + YH2O(I+1,J+1)
     1                   +   YH2O(I,J+1) + YH2O(I+1,J))
      Y3S          = .25 * (YH2O(I+1,J) + YH2O(I,J)
     1                   +   YH2O(I,J-1) + YH2O(I+1,J-1))
C
C ---   YYN2
C
      Y4E          = YYN2(I+1,J)
      Y4W          = YYN2(I,J)
      Y4N          = .25 * (YYN2(I+1,J) + YYN2(I+1,J+1)
     1                   +   YYN2(I,J+1) + YYN2(I+1,J))
      Y4S          = .25 * (YYN2(I+1,J) + YYN2(I,J)
     1                   +   YYN2(I,J-1) + YYN2(I+1,J-1))
C
C ---   CALCULATE THE GRADIENT TERMS
C
```

```
C
C ---    X  GRADIENTS
C
        DUDX        = (UE * DYE(1,I,J) + UN * DYN(1,I,J)
     1                + UW * DYW(1,I,J) + US * DYS(1,I,J)) / AV
        DVDX        = (VE * DYE(1,I,J) + VN * DYN(1,I,J)
     1                + VW * DYW(1,I,J) + VS * DYS(1,I,J)) / AV
        DTDX        = (TE * DYE(1,I,J) + TN * DYN(1,I,J)
     1                + TW * DYW(1,I,J) + TS * DYS(1,I,J)) / AV
        DY1DX       = (Y1E * DYE(1,I,J) + Y1N * DYN(1,I,J)
     1                + Y1W * DYW(1,I,J) + Y1S * DYS(1,I,J)) / AV
        DY2DX       = (Y2E * DYE(1,I,J) + Y2N * DYN(1,I,J)
     1                + Y2W * DYW(1,I,J) + Y2S * DYS(1,I,J)) / AV
        DY3DX       = (Y3E * DYE(1,I,J) + Y3N * DYN(1,I,J)
     1                + Y3W * DYW(1,I,J) + Y3S * DYS(1,I,J)) / AV
        DY4DX       = (Y4E * DYE(1,I,J) + Y4N * DYN(1,I,J)
     1                + Y4W * DYW(1,I,J) + Y4S * DYS(1,I,J)) / AV
        DY5DX       = -DY1DX - DY2DX - DY3DX - DY4DX
C
C ---    Y  GRADIENTS
C
        DUDY        = - (UE * DXE(1,I,J) + UN * DXN(1,I,J)
     1                  + UW * DXW(1,I,J) + US * DXS(1,I,J)) / AV
        DVDY        = - (VE * DXE(1,I,J) + VN * DXN(1,I,J)
     1                  + VW * DXW(1,I,J) + VS * DXS(1,I,J)) / AV
        DTDY        = - (TE * DXE(1,I,J) + TN * DXN(1,I,J)
     1                  + TW * DXW(1,I,J) + TS * DXS(1,I,J)) / AV
        DY1DY       = - (Y1E * DXE(1,I,J) + Y1N * DXN(1,I,J)
     1                  + Y1W * DXW(1,I,J) + Y1S * DXS(1,I,J)) / AV
        DY2DY       = - (Y2E * DXE(1,I,J) + Y2N * DXN(1,I,J)
     1                  + Y2W * DXW(1,I,J) + Y2S * DXS(1,I,J)) / AV
        DY3DY       = -(Y3E * DXE(1,I,J) + Y3N * DXN(1,I,J)
     1                  + Y3W * DXW(1,I,J) + Y3S * DXS(1,I,J)) / AV
        DY4DY       = -(Y4E * DXE(1,I,J) + Y4N * DXN(1,I,J)
     1                  + Y4W * DXW(1,I,J) + Y4S * DXS(1,I,J)) / AV
        DY5DY       = - DY1DY - DY2DY - DY3DY - DY4DY
C
C ---    COMPUTE THE FULL SHEAR STRESS TERMS
C
        TXX         = - LAMB * VIS(I,J) * (DUDX + DVDY)
                      - 2.0  * VIS(I,J) * DUDX / REN
        TXY         = - VIS(I,J) / REN  * (DUDY + DVDX)
        TYY         = - LAMB * VIS(I,J) * (DUDX + DVDY)
                      - 2.0  * VIS(I,J) * DVDY / REN
C
C ---    COMPUTE THE VISCOUS CONTRIBUTIONS TO "F" AND "G"
C
        FV(1,1,I,J)  = 0.0
        FV(1,2,I,J)  = TXX
        FV(1,3,I,J)  = TXY
```

```
      FV(1,4,I,J)  =   .5 * (UVEL(I,J) + UVEL(I+1,J))*TXX
     1                 + .5 * (VVEL(I,J) + VVEL(I+1,J))*TXY
     2                 - SDIFF * FACT1 * DTDX
     3                 - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2
     4                 + CPH2 * T1 * TEMP(I,J)) *  DY1DX
     5                 - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFO2
     6                 + CPO2 * T1 * TEMP(I,J)) *  DY2DX
     7                 - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2O
     8                 + CPH2O * T1 * TEMP(I,J)) *  DY3DX
     9                 - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2
     10                + CPN2 * T1 * TEMP(I,J)) *  DY4DX
     11                - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFOH
     12                + CPOH * T1 * TEMP(I,J)) *  DY5DX
      FV(1,5,I,J)  = - SDIFF * DY1DX
      FV(1,6,I,J)  = - SDIFF * DY2DX
      FV(1,7,I,J)  = - SDIFF * DY3DX
      FV(1,8,I,J)  = - SDIFF * DY4DX
      GV(1,1,I,J)  = 0.0
      GV(1,2,I,J)  = TXY
      GV(1,3,I,J)  = TYY
      GV(1,4,I,J)  =   .5 * (VVEL(I,J) + VVEL(I+1,J))*TYY
     1                 + .5 * (UVEL(I,J) + UVEL(I+1,J))*TXY
     2                 - SDIFF * FACT1 * DTDY
     3                 - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2
     4                 + CPH2 * T1 * TEMP(I,J)) *  DY1DY
     5                 - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFO2
     6                 + CPO2 * T1 * TEMP(I,J)) *  DY2DY
     7                 - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2O
     8                 + CPH2O * T1 * TEMP(I,J)) *  DY3DY
     9                 - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2
     10                + CPN2 * T1 * TEMP(I,J)) *  DY4DY
     11                - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFOH
     12                + CPOH * T1 * TEMP(I,J)) *  DY5DY
      GV(1,5,I,J)  = - SDIFF * DY1DY
      GV(1,6,I,J)  = - SDIFF * DY2DY
      GV(1,7,I,J)  = - SDIFF * DY3DY
      GV(1,8,I,J)  = - SDIFF * DY4DY
20    CONTINUE
C
C                 ------------------
C ---             \   NORTH  FACE   \
C                 ------------------
C
      DO 30  J    = 2 , NYYY
      DO 30  I    = 2 , NXXX
      AV          = .5 * (AREA(I,J) + AREA(I,J+1))
C
C ---   U-VELOCITY
C
      UE          = .25 * (UVEL(I+1,J) + UVEL(I+1,J+1)
```

```
     1                   +  UVEL(I,J+1) + UVEL(I,J))
       UN           = UVEL(I,J+1)
       UW           = .25 * (UVEL(I,J+1) + UVEL(I-1,J+1)
     1                   +  UVEL(I-1,J) + UVEL(I,J))
       US           = UVEL(I,J)
C
C ---   V-VELOCITY
C
       VE           = .25 * (VVEL(I+1,J) + VVEL(I+1,J+1)
     1                   +  VVEL(I,J+1) + VVEL(I,J))
       VN           = VVEL(I,J+1)
       VW           = .25 * (VVEL(I,J+1) + VVEL(I-1,J+1)
     1                   +  VVEL(I-1,J) + VVEL(I,J))
       VS           = VVEL(I,J)
C
C ---   TEMPERATURE
C
       TE           = .25 * (TEMP(I+1,J) + TEMP(I+1,J+1)
     1                   +  TEMP(I,J+1) + TEMP(I,J))
       TN           = TEMP(I,J+1)
       TW           = .25 * (TEMP(I,J+1) + TEMP(I-1,J+1)
     1                   +  TEMP(I-1,J) + TEMP(I,J))
       TS           = TEMP(I,J)
C
C ---   YH2
C
       Y1E           = .25 * (YH2(I+1,J) + YH2(I+1,J+1)
     1                   +   YH2(I,J+1) + YH2(I,J))
       Y1N           = YH2(I,J+1)
       Y1W           = .25 * (YH2(I,J+1) + YH2(I-1,J+1)
     1                   +   YH2(I-1,J) + YH2(I,J))
       Y1S           = YH2(I,J)
C
C ---   YO2
C
       Y2E           = .25 * (YO2(I+1,J) + YO2(I+1,J+1)
     1                   +   YO2(I,J+1) + YO2(I,J))
       Y2N           = YO2(I,J+1)
       Y2W           = .25 * (YO2(I,J+1) + YO2(I-1,J+1)
     1                  +    YO2(I-1,J) + YO2(I,J))
       Y2S           = YO2(I,J)
C
C ---   YH2O
C
       Y3E           = .25 * (YH2O(I+1,J) + YH2O(I+1,J+1)
     1                   +   YH2O(I,J+1) + YH2O(I,J))
       Y3N           = YH2O(I,J+1)
       Y3W           = .25 * (YH2O(I,J+1) + YH2O(I-1,J+1)
     1                   +   YH2O(I-1,J) + YH2O(I,J))
       Y3S           = YH2O(I,J)
```

```
C
C ---   YYN2
C
      Y4E          = .25 * (YYN2(I+1,J) + YYN2(I+1,J+1)
     1                 +   YYN2(I,J+1) + YYN2(I,J))
      Y4N          = YYN2(I,J+1)
      Y4W          = .25 * (YYN2(I,J+1) + YYN2(I-1,J+1)
     1                 +   YYN2(I-1,J) + YYN2(I,J))
      Y4S          = YYN2(I,J)
C
C ---   CALCULATE THE GRADIENT TERMS
C
C
C ---   X  GRADIENTS
C
      DUDX         = (UE * DYE(2,I,J) + UN * DYN(2,I,J)
     1              + UW * DYW(2,I,J) + US * DYS(2,I,J)) / AV
      DVDX         = (VE * DYE(2,I,J) + VN * DYN(2,I,J)
     1              + VW * DYW(2,I,J) + VS * DYS(2,I,J)) / AV
      DTDX         = (TE * DYE(2,I,J) + TN * DYN(2,I,J)
     1              + TW * DYW(2,I,J) + TS * DYS(2,I,J)) / AV
      DY1DX        = (Y1E * DYE(2,I,J) + Y1N * DYN(2,I,J)
     1              + Y1W * DYW(2,I,J) + Y1S * DYS(2,I,J)) / AV
      DY2DX        = (Y2E * DYE(2,I,J) + Y2N * DYN(2,I,J)
     1              + Y2W * DYW(2,I,J) + Y2S * DYS(2,I,J)) / AV
      DY3DX        = (Y3E * DYE(2,I,J) + Y3N * DYN(2,I,J)
     1              + Y3W * DYW(2,I,J) + Y3S * DYS(2,I,J)) / AV
      DY4DX        = (Y4E * DYE(2,I,J) + Y4N * DYN(2,I,J)
     1              + Y4W * DYW(2,I,J) + Y4S * DYS(2,I,J)) / AV
      DY5DX        = -DY1DX - DY2DX - DY3DX - DY4DX
C
C ---   Y  GRADIENTS
C
      DUDY         = - (UE * DXE(2,I,J) + UN * DXN(2,I,J)
     1                + UW * DXW(2,I,J) + US * DXS(2,I,J)) / AV
      DVDY         = - (VE * DXE(2,I,J) + VN * DXN(2,I,J)
     1                + VW * DXW(2,I,J) + VS * DXS(2,I,J)) / AV
      DTDY         = - (TE * DXE(2,I,J) + TN * DXN(2,I,J)
     1                + TW * DXW(2,I,J) + TS * DXS(2,I,J)) / AV
      DY1DY        = - (Y1E * DXE(2,I,J) + Y1N * DXN(2,I,J)
     1                + Y1W * DXW(2,I,J) + Y1S * DXS(2,I,J)) / AV
      DY2DY        = - (Y2E * DXE(2,I,J) + Y2N * DXN(2,I,J)
     1                + Y2W * DXW(2,I,J) + Y2S * DXS(2,I,J)) / AV
      DY3DY        = - (Y3E * DXE(2,I,J) + Y3N * DXN(2,I,J)
     1                + Y3W * DXW(2,I,J) + Y3S * DXS(2,I,J)) / AV
      DY4DY        = - (Y4E * DXE(2,I,J) + Y4N * DXN(2,I,J)
     1                + Y4W * DXW(2,I,J) + Y4S * DXS(2,I,J)) / AV
      DY5DY        = - DY1DY - DY2DY - DY3DY - DY4DY
C
C ---   COMPUTE THE FULL SHEAR STRESS TERMS
```

```
C
      TXX         = - LAMB * VIS(I,J) * (DUDX + DVDY)
                    - 2.0  * VIS(I,J) * DUDX / REN
      TXY         = - VIS(I,J) / REN  * (DUDY + DVDX)
      TYY         = - LAMB * VIS(I,J) * (DUDX + DVDY)
                    - 2.0  * VIS(I,J) * DVDY / REN
C
C ---   COMPUTE THE VISCOUS CONTRIBUTIONS TO "F" AND "G"
C
      FV(2,1,I,J)  = 0.0
      FV(2,2,I,J)  = TXX
      FV(2,3,I,J)  = TXY
      FV(2,4,I,J)  =   .5 * (UVEL(I,J) + UVEL(I,J+1))*TXX
     1                + .5 * (VVEL(I,J) + VVEL(I,J+1))*TXY
     2                - SDIFF * FACT1 * DTDX
     3                - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2
     4                + CPH2 * T1 * TEMP(I,J)) *  DY1DX
     5                - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFO2
     6                + CPO2 * T1 * TEMP(I,J)) *  DY2DX
     7                - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2O
     8                + CPH2O * T1 * TEMP(I,J)) *  DY3DX
     9                - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2
     10               + CPN2 * T1 * TEMP(I,J)) *  DY4DX
     11               - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFOH
     12               + CPOH * T1 * TEMP(I,J)) *  DY5DX
      FV(2,5,I,J)  = - SDIFF * DY1DX
      FV(2,6,I,J)  = - SDIFF * DY2DX
      FV(2,7,I,J)  = - SDIFF * DY3DX
      FV(2,8,I,J)  = - SDIFF * DY8DX
      GV(2,1,I,J)  = 0.0
      GV(2,2,I,J)  = TXY
      GV(2,3,I,J)  = TYY
      GV(2,4,I,J)  =   .5 * (VVEL(I,J) + VVEL(I,J+1))*TYY
     1                + .5 * (UVEL(I,J) + UVEL(I,J+1))*TXY
     2                - SDIFF * FACT1 * DTDY
     3                - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2
     4                + CPH2 * T1 * TEMP(I,J)) *  DY1DY
     5                - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFO2
     6                + CPO2 * T1 * TEMP(I,J)) *  DY2DY
     7                - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2O
     8                + CPH2O * T1 * TEMP(I,J)) *  DY3DY
     9                - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2
     10               + CPN2 * T1 * TEMP(I,J)) *  DY4DY
     11               - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFOH
     12               + CPOH * T1 * TEMP(I,J)) *  DY5DY
      GV(2,5,I,J)  = - SDIFF * DY1DY
      GV(2,6,I,J)  = - SDIFF * DY2DY
      GV(2,7,I,J)  = - SDIFF * DY3DY
      GV(2,8,I,J)  = - SDIFF * DY8DY
30    CONTINUE
```

```
C
C
C                    ---------------
C ---                \  WEST FACE  \
C                    ---------------
C
      DO 40  J    = 2 , NYYY
      DO 40  I    = 2 , NXXX
      AV          = .5 * (AREA(I,J) + AREA(I-1,J))
C
C ---   U-VELOCITY
C
      UE          = UVEL(I,J)
      UW          = UVEL(I-1,J)
      UN          = .25 * (UVEL(I,J)     + UVEL(I,J+1)
     1                 +  UVEL(I-1,J+1) + UVEL(I-1,J))
      US          = .25 * (UVEL(I,J)     + UVEL(I-1,J)
     1                 +  UVEL(I-1,J-1) + UVEL(I,J-1))
C
C ---   V-VELOCITY
C
      VE          = VVEL(I,J)
      VW          = VVEL(I-1,J)
      VN          = .25 * (VVEL(I,J)     + VVEL(I,J+1)
     1                 +  VVEL(I-1,J+1) + VVEL(I-1,J))
      VS          = .25 * (VVEL(I,J)     + VVEL(I-1,J)
     1                 +  VVEL(I-1,J-1) + VVEL(I,J-1))
C
C ---   TEMPERATURE
C
      TE          = TEMP(I,J)
      TW          = TEMP(I-1,J)
      TN          = .25 * (TEMP(I,J)     + TEMP(I,J+1)
     1                 +  TEMP(I-1,J+1) + TEMP(I-1,J))
      TS          = .25 * (TEMP(I,J)     + TEMP(I-1,J)
     1                 +  TEMP(I-1,J-1) + TEMP(I,J-1))
C
C ---   YH2
C
      Y1E          = YH2(I,J)
      Y1W          = YH2(I-1,J)
      Y1N          = .25 * (YH2(I,J)     + YH2(I,J+1)
     1                 +   YH2(I-1,J+1) + YH2(I-1,J))
      Y1S          = .25 * (YH2(I,J)     + YH2(I-1,J)
     1                 +   YH2(I-1,J-1) + YH2(I,J-1))
C
C ---   YO2
C
      Y2E          = YO2(I,J)
      Y2W          = YO2(I-1,J)
```

```
      Y2N           = .25 * (YO2(I,J)      + YO2(I,J+1)
     1                  +   YO2(I-1,J+1) + YO2(I-1,J))
      Y2S           = .25 * (YO2(I,J)      + YO2(I-1,J)
     1                  +   YO2(I-1,J-1) + YO2(I,J-1))
C
C ---   YH2O
C
      Y3E           = YH2O(I,J)
      Y3W           = YH2O(I-1,J)
      Y3N           = .25 * (YH2O(I,J)      + YH2O(I,J+1)
     1                  +   YH2O(I-1,J+1) + YH2O(I-1,J))
      Y3S           = .25 * (YH2O(I,J)      + YH2O(I-1,J)
     1                  +   YH2O(I-1,J-1) + YH2O(I,J-1))
C
C ---   YYN2
C
      Y4E           = YYN2(I,J)
      Y4W           = YYN2(I-1,J)
      Y4N           = .25 * (YYN2(I,J)      + YYN2(I,J+1)
     1                  +   YYN2(I-1,J+1) + YYN2(I-1,J))
      Y4S           = .25 * (YYN2(I,J)      + YYN2(I-1,J)
     1                  +   YYN2(I-1,J-1) + YYN2(I,J-1))
C
C ---   CALCULATE THE GRADIENT TERMS
C
C
C ---   X  GRADIENTS
C
      DUDX        = (UE * DYE(3,I,J) + UN * DYN(3,I,J)
     1             + UW * DYW(3,I,J) + US * DYS(3,I,J)) / AV
      DVDX        = (VE * DYE(3,I,J) + VN * DYN(3,I,J)
     1             + VW * DYW(3,I,J) + VS * DYS(3,I,J)) / AV
      DTDX        = (TE * DYE(3,I,J) + TN * DYN(3,I,J)
     1             + TW * DYW(3,I,J) + TS * DYS(3,I,J)) / AV
      DY1DX       = (Y1E * DYE(3,I,J) + Y1N * DYN(3,I,J)
     1             + Y1W * DYW(3,I,J) + Y1S * DYS(3,I,J)) / AV
      DY2DX       = (Y2E * DYE(3,I,J) + Y2N * DYN(3,I,J)
     1             + Y2W * DYW(3,I,J) + Y2S * DYS(3,I,J)) / AV
      DY3DX       = (Y3E * DYE(3,I,J) + Y3N * DYN(3,I,J)
     1             + Y3W * DYW(3,I,J) + Y3S * DYS(3,I,J)) / AV
      DY4DX       = (Y4E * DYE(3,I,J) + Y4N * DYN(3,I,J)
     1             + Y4W * DYW(3,I,J) + Y4S * DYS(3,I,J)) / AV
      DY5DX       = -DY1DX - DY2DX - DY3DX - DY4DX
C
C ---   Y  GRADIENTS
C
      DUDY        = - (UE * DXE(3,I,J) + UN * DXN(3,I,J)
     1               + UW * DXW(3,I,J) + US * DXS(3,I,J)) / AV
      DVDY        = - (VE * DXE(3,I,J) + VN * DXN(3,I,J)
     1               + VW * DXW(3,I,J) + VS * DXS(3,I,J)) / AV
```

```
      DTDY         = - (TE * DXE(3,I,J) + TN * DXN(3,I,J)
     1                + TW * DXW(3,I,J) + TS * DXS(3,I,J)) / AV
      DY1DY        = - (Y1E * DXE(3,I,J) + Y1N * DXN(3,I,J)
     1                + Y1W * DXW(3,I,J) + Y1S * DXS(3,I,J)) / AV
      DY2DY        = - (Y2E * DXE(3,I,J) + Y2N * DXN(3,I,J)
     1                + Y2W * DXW(3,I,J) + Y2S * DXS(3,I,J)) / AV
      DY3DY        = - (Y3E * DXE(3,I,J) + Y3N * DXN(3,I,J)
     1                + Y3W * DXW(3,I,J) + Y3S * DXS(3,I,J)) / AV
      DY4DY        = - (Y4E * DXE(3,I,J) + Y4N * DXN(3,I,J)
     1                + Y4W * DXW(3,I,J) + Y4S * DXS(3,I,J)) / AV
      DY5DY        = - DY1DY - DY2DY - DY3DY - DY4DY
C
C ---   COMPUTE THE FULL SHEAR STRESS TERMS
C
      TXX          = - LAMB * VIS(I,J) * (DUDX + DVDY)
                    - 2.0  * VIS(I,J) * DUDX / REN
      TXY          = - VIS(I,J) / REN  * (DUDY + DVDX)
      TYY          = - LAMB * VIS(I,J) * (DUDX + DVDY)
                    - 2.0  * VIS(I,J) * DVDY / REN
C
C ---   COMPUTE THE VISCOUS CONTRIBUTIONS TO "F" AND "G"
C
      FV(3,1,I,J)  = 0.0
      FV(3,2,I,J)  = TXX
      FV(3,3,I,J)  = TXY
      FV(3,4,I,J)  =   .5 * (UVEL(I,J) + UVEL(I-1,J))*TXX
     1                 + .5 * (VVEL(I,J) + VVEL(I-1,J))*TXY
     2                 - SDIFF * FACT1 * DTDX
     3                 - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2
     4                 + CPH2 * T1 * TEMP(I,J)) *  DY1DX
     5                 - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFO2
     6                 + CPO2 * T1 * TEMP(I,J)) *  DY2DX
     7                 - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2O
     8                 + CPH2O * T1 * TEMP(I,J)) *  DY3DX
     9                 - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2
     10                + CPN2 * T1 * TEMP(I,J)) *  DY4DX
     11                - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFOH
     12                + CPOH * T1 * TEMP(I,J)) *  DY5DX
      FV(3,5,I,J)  = - SDIFF * DY1DX
      FV(3,6,I,J)  = - SDIFF * DY2DX
      FV(3,7,I,J)  = - SDIFF * DY3DX
      FV(3,8,I,J)  = - SDIFF * DY4DX
      GV(3,1,I,J)  = 0.0
      GV(3,2,I,J)  = TXY
      GV(3,3,I,J)  = TYY
      GV(3,4,I,J)  =   .5 * (VVEL(I,J) + VVEL(I-1,J))*TYY
     1                 + .5 * (UVEL(I,J) + UVEL(I-1,J))*TXY
     2                 - SDIFF * FACT1 * DTDY
     3                 - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2
     4                 + CPH2 * T1 * TEMP(I,J)) *  DY1DY
```

```
     5                   - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFO2
     6                   + CPO2 * T1 * TEMP(I,J)) *  DY2DY
     7                   - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2O
     8                   + CPH2O * T1 * TEMP(I,J)) *  DY3DY
     9                   - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2
     10                  + CPN2 * T1 * TEMP(I,J)) *  DY4DY
     11                  - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFOH
     12                  + CPOH * T1 * TEMP(I,J)) *  DY5DY
      GV(3,5,I,J)  = - SDIFF * DY1DY
      GV(3,6,I,J)  = - SDIFF * DY2DY
      GV(3,7,I,J)  = - SDIFF * DY3DY
      GV(3,8,I,J)  = - SDIFF * DY4DY
40    CONTINUE
C
C
C                  ----------------
C ---              \  SOUTH FACE  \
C                  ----------------
C
      DO 50  J    = 2 , NYYY
      DO 50  I    = 2 , NXXX
      AV          = .5 * (AREA(I,J) + AREA(I,J-1))
C
C ---  U-VELOCITY
C
      UE          = .25 * (UVEL(I+1,J)   + UVEL(I,J)
     1                  +  UVEL(I,J-1)   + UVEL(I+1,J-1))
      UN          = UVEL(I,J)
      UW          = .25 * (UVEL(I,J)     + UVEL(I-1,J)
     1                  +  UVEL(I-1,J-1) + UVEL(I,J-1))
      US          = UVEL(I,J-1)
C
C ---  V-VELOCITY
C
      VE          = .25 * (VVEL(I+1,J)   + VVEL(I,J)
     1                  +  VVEL(I,J-1)   + VVEL(I+1,J-1))
      VN          = VVEL(I,J)
      VW          = .25 * (VVEL(I,J)     + VVEL(I-1,J)
     1                  +  VVEL(I-1,J-1) + VVEL(I,J-1))
      VS          = VVEL(I,J-1)
C
C ---  TEMPERATURE
C
      TE          = .25 * (TEMP(I+1,J)   + TEMP(I,J)
     1                  +  TEMP(I,J-1)   + TEMP(I+1,J-1))
      TN          = TEMP(I,J)
      TW          = .25 * (TEMP(I,J)     + TEMP(I-1,J)
     1                  +  TEMP(I-1,J-1) + TEMP(I,J-1))
      TS          = TEMP(I,J-1)
C
```

```
C ---   YH2
C
        Y1E         = .25 * (YH2(I+1,J)   + YH2(I,J)
     1                  +   YH2(I,J-1)   + YH2(I+1,J-1))
        Y1N         = YH2(I,J)
        Y1W         = .25 * (YH2(I,J)     + YH2(I-1,J)
     1                  +   YH2(I-1,J-1) + YH2(I,J-1))
        Y1S         = YH2(I,J-1)
C
C ---   YO2
C
        Y2E         = .25 * (YO2(I+1,J)   + YO2(I,J)
     1                  +   YO2(I,J-1)   + YO2(I+1,J-1))
        Y2N         = YO2(I,J)
        Y2W         = .25 * (YO2(I,J)     + YO2(I-1,J)
     1                  +   YO2(I-1,J-1) + YO2(I,J-1))
        Y2S         = YO2(I,J-1)
C
C ---   YH2O
C
        Y3E         = .25 * (YH2O(I+1,J)   + YH2O(I,J)
     1                  +   YH2O(I,J-1)   + YH2O(I+1,J-1))
        Y3N         = YH2O(I,J)
        Y3W         = .25 * (YH2O(I,J)     + YH2O(I-1,J)
     1                 +    YH2O(I-1,J-1) + YH2O(I,J-1))
        Y3S         = YH2O(I,J-1)
C
C ---   YYN2
C
        Y4E         = .25 * (YYN2(I+1,J)   + YYN2(I,J)
     1                  +   YYN2(I,J-1)   + YYN2(I+1,J-1))
        Y4N         = YYN2(I,J)
        Y4W         = .25 * (YYN2(I,J)     + YYN2(I-1,J)
     1                 +    YYN2(I-1,J-1) + YYN2(I,J-1))
        Y4S         = YYN2(I,J-1)
C
C ---   CALCULATE THE GRADIENT TERMS
C
C
C ---   X  GRADIENTS
C
        DUDX      = (UE * DYE(4,I,J) + UN * DYN(4,I,J)
     1             + UW * DYW(4,I,J) + US * DYS(4,I,J)) / AV
        DVDX      = (VE * DYE(4,I,J) + VN * DYN(4,I,J)
     1             + VW * DYW(4,I,J) + VS * DYS(4,I,J)) / AV
        DTDX      = (TE * DYE(4,I,J) + TN * DYN(4,I,J)
     1             + TW * DYW(4,I,J) + TS * DYS(4,I,J)) / AV
        DY1DX     = (Y1E * DYE(4,I,J) + Y1N * DYN(4,I,J)
     1             + Y1W * DYW(4,I,J) + Y1S * DYS(4,I,J)) / AV
        DY2DX     = (Y2E * DYE(4,I,J) + Y2N * DYN(4,I,J)
```

```
     1             + Y2W * DYW(4,I,J) + Y2S * DYS(4,I,J)) / AV
       DY3DX      = (Y3E * DYE(4,I,J) + Y3N * DYN(4,I,J)
     1             + Y3W * DYW(4,I,J) + Y3S * DYS(4,I,J)) / AV
       DY4DX      = (Y4E * DYE(4,I,J) + Y4N * DYN(4,I,J)
     1             + Y4W * DYW(4,I,J) + Y4S * DYS(4,I,J)) / AV
       DY5DX      = -DY1DX - DY2DX - DY3DX - DY4DX
C
C ---   Y GRADIENTS
C
       DUDY       = - (UE * DXE(4,I,J) + UN * DXN(4,I,J)
     1               + UW * DXW(4,I,J) + US * DXS(4,I,J)) / AV
       DVDY       = - (VE * DXE(4,I,J) + VN * DXN(4,I,J)
     1               + VW * DXW(4,I,J) + VS * DXS(4,I,J)) / AV
       DTDY       = - (TE * DXE(4,I,J) + TN * DXN(4,I,J)
     1               + TW * DXW(4,I,J) + TS * DXS(4,I,J)) / AV
       DY1DY      = - (Y1E * DXE(4,I,J) + Y1N * DXN(4,I,J)
     1               + Y1W * DXW(4,I,J) + Y1S * DXS(4,I,J)) / AV
       DY2DY      = - (Y2E * DXE(4,I,J) + Y2N * DXN(4,I,J)
     1               + Y2W * DXW(4,I,J) + Y2S * DXS(4,I,J)) / AV
       DY3DY      = - (Y3E * DXE(4,I,J) + Y3N * DXN(4,I,J)
     1               + Y3W * DXW(4,I,J) + Y3S * DXS(4,I,J)) / AV
       DY4DY      = - (Y4E * DXE(4,I,J) + Y4N * DXN(4,I,J)
     1               + Y4W * DXW(4,I,J) + Y4S * DXS(4,I,J)) / AV
       DY5DY      = - DY1DY - DY2DY - DY3DY - DY4DY
C
C ---   COMPUTE THE FULL SHEAR STRESS TERMS
C
       TXX        = - LAMB * VIS(I,J) * (DUDX + DVDY)
                    - 2.0  * VIS(I,J) * DUDX / REN
       TXY        = - VIS(I,J) / REN  * (DUDY + DVDX)
       TYY        = - LAMB * VIS(I,J) * (DUDX + DVDY)
                    - 2.0  * VIS(I,J) * DVDY / REN
C
C ---   COMPUTE THE VISCOUS CONTRIBUTIONS TO "F" AND "G"
C
       FV(4,1,I,J)  = 0.0
       FV(4,2,I,J)  = TXX
       FV(4,3,I,J)  = TXY
       FV(4,4,I,J)  =   .5 * (UVEL(I,J) + UVEL(I,J-1))*TXX
     1                + .5 * (VVEL(I,J) + VVEL(I,J-1))*TXY
     2                - SDIFF * FACT1 * DTDX
     3                - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2
     4                + CPH2 * T1 * TEMP(I,J)) *  DY1DX
     5                - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFO2
     6                + CPO2 * T1 * TEMP(I,J)) *  DY2DX
     7                - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2O
     8                + CPH2O * T1 * TEMP(I,J)) *  DY3DX
     9                - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2
     10               + CPN2 * T1 * TEMP(I,J)) *  DY4DX
     11               - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFOH
```

```
     12                + CPOH * T1 * TEMP(I,J)) *  DY5DX
       FV(4,5,I,J)  = - SDIFF * DY1DX
       FV(4,6,I,J)  = - SDIFF * DY2DX
       FV(4,7,I,J)  = - SDIFF * DY3DX
       FV(4,8,I,J)  = - SDIFF * DY4DX
       GV(4,1,I,J)  = 0.0
       GV(4,2,I,J)  = TXY
       GV(4,3,I,J)  = TYY
       GV(4,4,I,J)  =    .5 * (VVEL(I,J) + VVEL(I,J-1))*TYY
     1                + .5 * (UVEL(I,J) + UVEL(I,J-1))*TXY
     2                - SDIFF * FACT1 * DTDY
     3                - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2
     4                + CPH2 * T1 * TEMP(I,J)) *  DY1DY
     5                - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFO2
     6                + CPO2 * T1 * TEMP(I,J)) *  DY2DY
     7                - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2O
     8                + CPH2O * T1 * TEMP(I,J)) *  DY3DY
     9                - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2
     10               + CPN2 * T1 * TEMP(I,J)) *  DY4DY
     11               - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFOH
     12               + CPOH * T1 * TEMP(I,J)) *  DY5DY
       GV(4,5,I,J)  = - SDIFF * DY1DY
       GV(4,6,I,J)  = - SDIFF * DY2DY
       GV(4,7,I,J)  = - SDIFF * DY3DY
       GV(4,8,I,J)  = - SDIFF * DY4DY
50     CONTINUE
100    CONTINUE
       RETURN
       END
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
       SUBROUTINE LOWERBD
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C ---   COMPUTE FLUXES THROUGH THE LOWER WALL CELLS
C
       COMMON/VAR0/U(8,53,33,2),FI(8,53,33),GI(8,53,33),AH(8,53,33)
       COMMON/VAR00/FV(4,8,53,33),GV(4,8,53,33)
       COMMON/VAR1/UVEL(53,33),VVEL(53,33),PRES(53,33),TEMP(53,33)
       COMMON/VAR111/YH2(53,33),YO2(53,33),YH2O(53,33),YOH(53,33)
       COMMON/VAR2/DEN(53,33),SOUND(53,33),AINTE(53,33),AMACH(53,33)
       COMMON/VAR3/ENTHP(53,33),VIS(53,33),YYN2(53,33),CPND(53,33)
       COMMON/VAR4/DXX,DYY,X(55,35),Y(55,35),AREA(53,33)
       COMMON/VAR5/DT(53,33)
       COMMON/VAR6/NX,NXX,NXXX,NY,NYY,NYYY,IRES,IEQ
       COMMON/VAR7/IVIS,NITER,NOITER
       COMMON/VAR8/P1,T1,AM1,VISL,U1,V1,AK1,CV,CP,R,GAMA,DEN1,E1
       COMMON/VAR9/P11,C11,E11
       COMMON/VAR10/DFH2,DFO2,DFH2O,DFOH,DFN2
       COMMON/VAR11/CPH2,CPO2,CPH2O,CPOH,CPN2,CVH2,CVO2,CVH2O,CVOH,CVN2
       COMMON/VAR12/CONH2,CONO2,CONH2O,CONOH,CONN2
```

```
      COMMON/VAR13/COND,CFL,DCOFF,AL,VELO1
      COMMON/VAR14/RES(8,53,33)
      COMMON/VAR15/ALPHA(4)
      COMMON/VAR16/DXE(4,53,33),DXN(4,53,33),DXW(4,53,33),DXS(4,53,33)
      COMMON/VAR17/DYE(4,53,33),DYN(4,53,33),DYW(4,53,33),DYS(4,53,33)
      COMMON/VAR18/DD(2,8,53,33)
      COMMON/VAR20/NSX,NSXB,NSXA,NSY,NSYB,NSYA
      COMMON/VAR21/ACOM
      COMMON/VAR22/TRIGTEMP,PHI
      COMMON/VAR23/REN,PR,FACT1,LAMB,SDIFF
C                  ---------------
C ---              \  EAST FACE  \
C                  ---------------
C
      J = 1
      DO 20  I    = 2 , NXXX
      TWALL  = TEMP(I,1)
      Y1WALL = YH2(I,1)
      Y2WALL = YO2(I,1)
      Y3WALL = YH2O(I,1)
      Y4WALL = YYN2(I,1)
      AV          = .5 * (AREA(I,J) + AREA(I+1,J))
C
C ---   U-VELOCITY
C
      UE          = UVEL(I+1,J)
      UW          = UVEL(I,J)
      UN          = .25 * (UVEL(I+1,J) + UVEL(I+1,J+1)
     1                  +  UVEL(I,J+1) + UVEL(I,J))
      US          = 0.0
C
C ---   V-VELOCITY
C
      VE          = VVEL(I+1,J)
      VW          = VVEL(I,J)
      VN          = .25 * (VVEL(I+1,J) + VVEL(I+1,J+1)
     1                  +  VVEL(I,J+1) + VVEL(I+1,J))
      VS          = 0.0
C
C ---   TEMPERATURE
C
      TE          = TEMP(I+1,J)
      TW          = TEMP(I,J)
      TN          = .25 * (TEMP(I+1,J) + TEMP(I+1,J+1)
     1                  +  TEMP(I,J+1) + TEMP(I+1,J))
      TS          = TWALL
C
C ---   YH2
C
      Y1E          = YH2(I+1,J)
```

```
      Y1W          = YH2(I,J)
      Y1N          = .25 * (YH2(I+1,J) + YH2(I+1,J+1)
     1                 +   YH2(I,J+1) + YH2(I+1,J))
      Y1S          = Y1WALL
C
C ---   YO2
C
      Y2E          = YO2(I+1,J)
      Y2W          = YO2(I,J)
      Y2N          = .25 * (YO2(I+1,J) + YO2(I+1,J+1)
     1                 +   YO2(I,J+1) + YO2(I+1,J))
      Y2S          = Y2WALL
C
C ---   YH2O
C
      Y3E          = YH2O(I+1,J)
      Y3W          = YH2O(I,J)
      Y3N          = .25 * (YH2O(I+1,J) + YH2O(I+1,J+1)
     1                 +   YH2O(I,J+1) + YH2O(I+1,J))
      Y3S          = Y3WALL
C
C ---   YYN2
C
      Y4E          = YYN2(I+1,J)
      Y4W          = YYN2(I,J)
      Y4N          = .25 * (YYN2(I+1,J) + YYN2(I+1,J+1)
     1                 +   YYN2(I,J+1) + YYN2(I+1,J))
      Y4S          = Y4WALL
C
C ---   CALCULATE THE GRADIENT TERMS
C
C
C ---   X  GRADIENTS
C
      DUDX        = (UE * DYE(1,I,J) + UN * DYN(1,I,J)
     1              + UW * DYW(1,I,J) + US * DYS(1,I,J)) / AV
      DVDX        = (VE * DYE(1,I,J) + VN * DYN(1,I,J)
     1              + VW * DYW(1,I,J) + VS * DYS(1,I,J)) / AV
      DTDX        = (TE * DYE(1,I,J) + TN * DYN(1,I,J)
     1              + TW * DYW(1,I,J) + TS * DYS(1,I,J)) / AV
      DY1DX       = (Y1E * DYE(1,I,J) + Y1N * DYN(1,I,J)
     1              + Y1W * DYW(1,I,J) + Y1S * DYS(1,I,J)) / AV
      DY2DX       = (Y2E * DYE(1,I,J) + Y2N * DYN(1,I,J)
     1              + Y2W * DYW(1,I,J) + Y2S * DYS(1,I,J)) / AV
      DY3DX       = (Y3E * DYE(1,I,J) + Y3N * DYN(1,I,J)
     1              + Y3W * DYW(1,I,J) + Y3S * DYS(1,I,J)) / AV
      DY4DX       = (Y4E * DYE(1,I,J) + Y4N * DYN(1,I,J)
     1              + Y4W * DYW(1,I,J) + Y4S * DYS(1,I,J)) / AV
      DY5DX       = -DY1DX - DY2DX - DY3DX - DY4DX
C
```

```
C ---   Y  GRADIENTS
C
      DUDY        = - (UE * DXE(1,I,J) + UN * DXN(1,I,J)
     1                + UW * DXW(1,I,J) + US * DXS(1,I,J)) / AV
      DVDY        = - (VE * DXE(1,I,J) + VN * DXN(1,I,J)
     1                + VW * DXW(1,I,J) + VS * DXS(1,I,J)) / AV
      DTDY        = - (TE * DXE(1,I,J) + TN * DXN(1,I,J)
     1                + TW * DXW(1,I,J) + TS * DXS(1,I,J)) / AV
      DY1DY       = - (Y1E * DXE(1,I,J) + Y1N * DXN(1,I,J)
     1                + Y1W * DXW(1,I,J) + Y1S * DXS(1,I,J)) / AV
      DY2DY       = - (Y2E * DXE(1,I,J) + Y2N * DXN(1,I,J)
     1                + Y2W * DXW(1,I,J) + Y2S * DXS(1,I,J)) / AV
      DY3DY       = - (Y3E * DXE(1,I,J) + Y3N * DXN(1,I,J)
     1                + Y3W * DXW(1,I,J) + Y3S * DXS(1,I,J)) / AV
      DY4DY       = - (Y4E * DXE(1,I,J) + Y4N * DXN(1,I,J)
     1                + Y4W * DXW(1,I,J) + Y4S * DXS(1,I,J)) / AV
      DY5DY       = - DY1DY - DY2DY - DY3DY - DY4DY
C
C ---   COMPUTE THE FULL SHEAR STRESS TERMS
C
      TXX         = - LAMB * VIS(I,J) * (DUDX + DVDY)
                   - 2.0  * VIS(I,J) * DUDX / REN
      TXY         = - VIS(I,J) / REN  * (DUDY + DVDX)
      TYY         = - LAMB * VIS(I,J) * (DUDX + DVDY)
                   - 2.0  * VIS(I,J) * DVDY / REN
C
C ---   COMPUTE THE VISCOUS CONTRIBUTIONS TO "F" AND "G"
C
      FV(1,1,I,J)  = 0.0
      FV(1,2,I,J)  = TXX
      FV(1,3,I,J)  = TXY
      FV(1,4,I,J)  =   .5 * (UVEL(I,J) + UVEL(I+1,J))*TXX
     1                 + .5 * (VVEL(I,J) + VVEL(I+1,J))*TXY
     2                 - SDIFF * FACT1 * DTDX
     3                 - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2
     4                 + CPH2 * T1 * TEMP(I,J)) *  DY1DX
     5                 - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFO2
     6                 + CPO2 * T1 * TEMP(I,J)) *  DY2DX
     7                 - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2O
     8                 + CPH2O * T1 * TEMP(I,J)) *  DY3DX
     9                 - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2
     10                + CPN2 * T1 * TEMP(I,J)) *  DY4DX
     11                - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFOH
     12                + CPOH * T1 * TEMP(I,J)) *  DY5DX
      FV(1,5,I,J)  = - SDIFF * DY1DX
      FV(1,6,I,J)  = - SDIFF * DY2DX
      FV(1,7,I,J)  = - SDIFF * DY3DX
      FV(1,8,I,J)  = - SDIFF * DY4DX
      GV(1,1,I,J)  = 0.0
      GV(1,2,I,J)  = TXY
```

```
      GV(1,3,I,J)  = TYY
      GV(1,4,I,J)  =    .5 * (VVEL(I,J) + VVEL(I+1,J))*TYY
     1                + .5 * (UVEL(I,J) + UVEL(I+1,J))*TXY
     2                - SDIFF * FACT1 * DTDY
     3                - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2
     4                + CPH2 * T1 * TEMP(I,J)) *  DY1DY
     5                - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFO2
     6                + CPO2 * T1 * TEMP(I,J)) *  DY2DY
     7                - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2O
     8                + CPH2O * T1 * TEMP(I,J)) *  DY3DY
     9                - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2
     10               + CPN2 * T1 * TEMP(I,J)) *  DY4DY
     11               - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFOH
     12               + CPOH * T1 * TEMP(I,J)) *  DY5DY
      GV(1,5,I,J)  = - SDIFF * DY1DY
      GV(1,6,I,J)  = - SDIFF * DY2DY
      GV(1,7,I,J)  = - SDIFF * DY3DY
      GV(1,8,I,J)  = - SDIFF * DY4DY
20    CONTINUE
C
C                ------------------
C ---            \   NORTH  FACE  \
C                ------------------
C
      J          = 1
      DO 30  I   = 2 , NXXX
      TWALL  = TEMP(I,1)
      Y1WALL = YH2(I,1)
      Y2WALL = YO2(I,1)
      Y3WALL = YH2O(I,1)
      Y4WALL = YYN2(I,1)
      AV         = .5 * (AREA(I,J) + AREA(I,J+1))
C
C ---   U-VELOCITY
C
      UE         = .25 * (UVEL(I+1,J) + UVEL(I+1,J+1)
     1                  +  UVEL(I,J+1) + UVEL(I,J))
      UN         = UVEL(I,J+1)
      UW         = .25 * (UVEL(I,J+1) + UVEL(I-1,J+1)
     1                  +  UVEL(I-1,J) + UVEL(I,J))
      US         = UVEL(I,J)
C
C ---   V-VELOCITY
C
      VE         = .25 * (VVEL(I+1,J) + VVEL(I+1,J+1)
     1                  +  VVEL(I,J+1) + VVEL(I,J))
      VN         = VVEL(I,J+1)
      VW         = .25 * (VVEL(I,J+1) + VVEL(I-1,J+1)
     1                  +  VVEL(I-1,J) + VVEL(I,J))
      VS         = VVEL(I,J)
```

```
C
C ---   TEMPERATURE
C
        TE          = .25 * (TEMP(I+1,J) + TEMP(I+1,J+1)
     1                   +  TEMP(I,J+1) + TEMP(I,J))
        TN          = TEMP(I,J+1)
        TW          = .25 * (TEMP(I,J+1) + TEMP(I-1,J+1)
     1                   +  TEMP(I-1,J) + TEMP(I,J))
        TS          = TEMP(I,J)
C
C ---   YH2
C
        Y1E          = .25 * (YH2(I+1,J)  + YH2(I+1,J+1)
     1                    +  YH2(I,J+1)  + YH2(I,J))
        Y1N          = YH2(I,J+1)
        Y1W          = .25 * (YH2(I,J+1) + YH2(I-1,J+1)
     1                    +  YH2(I-1,J) + YH2(I,J))
        Y1S          = YH2(I,J)
C
C ---   YO2
C
        Y2E          = .25 * (YO2(I+1,J)  + YO2(I+1,J+1)
     1                    +  YO2(I,J+1)  + YO2(I,J))
        Y2N          = YO2(I,J+1)
        Y2W          = .25 * (YO2(I,J+1) + YO2(I-1,J+1)
     1                    +  YO2(I-1,J) + YO2(I,J))
        Y2S          = YO2(I,J)
C
C ---   YH2O
C
        Y3E          = .25 * (YH2O(I+1,J)  + YH2O(I+1,J+1)
     1                    +  YH2O(I,J+1)  + YH2O(I,J))
        Y3N          = YH2O(I,J+1)
        Y3W          = .25 * (YH2O(I,J+1) + YH2O(I-1,J+1)
     1                    +  YH2O(I-1,J) + YH2O(I,J))
        Y3S          = YH2O(I,J)
C
C ---   YYN2
C
        Y4E          = .25 * (YYN2(I+1,J)  + YYN2(I+1,J+1)
     1                    +  YYN2(I,J+1)  + YYN2(I,J))
        Y4N          = YYN2(I,J+1)
        Y4W          = .25 * (YYN2(I,J+1) + YYN2(I-1,J+1)
     1                    +  YYN2(I-1,J) + YYN2(I,J))
        Y4S          = YYN2(I,J)
C
C ---   CALCULATE THE GRADIENT TERMS
C
C
C ---   X  GRADIENTS
```

```
C
      DUDX        = (UE * DYE(2,I,J) + UN * DYN(2,I,J)
     1             + UW * DYW(2,I,J) + US * DYS(2,I,J)) / AV
      DVDX        = (VE * DYE(2,I,J) + VN * DYN(2,I,J)
     1             + VW * DYW(2,I,J) + VS * DYS(2,I,J)) / AV
      DTDX        = (TE * DYE(2,I,J) + TN * DYN(2,I,J)
     1             + TW * DYW(2,I,J) + TS * DYS(2,I,J)) / AV
      DY1DX       = (Y1E * DYE(2,I,J) + Y1N * DYN(2,I,J)
     1             + Y1W * DYW(2,I,J) + Y1S * DYS(2,I,J)) / AV
      DY2DX       = (Y2E * DYE(2,I,J) + Y2N * DYN(2,I,J)
     1             + Y2W * DYW(2,I,J) + Y2S * DYS(2,I,J)) / AV
      DY3DX       = (Y3E * DYE(2,I,J) + Y3N * DYN(2,I,J)
     1             + Y3W * DYW(2,I,J) + Y3S * DYS(2,I,J)) / AV
      DY4DX       = (Y4E * DYE(2,I,J) + Y4N * DYN(2,I,J)
     1             + Y4W * DYW(2,I,J) + Y4S * DYS(2,I,J)) / AV
      DY5DX       = -DY1DX - DY2DX - DY3DX - DY4DX
C
C ---   Y  GRADIENTS
C
      DUDY        = - (UE * DXE(2,I,J) + UN * DXN(2,I,J)
     1               + UW * DXW(2,I,J) + US * DXS(2,I,J)) / AV
      DVDY        = - (VE * DXE(2,I,J) + VN * DXN(2,I,J)
     1               + VW * DXW(2,I,J) + VS * DXS(2,I,J)) / AV
      DTDY        = - (TE * DXE(2,I,J) + TN * DXN(2,I,J)
     1               + TW * DXW(2,I,J) + TS * DXS(2,I,J)) / AV
      DY1DY       = - (Y1E * DXE(2,I,J) + Y1N * DXN(2,I,J)
     1               + Y1W * DXW(2,I,J) + Y1S * DXS(2,I,J)) / AV
      DY2DY       = - (Y2E * DXE(2,I,J) + Y2N * DXN(2,I,J)
     1               + Y2W * DXW(2,I,J) + Y2S * DXS(2,I,J)) / AV
      DY3DY       = - (Y3E * DXE(2,I,J) + Y3N * DXN(2,I,J)
     1               + Y3W * DXW(2,I,J) + Y3S * DXS(2,I,J)) / AV
      DY4DY       = - (Y4E * DXE(2,I,J) + Y4N * DXN(2,I,J)
     1               + Y4W * DXW(2,I,J) + Y4S * DXS(2,I,J)) / AV
      DY5DY       = - DY1DY - DY2DY - DY3DY - DY4DY
C
C ---   COMPUTE THE FULL SHEAR STRESS TERMS
C
      TXX         = - LAMB * VIS(I,J) * (DUDX + DVDY)
                    - 2.0  * VIS(I,J) * DUDX / REN
      TXY         = - VIS(I,J) / REN  * (DUDY + DVDX)
      TYY         = - LAMB * VIS(I,J) * (DUDX + DVDY)
                    - 2.0  * VIS(I,J) * DVDY / REN
C
C ---   COMPUTE THE VISCOUS CONTRIBUTIONS TO "F" AND "G"
C
      FV(2,1,I,J)  = 0.0
      FV(2,2,I,J)  = TXX
      FV(2,3,I,J)  = TXY
      FV(2,4,I,J)  =   .5 * (UVEL(I,J) + UVEL(I,J+1))*TXX
     1                + .5 * (VVEL(I,J) + VVEL(I,J+1))*TXY
```

```
     2                   - SDIFF * FACT1 * DTDX
     3                   - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2
     4                   + CPH2 * T1 * TEMP(I,J)) *  DY1DX
     5                   - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFO2
     6                   + CPO2 * T1 * TEMP(I,J)) *  DY2DX
     7                   - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2O
     8                   + CPH2O * T1 * TEMP(I,J)) *  DY3DX
     9                   - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2
     10                  + CPN2 * T1 * TEMP(I,J)) *  DY4DX
     11                  - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFOH
     12                  + CPOH * T1 * TEMP(I,J)) *  DY5DX
       FV(2,5,I,J)  = - SDIFF * DY1DX
       FV(2,6,I,J)  = - SDIFF * DY2DX
       FV(2,7,I,J)  = - SDIFF * DY3DX
       FV(2,8,I,J)  = - SDIFF * DY4DX
       GV(2,1,I,J)  = 0.0
       GV(2,2,I,J)  = TXY
       GV(2,3,I,J)  = TYY
       GV(2,4,I,J)  =   .5 * (VVEL(I,J) + VVEL(I,J+1))*TYY
     1                 + .5 * (UVEL(I,J) + UVEL(I,J+1))*TXY
     2                 - SDIFF * FACT1 * DTDY
     3                 - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2
     4                 + CPH2 * T1 * TEMP(I,J)) *  DY1DY
     5                 - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFO2
     6                 + CPO2 * T1 * TEMP(I,J)) *  DY2DY
     7                 - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2O
     8                 + CPH2O * T1 * TEMP(I,J)) *  DY3DY
     9                 - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2
     10                + CPN2 * T1 * TEMP(I,J)) *  DY4DY
     11                - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFOH
     12                + CPOH * T1 * TEMP(I,J)) *  DY5DY
       GV(2,5,I,J)  = - SDIFF * DY1DY
       GV(2,6,I,J)  = - SDIFF * DY2DY
       GV(2,7,I,J)  = - SDIFF * DY3DY
       GV(2,8,I,J)  = - SDIFF * DY4DY
30     CONTINUE
C
C
C                   ---------------
C ---               \  WEST FACE  \
C                   ---------------
C
       J          = 1
       DO 40  I   = 2 , NXXX
       TWALL  = TEMP(I,1)
       Y1WALL = YH2(I,1)
       Y2WALL = YO2(I,1)
       Y3WALL = YH2O(I,1)
       Y4WALL = YYN2(I,1)
       AV         = .5 * (AREA(I,J) + AREA(I-1,J))
```

```
C
C ---   U-VELOCITY
C
        UE          = UVEL(I,J)
        UW          = UVEL(I-1,J)
        UN          = .25 * (UVEL(I,J)     + UVEL(I,J+1)
     1                     + UVEL(I-1,J+1) + UVEL(I-1,J))
        US          = 0.0
C
C ---   V-VELOCITY
C
        VE          = VVEL(I,J)
        VW          = VVEL(I-1,J)
        VN          = .25 * (VVEL(I,J)     + VVEL(I,J+1)
     1                     + VVEL(I-1,J+1) + VVEL(I-1,J))
        VS          = 0.0
C
C ---   TEMPERATURE
C
        TE          = TEMP(I,J)
        TW          = TEMP(I-1,J)
        TN          = .25 * (TEMP(I,J)     + TEMP(I,J+1)
     1                     + TEMP(I-1,J+1) + TEMP(I-1,J))
        TS          = TWALL
C
C ---   YH2
C
        Y1E          = YH2(I,J)
        Y1W          = YH2(I-1,J)
        Y1N          = .25 * (YH2(I,J)     + YH2(I,J+1)
     1                      + YH2(I-1,J+1) + YH2(I-1,J))
        Y1S          = Y1WALL
C
C ---   YO2
C
        Y2E          = YO2(I,J)
        Y2W          = YO2(I-1,J)
        Y2N          = .25 * (YO2(I,J)     + YO2(I,J+1)
     1                      + YO2(I-1,J+1) + YO2(I-1,J))
        Y2S          = Y2WALL
C
C ---   YH2O
C
        Y3E          = YH2O(I,J)
        Y3W          = YH2O(I-1,J)
        Y3N          = .25 * (YH2O(I,J)     + YH2O(I,J+1)
     1                      + YH2O(I-1,J+1) + YH2O(I-1,J))
        Y3S          = Y3WALL
C
C ---   YYN2
```

```
C
      Y4E          = YYN2(I,J)
      Y4W          = YYN2(I-1,J)
      Y4N          = .25 * (YYN2(I,J)     + YYN2(I,J+1)
     1                  +  YYN2(I-1,J+1) + YYN2(I-1,J))
      Y4S          = Y4WALL
C
C ---   CALCULATE THE GRADIENT TERMS
C
C
C ---   X  GRADIENTS
C
      DUDX        = (UE * DYE(3,I,J) + UN * DYN(3,I,J)
     1             + UW * DYW(3,I,J) + US * DYS(3,I,J)) / AV
      DVDX        = (VE * DYE(3,I,J) + VN * DYN(3,I,J)
     1             + VW * DYW(3,I,J) + VS * DYS(3,I,J)) / AV
      DTDX        = (TE * DYE(3,I,J) + TN * DYN(3,I,J)
     1             + TW * DYW(3,I,J) + TS * DYS(3,I,J)) / AV
      DY1DX       = (Y1E * DYE(3,I,J) + Y1N * DYN(3,I,J)
     1             + Y1W * DYW(3,I,J) + Y1S * DYS(3,I,J)) / AV
      DY2DX       = (Y2E * DYE(3,I,J) + Y2N * DYN(3,I,J)
     1             + Y2W * DYW(3,I,J) + Y2S * DYS(3,I,J)) / AV
      DY3DX       = (Y3E * DYE(3,I,J) + Y3N * DYN(3,I,J)
     1             + Y3W * DYW(3,I,J) + Y3S * DYS(3,I,J)) / AV
      DY4DX       = (Y4E * DYE(3,I,J) + Y4N * DYN(3,I,J)
     1             + Y4W * DYW(3,I,J) + Y4S * DYS(3,I,J)) / AV
      DY5DX       = -DY1DX - DY2DX - DY3DX - DY4DX
C
C ---   Y  GRADIENTS
C
      DUDY        = - (UE * DXE(3,I,J) + UN * DXN(3,I,J)
     1               + UW * DXW(3,I,J) + US * DXS(3,I,J)) / AV
      DVDY        = - (VE * DXE(3,I,J) + VN * DXN(3,I,J)
     1               + VW * DXW(3,I,J) + VS * DXS(3,I,J)) / AV
      DTDY        = - (TE * DXE(3,I,J) + TN * DXN(3,I,J)
     1               + TW * DXW(3,I,J) + TS * DXS(3,I,J)) / AV
      DY1DY       = - (Y1E * DXE(3,I,J) + Y1N * DXN(3,I,J)
     1               + Y1W * DXW(3,I,J) + Y1S * DXS(3,I,J)) / AV
      DY2DY       = - (Y2E * DXE(3,I,J) + Y2N * DXN(3,I,J)
     1               + Y2W * DXW(3,I,J) + Y2S * DXS(3,I,J)) / AV
      DY3DY       = - (Y3E * DXE(3,I,J) + Y3N * DXN(3,I,J)
     1               + Y3W * DXW(3,I,J) + Y3S * DXS(3,I,J)) / AV
      DY4DY       = - (Y4E * DXE(3,I,J) + Y4N * DXN(3,I,J)
     1               + Y4W * DXW(3,I,J) + Y4S * DXS(3,I,J)) / AV
      DY5DY       = - DY1DY - DY2DY - DY3DY - DY4DY
C
C ---   COMPUTE THE FULL SHEAR STRESS TERMS
C
      TXX         = - LAMB * VIS(I,J) * (DUDX + DVDY)
                    - 2.0  * VIS(I,J) * DUDX / REN
```

```
      TXY         = - VIS(I,J) / REN   * (DUDY + DVDX)
      TYY         = - LAMB * VIS(I,J) * (DUDX + DVDY)
                   - 2.0  * VIS(I,J) * DVDY / REN
C
C ---  COMPUTE THE VISCOUS CONTRIBUTIONS TO "F" AND "G"
C
      FV(3,1,I,J)  = 0.0
      FV(3,2,I,J)  = TXX
      FV(3,3,I,J)  = TXY
      FV(3,4,I,J)  =   .5 * (UVEL(I,J) + UVEL(I-1,J))*TXX
     1                + .5 * (VVEL(I,J) + VVEL(I-1,J))*TXY
     2                - SDIFF * FACT1 * DTDX
     3                - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2
     4                + CPH2 * T1 * TEMP(I,J)) *  DY1DX
     5                - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFO2
     6                + CPO2 * T1 * TEMP(I,J)) *  DY2DX
     7                - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2O
     8                + CPH2O * T1 * TEMP(I,J)) *  DY3DX
     9                - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2
     10               + CPN2 * T1 * TEMP(I,J)) *  DY4DX
     11               - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFOH
     12               + CPOH * T1 * TEMP(I,J)) *  DY5DX
      FV(3,5,I,J)  = - SDIFF * DY1DX
      FV(3,6,I,J)  = - SDIFF * DY2DX
      FV(3,7,I,J)  = - SDIFF * DY3DX
      FV(3,8,I,J)  = - SDIFF * DY4DX
      GV(3,1,I,J)  = 0.0
      GV(3,2,I,J)  = TXY
      GV(3,3,I,J)  = TYY
      GV(3,4,I,J)  =   .5 * (VVEL(I,J) + VVEL(I-1,J))*TYY
     1                + .5 * (UVEL(I,J) + UVEL(I-1,J))*TXY
     2                - SDIFF * FACT1 * DTDY
     3                - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2
     4                + CPH2 * T1 * TEMP(I,J)) *  DY1DY
     5                - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFO2
     6                + CPO2 * T1 * TEMP(I,J)) *  DY2DY
     7                - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2O
     8                + CPH2O * T1 * TEMP(I,J)) *  DY3DY
     9                - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2
     10               + CPN2 * T1 * TEMP(I,J)) *  DY4DY
     11               - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFOH
     12               + CPOH * T1 * TEMP(I,J)) *  DY5DY
      GV(3,5,I,J)  = - SDIFF * DY1DY
      GV(3,6,I,J)  = - SDIFF * DY2DY
      GV(3,7,I,J)  = - SDIFF * DY3DY
      GV(3,8,I,J)  = - SDIFF * DY4DY
40    CONTINUE
C
C
C                ----------------
```

```
C ---              \  SOUTH FACE  \
C                  ----------------
C
      J          = 1
      DO 50  I   = 2 , NXXX
      TWALL  = TEMP(I,1)
      Y1WALL = YH2(I,1)
      Y2WALL = YO2(I,1)
      Y3WALL = YH2O(I,1)
      Y4WALL = YYN2(I,1)
      AV         = AREA(I,J)
C
C ---   U-VELOCITY
C
      UE         = 0.0
      UN         = UVEL(I,J)
      UW         = 0.0
      US         = -UVEL(I,J)
C
C ---   V-VELOCITY
C
      VE         = 0.0
      VN         = VVEL(I,J)
      VW         = 0.0
      VS         = -VVEL(I,J)
C
C ---   TEMPERATURE
C
      TE         = TWALL
      TN         = TEMP(I,J)
      TW         = TWALL
      TS         = TEMP(I,J)
C
C ---   YH2
C
      Y1E         = Y1WALL
      Y1N         = YH2(I,J)
      Y1W         = Y1WALL
      Y1S         = YH2(I,J)
C
C ---   YO2
C
      Y2E         = Y2WALL
      Y2N         = YO2(I,J)
      Y2W         = Y2WALL
      Y2S         = YO2(I,J)
C
C ---   YH2O
C
      Y3E         = Y3WALL
```

```
      Y3N          = YH2O(I,J)
      Y3W          = Y3WALL
      Y3S          = YH2O(I,J)
C
C ---   YYN2
C
      Y4E          = Y4WALL
      Y4N          = YYN2(I,J)
      Y4W          = Y4WALL
      Y4S          = YYN2(I,J)
C
C ---   CALCULATE THE GRADIENT TERMS
C
C
C ---   X  GRADIENTS
C
      DUDX        = (UE * DYE(4,I,J) + UN * DYN(4,I,J)
     1              + UW * DYW(4,I,J) + US * DYS(4,I,J)) / AV
      DVDX        = (VE * DYE(4,I,J) + VN * DYN(4,I,J)
     1              + VW * DYW(4,I,J) + VS * DYS(4,I,J)) / AV
      DTDX        = (TE * DYE(4,I,J) + TN * DYN(4,I,J)
     1              + TW * DYW(4,I,J) + TS * DYS(4,I,J)) / AV
      DY1DX       = (Y1E * DYE(4,I,J) + Y1N * DYN(4,I,J)
     1              + Y1W * DYW(4,I,J) + Y1S * DYS(4,I,J)) / AV
      DY2DX       = (Y2E * DYE(4,I,J) + Y2N * DYN(4,I,J)
     1              + Y2W * DYW(4,I,J) + Y2S * DYS(4,I,J)) / AV
      DY3DX       = (Y3E * DYE(4,I,J) + Y3N * DYN(4,I,J)
     1              + Y3W * DYW(4,I,J) + Y3S * DYS(4,I,J)) / AV
      DY4DX       = (Y4E * DYE(4,I,J) + Y4N * DYN(4,I,J)
     1              + Y4W * DYW(4,I,J) + Y4S * DYS(4,I,J)) / AV
      DY5DX       = -DY1DX - DY2DX - DY3DX - DY4DX
C
C ---   Y  GRADIENTS
C
      DUDY        = - (UE * DXE(4,I,J) + UN * DXN(4,I,J)
     1                + UW * DXW(4,I,J) + US * DXS(4,I,J)) / AV
      DVDY        = - (VE * DXE(4,I,J) + VN * DXN(4,I,J)
     1                + VW * DXW(4,I,J) + VS * DXS(4,I,J)) / AV
      DTDY        = - (TE * DXE(4,I,J) + TN * DXN(4,I,J)
     1                + TW * DXW(4,I,J) + TS * DXS(4,I,J)) / AV
      DY1DY       = - (Y1E * DXE(4,I,J) + Y1N * DXN(4,I,J)
     1                + Y1W * DXW(4,I,J) + Y1S * DXS(4,I,J)) / AV
      DY2DY       = - (Y2E * DXE(4,I,J) + Y2N * DXN(4,I,J)
     1                + Y2W * DXW(4,I,J) + Y2S * DXS(4,I,J)) / AV
      DY3DY       = - (Y3E * DXE(4,I,J) + Y3N * DXN(4,I,J)
     1                + Y3W * DXW(4,I,J) + Y3S * DXS(4,I,J)) / AV
      DY4DY       = - (Y4E * DXE(4,I,J) + Y4N * DXN(4,I,J)
     1                + Y4W * DXW(4,I,J) + Y4S * DXS(4,I,J)) / AV
C
C ---   COMPUTE THE FULL SHEAR STRESS TERMS
```

```
C
C
      DUDX        = 0.0
      DVDX        = 0.0
      DTDX        = 0.0
      DY1DX       = 0.0
      DY2DX       = 0.0
      DY3DX       = 0.0
      DY4DX       = 0.0
C
      DUDY        = 2.0*UVEL(I,1)/(Y(I,2)-Y(I,1))
      DVDY        = 2.0*VVEL(I,1)/(Y(I,2)-Y(I,1))
      DTDY        = 2.0*(TEMP(I,1)-TWALL)/(Y(I,2)-Y(I,1))
      DY1DY       = 2.0*(YH2(I,1)-Y1WALL) /(Y(I,2)-Y(I,1))
      DY2DY       = 2.0*(YO2(I,1)-Y2WALL) /(Y(I,2)-Y(I,1))
      DY3DY       = 2.0*(YH2O(I,1)-Y3WALL)/(Y(I,2)-Y(I,1))
      DY4DY       = 2.0*(YYN2(I,1)-Y4WALL) /(Y(I,2)-Y(I,1))
      TXX         = - LAMB * VIS(I,J) * (DUDX + DVDY)
                     - 2.0  * VIS(I,J) * DUDX / REN
      TXY         = - VIS(I,J) / REN  * (DUDY + DVDX)
      TYY         = - LAMB * VIS(I,J) * (DUDX + DVDY)
                     - 2.0  * VIS(I,J) * DVDY / REN
C
C ---   COMPUTE THE VISCOUS CONTRIBUTIONS TO "F" AND "G"
C
      FV(4,1,I,J)  = 0.0
      FV(4,2,I,J)  = TXX
      FV(4,3,I,J)  = TXY
      FV(4,4,I,J)  =    - SDIFF * FACT1 * DTDX
     3                  - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2
     4                  + CPH2 * T1 * TEMP(I,J)) *  DY1DX
     5                  - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFO2
     6                  + CPO2 * T1 * TEMP(I,J)) *  DY2DX
     7                  - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2O
     8                  + CPH2O * T1 * TEMP(I,J)) *  DY3DX
     9                  - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2
     10                 + CPN2 * T1 * TEMP(I,J)) *  DY4DX
     11                 - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFOH
     12                 + CPOH * T1 * TEMP(I,J)) *  DY5DX
      FV(4,5,I,J)  = - SDIFF * DY1DX
      FV(4,6,I,J)  = - SDIFF * DY2DX
      FV(4,7,I,J)  = - SDIFF * DY3DX
      FV(4,8,I,J)  = - SDIFF * DY4DX
      GV(4,1,I,J)  = 0.0
      GV(4,2,I,J)  = TXY
      GV(4,3,I,J)  = TYY
      GV(4,4,I,J)  =    - SDIFF * FACT1 * DTDY
     3                  - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2
     4                  + CPH2 * T1 * TEMP(I,J)) *  DY1DY
     5                  - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFO2
```

```
     6                    + CPO2 * T1 * TEMP(I,J)) *  DY2DY
     7                    - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2O
     8                    + CPH2O * T1 * TEMP(I,J)) *  DY3DY
     9                    - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2
     10                   + CPN2 * T1 * TEMP(I,J)) *  DY4DY
     11                   - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFOH
     12                   + CPOH * T1 * TEMP(I,J)) *  DY5DY
      GV(4,5,I,J)  = - SDIFF * DY1DY
      GV(4,6,I,J)  = - SDIFF * DY2DY
      GV(4,7,I,J)  = - SDIFF * DY3DY
      GV(4,8,I,J)  = - SDIFF * DY4DY
50    CONTINUE
100   CONTINUE
      RETURN
      END
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      SUBROUTINE UPPERBD
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C ---   COMPUTE FLUXES THROUGH THE UPPER WALL CELLS
C
      COMMON/VAR0/U(8,53,33,2),FI(8,53,33),GI(8,53,33),AH(8,53,33)
      COMMON/VAR00/FV(4,8,53,33),GV(4,8,53,33)
      COMMON/VAR1/UVEL(53,33),VVEL(53,33),PRES(53,33),TEMP(53,33)
      COMMON/VAR111/YH2(53,33),YO2(53,33),YH2O(53,33),YOH(53,33)
      COMMON/VAR2/DEN(53,33),SOUND(53,33),AINTE(53,33),AMACH(53,33)
      COMMON/VAR3/ENTHP(53,33),VIS(53,33),YYN2(53,33),CPND(53,33)
      COMMON/VAR4/DXX,DYY,X(55,35),Y(55,35),AREA(53,33)
      COMMON/VAR5/DT(53,33)
      COMMON/VAR6/NX,NXX,NXXX,NY,NYY,NYYY,IRES,IEQ
      COMMON/VAR7/IVIS,NITER,NOITER
      COMMON/VAR8/P1,T1,AM1,VISL,U1,V1,AK1,CV,CP,R,GAMA,DEN1,E1
      COMMON/VAR9/P11,C11,E11
      COMMON/VAR10/DFH2,DFO2,DFH2O,DFOH,DFN2
      COMMON/VAR11/CPH2,CPO2,CPH2O,CPOH,CPN2,CVH2,CVO2,CVH2O,CVOH,CVN2
      COMMON/VAR12/CONH2,CONO2,CONH2O,CONOH,CONN2
      COMMON/VAR13/COND,CFL,DCOFF,AL,VELO1
      COMMON/VAR14/RES(8,53,33)
      COMMON/VAR15/ALPHA(4)
      COMMON/VAR16/DXE(4,53,33),DXN(4,53,33),DXW(4,53,33),DXS(4,53,33)
      COMMON/VAR17/DYE(4,53,33),DYN(4,53,33),DYW(4,53,33),DYS(4,53,33)
      COMMON/VAR18/DD(2,8,53,33)
      COMMON/VAR20/NSX,NSXB,NSXA,NSY,NSYB,NSYA
      COMMON/VAR21/ACOM
      COMMON/VAR22/TRIGTEMP,PHI
      COMMON/VAR23/REN,PR,FACT1,LAMB,SDIFF
C
C                  ---------------
C ---              \  EAST FACE  \
C                  ---------------
```

```
C
      J           = NYY
      DO 20  I    = 2 , NXXX
      AV          = .5 * (AREA(I,J) + AREA(I+1,J))
C
C ---   U-VELOCITY
C
      UE          = UVEL(I+1,J)
      UW          = UVEL(I,J)
      UN          = 0.0
      US          = .25 * (UVEL(I+1,J) + UVEL(I,J)
     1                 +  UVEL(I,J-1) + UVEL(I+1,J-1))
C
C ---   V-VELOCITY
C
      VE          = VVEL(I+1,J)
      VW          = VVEL(I,J)
      VN          = 0.0
      VS          = .25 * (VVEL(I+1,J) + VVEL(I,J)
     1                 +  VVEL(I,J-1) + VVEL(I+1,J-1))
C
C ---   TEMPERATURE
C
      TE          = TEMP(I+1,J)
      TW          = TEMP(I,J)
      TN          = TEMP(I,NYY)
      TS          = .25 * (TEMP(I+1,J) + TEMP(I,J)
     1                 +  TEMP(I,J-1) + TEMP(I+1,J-1))
C
C ---   YH2
C
      Y1E          = YH2(I+1,J)
      Y1W          = YH2(I,J)
      Y1N          = YH2(I,NYY)
      Y1S          = .25 * (YH2(I+1,J) + YH2(I,J)
     1                  +  YH2(I,J-1) + YH2(I+1,J-1))
C
C ---   YO2
C
      Y2E          = YO2(I+1,J)
      Y2W          = YO2(I,J)
      Y2N          = YO2(I,NYY)
      Y2S          = .25 * (YO2(I+1,J) + YO2(I,J)
     1                  +  YO2(I,J-1) + YO2(I+1,J-1))
C
C ---   YH2O
C
      Y3E          = YH2O(I+1,J)
      Y3W          = YH2O(I,J)
      Y3N          = YH2O(I,NYY)
```

```
      Y3S          = .25 * (YH2O(I+1,J) + YH2O(I,J)
     1                   +  YH2O(I,J-1) + YH2O(I+1,J-1))
C
C ---   YYN2
C
      Y4E          = YYN2(I+1,J)
      Y4W          = YYN2(I,J)
      Y4N          = YYN2(I,NYY)
      Y4S          = .25 * (YYN2(I+1,J) + YYN2(I,J)
     1                   +  YYN2(I,J-1) + YYN2(I+1,J-1))
C
C ---   CALCULATE THE GRADIENT TERMS
C
C
C ---   X  GRADIENTS
C
      DUDX         = (UE * DYE(1,I,J) + UN * DYN(1,I,J)
     1              + UW * DYW(1,I,J) + US * DYS(1,I,J)) / AV
      DVDX         = (VE * DYE(1,I,J) + VN * DYN(1,I,J)
     1              + VW * DYW(1,I,J) + VS * DYS(1,I,J)) / AV
      DTDX         = (TE * DYE(1,I,J) + TN * DYN(1,I,J)
     1              + TW * DYW(1,I,J) + TS * DYS(1,I,J)) / AV
      DY1DX        = (Y1E * DYE(1,I,J) + Y1N * DYN(1,I,J)
     1              + Y1W * DYW(1,I,J) + Y1S * DYS(1,I,J)) / AV
      DY2DX        = (Y2E * DYE(1,I,J) + Y2N * DYN(1,I,J)
     1              + Y2W * DYW(1,I,J) + Y2S * DYS(1,I,J)) / AV
      DY3DX        = (Y3E * DYE(1,I,J) + Y3N * DYN(1,I,J)
     1              + Y3W * DYW(1,I,J) + Y3S * DYS(1,I,J)) / AV
      DY4DX        = (Y4E * DYE(1,I,J) + Y4N * DYN(1,I,J)
     1              + Y4W * DYW(1,I,J) + Y4S * DYS(1,I,J)) / AV
      DY5DX        = -DY1DX - DY2DX - DY3DX - DY4DX
C
C ---   Y  GRADIENTS
C
      DUDY         = - (UE * DXE(1,I,J) + UN * DXN(1,I,J)
     1                + UW * DXW(1,I,J) + US * DXS(1,I,J)) / AV
      DVDY         = - (VE * DXE(1,I,J) + VN * DXN(1,I,J)
     1                + VW * DXW(1,I,J) + VS * DXS(1,I,J)) / AV
      DTDY         = - (TE * DXE(1,I,J) + TN * DXN(1,I,J)
     1                + TW * DXW(1,I,J) + TS * DXS(1,I,J)) / AV
      DY1DY        = - (Y1E * DXE(1,I,J) + Y1N * DXN(1,I,J)
     1                + Y1W * DXW(1,I,J) + Y1S * DXS(1,I,J)) / AV
      DY2DY        = - (Y2E * DXE(1,I,J) + Y2N * DXN(1,I,J)
     1                + Y2W * DXW(1,I,J) + Y2S * DXS(1,I,J)) / AV
      DY3DY        = - (Y3E * DXE(1,I,J) + Y3N * DXN(1,I,J)
     1                + Y3W * DXW(1,I,J) + Y3S * DXS(1,I,J)) / AV
      DY4DY        = - (Y4E * DXE(1,I,J) + Y4N * DXN(1,I,J)
     1                + Y4W * DXW(1,I,J) + Y4S * DXS(1,I,J)) / AV
      DY5DY        = - DY1DY - DY2DY - DY3DY - DY4DY
C
```

```
C ---   COMPUTE THE FULL SHEAR STRESS TERMS
C
      TXX         = - LAMB * VIS(I,J) * (DUDX + DVDY)
                   - 2.0   * VIS(I,J) * DUDX / REN
      TXY         = - VIS(I,J) / REN   * (DUDY + DVDX)
      TYY         = - LAMB * VIS(I,J) * (DUDX + DVDY)
                   - 2.0   * VIS(I,J) * DVDY / REN
C
C ---   COMPUTE THE VISCOUS CONTRIBUTIONS TO "F" AND "G"
C
      FV(1,1,I,J)  = 0.0
      FV(1,2,I,J)  = TXX
      FV(1,3,I,J)  = TXY
      FV(1,4,I,J)  =   .5 * (UVEL(I,J) + UVEL(I+1,J))*TXX
     1                + .5 * (VVEL(I,J) + VVEL(I+1,J))*TXY
     2                - SDIFF * FACT1 * DTDX
     3                - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2
     4                + CPH2 * T1 * TEMP(I,J)) *  DY1DX
     5                - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFO2
     6                + CPO2 * T1 * TEMP(I,J)) *  DY2DX
     7                - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2O
     8                + CPH2O * T1 * TEMP(I,J)) *  DY3DX
     9                - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2
     10               + CPN2 * T1 * TEMP(I,J)) *  DY4DX
     11               - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFOH
     12               + CPOH * T1 * TEMP(I,J)) *  DY5DX
      FV(1,5,I,J)  = - SDIFF * DY1DX
      FV(1,6,I,J)  = - SDIFF * DY2DX
      FV(1,7,I,J)  = - SDIFF * DY3DX
      FV(1,8,I,J)  = - SDIFF * DY4DX
      GV(1,1,I,J)  = 0.0
      GV(1,2,I,J)  = TXY
      GV(1,3,I,J)  = TYY
      GV(1,4,I,J)  =   .5 * (VVEL(I,J) + VVEL(I+1,J))*TYY
     1                + .5 * (UVEL(I,J) + UVEL(I+1,J))*TXY
     2                - SDIFF * FACT1 * DTDY
     3                - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2
     4                + CPH2 * T1 * TEMP(I,J)) *  DY1DY
     5                - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFO2
     6                + CPO2 * T1 * TEMP(I,J)) *  DY2DY
     7                - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2O
     8                + CPH2O * T1 * TEMP(I,J)) *  DY3DY
     9                - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2
     10               + CPN2 * T1 * TEMP(I,J)) *  DY4DY
     11               - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFOH
     12               + CPOH * T1 * TEMP(I,J)) *  DY5DY
      GV(1,5,I,J)  = - SDIFF * DY1DY
      GV(1,6,I,J)  = - SDIFF * DY2DY
      GV(1,7,I,J)  = - SDIFF * DY3DY
      GV(1,8,I,J)  = - SDIFF * DY4DY
```

```
20      CONTINUE
C
C                      ------------------
C ---                  \   NORTH  FACE  \
C                      ------------------
C
        J              = NYY
        DO 30  I       = 2 , NXXX
        AV             = AREA(I,J)
C
C ---    U-VELOCITY
C
        UE             = 0.0
        UN             = -UVEL(I,J)
        UW             = 0.0
        US             = UVEL(I,J)
C
C ---    V-VELOCITY
C
        VE             = 0.0
        VN             = -VVEL(I,J)
        VW             = 0.0
        VS             = VVEL(I,J)
C
C ---    TEMPERATURE
C
        TE             = TEMP(I,J)
        TN             = TEMP(I,J)
        TW             = TEMP(I,J)
        TS             = TEMP(I,J)
C
C ---    YH2
C
        Y1E             = YH2(I,J)
        Y1N             = YH2(I,J)
        Y1W             = YH2(I,J)
        Y1S             = YH2(I,J)
C
C ---    YO2
C
        Y2E             = YO2(I,J)
        Y2N             = YO2(I,J)
        Y2W             = YO2(I,J)
        Y2S             = YO2(I,J)
C
C ---    YH2O
C
        Y3E             = YH2O(I,J)
        Y3N             = YH2O(I,J)
        Y3W             = YH2O(I,J)
```

```
      Y3S          = YH2O(I,J)
C
C ---   YYN2
C
      Y4E          = YYN2(I,J)
      Y4N          = YYN2(I,J)
      Y4W          = YYN2(I,J)
      Y4S          = YYN2(I,J)
C
C ---   CALCULATE THE GRADIENT TERMS
C
C
C ---   X  GRADIENTS
C
      DUDX        = (UE * DYE(2,I,J) + UN * DYN(2,I,J)
     1              + UW * DYW(2,I,J) + US * DYS(2,I,J)) / AV
      DVDX        = (VE * DYE(2,I,J) + VN * DYN(2,I,J)
     1              + VW * DYW(2,I,J) + VS * DYS(2,I,J)) / AV
      DTDX        = (TE * DYE(2,I,J) + TN * DYN(2,I,J)
     1              + TW * DYW(2,I,J) + TS * DYS(2,I,J)) / AV
      DY1DX       = (Y1E * DYE(2,I,J) + Y1N * DYN(2,I,J)
     1              + Y1W * DYW(2,I,J) + Y1S * DYS(2,I,J)) / AV
      DY2DX       = (Y2E * DYE(2,I,J) + Y2N * DYN(2,I,J)
     1              + Y2W * DYW(2,I,J) + Y2S * DYS(2,I,J)) / AV
      DY3DX       = (Y3E * DYE(2,I,J) + Y3N * DYN(2,I,J)
     1              + Y3W * DYW(2,I,J) + Y3S * DYS(2,I,J)) / AV
      DY4DX       = (Y4E * DYE(2,I,J) + Y4N * DYN(2,I,J)
     1              + Y4W * DYW(2,I,J) + Y4S * DYS(2,I,J)) / AV
      DY5DX       = -DY1DX - DY2DX - DY3DX - DY4DX
C
C ---   Y  GRADIENTS
C
      DUDY        = - (UE * DXE(2,I,J) + UN * DXN(2,I,J)
     1                + UW * DXW(2,I,J) + US * DXS(2,I,J)) / AV
      DVDY        = - (VE * DXE(2,I,J) + VN * DXN(2,I,J)
     1                + VW * DXW(2,I,J) + VS * DXS(2,I,J)) / AV
      DTDY        = - (TE * DXE(2,I,J) + TN * DXN(2,I,J)
     1                + TW * DXW(2,I,J) + TS * DXS(2,I,J)) / AV
      DY1DY       = - (Y1E * DXE(2,I,J) + Y1N * DXN(2,I,J)
     1                + Y1W * DXW(2,I,J) + Y1S * DXS(2,I,J)) / AV
      DY2DY       = - (Y2E * DXE(2,I,J) + Y2N * DXN(2,I,J)
     1                + Y2W * DXW(2,I,J) + Y2S * DXS(2,I,J)) / AV
      DY3DY       = - (Y3E * DXE(2,I,J) + Y3N * DXN(2,I,J)
     1                + Y3W * DXW(2,I,J) + Y3S * DXS(2,I,J)) / AV
      DY4DY       = - (Y4E * DXE(2,I,J) + Y4N * DXN(2,I,J)
     1                + Y4W * DXW(2,I,J) + Y4S * DXS(2,I,J)) / AV
      DY5DY       = - DY1DY - DY2DY - DY3DY - DY4DY
C
C ---   COMPUTE THE FULL SHEAR STRESS TERMS
C
```

```
      TXX          = - LAMB * VIS(I,J) * (DUDX + DVDY)
                      - 2.0  * VIS(I,J) * DUDX / REN
      TXY          = - VIS(I,J) / REN  * (DUDY + DVDX)
      TYY          = - LAMB * VIS(I,J) * (DUDX + DVDY)
                      - 2.0  * VIS(I,J) * DVDY / REN
C
C ---   COMPUTE THE VISCOUS CONTRIBUTIONS TO "F" AND "G"
C
      FV(2,1,I,J)  = 0.0
      FV(2,2,I,J)  = TXX
      FV(2,3,I,J)  = TXY
      FV(2,4,I,J)  =    - SDIFF * FACT1 * DTDX
     3                  - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2
     4                  + CPH2 * T1 * TEMP(I,J)) *  DY1DX
     5                  - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFO2
     6                  + CPO2 * T1 * TEMP(I,J)) *  DY2DX
     7                  - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2O
     8                  + CPH2O * T1 * TEMP(I,J)) *  DY3DX
     9                  - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2
     10                 + CPN2 * T1 * TEMP(I,J)) *  DY4DX
     11                 - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFOH
     12                 + CPOH * T1 * TEMP(I,J)) *  DY5DX
      FV(2,5,I,J)  = - SDIFF * DY1DX
      FV(2,6,I,J)  = - SDIFF * DY2DX
      FV(2,7,I,J)  = - SDIFF * DY3DX
      FV(2,8,I,J)  = - SDIFF * DY4DX
      GV(2,1,I,J)  = 0.0
      GV(2,2,I,J)  = TXY
      GV(2,3,I,J)  = TYY
      GV(2,4,I,J)  =    - SDIFF * FACT1 * DTDY
     3                  - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2
     4                  + CPH2 * T1 * TEMP(I,J)) *  DY1DY
     5                  - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFO2
     6                  + CPO2 * T1 * TEMP(I,J)) *  DY2DY
     7                  - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2O
     8                  + CPH2O * T1 * TEMP(I,J)) *  DY3DY
     9                  - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2
     10                 + CPN2 * T1 * TEMP(I,J)) *  DY4DY
     11                 - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFOH
     12                 + CPOH * T1 * TEMP(I,J)) *  DY5DY
      GV(2,5,I,J)  = - SDIFF * DY1DY
      GV(2,6,I,J)  = - SDIFF * DY2DY
      GV(2,7,I,J)  = - SDIFF * DY3DY
      GV(2,8,I,J)  = - SDIFF * DY4DY
30    CONTINUE
C
C
C                     ---------------
C ---                 \  WEST FACE  \
C                     ---------------
```

```
C
      J           = NYY
      DO 40  I    = 2 , NXXX
      AV          = .5 * (AREA(I,J) + AREA(I-1,J))
C
C ---   U-VELOCITY
C
      UE          = UVEL(I,J)
      UW          = UVEL(I-1,J)
      UN          = 0.0
      US          = .25 * (UVEL(I,J)     + UVEL(I-1,J)
     1                  +  UVEL(I-1,J-1) + UVEL(I,J-1))
C
C ---   V-VELOCITY
C
      VE          = VVEL(I,J)
      VW          = VVEL(I-1,J)
      VN          = 0.0
      VS          = .25 * (VVEL(I,J)     + VVEL(I-1,J)
     1                  +  VVEL(I-1,J-1) + VVEL(I,J-1))
C
C ---   TEMPERATURE
C
      TE          = TEMP(I,J)
      TW          = TEMP(I-1,J)
      TN          = TEMP(I,J)
      TS          = .25 * (TEMP(I,J)     + TEMP(I-1,J)
     1                  +  TEMP(I-1,J-1) + TEMP(I,J-1))
C
C ---   YH2
C
      Y1E          = YH2(I,J)
      Y1W          = YH2(I-1,J)
      Y1N          = YH2(I,J)
      Y1S          = .25 * (YH2(I,J)     + YH2(I-1,J)
     1                   +  YH2(I-1,J-1) + YH2(I,J-1))
C
C ---   YO2
C
      Y2E          = YO2(I,J)
      Y2W          = YO2(I-1,J)
      Y2N          = YO2(I,J)
      Y2S          = .25 * (YO2(I,J)     + YO2(I-1,J)
     1                   +  YO2(I-1,J-1) + YO2(I,J-1))
C
C ---   YH2O
C
      Y3E          = YH2O(I,J)
      Y3W          = YH2O(I-1,J)
      Y3N          = YH2O(I,J)
```

```
      Y3S          = .25 * (YH2O(I,J)      + YH2O(I-1,J)
     1                   +  YH2O(I-1,J-1) + YH2O(I,J-1))
C
C ---   YYN2
C
      Y4E          = YYN2(I,J)
      Y4W          = YYN2(I-1,J)
      Y4N          = YYN2(I,J)
      Y4S          = .25 * (YYN2(I,J)      + YYN2(I-1,J)
     1                   +  YYN2(I-1,J-1) + YYN2(I,J-1))
C
C ---   CALCULATE THE GRADIENT TERMS
C
C
C ---   X  GRADIENTS
C
      DUDX        = (UE * DYE(3,I,J) + UN * DYN(3,I,J)
     1             + UW * DYW(3,I,J) + US * DYS(3,I,J)) / AV
      DVDX        = (VE * DYE(3,I,J) + VN * DYN(3,I,J)
     1             + VW * DYW(3,I,J) + VS * DYS(3,I,J)) / AV
      DTDX        = (TE * DYE(3,I,J) + TN * DYN(3,I,J)
     1             + TW * DYW(3,I,J) + TS * DYS(3,I,J)) / AV
      DY1DX       = (Y1E * DYE(3,I,J) + Y1N * DYN(3,I,J)
     1             + Y1W * DYW(3,I,J) + Y1S * DYS(3,I,J)) / AV
      DY2DX       = (Y2E * DYE(3,I,J) + Y2N * DYN(3,I,J)
     1             + Y2W * DYW(3,I,J) + Y2S * DYS(3,I,J)) / AV
      DY3DX       = (Y3E * DYE(3,I,J) + Y3N * DYN(3,I,J)
     1             + Y3W * DYW(3,I,J) + Y3S * DYS(3,I,J)) / AV
      DY4DX       = (Y4E * DYE(3,I,J) + Y4N * DYN(3,I,J)
     1             + Y4W * DYW(3,I,J) + Y4S * DYS(3,I,J)) / AV
      DY5DX       = -DY1DX - DY2DX - DY3DX - DY4DX
C
C ---   Y  GRADIENTS
C
      DUDY        = - (UE * DXE(3,I,J) + UN * DXN(3,I,J)
     1               + UW * DXW(3,I,J) + US * DXS(3,I,J)) / AV
      DVDY        = - (VE * DXE(3,I,J) + VN * DXN(3,I,J)
     1               + VW * DXW(3,I,J) + VS * DXS(3,I,J)) / AV
      DTDY        = - (TE * DXE(3,I,J) + TN * DXN(3,I,J)
     1               + TW * DXW(3,I,J) + TS * DXS(3,I,J)) / AV
      DY1DY       = - (Y1E * DXE(3,I,J) + Y1N * DXN(3,I,J)
     1               + Y1W * DXW(3,I,J) + Y1S * DXS(3,I,J)) / AV
      DY2DY       = - (Y2E * DXE(3,I,J) + Y2N * DXN(3,I,J)
     1               + Y2W * DXW(3,I,J) + Y2S * DXS(3,I,J)) / AV
      DY3DY       = - (Y3E * DXE(3,I,J) + Y3N * DXN(3,I,J)
     1               + Y3W * DXW(3,I,J) + Y3S * DXS(3,I,J)) / AV
      DY4DY       = - (Y4E * DXE(3,I,J) + Y4N * DXN(3,I,J)
     1               + Y4W * DXW(3,I,J) + Y4S * DXS(3,I,J)) / AV
      DY5DY       = - DY1DY - DY2DY - DY3DY - DY4DY
C
```

```
C ---   COMPUTE THE FULL SHEAR STRESS TERMS
C
      TXX          = - LAMB * VIS(I,J) * (DUDX + DVDY)
                      - 2.0  * VIS(I,J) * DUDX / REN
      TXY          = - VIS(I,J) / REN  * (DUDY + DVDX)
      TYY          = - LAMB * VIS(I,J) * (DUDX + DVDY)
                      - 2.0  * VIS(I,J) * DVDY / REN
C
C ---   COMPUTE THE VISCOUS CONTRIBUTIONS TO "F" AND "G"
C
      FV(3,1,I,J)  = 0.0
      FV(3,2,I,J)  = TXX
      FV(3,3,I,J)  = TXY
      FV(3,4,I,J)  =   .5 * (UVEL(I,J) + UVEL(I-1,J))*TXX
     1                 + .5 * (VVEL(I,J) + VVEL(I-1,J))*TXY
     2                 - SDIFF * FACT1 * DTDX
     3                 - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2
     4                 + CPH2 * T1 * TEMP(I,J)) *  DY1DX
     5                 - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFO2
     6                 + CPO2 * T1 * TEMP(I,J)) *  DY2DX
     7                 - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2O
     8                 + CPH2O * T1 * TEMP(I,J)) *  DY3DX
     9                 - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2
     10                + CPN2 * T1 * TEMP(I,J)) *  DY4DX
     11                - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFOH
     12                + CPOH * T1 * TEMP(I,J)) *  DY5DX
      FV(3,5,I,J)  = - SDIFF * DY1DX
      FV(3,6,I,J)  = - SDIFF * DY2DX
      FV(3,7,I,J)  = - SDIFF * DY3DX
      FV(3,8,I,J)  = - SDIFF * DY4DX
      GV(3,1,I,J)  = 0.0
      GV(3,2,I,J)  = TXY
      GV(3,3,I,J)  = TYY
      GV(3,4,I,J)  =   .5 * (VVEL(I,J) + VVEL(I-1,J))*TYY
     1                 + .5 * (UVEL(I,J) + UVEL(I-1,J))*TXY
     2                 - SDIFF * FACT1 * DTDY
     3                 - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2
     4                 + CPH2 * T1 * TEMP(I,J)) *  DY1DY
     5                 - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFO2
     6                 + CPO2 * T1 * TEMP(I,J)) *  DY2DY
     7                 - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2O
     8                 + CPH2O * T1 * TEMP(I,J)) *  DY3DY
     9                 - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2
     10                + CPN2 * T1 * TEMP(I,J)) *  DY4DY
     11                - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFOH
     12                + CPOH * T1 * TEMP(I,J)) *  DY5DY
      GV(3,5,I,J)  = - SDIFF * DY1DY
      GV(3,6,I,J)  = - SDIFF * DY2DY
      GV(3,7,I,J)  = - SDIFF * DY3DY
      GV(3,8,I,J)  = - SDIFF * DY4DY
```

```
40      CONTINUE
C
C
C                       ----------------
C ---                   \  SOUTH FACE  \
C                       ----------------
C
        J               = NYY
        DO 50  I        = 2 , NXXX
        AV              = .5 * (AREA(I,J) + AREA(I,J-1))
C
C ---    U-VELOCITY
C
        UE              = .25 * (UVEL(I+1,J)    + UVEL(I,J)
     1                        +  UVEL(I,J-1)    + UVEL(I+1,J-1))
        UN              = UVEL(I,J)
        UW              = .25 * (UVEL(I,J)      + UVEL(I-1,J)
     1                        +  UVEL(I-1,J-1)  + UVEL(I,J-1))
        US              = UVEL(I,J-1)
C
C ---    V-VELOCITY
C
        VE              = .25 * (VVEL(I+1,J)    + VVEL(I,J)
     1                        +  VVEL(I,J-1)    + VVEL(I+1,J-1))
        VN              = VVEL(I,J)
        VW              = .25 * (VVEL(I,J)      + VVEL(I-1,J)
     1                        +  VVEL(I-1,J-1)  + VVEL(I,J-1))
        VS              = VVEL(I,J-1)
C
C ---    TEMPERATURE
C
        TE              = .25 * (TEMP(I+1,J)    + TEMP(I,J)
     1                        +  TEMP(I,J-1)    + TEMP(I+1,J-1))
        TN              = TEMP(I,J)
        TW              = .25 * (TEMP(I,J)      + TEMP(I-1,J)
     1                        +  TEMP(I-1,J-1)  + TEMP(I,J-1))
        TS              = TEMP(I,J-1)
C
C ---    YH2
C
        Y1E              = .25 * (YH2(I+1,J)    + YH2(I,J)
     1                         +  YH2(I,J-1)    + YH2(I+1,J-1))
        Y1N              = YH2(I,J)
        Y1W              = .25 * (YH2(I,J)      + YH2(I-1,J)
     1                         +  YH2(I-1,J-1)  + YH2(I,J-1))
        Y1S              = YH2(I,J-1)
C
C ---    YO2
C
        Y2E              = .25 * (YO2(I+1,J)    + YO2(I,J)
```

```
     1                  +  YO2(I,J-1)    + YO2(I+1,J-1))
       Y2N          = YO2(I,J)
       Y2W          = .25 * (YO2(I,J)      + YO2(I-1,J)
     1                  +  YO2(I-1,J-1) + YO2(I,J-1))
       Y2S          = YO2(I,J-1)
C
C ---   YH2O
C
       Y3E          = .25 * (YH2O(I+1,J)    + YH2O(I,J)
     1                  +  YH2O(I,J-1)    + YH2O(I+1,J-1))
       Y3N          = YH2O(I,J)
       Y3W          = .25 * (YH2O(I,J)      + YH2O(I-1,J)
     1                  +  YH2O(I-1,J-1) + YH2O(I,J-1))
       Y3S          = YH2O(I,J-1)
C
C ---   YYN2
C
       Y4E          = .25 * (YYN2(I+1,J)    + YYN2(I,J)
     1                  +  YYN2(I,J-1)    + YYN2(I+1,J-1))
       Y4N          = YYN2(I,J)
       Y4W          = .25 * (YYN2(I,J)      + YYN2(I-1,J)
     1                  +  YYN2(I-1,J-1) + YYN2(I,J-1))
       Y4S          = YYN2(I,J-1)
C
C ---   CALCULATE THE GRADIENT TERMS
C
C
C ---   X  GRADIENTS
C
       DUDX        = (UE * DYE(4,I,J) + UN * DYN(4,I,J)
     1              + UW * DYW(4,I,J) + US * DYS(4,I,J)) / AV
       DVDX        = (VE * DYE(4,I,J) + VN * DYN(4,I,J)
     1              + VW * DYW(4,I,J) + VS * DYS(4,I,J)) / AV
       DTDX        = (TE * DYE(4,I,J) + TN * DYN(4,I,J)
     1              + TW * DYW(4,I,J) + TS * DYS(4,I,J)) / AV
       DY1DX       = (Y1E * DYE(4,I,J) + Y1N * DYN(4,I,J)
     1              + Y1W * DYW(4,I,J) + Y1S * DYS(4,I,J)) / AV
       DY2DX       = (Y2E * DYE(4,I,J) + Y2N * DYN(4,I,J)
     1              + Y2W * DYW(4,I,J) + Y2S * DYS(4,I,J)) / AV
       DY3DX       = (Y3E * DYE(4,I,J) + Y3N * DYN(4,I,J)
     1              + Y3W * DYW(4,I,J) + Y3S * DYS(4,I,J)) / AV
       DY4DX       = (Y4E * DYE(4,I,J) + Y4N * DYN(4,I,J)
     1              + Y4W * DYW(4,I,J) + Y4S * DYS(4,I,J)) / AV
       DY5DX       = -DY1DX - DY2DX - DY3DX - DY4DX
C
C ---   Y  GRADIENTS
C
       DUDY        = - (UE * DXE(4,I,J) + UN * DXN(4,I,J)
     1                + UW * DXW(4,I,J) + US * DXS(4,I,J)) / AV
       DVDY        = - (VE * DXE(4,I,J) + VN * DXN(4,I,J)
```

```
     1              + VW * DXW(4,I,J) + VS * DXS(4,I,J)) / AV
       DTDY        = - (TE * DXE(4,I,J) + TN * DXN(4,I,J)
     1              + TW * DXW(4,I,J) + TS * DXS(4,I,J)) / AV
       DY1DY       = - (Y1E * DXE(4,I,J) + Y1N * DXN(4,I,J)
     1              + Y1W * DXW(4,I,J) + Y1S * DXS(4,I,J)) / AV
       DY2DY       = - (Y2E * DXE(4,I,J) + Y2N * DXN(4,I,J)
     1              + Y2W * DXW(4,I,J) + Y2S * DXS(4,I,J)) / AV
       DY3DY       = - (Y3E * DXE(4,I,J) + Y3N * DXN(4,I,J)
     1              + Y3W * DXW(4,I,J) + Y3S * DXS(4,I,J)) / AV
       DY4DY       = - (Y4E * DXE(4,I,J) + Y4N * DXN(4,I,J)
     1              + Y4W * DXW(4,I,J) + Y4S * DXS(4,I,J)) / AV
       DY5DY       = - DY1DY - DY2DY - DY3DY - DY4DY
C
C ---   COMPUTE THE FULL SHEAR STRESS TERMS
C
       TXX         = - LAMB * VIS(I,J) * (DUDX + DVDY)
                     - 2.0  * VIS(I,J) * DUDX / REN
       TXY         = - VIS(I,J) / REN  * (DUDY + DVDX)
       TYY         = - LAMB * VIS(I,J) * (DUDX + DVDY)
                     - 2.0  * VIS(I,J) * DVDY / REN
C
C ---   COMPUTE THE VISCOUS CONTRIBUTIONS TO "F" AND "G"
C
       FV(4,1,I,J)  = 0.0
       FV(4,2,I,J)  = TXX
       FV(4,3,I,J)  = TXY
       FV(4,4,I,J)  =   .5 * (UVEL(I,J) + UVEL(I,J-1))*TXX
     1                 + .5 * (VVEL(I,J) + VVEL(I,J-1))*TXY
     2                 - SDIFF * FACT1 * DTDX
     3                 - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2
     4                 + CPH2 * T1 * TEMP(I,J)) *  DY1DX
     5                 - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFO2
     6                 + CPO2 * T1 * TEMP(I,J)) *  DY2DX
     7                 - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2O
     8                 + CPH2O * T1 * TEMP(I,J)) *  DY3DX
     9                 - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2
    10                 + CPN2 * T1 * TEMP(I,J)) *  DY4DX
    11                 - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFOH
    12                 + CPOH * T1 * TEMP(I,J)) *  DY5DX
       FV(4,5,I,J)  = - SDIFF * DY1DX
       FV(4,6,I,J)  = - SDIFF * DY2DX
       FV(4,7,I,J)  = - SDIFF * DY3DX
       FV(4,8,I,J)  = - SDIFF * DY4DX
       GV(4,1,I,J)  = 0.0
       GV(4,2,I,J)  = TXY
       GV(4,3,I,J)  = TYY
       GV(4,4,I,J)  =   .5 * (VVEL(I,J) + VVEL(I,J-1))*TYY
     1                 + .5 * (UVEL(I,J) + UVEL(I,J-1))*TXY
     2                 - SDIFF * FACT1 * DTDY
     3                 - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2
```

```
     4                    + CPH2 * T1 * TEMP(I,J)) *  DY1DY
     5                    - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFO2
     6                    + CPO2 * T1 * TEMP(I,J)) *  DY2DY
     7                    - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2O
     8                    + CPH2O * T1 * TEMP(I,J)) *  DY3DY
     9                    - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2
     10                   + CPN2 * T1 * TEMP(I,J)) *  DY4DY
     11                   - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFOH
     12                   + CPOH * T1 * TEMP(I,J)) *  DY5DY
      GV(4,5,I,J)  = - SDIFF * DY1DY
      GV(4,6,I,J)  = - SDIFF * DY2DY
      GV(4,7,I,J)  = - SDIFF * DY3DY
      GV(4,8,I,J)  = - SDIFF * DY4DY
50    CONTINUE
100   CONTINUE
      RETURN
      END
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      SUBROUTINE CONV
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C ---  DETERMINE THE CONVERGENCE HISTORY OF THE SOLUTION METHOD
C
      COMMON/VAR0/U(8,53,33,2),FI(8,53,33),GI(8,53,33),AH(8,53,33)
      COMMON/VAR00/FV(4,8,53,33),GV(4,8,53,33)
      COMMON/VAR1/UVEL(53,33),VVEL(53,33),PRES(53,33),TEMP(53,33)
      COMMON/VAR111/YH2(53,33),YO2(53,33),YH2O(53,33),YOH(53,33)
      COMMON/VAR2/DEN(53,33),SOUND(53,33),AINTE(53,33),AMACH(53,33)
      COMMON/VAR3/ENTHP(53,33),VIS(53,33),YYN2(53,33),CPND(53,33)
      COMMON/VAR4/DXX,DYY,X(55,35),Y(55,35),AREA(53,33)
      COMMON/VAR5/DT(53,33)
      COMMON/VAR6/NX,NXX,NXXX,NY,NYY,NYYY,IRES,IEQ
      COMMON/VAR7/IVIS,NITER,NOITER
      COMMON/VAR8/P1,T1,AM1,VISL,U1,V1,AK1,CV,CP,R,GAMA,DEN1,E1
      COMMON/VAR9/P11,C11,E11
      COMMON/VAR10/DFH2,DFO2,DFH2O,DFOH,DFN2
      COMMON/VAR11/CPH2,CPO2,CPH2O,CPOH,CPN2,CVH2,CVO2,CVH2O,CVOH,CVN2
      COMMON/VAR12/CONH2,CONO2,CONH2O,CONOH,CONN2
      COMMON/VAR13/COND,CFL,DCOFF,AL,VELO1
      COMMON/VAR14/RES(8,53,33)
      COMMON/VAR15/ALPHA(4)
      COMMON/VAR16/DXE(4,53,33),DXN(4,53,33),DXW(4,53,33),DXS(4,53,33)
      COMMON/VAR17/DYE(4,53,33),DYN(4,53,33),DYW(4,53,33),DYS(4,53,33)
      COMMON/VAR18/DD(2,8,53,33)
      COMMON/VAR20/NSX,NSXB,NSXA,NSY,NSYB,NSYA
      COMMON/VAR21/ACOM
      COMMON/VAR22/TRIGTEMP,PHI
      COMMON/VAR23/REN,PR,FACT1,LAMB,SDIFF
      NXX      = NX - 1
      NYY      = NY - 1
```

```
      IRES    = 0
      TRES1   = 0.0
      TRES2   = 0.0
      TRES3   = 0.0
      TRES4   = 0.0
      TRES5   = 0.0
      TRES6   = 0.0
      TRES7   = 0.0
      NPT     = NXX * NYY  +  NSX * NSY
      DO 1 I  = 1 , NXX
      DO 1 J  = 1 , NYY
      RES1    = (U(1,I,J,2) - U(1,I,J,1)) / DT(I,J)
      RES2    = (U(2,I,J,2) - U(2,I,J,1)) / DT(I,J)
      RES3    = (U(3,I,J,2) - U(3,I,J,1)) / DT(I,J)
      RES4    = (U(4,I,J,2) - U(4,I,J,1)) / DT(I,J)
      RES5    = (U(5,I,J,2) - U(5,I,J,1)) / DT(I,J)
      RES6    = (U(6,I,J,2) - U(6,I,J,1)) / DT(I,J)
      RES7    = (U(7,I,J,2) - U(7,I,J,1)) / DT(I,J)
      TRES1   = TRES1 + ABS(RES1)
      TRES2   = TRES2 + ABS(RES2)
      TRES3   = TRES3 + ABS(RES3)
      TRES4   = TRES4 + ABS(RES4)
      TRES5   = TRES5 + ABS(RES5)
      TRES6   = TRES6 + ABS(RES6)
      TRES7   = TRES7 + ABS(RES7)
C
1     CONTINUE
C
      TRES1   = TRES1 / NPT
      TRES2   = TRES2 / NPT
      TRES3   = TRES3 / NPT
      TRES4   = TRES4 / NPT
      TRES5   = TRES5 / NPT
      TRES6   = TRES6 / NPT
      TRES7   = TRES7 / NPT
      TRES8   = TRES8 / NPT
C
C ---  PRINT OUT THE RESIDUAL HISTORIES FOR EACH ITERATION
C
      WRITE(6,10)TRES1,TRES2,TRES3,TRES4,TRES5,TRES6,TRES7,
     .          TRES8
10    FORMAT(2X,8(E10.4))
C
C????????????????????????????????????????????????????????????
C
C ---   CONVERGENCE TEST
C
C????????????????????????????????????????????????????????????
      IRES   = 0
      IF(TRES1.GT.RESCONV) IRES = 1
```

```
C?????????????????????????????????????????????????????????
      RETURN
      END
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      SUBROUTINE OUT
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C --- LOAD OUTPUT DATA FILES
C
      COMMON/VAR0/U(8,53,33,2),FI(8,53,33),GI(8,53,33),AH(8,53,33)
      COMMON/VAR00/FV(4,8,53,33),GV(4,8,53,33)
      COMMON/VAR1/UVEL(53,33),VVEL(53,33),PRES(53,33),TEMP(53,33)
      COMMON/VAR111/YH2(53,33),YO2(53,33),YH2O(53,33),YOH(53,33)
      COMMON/VAR2/DEN(53,33),SOUND(53,33),AINTE(53,33),AMACH(53,33)
      COMMON/VAR3/ENTHP(53,33),VIS(53,33),YYN2(53,33),CPND(53,33)
      COMMON/VAR4/DXX,DYY,X(55,35),Y(55,35),AREA(53,33)
      COMMON/VAR5/DT(53,33)
      COMMON/VAR6/NX,NXX,NXXX,NY,NYY,NYYY,IRES,IEQ
      COMMON/VAR7/IVIS,NITER,NOITER
      COMMON/VAR8/P1,T1,AM1,VISL,U1,V1,AK1,CV,CP,R,GAMA,DEN1,E1
      COMMON/VAR9/P11,C11,E11
      COMMON/VAR10/DFH2,DFO2,DFH2O,DFOH,DFN2
      COMMON/VAR11/CPH2,CPO2,CPH2O,CPOH,CPN2,CVH2,CVO2,CVH2O,CVOH,CVN2
      COMMON/VAR12/CONH2,CONO2,CONH2O,CONOH,CONN2
      COMMON/VAR13/COND,CFL,DCOFF,AL,VELO1
      COMMON/VAR14/RES(8,53,33)
      COMMON/VAR15/ALPHA(4)
      COMMON/VAR16/DXE(4,53,33),DXN(4,53,33),DXW(4,53,33),DXS(4,53,33)
      COMMON/VAR17/DYE(4,53,33),DYN(4,53,33),DYW(4,53,33),DYS(4,53,33)
      COMMON/VAR18/DD(2,8,53,33)
      COMMON/VAR20/NSX,NSXB,NSXA,NSY,NSYB,NSYA
      COMMON/VAR21/ACOM
      COMMON/VAR22/TRIGTEMP,PHI
      COMMON/VAR23/REN,PR,FACT1,LAMB,SDIFF
C
C --- NUMBER OF ITERATIONS
C
     WRITE(6,8)NOITER
8    FORMAT(2X,I4)
C
C     GRID METRIX AND REACTION RATES
C
       DO 6 J = 1 , NY
       DO 6 I = 1 , NX
6      WRITE(6,7)X(I,J),Y(I,J),AH(5,I,J),AH(6,I,J),AH(7,I,J)
7      FORMAT(2X,5(E10.4))
C
C ---  LOAD RESTART FILE " DSTEP.DAT "
C
      DO 5 J = 1 , NYY
```

```
      DO 5 I = 1 , NXX
      WRITE(6,14)U(1,I,J,2),U(2,I,J,2),U(3,I,J,2),U(4,I,J,2),
     1           U(5,I,J,2),U(6,I,J,2),U(7,I,J,2),U(8,I,J,2)
5     CONTINUE
14    FORMAT(2X,8(E10.4))
C
      DO 3000 I = 1 , NXX
      DO 3000 J = 1 , NYY
      RESN1 = ABS(U(1,I,J,2) - U(1,I,J,1)) / DT(I,J)
      RESN5 = ABS(U(5,I,J,2) - U(5,I,J,1)) / DT(I,J)
      WRITE(6,3001)RESN1,RESN5
3000  CONTINUE
3001  FORMAT(2X,2(E10.4))
      RETURN
      END
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      SUBROUTINE SOURCE
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C ---          COMPUTE THE SOURCE TERMS
C ---     THE H2 - O2 CHEMISTRY SOURCE TERMS ARE AS
C ---     DESCRIBED BY ROGERS/CHINITZ (AIAA-82-0112)
C
      COMMON/VAR0/U(8,53,33,2),FI(8,53,33),GI(8,53,33),AH(8,53,33)
      COMMON/VAR00/FV(4,8,53,33),GV(4,8,53,33)
      COMMON/VAR1/UVEL(53,33),VVEL(53,33),PRES(53,33),TEMP(53,33)
      COMMON/VAR111/YH2(53,33),YO2(53,33),YH2O(53,33),YOH(53,33)
      COMMON/VAR2/DEN(53,33),SOUND(53,33),AINTE(53,33),AMACH(53,33)
      COMMON/VAR3/ENTHP(53,33),VIS(53,33),YYN2(53,33),CPND(53,33)
      COMMON/VAR4/DXX,DYY,X(55,35),Y(55,35),AREA(53,33)
      COMMON/VAR5/DT(53,33)
      COMMON/VAR6/NX,NXX,NXXX,NY,NYY,NYYY,IRES,IEQ
      COMMON/VAR7/IVIS,NITER,NOITER
      COMMON/VAR8/P1,T1,AM1,VISL,U1,V1,AK1,CV,CP,R,GAMA,DEN1,E1
      COMMON/VAR9/P11,C11,E11
      COMMON/VAR10/DFH2,DFO2,DFH2O,DFOH,DFN2
      COMMON/VAR11/CPH2,CPO2,CPH2O,CPOH,CPN2,CVH2,CVO2,CVH2O,CVOH,CVN2
      COMMON/VAR12/CONH2,CONO2,CONH2O,CONOH,CONN2
      COMMON/VAR13/COND,CFL,DCOFF,AL,VELO1
      COMMON/VAR14/RES(8,53,33)
      COMMON/VAR15/ALPHA(4)
      COMMON/VAR16/DXE(4,53,33),DXN(4,53,33),DXW(4,53,33),DXS(4,53,33)
      COMMON/VAR17/DYE(4,53,33),DYN(4,53,33),DYW(4,53,33),DYS(4,53,33)
      COMMON/VAR18/DD(2,8,53,33)
      COMMON/VAR20/NSX,NSXB,NSXA,NSY,NSYB,NSYA
      COMMON/VAR21/ACOM
      COMMON/VAR22/TRIGTEMP,PHI
      COMMON/VAR23/REN,PR,FACT1,LAMB,SDIFF
C
C ---     GAS PROPERITY INPUT
```

```
C
      AWH2    =  2.0E-3
      AWO2    =  32.0E-3
      AWH2O   =  18.0E-3
      AWOH    =  17.0E-3
      AWN2    =  28.0E-3
      RUCGS   = 1.987
      WSCALE  = AL / (VELO1 * DEN1)
C
C ---    SWEEP THROUGH ALL GRID POINTS TO COMPUTE "WDOT"
C
      FACT4   = ACOM * (1.0E+21/(T1**5)) * (1.0E+20/(T1**5))
      FACT5   = ACOM * (1.0E+20/(T1**5)) * (1.0E+20/(T1**5))
     1                * (1.0E+12/(T1**3))
      S5      = 1.0
C
C
C ---    INITIALIZE CHEMICAL SOURCE TERM ARRAYS
C
      DO 100 K  = 1 , IEQ
      DO 100 J  = 1 , NYY
      DO 100 I  = 1 , NXX
      AH(K,I,J) = 0.0
100   CONTINUE
C
C ---    TEST IF CASE IS REACTING OR NONREACTING- DO ONLY IF
C ---    REACTING CASE
C
      IF(ACOM.EQ.0.0) GO TO 500
C
      IF(NOITER.GT.200) GO TO 977
C
C
C ---  IGNITION TRIGGER - REMOVE FOR FLAT PLATE CALCULATIONS
C
      DO 978 IS = 11 , 17
      DO 978 JS = 1   , 7
978   TEMP(IS,JS) = 2.5
977   CONTINUE
C
      DO 1 J  = 1 , NYY
      DO 1 I  = 1 , NXX
      ATEMP = TEMP(I,J) * T1
      IF(ATEMP.LE.TRIGTEMP) GO TO 1
C
C ---    EXPONENTIAL FACTOR
C
      EQUIL4  = 26.164 * EXP( -8992. / (TEMP(I,J) * T1))
      EQUIL5  = 3.269E-8 * EXP( 69415. / (TEMP(I,J) * T1))
      CONST4  = - 4865. /(1.987 * T1 * TEMP(I,J))
```

```
      CONSTS    = - 42500./(1.987 * T1 * TEMP(I,J))
      APHI4     = (8.917 * PHI + 31.433 / PHI - 28.950     )
      APHI5     = (2.000       + 1.333  / PHI - .8333 * PHI)
C
C ---   RATE CONSTANTS
C
C       WRITE(5,*)TEMP(I,J),T1
      AF4       = APHI4 / (TEMP(I,J)**5) *  FACT4
     1            *  EXP(CONST4) / (TEMP(I,J)**5)
      AF5       = APHI5 / (TEMP(I,J)**6) * FACT5
     1            *  EXP(CONST5) / (TEMP(I,J)**7)
      AB4       = AF4 /  EQUIL4
      AB5       = AF5 / (EQUIL5 * RUCGS * T1 * TEMP(I,J))
C       WRITE(5,*)'AF4,AF5,AB4,AB5'
C       WRITE(5,*)I,J, AF4,AF5,AB4,AB5
C
C ---   SPECIES CONCENTRATIONS
C
      UDEN      = U(1,I,J,2)
      UH2       = U(5,I,J,2)
      UO2       = U(6,I,J,2)
      UH2O      = U(7,I,J,2)
      UN2       = U(8,I,J,2)
      UOH       = UDEN - UH2 - UO2 - UH2O - UN2
      UOH       = ABS(UOH)
      UH2       = UH2 * DEN1              / AWH2
      UO2       = UO2 * DEN1              / AWO2
      UH2O      = UH2O* DEN1              / AWH2O
      UOH       = UOH * DEN1              / AWOH
      UN2       = UN2 * DEN1              / AWN2
C
C ---   PRODUCTION RATES - 'H'  TERM
C
      S5          = 1.0
      AH(5,I,J)   = AWH2  * (- AF4 * UH2 * UO2    +  AB4 * UOH**2
     1                       - AF5 * UOH**2 *UH2 * S5
     2                       + AB5 * UH2O**2 * S5 )
      S5          = 1.0
      AH(6,I,J)   = AWO2  * (- AF4 * UH2 * UO2    +  AB4 * UOH**2  )
      S5          = 1.0
      AH(7,I,J)   = AWH2O * 2.0 * (  AF5 * UOH**2 * UH2 * S5
     1                             - AB5 * UH2O**2 * S5            )
C       WRITE(5,*)I ,J,AH(5,I,J),AH(6,I,J),AH(7,I,J)
C
C ---    ADD THE SOURCE TERMS TO THE RESIDUALS COMPUTED IN THE
C ---    FLUX BALANCE ROUTINE " SUBROUTINE FLUX"
C ---    NOTE 'WSCALE' IS A PARAMETER USED TO NON-DIMENSIONALIZE WDOT
C
      RES(5,I,J)  = RES(5,I,J) - AH(5,I,J) * WSCALE
      RES(6,I,J)  = RES(6,I,J) - AH(6,I,J) * WSCALE
```

```
      RES(7,I,J)  = RES(7,I,J) - AH(7,I,J) * WSCALE
1     CONTINUE
500   CONTINUE
      RETURN
      END
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      SUBROUTINE NSSOLVE(IA)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C============================================================C
C                                                            C
C  M A T R I X   E Q U A T I O N   T O   B E   S O L V E D   C
C                                                            C
C                                                            C
C    [                  ] [       ]    [        ]            C
C    [   A    B    C    ] [  X1   ]    [   R1   ]            C
C    [                  ] [       ]    [        ]            C
C    [   D    E    F    ] [  X2   ] =  [   R2   ]            C
C    [                  ] [       ]    [        ]            C
C    [   G    H    I    ] [  X3   ]    [   R3   ]            C
C    [                  ] [       ]    [        ]            C
C                                                            C
C============================================================C
C
C ---
C ---   SYSTEM OF LINEAR EQUATIONS SOLVED BY
C ---   GAUSSIAN ELIMINATION - GLOBAL CHEMISTRY MODEL
C ---
C
C ---   COMPUTE THE  TIME SCALING DERIVATIVES OF THE
C ---   \ S \ MATRIX
C
      COMMON/VAR0/U(8,53,33,2),FI(8,53,33),GI(8,53,33),AH(8,53,33)
      COMMON/VAR00/FV(4,8,53,33),GV(4,8,53,33)
      COMMON/VAR1/UVEL(53,33),VVEL(53,33),PRES(53,33),TEMP(53,33)
      COMMON/VAR111/YH2(53,33),YO2(53,33),YH2O(53,33),YOH(53,33)
      COMMON/VAR2/DEN(53,33),SOUND(53,33),AINTE(53,33),AMACH(53,33)
      COMMON/VAR3/ENTHP(53,33),VIS(53,33),YYN2(53,33),CPND(53,33)
      COMMON/VAR4/DXX,DYY,X(55,35),Y(55,35),AREA(53,33)
      COMMON/VAR5/DT(53,33)
      COMMON/VAR6/NX,NXX,NXXX,NY,NYY,NYYY,IRES,IEQ
      COMMON/VAR7/IVIS,NITER,NCITER
      COMMON/VAR8/P1,T1,AM1,VISL,U1,V1,AK1,CV,CP,R,GAMA,DEN1,E1
      COMMON/VAR9/P11,C11,E11
      COMMON/VAR10/DFH2,DFO2,DFH2O,DFOH,DFN2
      COMMON/VAR11/CPH2,CPO2,CPH2O,CPOH,CPN2,CVH2,CVO2,CVH2O,CVOH,CVN2
      COMMON/VAR12/CONH2,CONO2,CONH2O,CONOH,CONN2
      COMMON/VAR13/COND,CFL,DCOFF,AL,VELO1
      COMMON/VAR14/RES(8,53,33)
      COMMON/VAR15/ALPHA(4)
      COMMON/VAR16/DXE(4,53,33),DXN(4,53,33),DXW(4,53,33),DXS(4,53,33)
```

```
      COMMON/VAR17/DYE(4,53,33),DYN(4,53,33),DYW(4,53,33),DYS(4,53,33)
      COMMON/VAR18/DD(2,8,53,33)
      COMMON/VAR20/NSX,NSXB,NSXA,NSY,NSYB,NSYA
      COMMON/VAR21/ACOM
      COMMON/VAR22/TRIGTEMP,PHI
      COMMON/VAR23/REN,PR,FACT1,LAMB,SDIFF
C
C ---    GAS PROPERITY INPUT
C
      AWH2    =  2.0E-3
      AWO2    =  32.0E-3
      AWH2O   =  18.0E-3
      AWOH    =  17.0E-3
      AWN2    =  28.0E-3
      RUCGS   = 1.987
      EQUIL4  = 5.76E-1
      EQUIL5  = 3.47E+3
      WSCALE  = AL / (VELO1 * DEN1)
C
C ---    SWEEP THROUGH ALL GRID POINTS TO COMPUTE "WDOT"
C
      FACT4   = ACOM * (1.0E+21/(T1**5)) * (1.0E+20/(T1**5))
      FACT5   = ACOM * (1.0E+20/(T1**5)) * (1.0E+20/(T1**5))
     1               * (1.0E+12/(T1**3))
      S5      = 1.0
C
C ---    DETERMINE OPTIMUM TIME STEP FOR STABILITY
C
      CALL STAB
C
      IF(NOITER.GT.200) GO TO 977
C
C
C ---  IGNITION TRIGGER - REMOVE FOR FLAT PLATE CALCULATIONS
C
      DO 978 IS = 11 , 17
      DO 978 JS = 1  , 7
978   TEMP(IS,JS) = 2.5
977   CONTINUE
C
C ---  FREE STREAM SPECIFIC HEAT
C
      CPFS   = CPN2 * CONN2 + CPO2 * CONO2 + CPH2 * CONH2
C
      DO 1 J = 1 , NYY
      DO 1 I = 1 , NXX
      IF(I.LT.NSX.AND.J.LT.NSY) GO TO 1
C
      IF(I.LT.NSX) GO TO 1
C
```

```
      DDT     =  DT(I,J)
C
C ---     IGNITION TEMPERATURE TEST
C
      ATEMP = TEMP(I,J) * T1
      IF(ATEMP.LE.TRIGTEMP.OR.ACOM.EQ.0.0) GO TO 500
C
C ---     EXPONENTIAL FACTOR
C
      EQUIL4   = 26.164 * EXP( -8992. / (TEMP(I,J) * T1))
      EQUIL5   = 3.269E-8 * EXP( 69415. / (TEMP(I,J) * T1))
      CONST4   = - 4565. /(1.987 * T1 * TEMP(I,J))
      CONST5   = - 42500./(1.987 * T1 * TEMP(I,J))
      APHI4    = (8.917 * PHI + 31.433 / PHI - 28.950      )
      APHI5    = (2.000       + 1.333  / PHI - .8333 * PHI)
C
C ---   RATE CONSTANTS
C
      AF4      = APHI4 / (TEMP(I,J)**5) *  FACT4
     1           *  EXP(CONST4) / (TEMP(I,J)**5)
      AF5      = APHI5 / (TEMP(I,J)**6) * FACT5
     1           * EXP(CONST5) / (TEMP(I,J)**7)
      AB4      = AF4 /  EQUIL4
      AB5      = AF5 / (EQUIL5 * RUCGS * T1 * TEMP(I,J))
C
C ---   DK / DT
C
C       WRITE(5,*)I,J,FACT4,FACT5,CONST4,CONST5
      PAF4     = APHI4 / (TEMP(I,J)**5) *  FACT4
     1           * EXP(CONST4) / (TEMP(I,J)**6)
     2           * ( -10. - CONST4 )
      PAF5     = APHI5 / (TEMP(I,J)**6) * FACT5
     1           * EXP(CONST5) / (TEMP(I,J)**8)
     2           * ( -13. - CONST5 )
C
C ---   SPECIES CONCENTRATIONS
C
      UDEN     = U(1,I,J,2)
      UH2      = U(5,I,J,2)
      UO2      = U(6,I,J,2)
      UH2O     = U(7,I,J,2)
      UN2      = U(8,I,J,2)
      UOH      = UDEN - UH2 - UO2 - UH2O - UN2
      UOH      = ABS(UOH)
      UH2      = UH2 * DEN1            / AWH2
      UO2      = UO2 * DEN1            / AWO2
      UH2O     = UH2O* DEN1            / AWH2O
      UOH      = UOH * DEN1            / AWOH
      UN2      = UN2 * DEN1            / AWN2
      CV         = (  YH2(I,J)  * CVH2  + YO2(I,J)  * CVO2
```

```
     1               + YH2O(I,J) * CVH2O + YOH(I,J)  * CVOH
     2               + YYN2(I,J)  * CVN2 )
C
C ---   COMPUTE \ S \ MATRIX ELEMENTS     DH / DU
C ---   IE THE CHARACTERISTIC CHEMICAL TIMES
C
C
C
C       -------------------------
C ---   CHARACTERISTIC TIME SCALES
C       -------------------------
C
      S1        =  1.0 / AWH2
      S2        =  1.0 / AWO2
      S3        =  1.0 / AWH2O
      S4        =  1.0 / AWOH
C
C        -------------
C ---    H2 TIME SCALES
C        -------------
C
      DH2UH2    =  AWH2 *     (- AF4*UO2*S1 - 2.0*AB4*UOH*S4
     1                         + 2.0*AF5*UOH*UH2*S4
     2                         - AF5*UOH**2*S1                    )
      DH2UO2    =  AWH2 *     (- AF4*UH2*S2 - 2.0*AB4*UOH*S4
     1                         + 2.0*AF5*UOH*UH2*S4               )
      DH2UW     =  AWH2 * 2.0*(- AB4*UOH*S4 + AF5*UOH*UH2*S4
     1                         + AB5*UH2O*S3                      )
      DH2UD     =  AWH2 * 2.0*(+ AB4*UOH*S4 - AF5*UOH*UH2*S4      )
      DH2UN2    =  - DH2UD
      DH2UE1    =  (-UH2*UO2    + UOH**2/EQUIL4 ) * PAF4
      EQUILS    =  EQUIL5*RUCGS*T1*TEMP(I,J)
      DH2UE2    =  (-UOH**2*UH2 + UH2O**2/EQUILS) * PAF5
      DH2UE     =  AWH2*VELO1**2/(T1*CV*DEN(I,J))*(DH2UE1+DH2UE2)
C
C        -------------
C ---    O2 TIME SCALES
C        -------------
C
      DO2UH2    =  AWO2  *     (- AF4*UO2*S1 - 2.0*AB4*UOH*S4     )
      DO2UO2    =  AWO2  *     (- AF4*UH2*S2 - 2.0*AB4*UOH*S4     )
      DO2UW     =  AWO2  *     (- 2.0*AB4*UOH*S4                  )
      DO2UD     =  AWO2  *     (  2.0*AB4*UOH*S4                  )
      DO2UN2    =  - DO2UD
      DO2UE1    = (-UH2*UO2 + UOH**2/EQUIL4) * PAF4
      DO2UE     = AWO2*VELO1**2/(T1*CV*DEN(I,J))*DO2UE1
C
C        --------------
C ---    H2O TIME SCALES
C        --------------
```

```
C
      DH2OUH2  =  AWH2O *2.0*(- 2.0*AF5*UOH*UH2*S4
     1                                   + AF5*UOH**2*S1    )
      DH2OUO2  =  AWH2O  *2.0*(- 2.0*AF5*UOH*UH2*S4         )
      DH2OUW   =  AWH2O*2.0*(- 2.0*AF5*UOH*UH2*S4
     1                                   - 2.0*AB5*UH2O*S3 )
      DH2OUD   =  AWH2O *2.0*(  2.0*AF5*UOH*UH2*S4         )
      DH2OUN2  = - DH2OUD
      EQUILS   =  EQUILS*RUCGS*T1*TEMP(I,J)
      DH2OUE1  =  2.0*(UOH**2*UH2 - UH2O**2/EQUILS) * PAF5
      DH2OUE   =  AWH2O*VELO1**2/(T1*CV*DEN(I,J))*DH2OUE1
C
C
C ---   DEFINE THE COEFFICIENTS OF THE COEFFICIENT MATRIX \\ S \\
C
      A    = 1.0 - DH2UH2   * DDT * ALPHA(IA) * WSCALE
      B    =     - DH2UO2   * DDT * ALPHA(IA) * WSCALE
      C    =     - DH2UW    * DDT * ALPHA(IA) * WSCALE
      D    =     - DO2UH2   * DDT * ALPHA(IA) * WSCALE
      E    = 1.0 - DO2UO2   * DDT * ALPHA(IA) * WSCALE
      F    =     - DO2UW    * DDT * ALPHA(IA) * WSCALE
      AAG  =     - DH2OUH2  * DDT * ALPHA(IA) * WSCALE
      AAH  =     - DH2OUO2  * DDT * ALPHA(IA) * WSCALE
      AAI  = 1.0 - DH2OUW   * DDT * ALPHA(IA) * WSCALE
C
C ---    NEXT DEFINE THE RESIDUAL VECTOR
C
      DU1   = - ALPHA(IA) * DDT * RES(1,I,J)
      DU1   = DU1 * WSCALE
      DU4   = - ALPHA(IA) * DDT * RES(4,I,J)
      DU4   = DU4 * WSCALE
      DU8   = - ALPHA(IA) * DDT * RES(8,I,J)
      DU8   = DU8 * WSCALE
C
C ---    DU1 * DH/DU  ACCOUNTS THE DEPENDANCE OF WDOT ON
C ---                 DENSITY
C
      R1    = - ALPHA(IA) * DDT * (RES(5,I,J) - DH2UD  * DU1
     1          - DH2UE    * DU4 - DH2UN2 * DU8)
      R2    = - ALPHA(IA) * DDT * (RES(6,I,J) - DO2UD  * DU1
     1          - DO2UE    * DU4 - DO2UN2 * DU8)
      R3    = - ALPHA(IA) * DDT * (RES(7,I,J) - DH2OUD * DU1
     1          - DH2OUE   * DU4 - DH2OUN2 * DU8)
C
C ---    NORMALIZE THE MATRIX ELEMENTS SUCH THAT NONE IS GREATER
C ---    THAN ONE
C
C ---    FIND NORMALIZING VALUES
C
       SC1   = ABS(A)
```

```
      IF(ABS(B).GT.SC1) SC1 = ABS(B)
      IF(ABS(C).GT.SC1) SC1 = ABS(C)
      SC2   = ABS(D)
      IF(ABS(E).GT.SC2) SC2 = ABS(E)
      IF(ABS(F).GT.SC2) SC2 = ABS(F)
      SC3   = ABS(AAG)
      IF(ABS(AAH).GT.SC3) SC3 = ABS(AAH)
      IF(ABS(AAI).GT.SC3) SC3 = ABS(AAI)
C
C ---     NORMALIZE
C
      A   = A / SC1
      B   = B / SC1
      C   = C / SC1
      D   = D / SC2
      E   = E / SC2
      F   = F / SC2
      AAG = AAG / SC3
      AAH = AAH / SC3
      AAI = AAI / SC3
C
      R1  = R1  / SC1
      R2  = R2  / SC2
      R3  = R3  / SC3
C
C        --------------------------------
C ---    SOLVE MATRIX SYSTEM OF EQUATIONS
C        --------------------------------
C
C ---    REDUCE TO DIAGONAL FORM
C
      EP    = 1.0E-15
      TEST  = ABS(D)
      IF(TEST.LT.EP) GO TO 100
      DIV   = A / D
      E     = B  - DIV * E
      F     = C  - DIV * F
      R2    = R1 - DIV * R2
100   CONTINUE
      TEST  = ABS(AAG)
      IF(TEST.LT.EP) GO TO 101
      DIV   = A / AAG
      AAH   = B  - DIV * AAH
      AAI   = C  - DIV * AAI
      R3    = R1 - DIV * R3
101   CONTINUE
      TEST  = ABS(AAH)
      IF(TEST.LT.EP) GO TO 102
      DIV   = E / AAH
      AAI   = F  - DIV * AAI
```

```
      R3     = R2 - DIV * R3
102   CONTINUE
C
C ---  NOW COMPUTE THE X'S VIA BACK SUBSTITUTION
C
      X3     = R3 / AAI
      X2     = (R2 - F * X3) / E
      X1     = (R1 - B * X2 - C * X3) / A
500   CONTINUE
C
C      ----------------------------
C ---  COMPUTE NEW FLUID    " U 'S "
C      ----------------------------
C
      U(1,I,J,2) = U(1,I,J,1) - ALPHA(IA) * DDT * RES(1,I,J)
      U(2,I,J,2) = U(2,I,J,1) - ALPHA(IA) * DDT * RES(2,I,J)
      U(3,I,J,2) = U(3,I,J,1) - ALPHA(IA) * DDT * RES(3,I,J)
      U(4,I,J,2) = U(4,I,J,1) - ALPHA(IA) * DDT * RES(4,I,J)
      U(8,I,J,2) = U(8,I,J,1) - ALPHA(IA) * DDT * RES(8,I,J)
C
C      ---------------------------------------------------------
C ---  COMPUTE THE NEW SPECIES STATE QUANTITIES IE "  U'S "
C      ---------------------------------------------------------
C
      IF(ATEMP.LE.TRIGTEMP.OR.ACOM.EQ.0.0) GO TO 501
      U(5,I,J,2) = U(5,I,J,1) + X1
      U(6,I,J,2) = U(6,I,J,1) + X2
      U(7,I,J,2) = U(7,I,J,1) + X3
      GO TO 502
501   CONTINUE
      U(5,I,J,2) = U(5,I,J,1) - ALPHA(IA) * DDT * RES(5,I,J)
      U(6,I,J,2) = U(6,I,J,1) - ALPHA(IA) * DDT * RES(6,I,J)
      U(7,I,J,2) = U(7,I,J,1) - ALPHA(IA) * DDT * RES(7,I,J)
502   CONTINUE
C
C       ----------------------
C ---   UPDATE FLUID PROPERTIES
C       ----------------------
C
      DEN(I,J)    = U(1,I,J,2)
      ODEN        = 1.0 / DEN(I,J)
      UVEL(I,J)   = U(2,I,J,2) * ODEN
      VVEL(I,J)   = U(3,I,J,2) * ODEN
      AINTE(I,J)  = U(4,I,J,2) * ODEN
C
C      ----------------------------
C ---  UPDATE SPECIES MASS FRACTIONS
C      ----------------------------
C
      YH2(I,J)    = U(5,I,J,2) * ODEN
```

```
      YO2(I,J)    = U(6,I,J,2) * ODEN
      YH2O(I,J)   = U(7,I,J,2) * ODEN
      YYN2(I,J)   = U(8,I,J,2) * ODEN
C
C        --------------------------
C ---    COMPUTE REMAINING UNKNOWNS
C        --------------------------
C
      UOH         = 1.0 - YH2(I,J) - YO2(I,J) - YH2O(I,J) - YYN2(I,J)
      YOH(I,J)    = UOH
C      WRITE(5,*)I,J,YH2(I,J),YO2(I,J),YH2O(I,J),YOH(I,J),CONN2,
C     1   TEMP(I,J)
      CP          = (  YH2(I,J)  * CPH2  + YO2(I,J)  * CPO2
     1                + YH2O(I,J) * CPH2O + UOH       * CPOH
     2                + YYN2(I,J)  * CPN2 )
C
C ---  NON-DIMENSIONAL CP
C
      CPND(I,J)   = CP / CPFS
C
      CV          = (  YH2(I,J)  * CVH2  + YO2(I,J)  * CVO2
     1                + YH2O(I,J) * CVH2O + UOH       * CVOH
     2                + YYN2(I,J)  * CVN2 )
C      WRITE(5,*)I,J,CP,CV,YH2O(I,J),YO2(I,J),YH2O(I,J),YOH(I,J)
      R           = CP - CV
      GAMA        = CP / CV
      DHEATF      = YH2(I,J)*DFH2 + YO2(I,J)* DFO2 + YH2O(I,J)*DFH2O
     1               + UOH*DFOH    + YYN2(I,J)* DFN2
      VELO        = UVEL(I,J)**2 + VVEL(I,J)**2
      TEMP(I,J)   = (VELO1**2/(CV*T1))*(AINTE(I,J) - .5*VELO
     1                - DHEATF/(VELO1**2))
      TEMP(I,J)   = ABS(TEMP(I,J))
      SOUND(I,J)  = SQRT(GAMA * R * T1 * TEMP(I,J))/VELO1
      AMACH(I,J)  = SQRT(VELO)/SOUND(I,J)
      PRES(I,J)   = (T1 /VELO1**2) * R * DEN(I,J) * TEMP(I,J)
      ENTHP(I,J)  = CP * T1 / VELO1**2 *TEMP(I,J) + .5 * VELO
     1               + DHEATF / VELO1**2
1     CONTINUE
      RETURN
      END
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      SUBROUTINE VISS
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C ---   DETERNINE THE LAMINAR VISCOSITY USING SUTHERLANDS LAW
C
      COMMON/VAR0/U(8,53,33,2),FI(8,53,33),GI(8,53,33),AH(8,53,33)
      COMMON/VAR00/FV(4,8,53,33),GV(4,8,53,33)
      COMMON/VAR1/UVEL(53,33),VVEL(53,33),PRES(53,33),TEMP(53,33)
      COMMON/VAR111/YH2(53,33),YO2(53,33),YH2O(53,33),YOH(53,33)
```

```
      COMMON/VAR2/DEN(53,33),SOUND(53,33),AINTE(53,33),AMACH(53,33)
      COMMON/VAR3/ENTHP(53,33),VIS(53,33),YYN2(53,33),CPND(53,33)
      COMMON/VAR4/DXX,DYY,X(55,35),Y(55,35),AREA(53,33)
      COMMON/VAR5/DT(53,33)
      COMMON/VAR6/NX,NXX,NXXX,NY,NYY,NYYY,IRES,IEQ
      COMMON/VAR7/IVIS,NITER,NOITER
      COMMON/VAR8/P1,T1,AM1,VISL,U1,V1,AK1,CV,CP,R,GAMA,DEN1,E1
      COMMON/VAR9/P11,C11,E11
      COMMON/VAR10/DFH2,DFO2,DFH2O,DFOH,DFN2
      COMMON/VAR11/CPH2,CPO2,CPH2O,CPOH,CPN2,CVH2,CVO2,CVH2O,CVOH,CVN2
      COMMON/VAR12/CONH2,CONO2,CONH2O,CONOH,CONN2
      COMMON/VAR13/COND,CFL,DCOFF,AL,VELO1
      COMMON/VAR14/RES(8,53,33)
      COMMON/VAR15/ALPHA(4)
      COMMON/VAR16/DXE(4,53,33),DXN(4,53,33),DXW(4,53,33),DXS(4,53,33)
      COMMON/VAR17/DYE(4,53,33),DYN(4,53,33),DYW(4,53,33),DYS(4,53,33)
      COMMON/VAR18/DD(2,8,53,33)
      COMMON/VAR20/NSX,NSXB,NSXA,NSY,NSYB,NSYA
      COMMON/VAR21/ACOM
      COMMON/VAR22/TRIGTEMP,PHI
      COMMON/VAR23/REN,PR,FACT1,LAMB,SDIFF
C
C ---   VIS(I,J) IS A NON-DIMENSIONAL LAMINAR VISCOSITY
C
      S1           = 110
C
      DO 1 J       = 1 , NYY
      DO 1 I       = 1 , NXX
      VIS(I,J)     = (T1 * TEMP(I,J))**1.5 / (TEMP(I,J)*T1 + S1)
1     CONTINUE
      RETURN
      END
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      SUBROUTINE PARAM
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                    C
C --- COMPUTE AND PRINT OUT THE GROSS FEATURES OF THE PROBLEM        C
C --- FOR EXAMPLE CALCULATE THE PERCENT OF H2 AND O2CONSUMED OR THE  C
C --- RATIO OF THE HEAT RELEASED TO THE TOTAL HEAT AVAILABLE         C
C                                                                    C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      COMMON/VAR0/U(8,53,33,2),FI(8,53,33),GI(8,53,33),AH(8,53,33)
      COMMON/VAR00/FV(4,8,53,33),GV(4,8,53,33)
      COMMON/VAR1/UVEL(53,33),VVEL(53,33),PRES(53,33),TEMP(53,33)
      COMMON/VAR111/YH2(53,33),YO2(53,33),YH2O(53,33),YOH(53,33)
      COMMON/VAR2/DEN(53,33),SOUND(53,33),AINTE(53,33),AMACH(53,33)
      COMMON/VAR3/ENTHP(53,33),VIS(53,33),YYN2(53,33),CPND(53,33)
      COMMON/VAR4/DXX,DYY,X(55,35),Y(55,35),AREA(53,33)
      COMMON/VAR5/DT(53,33)
      COMMON/VAR6/NX,NXX,NXXX,NY,NYY,NYYY,IRES,IEQ
```

```
      COMMON/VAR7/IVIS,NITER,NOITER
      COMMON/VAR8/P1,T1,AM1,VISL,U1,V1,AK1,CV,CP,R,GAMA,DEN1,E1
      COMMON/VAR9/P11,C11,E11
      COMMON/VAR10/DFH2,DFO2,DFH2O,DFOH,DFN2
      COMMON/VAR11/CPH2,CPO2,CPH2O,CPOH,CPN2,CVH2,CVO2,CVH2O,CVOH,CVN2
      COMMON/VAR12/CONH2,CONO2,CONH2O,CONOH,CONN2
      COMMON/VAR13/COND,CFL,DCOFF,AL,VELO1
      COMMON/VAR14/RES(8,53,33)
      COMMON/VAR15/ALPHA(4)
      COMMON/VAR16/DXE(4,53,33),DXN(4,53,33),DXW(4,53,33),DXS(4,53,33)
      COMMON/VAR17/DYE(4,53,33),DYN(4,53,33),DYW(4,53,33),DYS(4,53,33)
      COMMON/VAR18/DD(2,8,53,33)
      COMMON/VAR20/NSX,NSXB,NSXA,NSY,NSYB,NSYA
      COMMON/VAR21/ACOM
      COMMON/VAR22/TRIGTEMP,PHI
      COMMON/VAR23/REN,PR,FACT1,LAMB,SDIFF
C*****************************************************************
C
C ---    COMPUTE THE PERCENT OF H2 CONSUMED
C
C*****************************************************************
C ---     AMASS1  =  MASS OF H2 IN
C ---     AMASS2  =  MASS OF H2 OUT
C
      AMASS1      =  Y(NSY,NYY) * DEN1 * CONH2 * 1.0
C
      AMASS2      = 0.0
      DO 1 J      = 1 , NYY
      DELTAY      = Y(NXX,J+1) - Y(NXX,J)
      AVEL        = SQRT(UVEL(NXX,J)**2 + VVEL(NXX,J)**2)
      DM          = DELTAY * DEN1 * DEN(NXX,J) * YH2(NXX,J) * AVEL
      AMASS2      = AMASS2 + DM
1     CONTINUE
C
C ---   PERCENT OF H2 CONSUMED   PH2
C
      PH2         =  100. * ( 1.0 - AMASS2  / AMASS1 )
C
C*****************************************************************
C
C ---    COMPUTE THE PERCENT OF O2 CONSUMED
C
C*****************************************************************
C ---     AMASS1  =  MASS OF O2 IN
C ---     AMASS2  =  MASS OF O2 OUT
C
      AMASS1      =  Y(NSY,NYY) * DEN1 * CONO2 * 1.0
C
      AMASS2      = 0.0
      DO 2 J      = 1 , NYY
```

```
      DELTAY       = Y(NXX,J+1) - Y(NXX,J)
      AVEL         = SQRT(UVEL(NXX,J)**2 + VVEL(NXX,J)**2)
      DM           = DELTAY * DEN1 * DEN(NXX,J) * YO2(NXX,J) * AVEL
      AMASS2       = AMASS2 + DM
2     CONTINUE
C
C ---   PERCENT OF O2 COMSUMED  PO2
C
      PO2          = 100. * ( 1.0 - AMASS2  / AMASS1 )
C
C
C***************************************************************
C
C ---    COMPUTE THE HEAT RELEASE PARAMETER   PH = DHF / HTO
C
C**************************************************************
C ---     HTO      =  ENTERING TOTAL ENTHALPY
C ---     DHF      =  HEAT RELEASED DUE TO THE FORMATION OF H2O
C
      HTO          = (CPN2 * CONN2 + CPH2 * CONH2 + CPO2 * CONO2) * T1
     1                   +  .5
      DO 3 J       = 1 , NYY
      DM           = YH2O(NXX,J) * DFH2O
      DHF          = DHF  + DM
3     CONTINUE
      DHF          = DHF / (Y(NXX,NYY) - Y(NXX,1))
      PH           = DHF / HTO
C
C ---      PRINT OUTPUT
C
      WRITE(6,11) PH2 , PO2 , PH
11    FORMAT(2X,3(E10.4))
      RETURN
      END
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      SUBROUTINE LOWERST
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C
C                  **************************************
C
C                  S T E P     VERSION OF LOWER BOUNDARY
C
C                  **************************************
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
      COMMON/VARO/U(8,53,33,2),FI(8,53,33),GI(8,53,33),AH(8,53,33)
```

```
      COMMON/VAR00/FV(4,8,53,33),GV(4,8,53,33)
      COMMON/VAR1/UVEL(53,33),VVEL(53,33),PRES(53,33),TEMP(53,33)
      COMMON/VAR111/YH2(53,33),YO2(53,33),YH2O(53,33),YOH(53,33)
      COMMON/VAR2/DEN(53,33),SOUND(53,33),AINTE(53,33),AMACH(53,33)
      COMMON/VAR3/ENTHP(53,33),VIS(53,33),YYN2(53,33),CPND(53,33)
      COMMON/VAR4/DXX,DYY,X(55,35),Y(55,35),AREA(53,33)
      COMMON/VAR5/DT(53,33)
      COMMON/VAR6/NX,NXX,NXXX,NY,NYY,NYYY,IRES,IEQ
      COMMON/VAR7/IVIS,NITER,NOITER
      COMMON/VAR8/P1,T1,AM1,VISL,U1,V1,AK1,CV,CP,R,GAMA,DEN1,E1
      COMMON/VAR9/P11,C11,E11
      COMMON/VAR10/DFH2,DFO2,DFH2O,DFOH,DFN2
      COMMON/VAR11/CPH2,CPO2,CPH2O,CPOH,CPN2,CVH2,CVO2,CVH2O,CVOH,CVN2
      COMMON/VAR12/CONH2,CONO2,CONH2O,CONOH,CONN2
      COMMON/VAR13/COND,CFL,DCOFF,AL,VELO1
      COMMON/VAR14/RES(8,53,33)
      COMMON/VAR15/ALPHA(4)
      COMMON/VAR16/DXE(4,53,33),DXN(4,53,33),DXW(4,53,33),DXS(4,53,33)
      COMMON/VAR17/DYE(4,53,33),DYN(4,53,33),DYW(4,53,33),DYS(4,53,33)
      COMMON/VAR18/DD(2,8,53,33)
      COMMON/VAR20/NSX,NSXB,NSXA,NSY,NSYB,NSYA
      COMMON/VAR21/ACOM
      COMMON/VAR22/TRIGTEMP,PHI
      COMMON/VAR23/REN,PR,FACT1,LAMB,SDIFF
C
C
C
C ---   COMPUTE FLUXES THROUGH THE LOWER WALL CELLS
C
C                   ---------------
C ---               !  EAST FACE  !
C                   ---------------
C
      DO 20  I   = 2 , NXXX
      IF(I.EQ.NSX) GO TO 20
      IF(I.LT.NSX) J = NSY
      IF(I.GT.NSX) J = 1
      IF(I.LT.NSX) TWALL  = TEMP(I,NSY)
      IF(I.GT.NSX) TWALL  = TEMP(I,1)
      IF(I.LT.NSX) Y1WALL = YH2(I,NSY)
      IF(I.GT.NSX) Y1WALL = YH2(I,1)
      IF(I.LT.NSX) Y2WALL = YO2(I,NSY)
      IF(I.GT.NSX) Y2WALL = YO2(I,1)
      IF(I.LT.NSX) Y3WALL = YH2O(I,NSY)
      IF(I.GT.NSX) Y3WALL = YH2O(I,1)
      IF(I.LT.NSX) Y4WALL = YYN2(I,NSY)
      IF(I.GT.NSX) Y4WALL = YYN2(I,1)
      AV         = .5 * (AREA(I,J) + AREA(I+1,J))
C
C ---  U-VELOCITY
```

```
C
      UE          = UVEL(I+1,J)
      UW          = UVEL(I,J)
      UN          = .25 * (UVEL(I+1,J) + UVEL(I+1,J+1)
     1                  +  UVEL(I,J+1) + UVEL(I,J))
      US          = 0.0
C
C ---   V-VELOCITY
C
      VE          = VVEL(I+1,J)
      VW          = VVEL(I,J)
      VN          = .25 * (VVEL(I+1,J) + VVEL(I+1,J+1)
     1                  +  VVEL(I,J+1) + VVEL(I+1,J))
      VS          = 0.0
C
C ---   TEMPERATURE
C
      TE          = TEMP(I+1,J)
      TW          = TEMP(I,J)
      TN          = .25 * (TEMP(I+1,J) + TEMP(I+1,J+1)
     1                  +  TEMP(I,J+1) + TEMP(I+1,J))
      TS          = TWALL
C
C ---   YH2
C
      Y1E          = YH2(I+1,J)
      Y1W          = YH2(I,J)
      Y1N          = .25 * (YH2(I+1,J) + YH2(I+1,J+1)
     1                  +   YH2(I,J+1) + YH2(I+1,J))
      Y1S          = Y1WALL
C
C ---   YO2
C
      Y2E          = YO2(I+1,J)
      Y2W          = YO2(I,J)
      Y2N          = .25 * (YO2(I+1,J) + YO2(I+1,J+1)
     1                  +   YO2(I,J+1) + YO2(I+1,J))
      Y2S          = Y2WALL
C
C ---   YH2O
C
      Y3E          = YH2O(I+1,J)
      Y3W          = YH2O(I,J)
      Y3N          = .25 * (YH2O(I+1,J) + YH2O(I+1,J+1)
     1                  +   YH2O(I,J+1) + YH2O(I+1,J))
      Y3S          = Y3WALL
C
C ---   YYN2
C
      Y4E          = YYN2(I+1,J)
```

```
      Y4W          = YYN2(I,J)
      Y4N          = .25 * (YYN2(I+1,J) + YYN2(I+1,J+1)
     1                  +   YYN2(I,J+1) + YYN2(I+1,J))
      Y4S          = Y4WALL
C
C ---   CALCULATE THE GRADIENT TERMS
C
C
C ---   X  GRADIENTS
C
      DUDX        = (UE * DYE(1,I,J) + UN * DYN(1,I,J)
     1              + UW * DYW(1,I,J) + US * DYS(1,I,J)) / AV
      DVDX        = (VE * DYE(1,I,J) + VN * DYN(1,I,J)
     1              + VW * DYW(1,I,J) + VS * DYS(1,I,J)) / AV
      DTDX        = (TE * DYE(1,I,J) + TN * DYN(1,I,J)
     1              + TW * DYW(1,I,J) + TS * DYS(1,I,J)) / AV
      DY1DX       = (Y1E * DYE(1,I,J) + Y1N * DYN(1,I,J)
     1              + Y1W * DYW(1,I,J) + Y1S * DYS(1,I,J)) / AV
      DY2DX       = (Y2E * DYE(1,I,J) + Y2N * DYN(1,I,J)
     1              + Y2W * DYW(1,I,J) + Y2S * DYS(1,I,J)) / AV
      DY3DX       = (Y3E * DYE(1,I,J) + Y3N * DYN(1,I,J)
     1              + Y3W * DYW(1,I,J) + Y3S * DYS(1,I,J)) / AV
      DY4DX       = (Y4E * DYE(1,I,J) + Y4N * DYN(1,I,J)
     1              + Y4W * DYW(1,I,J) + Y4S * DYS(1,I,J)) / AV
      DY5DX       = -DY1DX - DY2DX - DY3DX - DY4DX
C
C ---   Y  GRADIENTS
C
      DUDY        = - (UE * DXE(1,I,J) + UN * DXN(1,I,J)
     1                + UW * DXW(1,I,J) + US * DXS(1,I,J)) / AV
      DVDY        = - (VE * DXE(1,I,J) + VN * DXN(1,I,J)
     1                + VW * DXW(1,I,J) + VS * DXS(1,I,J)) / AV
      DTDY        = - (TE * DXE(1,I,J) + TN * DXN(1,I,J)
     1                + TW * DXW(1,I,J) + TS * DXS(1,I,J)) / AV
      DY1DY       = - (Y1E * DXE(1,I,J) + Y1N * DXN(1,I,J)
     1                + Y1W * DXW(1,I,J) + Y1S * DXS(1,I,J)) / AV
      DY2DY       = - (Y2E * DXE(1,I,J) + Y2N * DXN(1,I,J)
     1                + Y2W * DXW(1,I,J) + Y2S * DXS(1,I,J)) / AV
      DY3DY       = - (Y3E * DXE(1,I,J) + Y3N * DXN(1,I,J)
     1                + Y3W * DXW(1,I,J) + Y3S * DXS(1,I,J)) / AV
      DY4DY       = - (Y4E * DXE(1,I,J) + Y4N * DXN(1,I,J)
     1                + Y4W * DXW(1,I,J) + Y4S * DXS(1,I,J)) / AV
      DY5DY       = - DY1DY - DY2DY - DY3DY - DY4DY
C
C ---   COMPUTE THE FULL SHEAR STRESS TERMS
C
      TXX         = - LAMB * VIS(I,J) * (DUDX + DVDY)
                    - 2.0  * VIS(I,J) * DUDX / REN
      TXY         = - VIS(I,J) / REN  * (DUDY + DVDX)
      TYY         = - LAMB * VIS(I,J) * (DUDX + DVDY)
```

```
                     - 2.0  * VIS(I,J) * DVDY / REN
C
C ---   COMPUTE THE VISCOUS CONTRIBUTIONS TO "F" AND "G"
C
      FV(1,1,I,J)  = 0.0
      FV(1,2,I,J)  = TXX
      FV(1,3,I,J)  = TXY
      FV(1,4,I,J)  =   .5 * (UVEL(I,J) + UVEL(I+1,J))*TXX
     1                + .5 * (VVEL(I,J) + VVEL(I+1,J))*TXY
     2                - SDIFF * FACT1 * DTDX
     3                - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2
     4                + CPH2 * T1 * TEMP(I,J)) *  DY1DX
     5                - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFO2
     6                + CPO2 * T1 * TEMP(I,J)) *  DY2DX
     7                - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2O
     8                + CPH2O * T1 * TEMP(I,J)) *  DY3DX
     9                - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2
     10               + CPN2 * T1 * TEMP(I,J)) *  DY4DX
     11               - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFOH
     12               + CPOH * T1 * TEMP(I,J)) *  DY5DX
      FV(1,5,I,J)  = - SDIFF * DY1DX
      FV(1,6,I,J)  = - SDIFF * DY2DX
      FV(1,7,I,J)  = - SDIFF * DY3DX
      FV(1,8,I,J)  = - SDIFF * DY4DX
C
      GV(1,1,I,J)  = 0.0
      GV(1,2,I,J)  = TXY
      GV(1,3,I,J)  = TYY
      GV(1,4,I,J)  =   .5 * (VVEL(I,J) + VVEL(I+1,J))*TYY
     1                + .5 * (UVEL(I,J) + UVEL(I+1,J))*TXY
     2                - SDIFF * FACT1 * DTDY
     3                - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2
     4                + CPH2 * T1 * TEMP(I,J)) *  DY1DY
     5                - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFO2
     6                + CPO2 * T1 * TEMP(I,J)) *  DY2DY
     7                - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2O
     8                + CPH2O * T1 * TEMP(I,J)) *  DY3DY
     9                - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2
     10               + CPN2 * T1 * TEMP(I,J)) *  DY4DY
     11               - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFOH
     12               + CPOH * T1 * TEMP(I,J)) *  DY5DY
      GV(1,5,I,J)  = - SDIFF * DY1DY
      GV(1,6,I,J)  = - SDIFF * DY2DY
      GV(1,7,I,J)  = - SDIFF * DY3DY
      GV(1,8,I,J)  = - SDIFF * DY4DY
C
20    CONTINUE
C
C                  -----------------
C ---              |  NORTH  FACE  |
```

```
C                  ------------------
C
      J            = 1
      DO 30  I     = 2 , NXXX
      IF(I.EQ.NSX) GO TO 30
      IF(I.LT.NSX) J = NSY
      IF(I.GT.NSX) J = 1
      IF(I.LT.NSX) TWALL = TEMP(I,NSY)
      IF(I.GT.NSX) TWALL = TEMP(I,1)
      IF(I.LT.NSX) Y1WALL = YH2(I,NSY)
      IF(I.GT.NSX) Y1WALL = YH2(I,1)
      IF(I.LT.NSX) Y2WALL = YO2(I,NSY)
      IF(I.GT.NSX) Y2WALL = YO2(I,1)
      IF(I.LT.NSX) Y3WALL = YH2O(I,NSY)
      IF(I.GT.NSX) Y3WALL = YH2O(I,1)
      IF(I.LT.NSX) Y4WALL = YYN2(I,NSY)
      IF(I.GT.NSX) Y4WALL = YYN2(I,1)
      AV           = .5 * (AREA(I,J) + AREA(I,J+1))
C
C ---   U-VELOCITY
C
      UE           = .25 * (UVEL(I+1,J) + UVEL(I+1,J+1)
     1                   +  UVEL(I,J+1) + UVEL(I,J))
      UN           = UVEL(I,J+1)
      UW           = .25 * (UVEL(I,J+1) + UVEL(I-1,J+1)
     1                   +  UVEL(I-1,J) + UVEL(I,J))
      US           = UVEL(I,J)
C
C ---   V-VELOCITY
C
      VE           = .25 * (VVEL(I+1,J) + VVEL(I+1,J+1)
     1                   +  VVEL(I,J+1) + VVEL(I,J))
      VN           = VVEL(I,J+1)
      VW           = .25 * (VVEL(I,J+1) + VVEL(I-1,J+1)
     1                   +  VVEL(I-1,J) + VVEL(I,J))
      VS           = VVEL(I,J)
C
C ---   TEMPERATURE
C
      TE           = .25 * (TEMP(I+1,J) + TEMP(I+1,J+1)
     1                   +  TEMP(I,J+1) + TEMP(I,J))
      TN           = TEMP(I,J+1)
      TW           = .25 * (TEMP(I,J+1) + TEMP(I-1,J+1)
     1                   +  TEMP(I-1,J) + TEMP(I,J))
      TS           = TEMP(I,J)
C
C ---   YH2
C
      Y1E           = .25 * (YH2(I+1,J)  + YH2(I+1,J+1)
     1                    +  YH2(I,J+1)  + YH2(I,J))
```

```
      Y1N         = YH2(I,J+1)
      Y1W         = .25 * (YH2(I,J+1) + YH2(I-1,J+1)
     1                 +  YH2(I-1,J) + YH2(I,J))
      Y1S         = YH2(I,J)
C
C ---   YO2
C
      Y2E         = .25 * (YO2(I+1,J)  + YO2(I+1,J+1)
     1                 +  YO2(I,J+1)  + YO2(I,J))
      Y2N         = YO2(I,J+1)
      Y2W         = .25 * (YO2(I,J+1) + YO2(I-1,J+1)
     1                 +  YO2(I-1,J) + YO2(I,J))
      Y2S         = YO2(I,J)
C
C ---   YH2O
C
      Y3E         = .25 * (YH2O(I+1,J)  + YH2O(I+1,J+1)
     1                 +  YH2O(I,J+1)  + YH2O(I,J))
      Y3N         = YH2O(I,J+1)
      Y3W         = .25 * (YH2O(I,J+1) + YH2O(I-1,J+1)
     1                 +  YH2O(I-1,J) + YH2O(I,J))
      Y3S         = YH2O(I,J)
C
C ---   YYN2
C
      Y4E         = .25 * (YYN2(I+1,J)  + YYN2(I+1,J+1)
     1                 +  YYN2(I,J+1)  + YYN2(I,J))
      Y4N         = YYN2(I,J+1)
      Y4W         = .25 * (YYN2(I,J+1) + YYN2(I-1,J+1)
     1                 +  YYN2(I-1,J) + YYN2(I,J))
      Y4S         = YYN2(I,J)
C
C ---   CALCULATE THE GRADIENT TERMS
C
C
C ---   X  GRADIENTS
C
      DUDX        = (UE * DYE(2,I,J) + UN * DYN(2,I,J)
     1             + UW * DYW(2,I,J) + US * DYS(2,I,J)) / AV
      DVDX        = (VE * DYE(2,I,J) + VN * DYN(2,I,J)
     1             + VW * DYW(2,I,J) + VS * DYS(2,I,J)) / AV
      DTDX        = (TE * DYE(2,I,J) + TN * DYN(2,I,J)
     1             + TW * DYW(2,I,J) + TS * DYS(2,I,J)) / AV
      DY1DX       = (Y1E * DYE(2,I,J) + Y1N * DYN(2,I,J)
     1             + Y1W * DYW(2,I,J) + Y1S * DYS(2,I,J)) / AV
      DY2DX       = (Y2E * DYE(2,I,J) + Y2N * DYN(2,I,J)
     1             + Y2W * DYW(2,I,J) + Y2S * DYS(2,I,J)) / AV
      DY3DX       = (Y3E * DYE(2,I,J) + Y3N * DYN(2,I,J)
     1             + Y3W * DYW(2,I,J) + Y3S * DYS(2,I,J)) / AV
      DY4DX       = (Y4E * DYE(2,I,J) + Y4N * DYN(2,I,J)
```

```
     1              + Y4W * DYW(2,I,J) + Y4S * DYS(2,I,J)) / AV
      DY5DX      = -DY1DX - DY2DX - DY3DX - DY4DX
C
C ---   Y  GRADIENTS
C
      DUDY       = - (UE * DXE(2,I,J) + UN * DXN(2,I,J)
     1              + UW * DXW(2,I,J) + US * DXS(2,I,J)) / AV
      DVDY       = - (VE * DXE(2,I,J) + VN * DXN(2,I,J)
     1              + VW * DXW(2,I,J) + VS * DXS(2,I,J)) / AV
      DTDY       = - (TE * DXE(2,I,J) + TN * DXN(2,I,J)
     1              + TW * DXW(2,I,J) + TS * DXS(2,I,J)) / AV
      DY1DY      = - (Y1E * DXE(2,I,J) + Y1N * DXN(2,I,J)
     1              + Y1W * DXW(2,I,J) + Y1S * DXS(2,I,J)) / AV
      DY2DY      = - (Y2E * DXE(2,I,J) + Y2N * DXN(2,I,J)
     1              + Y2W * DXW(2,I,J) + Y2S * DXS(2,I,J)) / AV
      DY3DY      = - (Y3E * DXE(2,I,J) + Y3N * DXN(2,I,J)
     1              + Y3W * DXW(2,I,J) + Y3S * DXS(2,I,J)) / AV
      DY4DY      = - (Y4E * DXE(2,I,J) + Y4N * DXN(2,I,J)
     1              + Y4W * DXW(2,I,J) + Y4S * DXS(2,I,J)) / AV
      DY5DY      = - DY1DY - DY2DY - DY3DY - DY4DY
C
C ---   COMPUTE THE FULL SHEAR STRESS TERMS
C
      TXX        = - LAMB * VIS(I,J) * (DUDX + DVDY)
                    - 2.0  * VIS(I,J) * DUDX / REN
      TXY        = - VIS(I,J) / REN  * (DUDY + DVDX)
      TYY        = - LAMB * VIS(I,J) * (DUDX + DVDY)
                    - 2.0  * VIS(I,J) * DVDY / REN
C
C ---   COMPUTE THE VISCOUS CONTRIBUTIONS TO "F" AND "G"
C
      FV(2,1,I,J)  = 0.0
      FV(2,2,I,J)  = TXX
      FV(2,3,I,J)  = TXY
      FV(2,4,I,J)  =   .5 * (UVEL(I,J) + UVEL(I+1,J))*TXX
     1                 + .5 * (VVEL(I,J) + VVEL(I+1,J))*TXY
     2                 - SDIFF * FACT1 * DTDX
     3                 - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2
     4                 + CPH2 * T1 * TEMP(I,J)) *  DY1DX
     5                 - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFO2
     6                 + CPO2 * T1 * TEMP(I,J)) *  DY2DX
     7                 - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2O
     8                 + CPH2O * T1 * TEMP(I,J)) *  DY3DX
     9                 - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2
     10                + CPN2 * T1 * TEMP(I,J)) *  DY4DX
     11                - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFOH
     12                + CPOH * T1 * TEMP(I,J)) *  DY5DX
      FV(2,5,I,J)  = - SDIFF * DY1DX
      FV(2,6,I,J)  = - SDIFF * DY2DX
      FV(2,7,I,J)  = - SDIFF * DY3DX
```

```
      FV(2,8,I,J)  = - SDIFF * DY4DX
C
      GV(2,1,I,J)  = 0.0
      GV(2,2,I,J)  = TXY
      GV(2,3,I,J)  = TYY
      GV(2,4,I,J)  =   .5 * (VVEL(I,J) + VVEL(I+1,J))*TYY
     1                 + .5 * (UVEL(I,J) + UVEL(I+1,J))*TXY
     2                 - SDIFF * FACT1 * DTDY
     3                 - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2
     4                 + CPH2 * T1 * TEMP(I,J)) *  DY1DY
     5                 - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFO2
     6                 + CPO2 * T1 * TEMP(I,J)) *  DY2DY
     7                 - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2O
     8                 + CPH2O * T1 * TEMP(I,J)) *  DY3DY
     9                 - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2
     10                + CPN2 * T1 * TEMP(I,J)) *  DY4DY
     11                - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFOH
     12                + CPOH * T1 * TEMP(I,J)) *  DY5DY
      GV(2,5,I,J)  = - SDIFF * DY1DY
      GV(2,6,I,J)  = - SDIFF * DY2DY
      GV(2,7,I,J)  = - SDIFF * DY3DY
      GV(2,8,I,J)  = - SDIFF * DY4DY
C
30    CONTINUE
C
C
C                    ---------------
C ---                |  WEST FACE  |
C                    ---------------
C
      J            = 1
      DO 40  I     = 2 , NXXX
      IF(I.EQ.NSX) GO TO 40
      IF(I.LT.NSX) J = NSY
      IF(I.GT.NSX) J = 1
      IF(I.LT.NSX) TWALL = TEMP(I,NSY)
      IF(I.GT.NSX) TWALL = TEMP(I,1)
      IF(I.LT.NSX) Y1WALL = YH2(I,NSY)
      IF(I.GT.NSX) Y1WALL = YH2(I,1)
      IF(I.LT.NSX) Y2WALL = YO2(I,NSY)
      IF(I.GT.NSX) Y2WALL = YO2(I,1)
      IF(I.LT.NSX) Y3WALL = YH2O(I,NSY)
      IF(I.GT.NSX) Y3WALL = YH2O(I,1)
      IF(I.LT.NSX) Y4WALL = YYN2(I,NSY)
      IF(I.GT.NSX) Y4WALL = YYN2(I,1)
      AV           = .5 * (AREA(I,J) + AREA(I-1,J))
C
C ---  U-VELOCITY
C
      UE           = UVEL(I,J)
```

```
      UW         = UVEL(I-1,J)
      UN         = .25 * (UVEL(I,J)      + UVEL(I,J+1)
     1                 +  UVEL(I-1,J+1) + UVEL(I-1,J))
      US         = 0.0
C
C ---   V-VELOCITY
C
      VE         = VVEL(I,J)
      VW         = VVEL(I-1,J)
      VN         = .25 * (VVEL(I,J)      + VVEL(I,J+1)
     1                 +  VVEL(I-1,J+1) + VVEL(I-1,J))
      VS         = 0.0
C
C ---   TEMPERATURE
C
      TE         = TEMP(I,J)
      TW         = TEMP(I-1,J)
      TN         = .25 * (TEMP(I,J)      + TEMP(I,J+1)
     1                 +  TEMP(I-1,J+1) + TEMP(I-1,J))
      TS         = TWALL
C
C ---   YH2
C
      Y1E        = YH2(I,J)
      Y1W        = YH2(I-1,J)
      Y1N        = .25 * (YH2(I,J)      + YH2(I,J+1)
     1                 +  YH2(I-1,J+1) + YH2(I-1,J))
      Y1S        = Y1WALL
C
C ---   YO2
C
      Y2E        = YO2(I,J)
      Y2W        = YO2(I-1,J)
      Y2N        = .25 * (YO2(I,J)      + YO2(I,J+1)
     1                 +  YO2(I-1,J+1) + YO2(I-1,J))
      Y2S        = Y2WALL
C
C ---   YH2O
C
      Y3E        = YH2O(I,J)
      Y3W        = YH2O(I-1,J)
      Y3N        = .25 * (YH2O(I,J)      + YH2O(I,J+1)
     1                 +  YH2O(I-1,J+1) + YH2O(I-1,J))
      Y3S        = Y3WALL
C
C ---   YYN2
C
      Y4E        = YYN2(I,J)
      Y4W        = YYN2(I-1,J)
      Y4N        = .25 * (YYN2(I,J)      + YYN2(I,J+1)
```

```
     1                    +  YYN2(I-1,J+1) + YYN2(I-1,J))
       Y4S          = Y4WALL
C
C ---   CALCULATE THE GRADIENT TERMS
C
C
C ---   X  GRADIENTS
C
       DUDX        = (UE * DYE(3,I,J) + UN * DYN(3,I,J)
     1              + UW * DYW(3,I,J) + US * DYS(3,I,J)) / AV
       DVDX        = (VE * DYE(3,I,J) + VN * DYN(3,I,J)
     1              + VW * DYW(3,I,J) + VS * DYS(3,I,J)) / AV
       DTDX        = (TE * DYE(3,I,J) + TN * DYN(3,I,J)
     1              + TW * DYW(3,I,J) + TS * DYS(3,I,J)) / AV
       DY1DX       = (Y1E * DYE(3,I,J) + Y1N * DYN(3,I,J)
     1              + Y1W * DYW(3,I,J) + Y1S * DYS(3,I,J)) / AV
       DY2DX       = (Y2E * DYE(3,I,J) + Y2N * DYN(3,I,J)
     1              + Y2W * DYW(3,I,J) + Y2S * DYS(3,I,J)) / AV
       DY3DX       = (Y3E * DYE(3,I,J) + Y3N * DYN(3,I,J)
     1              + Y3W * DYW(3,I,J) + Y3S * DYS(3,I,J)) / AV
       DY4DX       = (Y4E * DYE(3,I,J) + Y4N * DYN(3,I,J)
     1              + Y4W * DYW(3,I,J) + Y4S * DYS(3,I,J)) / AV
       DY5DX       = -DY1DX - DY2DX - DY3DX - DY4DX
C
C ---   Y  GRADIENTS
C
       DUDY        = - (UE * DXE(3,I,J) + UN * DXN(3,I,J)
     1                + UW * DXW(3,I,J) + US * DXS(3,I,J)) / AV
       DVDY        = - (VE * DXE(3,I,J) + VN * DXN(3,I,J)
     1                + VW * DXW(3,I,J) + VS * DXS(3,I,J)) / AV
       DTDY        = - (TE * DXE(3,I,J) + TN * DXN(3,I,J)
     1                + TW * DXW(3,I,J) + TS * DXS(3,I,J)) / AV
       DY1DY       = - (Y1E * DXE(3,I,J) + Y1N * DXN(3,I,J)
     1                + Y1W * DXW(3,I,J) + Y1S * DXS(3,I,J)) / AV
       DY2DY       = - (Y2E * DXE(3,I,J) + Y2N * DXN(3,I,J)
     1                + Y2W * DXW(3,I,J) + Y2S * DXS(3,I,J)) / AV
       DY3DY       = - (Y3E * DXE(3,I,J) + Y3N * DXN(3,I,J)
     1                + Y3W * DXW(3,I,J) + Y3S * DXS(3,I,J)) / AV
       DY4DY       = - (Y4E * DXE(3,I,J) + Y4N * DXN(3,I,J)
     1                + Y4W * DXW(3,I,J) + Y4S * DXS(3,I,J)) / AV
       DY5DY       = - DY1DY - DY2DY - DY3DY - DY4DY
C
C ---   COMPUTE THE FULL SHEAR STRESS TERMS
C
       TXX         = - LAMB * VIS(I,J) * (DUDX + DVDY)
                    - 2.0  * VIS(I,J) * DUDX / REN
       TXY         = - VIS(I,J) / REN  * (DUDY + DVDX)
       TYY         = - LAMB * VIS(I,J) * (DUDX + DVDY)
                    - 2.0  * VIS(I,J) * DVDY / REN
C
```

```
C ---   COMPUTE THE VISCOUS CONTRIBUTIONS TO "F" AND "G"
C
      FV(3,1,I,J)  = 0.0
      FV(3,2,I,J)  = TXX
      FV(3,3,I,J)  = TXY
      FV(3,4,I,J)  =   .5 * (UVEL(I,J) + UVEL(I+1,J))*TXX
     1                + .5 * (VVEL(I,J) + VVEL(I+1,J))*TXY
     2                - SDIFF * FACT1 * DTDX
     3                - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2
     4                + CPH2 * T1 * TEMP(I,J)) *  DY1DX
     5                - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFO2
     6                + CPO2 * T1 * TEMP(I,J)) *  DY2DX
     7                - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2O
     8                + CPH2O * T1 * TEMP(I,J)) *  DY3DX
     9                - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2
     10               + CPN2 * T1 * TEMP(I,J)) *  DY4DX
     11               - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFOH
     12               + CPOH * T1 * TEMP(I,J)) *  DY5DX
      FV(3,5,I,J)  = - SDIFF * DY1DX
      FV(3,6,I,J)  = - SDIFF * DY2DX
      FV(3,7,I,J)  = - SDIFF * DY3DX
      FV(3,8,I,J)  = - SDIFF * DY4DX
C
      GV(3,1,I,J)  = 0.0
      GV(3,2,I,J)  = TXY
      GV(3,3,I,J)  = TYY
      GV(3,4,I,J)  =   .5 * (VVEL(I,J) + VVEL(I+1,J))*TYY
     1                + .5 * (UVEL(I,J) + UVEL(I+1,J))*TXY
     2                - SDIFF * FACT1 * DTDY
     3                - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2
     4                + CPH2 * T1 * TEMP(I,J)) *  DY1DY
     5                - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFO2
     6                + CPO2 * T1 * TEMP(I,J)) *  DY2DY
     7                - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2O
     8                + CPH2O * T1 * TEMP(I,J)) *  DY3DY
     9                - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2
     10               + CPN2 * T1 * TEMP(I,J)) *  DY4DY
     11               - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFOH
     12               + CPOH * T1 * TEMP(I,J)) *  DY5DY
      GV(3,5,I,J)  = - SDIFF * DY1DY
      GV(3,6,I,J)  = - SDIFF * DY2DY
      GV(3,7,I,J)  = - SDIFF * DY3DY
      GV(3,8,I,J)  = - SDIFF * DY4DY
40    CONTINUE
C
C
C                  ----------------
C ---              |  SOUTH FACE  |
C                  ----------------
C
```

```
      J             = 1
      DO 50  I      = 2 , NXXX
      IF(I.EQ.NSX) GO TO 50
      IF(I.LT.NSX) J = NSY
      IF(I.GT.NSX) J = 1
      IF(I.LT.NSX) TWALL = TEMP(I,NSY)
      IF(I.GT.NSX) TWALL = TEMP(I,1)
      IF(I.LT.NSX) Y1WALL = YH2(I,NSY)
      IF(I.GT.NSX) Y1WALL = YH2(I,1)
      IF(I.LT.NSX) Y2WALL = YO2(I,NSY)
      IF(I.GT.NSX) Y2WALL = YO2(I,1)
      IF(I.LT.NSX) Y3WALL = YH2O(I,NSY)
      IF(I.GT.NSX) Y3WALL = YH2O(I,1)
      IF(I.LT.NSX) Y4WALL = YYN2(I,NSY)
      IF(I.GT.NSX) Y4WALL = YYN2(I,1)
      AV            = AREA(I,J)
C
C ---   U-VELOCITY
C
      UE            = 0.0
      UN            = UVEL(I,J)
      UW            = 0.0
      US            = -UVEL(I,J)
C
C ---   V-VELOCITY
C
      VE            = 0.0
      VN            = VVEL(I,J)
      VW            = 0.0
      VS            = -VVEL(I,J)
C
C ---   TEMPERATURE
C
      TE            = TWALL
      TN            = TEMP(I,J)
      TW            = TWALL
      TS            = TEMP(I,J)
C
C ---   YH2
C
      Y1E            = Y1WALL
      Y1N            = YH2(I,J)
      Y1W            = Y1WALL
      Y1S            = YH2(I,J)
C
C ---   YO2
C
      Y2E            = Y2WALL
      Y2N            = YO2(I,J)
      Y2W            = Y2WALL
```

```
      Y2S          = YO2(I,J)
C
C ---   YH2O
C
      Y3E          = Y3WALL
      Y3N          = YH2O(I,J)
      Y3W          = Y3WALL
      Y3S          = YH2O(I,J)
C
C ---   YYN2
C
      Y4E          = Y4WALL
      Y4N          = YYN2(I,J)
      Y4W          = Y4WALL
      Y4S          = YYN2(I,J)
C
C ---   CALCULATE THE GRADIENT TERMS
C
C
C ---   X  GRADIENTS
C
      DUDX        = (UE * DYE(4,I,J) + UN * DYN(4,I,J)
     1              + UW * DYW(4,I,J) + US * DYS(4,I,J)) / AV
      DVDX        = (VE * DYE(4,I,J) + VN * DYN(4,I,J)
     1              + VW * DYW(4,I,J) + VS * DYS(4,I,J)) / AV
      DTDX        = (TE * DYE(4,I,J) + TN * DYN(4,I,J)
     1              + TW * DYW(4,I,J) + TS * DYS(4,I,J)) / AV
      DY1DX       = (Y1E * DYE(4,I,J) + Y1N * DYN(4,I,J)
     1              + Y1W * DYW(4,I,J) + Y1S * DYS(4,I,J)) / AV
      DY2DX       = (Y2E * DYE(4,I,J) + Y2N * DYN(4,I,J)
     1              + Y2W * DYW(4,I,J) + Y2S * DYS(4,I,J)) / AV
      DY3DX       = (Y3E * DYE(4,I,J) + Y3N * DYN(4,I,J)
     1              + Y3W * DYW(4,I,J) + Y3S * DYS(4,I,J)) / AV
      DY4DX       = (Y4E * DYE(4,I,J) + Y4N * DYN(4,I,J)
     1              + Y4W * DYW(4,I,J) + Y4S * DYS(4,I,J)) / AV
      DY5DX       = -DY1DX - DY2DX - DY3DX - DY4DX
C
C ---   Y  GRADIENTS
C
      DUDY        = - (UE * DXE(4,I,J) + UN * DXN(4,I,J)
     1                + UW * DXW(4,I,J) + US * DXS(4,I,J)) / AV
      DVDY        = - (VE * DXE(4,I,J) + VN * DXN(4,I,J)
     1                + VW * DXW(4,I,J) + VS * DXS(4,I,J)) / AV
      DTDY        = - (TE * DXE(4,I,J) + TN * DXN(4,I,J)
     1                + TW * DXW(4,I,J) + TS * DXS(4,I,J)) / AV
      DY1DY       = - (Y1E * DXE(4,I,J) + Y1N * DXN(4,I,J)
     1                + Y1W * DXW(4,I,J) + Y1S * DXS(4,I,J)) / AV
      DY2DY       = - (Y2E * DXE(4,I,J) + Y2N * DXN(4,I,J)
     1                + Y2W * DXW(4,I,J) + Y2S * DXS(4,I,J)) / AV
      DY3DY       = - (Y3E * DXE(4,I,J) + Y3N * DXN(4,I,J)
```

```
     1                + Y3W * DXW(4,I,J) + Y3S * DXS(4,I,J)) / AV
       DY4DY      = - (Y4E * DXE(4,I,J) + Y4N * DXN(4,I,J)
     1                + Y4W * DXW(4,I,J) + Y4S * DXS(4,I,J)) / AV
       DY5DY      = - DY1DY - DY2DY - DY3DY - DY4DY
C
C ---   COMPUTE THE FULL SHEAR STRESS TERMS
C
       TXX        = - LAMB * VIS(I,J) * (DUDX + DVDY)
                    - 2.0  * VIS(I,J) * DUDX / REN
       TXY        = - VIS(I,J) / REN  * (DUDY + DVDX)
       TYY        = - LAMB * VIS(I,J) * (DUDX + DVDY)
                    - 2.0  * VIS(I,J) * DVDY / REN
C
C ---   COMPUTE THE VISCOUS CONTRIBUTIONS TO "F" AND "G"
C
       FV(4,1,I,J)  = 0.0
       FV(4,2,I,J)  = TXX
       FV(4,3,I,J)  = TXY
       FV(4,4,I,J)  =   - SDIFF * FACT1 * DTDX
     3                  - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2
     4                  + CPH2 * T1 * TEMP(I,J)) *  DY1DX
     5                  - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFO2
     6                  + CPO2 * T1 * TEMP(I,J)) *  DY2DX
     7                  - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2O
     8                  + CPH2O * T1 * TEMP(I,J)) *  DY3DX
     9                  - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2
     10                 + CPN2 * T1 * TEMP(I,J)) *  DY4DX
     11                 - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFOH
     12                 + CPOH * T1 * TEMP(I,J)) *  DY5DX
       FV(4,5,I,J)  = - SDIFF * DY1DX
       FV(4,6,I,J)  = - SDIFF * DY2DX
       FV(4,7,I,J)  = - SDIFF * DY3DX
       FV(4,8,I,J)  = - SDIFF * DY4DX
C
       GV(4,1,I,J)  = 0.0
       GV(4,2,I,J)  = TXY
       GV(4,3,I,J)  = TYY
       GV(4,4,I,J)  =   - SDIFF * FACT1 * DTDY
     3                  - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2
     4                  + CPH2 * T1 * TEMP(I,J)) *  DY1DY
     5                  - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFO2
     6                  + CPO2 * T1 * TEMP(I,J)) *  DY2DY
     7                  - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2O
     8                  + CPH2O * T1 * TEMP(I,J)) *  DY3DY
     9                  - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2
     10                 + CPN2 * T1 * TEMP(I,J)) *  DY4DY
     11                 - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFOH
     12                 + CPOH * T1 * TEMP(I,J)) *  DY5DY
       GV(4,5,I,J)  = - SDIFF * DY1DY
       GV(4,6,I,J)  = - SDIFF * DY2DY
```

```
      GV(4,7,I,J)  = - SDIFF * DY3DY
      GV(4,8,I,J)  = - SDIFF * DY4DY
50    CONTINUE
C
C
C ---    STEP UPPER EDGE CELL EVALUATION  I = NSX , J = NSY
C
C
C                  ---------------
C ---              !  EAST FACE  !
C                  ---------------
C
      I          = NSX
      J          = NSY
      AV         = .5 * (AREA(I,J) + AREA(I+1,J))
C
C ---   U-VELOCITY
C
      UE         = UVEL(I+1,J)
      UW         = UVEL(I,J)
      UN         = .25 * (UVEL(I+1,J) + UVEL(I+1,J+1)
     1                 +  UVEL(I,J+1) + UVEL(I,J))
      US         = .25 * (UVEL(I+1,J) + UVEL(I,J)
     1                 +  UVEL(I,J-1) + UVEL(I+1,J-1))
C
C ---   V-VELOCITY
C
      VE         = VVEL(I+1,J)
      VW         = VVEL(I,J)
      VN         = .25 * (VVEL(I+1,J) + VVEL(I+1,J+1)
     1                 +  VVEL(I,J+1) + VVEL(I+1,J))
      VS         = .25 * (VVEL(I+1,J) + VVEL(I,J)
     1                 +  VVEL(I,J-1) + VVEL(I+1,J-1))
C
C ---   TEMPERATURE
C
      TE         = TEMP(I+1,J)
      TW         = TEMP(I,J)
      TN         = .25 * (TEMP(I+1,J) + TEMP(I+1,J+1)
     1                 +  TEMP(I,J+1) + TEMP(I+1,J))
      TS         = .25 * (TEMP(I+1,J) + TEMP(I,J)
     1                 +  TEMP(I,J-1) + TEMP(I+1,J-1))
C
C ---   YH2
C
      Y1E         = YH2(I+1,J)
      Y1W         = YH2(I,J)
      Y1N         = .25 * (YH2(I+1,J) + YH2(I+1,J+1)
     1                  +  YH2(I,J+1) + YH2(I+1,J))
      Y1S         = .25 * (YH2(I+1,J) + YH2(I,J)
```

```
     1                 +  YH2(I,J-1) + YH2(I+1,J-1))
C
C ---   YO2
C
      Y2E        = YO2(I+1,J)
      Y2W        = YO2(I,J)
      Y2N        = .25 * (YO2(I+1,J) + YO2(I+1,J+1)
     1                 +  YO2(I,J+1) + YO2(I+1,J))
      Y2S        = .25 * (YO2(I+1,J) + YO2(I,J)
     1                 +  YO2(I,J-1) + YO2(I+1,J-1))
C
C ---   YH2O
C
      Y3E        = YH2O(I+1,J)
      Y3W        = YH2O(I,J)
      Y3N        = .25 * (YH2O(I+1,J) + YH2O(I+1,J+1)
     1                 +  YH2O(I,J+1) + YH2O(I+1,J))
      Y3S        = .25 * (YH2O(I+1,J) + YH2O(I,J)
     1                 +  YH2O(I,J-1) + YH2O(I+1,J-1))
C
C ---   YYN2
C
      Y4E        = YYN2(I+1,J)
      Y4W        = YYN2(I,J)
      Y4N        = .25 * (YYN2(I+1,J) + YYN2(I+1,J+1)
     1                 +  YYN2(I,J+1) + YYN2(I+1,J))
      Y4S        = .25 * (YYN2(I+1,J) + YYN2(I,J)
     1                 +  YYN2(I,J-1) + YYN2(I+1,J-1))
C
C ---   CALCULATE THE GRADIENT TERMS
C
C
C ---   X  GRADIENTS
C
      DUDX       = (UE * DYE(1,I,J) + UN * DYN(1,I,J)
     1            + UW * DYW(1,I,J) + US * DYS(1,I,J)) / AV
      DVDX       = (VE * DYE(1,I,J) + VN * DYN(1,I,J)
     1            + VW * DYW(1,I,J) + VS * DYS(1,I,J)) / AV
      DTDX       = (TE * DYE(1,I,J) + TN * DYN(1,I,J)
     1            + TW * DYW(1,I,J) + TS * DYS(1,I,J)) / AV
      DY1DX      = (Y1E * DYE(1,I,J) + Y1N * DYN(1,I,J)
     1            + Y1W * DYW(1,I,J) + Y1S * DYS(1,I,J)) / AV
      DY2DX      = (Y2E * DYE(1,I,J) + Y2N * DYN(1,I,J)
     1            + Y2W * DYW(1,I,J) + Y2S * DYS(1,I,J)) / AV
      DY3DX      = (Y3E * DYE(1,I,J) + Y3N * DYN(1,I,J)
     1            + Y3W * DYW(1,I,J) + Y3S * DYS(1,I,J)) / AV
      DY4DX      = (Y4E * DYE(1,I,J) + Y4N * DYN(1,I,J)
     1            + Y4W * DYW(1,I,J) + Y4S * DYS(1,I,J)) / AV
      DY5DX      = -DY1DX - DY2DX - DY3DX - DY4DX
C
```

```
C ---   Y  GRADIENTS
C
      DUDY        = - (UE * DXE(1,I,J) + UN * DXN(1,I,J)
     1                + UW * DXW(1,I,J) + US * DXS(1,I,J)) / AV
      DVDY        = - (VE * DXE(1,I,J) + VN * DXN(1,I,J)
     1                + VW * DXW(1,I,J) + VS * DXS(1,I,J)) / AV
      DTDY        = - (TE * DXE(1,I,J) + TN * DXN(1,I,J)
     1                + TW * DXW(1,I,J) + TS * DXS(1,I,J)) / AV
      DY1DY       = (Y1E * DXE(1,I,J) + Y1N * DXN(1,I,J)
     1              + Y1W * DXW(1,I,J) + Y1S * DXS(1,I,J)) / AV
      DY2DY       = (Y2E * DXE(1,I,J) + Y2N * DXN(1,I,J)
     1              + Y2W * DXW(1,I,J) + Y2S * DXS(1,I,J)) / AV
      DY3DY       = (Y3E * DXE(1,I,J) + Y3N * DXN(1,I,J)
     1              + Y3W * DXW(1,I,J) + Y3S * DXS(1,I,J)) / AV
      DY4DY       = (Y4E * DXE(1,I,J) + Y4N * DXN(1,I,J)
     1              + Y4W * DXW(1,I,J) + Y4S * DXS(1,I,J)) / AV
      DY5DY       = - DY1DY - DY2DY - DY3DY - DY4DY
C
C ---   COMPUTE THE FULL SHEAR STRESS TERMS
C
      TXX         = - LAMB * VIS(I,J) * (DUDX + DVDY)
                   - 2.0  * VIS(I,J) * DUDX / REN
      TXY         = - VIS(I,J) / REN  * (DUDY + DVDX)
      TYY         = - LAMB * VIS(I,J) * (DUDX + DVDY)
                   - 2.0  * VIS(I,J) * DVDY / REN
C
C ---   COMPUTE THE VISCOUS CONTRIBUTIONS TO "F" AND "G"
C
      FV(1,1,I,J)  = 0.0
      FV(1,2,I,J)  = TXX
      FV(1,3,I,J)  = TXY
      FV(1,4,I,J)  =   .5 * (UVEL(I,J) + UVEL(I+1,J))*TXX
     1                + .5 * (VVEL(I,J) + VVEL(I+1,J))*TXY
     2                - SDIFF * FACT1 * DTDX
     3                - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2
     4                + CPH2 * T1 * TEMP(I,J)) *  DY1DX
     5                - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFO2
     6                + CPO2 * T1 * TEMP(I,J)) *  DY2DX
     7                - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2O
     8                + CPH2O * T1 * TEMP(I,J)) *  DY3DX
     9                - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2
     10               + CPN2 * T1 * TEMP(I,J)) *  DY4DX
     11               - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFOH
     12               + CPOH * T1 * TEMP(I,J)) *  DY5DX
      FV(1,5,I,J)  = - SDIFF * DY1DX
      FV(1,6,I,J)  = - SDIFF * DY2DX
      FV(1,7,I,J)  = - SDIFF * DY3DX
      FV(1,8,I,J)  = - SDIFF * DY4DX
      GV(1,1,I,J)  = 0.0
      GV(1,2,I,J)  = TXY
```

```
      GV(1,3,I,J)   = TYY
      GV(1,4,I,J)   =    .5 * (VVEL(I,J) + VVEL(I+1,J))*TYY
     1                   + .5 * (UVEL(I,J) + UVEL(I+1,J))*TXY
     2                   - SDIFF * FACT1 * DTDY
     3                   - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2
     4                   + CPH2 * T1 * TEMP(I,J)) *  DY1DY
     5                   - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFO2
     6                   + CPO2 * T1 * TEMP(I,J)) *  DY2DY
     7                   - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2O
     8                   + CPH2O * T1 * TEMP(I,J)) *  DY3DY
     9                   - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2
     10                  + CPN2 * T1 * TEMP(I,J)) *  DY4DY
     11                  - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFOH
     12                  + CPOH * T1 * TEMP(I,J)) *  DY5DY
      GV(1,5,I,J)   = - SDIFF * DY1DY
      GV(1,6,I,J)   = - SDIFF * DY2DY
      GV(1,7,I,J)   = - SDIFF * DY3DY
      GV(1,8,I,J)   = - SDIFF * DY4DY
C
C                 ------------------
C ---             |   NORTH   FACE  |
C                 ------------------
C
      I           = NSX
      J           = NSY
      AV          = .5 * (AREA(I,J) + AREA(I,J+1))
C
C ---   U-VELOCITY
C
      UE          = .25 * (UVEL(I+1,J) + UVEL(I+1,J+1)
     1                 +  UVEL(I,J+1) + UVEL(I,J))
      UN          = UVEL(I,J+1)
      UW          = .25 * (UVEL(I,J+1) + UVEL(I-1,J+1)
     1                 +  UVEL(I-1,J) + UVEL(I,J))
      US          = UVEL(I,J)
C
C ---   V-VELOCITY
C
      VE          = .25 * (VVEL(I+1,J) + VVEL(I+1,J+1)
     1                 +  VVEL(I,J+1) + VVEL(I,J))
      VN          = VVEL(I,J+1)
      VW          = .25 * (VVEL(I,J+1) + VVEL(I-1,J+1)
     1                 +  VVEL(I-1,J) + VVEL(I,J))
      VS          = VVEL(I,J)
C
C ---   TEMPERATURE
C
      TE          = .25 * (TEMP(I+1,J) + TEMP(I+1,J+1)
     1                 +  TEMP(I,J+1) + TEMP(I,J))
      TN          = TEMP(I,J+1)
```

```
      TW          = .25 * (TEMP(I,J+1) + TEMP(I-1,J+1)
     1                +  TEMP(I-1,J) + TEMP(I,J))
      TS          = TEMP(I,J)
C
C ---   YH2
C
      Y1E         = .25 * (YH2(I+1,J) + YH2(I+1,J+1)
     1                +  YH2(I,J+1) + YH2(I,J))
      Y1N         = YH2(I,J+1)
      Y1W         = .25 * (YH2(I,J+1) + YH2(I-1,J+1)
     1                +  YH2(I-1,J) + YH2(I,J))
      Y1S         = YH2(I,J)
C
C ---   YO2
C
      Y2E         = .25 * (YO2(I+1,J) + YO2(I+1,J+1)
     1                +  YO2(I,J+1) + YO2(I,J))
      Y2N         = YO2(I,J+1)
      Y2W         = .25 * (YO2(I,J+1) + YO2(I-1,J+1)
     1                +  YO2(I-1,J) + YO2(I,J))
      Y2S         = YO2(I,J)
C
C ---   YH2O
C
      Y3E         = .25 * (YH2O(I+1,J) + YH2O(I+1,J+1)
     1                +  YH2O(I,J+1) + YH2O(I,J))
      Y3N         = YH2O(I,J+1)
      Y3W         = .25 * (YH2O(I,J+1) + YH2O(I-1,J+1)
     1                +  YH2O(I-1,J) + YH2O(I,J))
      Y3S         = YH2O(I,J)
C
C ---   YYN2
C
      Y4E         = .25 * (YYN2(I+1,J) + YYN2(I+1,J+1)
     1                +  YYN2(I,J+1) + YYN2(I,J))
      Y4N         = YYN2(I,J+1)
      Y4W         = .25 * (YYN2(I,J+1) + YYN2(I-1,J+1)
     1                +  YYN2(I-1,J) + YYN2(I,J))
      Y4S         = YYN2(I,J)
C
C ---   CALCULATE THE GRADIENT TERMS
C
C
C ---   X  GRADIENTS
C
      DUDX        = (UE * DYE(2,I,J) + UN * DYN(2,I,J)
     1              + UW * DYW(2,I,J) + US * DYS(2,I,J)) / AV
      DVDX        = (VE * DYE(2,I,J) + VN * DYN(2,I,J)
     1              + VW * DYW(2,I,J) + VS * DYS(2,I,J)) / AV
      DTDX        = (TE * DYE(2,I,J) + TN * DYN(2,I,J)
```

```
     1              + TW * DYW(2,I,J) + TS * DYS(2,I,J)) / AV
       DY1DX       = (Y1E * DYE(2,I,J) + Y1N * DYN(2,I,J)
     1              + Y1W * DYW(2,I,J) + Y1S * DYS(2,I,J)) / AV
       DY2DX       = (Y2E * DYE(2,I,J) + Y2N * DYN(2,I,J)
     1              + Y2W * DYW(2,I,J) + Y2S * DYS(2,I,J)) / AV
       DY3DX       = (Y3E * DYE(2,I,J) + Y3N * DYN(2,I,J)
     1              + Y3W * DYW(2,I,J) + Y3S * DYS(2,I,J)) / AV
       DY4DX       = (Y4E * DYE(2,I,J) + Y4N * DYN(2,I,J)
     1              + Y4W * DYW(2,I,J) + Y4S * DYS(2,I,J)) / AV
       DY5DX       = -DY1DX - DY2DX - DY3DX - DY4DX
C
C ---   Y  GRADIENTS
C
       DUDY        = - (UE * DXE(2,I,J) + UN * DXN(2,I,J)
     1                + UW * DXW(2,I,J) + US * DXS(2,I,J)) / AV
       DVDY        = - (VE * DXE(2,I,J) + VN * DXN(2,I,J)
     1                + VW * DXW(2,I,J) + VS * DXS(2,I,J)) / AV
       DTDY        = - (TE * DXE(2,I,J) + TN * DXN(2,I,J)
     1                + TW * DXW(2,I,J) + TS * DXS(2,I,J)) / AV
       DY1DY       = - (Y1E * DXE(2,I,J) + Y1N * DXN(2,I,J)
     1                + Y1W * DXW(2,I,J) + Y1S * DXS(2,I,J)) / AV
       DY2DY       = - (Y2E * DXE(2,I,J) + Y2N * DXN(2,I,J)
     1                + Y2W * DXW(2,I,J) + Y2S * DXS(2,I,J)) / AV
       DY3DY       = - (Y3E * DXE(2,I,J) + Y3N * DXN(2,I,J)
     1                + Y3W * DXW(2,I,J) + Y3S * DXS(2,I,J)) / AV
       DY4DY       = - (Y4E * DXE(2,I,J) + Y4N * DXN(2,I,J)
     1                + Y4W * DXW(2,I,J) + Y4S * DXS(2,I,J)) / AV
       DY5DY       = - DY1DY - DY2DY - DY3DY - DY4DY
C
C ---   COMPUTE THE FULL SHEAR STRESS TERMS
C
       TXX         = - LAMB * VIS(I,J) * (DUDX + DVDY)
                     - 2.0  * VIS(I,J) * DUDX / REN
       TXY         = - VIS(I,J) / REN  * (DUDY + DVDX)
       TYY         = - LAMB * VIS(I,J) * (DUDX + DVDY)
                     - 2.0  * VIS(I,J) * DVDY / REN
C
C ---   COMPUTE THE VISCOUS CONTRIBUTIONS TO "F" AND "G"
C
       FV(2,1,I,J)  = 0.0
       FV(2,2,I,J)  = TXX
       FV(2,3,I,J)  = TXY
       FV(2,4,I,J)  =   .5 * (UVEL(I,J) + UVEL(I,J+1))*TXX
     1                + .5 * (VVEL(I,J) + VVEL(I,J+1))*TXY
     2                - SDIFF * FACT1 * DTDX
     3                - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2
     4                + CPH2 * T1 * TEMP(I,J)) *  DY1DX
     5                - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFO2
     6                + CPO2 * T1 * TEMP(I,J)) *  DY2DX
     7                - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2O
```

```
     8                   + CPH2O * T1 * TEMP(I,J)) *  DY3DX
     9                   - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2
     10                  + CPN2 * T1 * TEMP(I,J)) *  DY4DX
     11                  - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFOH
     12                  + CPOH * T1 * TEMP(I,J)) *  DY5DX
        FV(2,5,I,J)  = - SDIFF * DY1DX
        FV(2,6,I,J)  = - SDIFF * DY2DX
        FV(2,7,I,J)  = - SDIFF * DY3DX
        FV(2,8,I,J)  = - SDIFF * DY4DX
C
        GV(2,1,I,J)  = 0.0
        GV(2,2,I,J)  = TXY
        GV(2,3,I,J)  = TYY
        GV(2,4,I,J)  =   .5 * (VVEL(I,J) + VVEL(I,J+1))*TYY
     1                   + .5 * (UVEL(I,J) + UVEL(I,J+1))*TXY
     2                   - SDIFF * FACT1 * DTDY
     3                   - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2
     4                   + CPH2 * T1 * TEMP(I,J)) *  DY1DY
     5                   - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFO2
     6                   + CPO2 * T1 * TEMP(I,J)) *  DY2DY
     7                   - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2O
     8                   + CPH2O * T1 * TEMP(I,J)) *  DY3DY
     9                   - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2
     10                  + CPN2 * T1 * TEMP(I,J)) *  DY4DY
     11                  - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFOH
     12                  + CPOH * T1 * TEMP(I,J)) *  DY5DY
        GV(2,5,I,J)  = - SDIFF * DY1DY
        GV(2,6,I,J)  = - SDIFF * DY2DY
        GV(2,7,I,J)  = - SDIFF * DY3DY
        GV(2,8,I,J)  = - SDIFF * DY4DY
C
C
C                   ---------------
C ---               |  WEST FACE  |
C                   ---------------
C
        I           = NSX
        J           = NSY
        AV          = .5 * (AREA(I,J) + AREA(I-1,J))
C
C ---   U-VELOCITY
C
        UE          = UVEL(I,J)
        UW          = UVEL(I-1,J)
        UN          = .25 * (UVEL(I,J)     + UVEL(I,J+1)
     1                   +  UVEL(I-1,J+1) + UVEL(I-1,J))
        US          = 0.0
C
C ---   V-VELOCITY
C
```

```
      VE         = VVEL(I,J)
      VW         = VVEL(I-1,J)
      VN         = .25 * (VVEL(I,J)     + VVEL(I,J+1)
     1                + VVEL(I-1,J+1) + VVEL(I-1,J))
      VS         = 0.0
C
C ---  TEMPERATURE
C
      TE         = TEMP(I,J)
      TW         = TEMP(I-1,J)
      TN         = .25 * (TEMP(I,J)     + TEMP(I,J+1)
     1                + TEMP(I-1,J+1) + TEMP(I-1,J))
      TS         = TEMP(I,J)
C
C ---  YH2
C
      Y1E         = YH2(I,J)
      Y1W         = YH2(I-1,J)
      Y1N         = .25 * (YH2(I,J)     + YH2(I,J+1)
     1                + YH2(I-1,J+1) + YH2(I-1,J))
      Y1S         = YH2(I,J)
C
C ---  YO2
C
      Y2E         = YO2(I,J)
      Y2W         = YO2(I-1,J)
      Y2N         = .25 * (YO2(I,J)     + YO2(I,J+1)
     1                + YO2(I-1,J+1) + YO2(I-1,J))
      Y2S         = YO2(I,J)
C
C ---  YH2O
C
      Y3E         = YH2O(I,J)
      Y3W         = YH2O(I-1,J)
      Y3N         = .25 * (YH2O(I,J)     + YH2O(I,J+1)
     1                + YH2O(I-1,J+1) + YH2O(I-1,J))
      Y3S         = YH2O(I,J)
C
C ---  YYN2
C
      Y4E         = YYN2(I,J)
      Y4W         = YYN2(I-1,J)
      Y4N         = .25 * (YYN2(I,J)     + YYN2(I,J+1)
     1                + YYN2(I-1,J+1) + YYN2(I-1,J))
      Y4S         = YYN2(I,J)
C
C ---  CALCULATE THE GRADIENT TERMS
C
C
C
```

```
C ---   X  GRADIENTS
C
        DUDX        = (UE * DYE(3,I,J) + UN * DYN(3,I,J)
     1               + UW * DYW(3,I,J) + US * DYS(3,I,J)) / AV
        DVDX        = (VE * DYE(3,I,J) + VN * DYN(3,I,J)
     1               + VW * DYW(3,I,J) + VS * DYS(3,I,J)) / AV
        DTDX        = (TE * DYE(3,I,J) + TN * DYN(3,I,J)
     1               + TW * DYW(3,I,J) + TS * DYS(3,I,J)) / AV
        DY1DX       = (Y1E * DYE(3,I,J) + Y1N * DYN(3,I,J)
     1               + Y1W * DYW(3,I,J) + Y1S * DYS(3,I,J)) / AV
        DY2DX       = (Y2E * DYE(3,I,J) + Y2N * DYN(3,I,J)
     1               + Y2W * DYW(3,I,J) + Y2S * DYS(3,I,J)) / AV
        DY3DX       = (Y3E * DYE(3,I,J) + Y3N * DYN(3,I,J)
     1               + Y3W * DYW(3,I,J) + Y3S * DYS(3,I,J)) / AV
        DY4DX       = (Y4E * DYE(3,I,J) + Y4N * DYN(3,I,J)
     1               + Y4W * DYW(3,I,J) + Y4S * DYS(3,I,J)) / AV
        DY5DX       = -DY1DX - DY2DX - DY3DX - DY4DX
C
C ---   Y  GRADIENTS
C
        DUDY        = - (UE * DXE(3,I,J) + UN * DXN(3,I,J)
     1                 + UW * DXW(3,I,J) + US * DXS(3,I,J)) / AV
        DVDY        = - (VE * DXE(3,I,J) + VN * DXN(3,I,J)
     1                 + VW * DXW(3,I,J) + VS * DXS(3,I,J)) / AV
        DTDY        = - (TE * DXE(3,I,J) + TN * DXN(3,I,J)
     1                 + TW * DXW(3,I,J) + TS * DXS(3,I,J)) / AV
        DY1DY       = - (Y1E * DXE(3,I,J) + Y1N * DXN(3,I,J)
     1                 + Y1W * DXW(3,I,J) + Y1S * DXS(3,I,J)) / AV
        DY2DY       = - (Y2E * DXE(3,I,J) + Y2N * DXN(3,I,J)
     1                 + Y2W * DXW(3,I,J) + Y2S * DXS(3,I,J)) / AV
        DY3DY       = - (Y3E * DXE(3,I,J) + Y3N * DXN(3,I,J)
     1                 + Y3W * DXW(3,I,J) + Y3S * DXS(3,I,J)) / AV
        DY4DY       = - (Y4E * DXE(3,I,J) + Y4N * DXN(3,I,J)
     1                 + Y4W * DXW(3,I,J) + Y4S * DXS(3,I,J)) / AV
        DY5DY       = - DY1DY - DY2DY - DY3DY - DY4DY
C
C ---   COMPUTE THE FULL SHEAR STRESS TERMS
C
        TXX         = - LAMB * VIS(I,J) * (DUDX + DVDY)
                     - 2.0  * VIS(I,J) * DUDX / REN
        TXY         = - VIS(I,J) / REN  * (DUDY + DVDX)
        TYY         = - LAMB * VIS(I,J) * (DUDX + DVDY)
                     - 2.0  * VIS(I,J) * DVDY / REN
C
C ---   COMPUTE THE VISCOUS CONTRIBUTIONS TO "F" AND "G"
C
        FV(3,1,I,J)  = 0.0
        FV(3,2,I,J)  = TXX
        FV(3,3,I,J)  = TXY
        FV(3,4,I,J)  =   .5 * (UVEL(I,J) + UVEL(I,J+1))*TXX
```

```
     1                   + .5 * (VVEL(I,J) + VVEL(I,J+1))*TXY
     2                   - SDIFF * FACT1 * DTDX
     3                   - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2
     4                   + CPH2 * T1 * TEMP(I,J)) *  DY1DX
     5                   - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFO2
     6                   + CPO2 * T1 * TEMP(I,J)) *  DY2DX
     7                   - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2O
     8                   + CPH2O * T1 * TEMP(I,J)) *  DY3DX
     9                   - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2
     10                  + CPN2 * T1 * TEMP(I,J)) *  DY4DX
     11                  - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFOH
     12                  + CPOH * T1 * TEMP(I,J)) *  DY5DX
       FV(3,5,I,J)  = - SDIFF * DY1DX
       FV(3,6,I,J)  = - SDIFF * DY2DX
       FV(3,7,I,J)  = - SDIFF * DY3DX
       FV(3,8,I,J)  = - SDIFF * DY4DX

       GV(3,1,I,J)  = 0.0
       GV(3,2,I,J)  = TXY
       GV(3,3,I,J)  = TYY
       GV(3,4,I,J)  =   .5 * (VVEL(I,J) + VVEL(I,J+1))*TYY
     1                   + .5 * (UVEL(I,J) + UVEL(I,J+1))*TXY
     2                   - SDIFF * FACT1 * DTDY
     3                   - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2
     4                   + CPH2 * T1 * TEMP(I,J)) *  DY1DY
     5                   - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFO2
     6                   + CPO2 * T1 * TEMP(I,J)) *  DY2DY
     7                   - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2O
     8                   + CPH2O * T1 * TEMP(I,J)) *  DY3DY
     9                   - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2
     10                  + CPN2 * T1 * TEMP(I,J)) *  DY4DY
     11                  - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFOH
     12                  + CPOH * T1 * TEMP(I,J)) *  DY5DY
       GV(3,5,I,J)  = - SDIFF * DY1DY
       GV(3,6,I,J)  = - SDIFF * DY2DY
       GV(3,7,I,J)  = - SDIFF * DY3DY
       GV(3,8,I,J)  = - SDIFF * DY4DY
C
C
C                  ----------------
C ---              |  SOUTH FACE  |
C                  ----------------
C
       I          = NSX
       J          = NSY
       AV         = .5 * (AREA(I,J) + AREA(I,J-1))
C
C ---   U-VELOCITY
C
       UE         = .25 * (UVEL(I+1,J)  + UVEL(I,J)
```

```
     1                  +  UVEL(I,J-1)  + UVEL(I+1,J-1))
      UN          = UVEL(I,J)
      UW          = 0.0
      US          = UVEL(I,J-1)
C
C ---   V-VELOCITY
C
      VE          = .25 * (VVEL(I+1,J)  + VVEL(I,J)
     1                  +  VVEL(I,J-1)  + VVEL(I+1,J-1))
      VN          = VVEL(I,J)
      VW          = 0.0
      VS          = VVEL(I,J-1)
C
C ---   TEMPERATURE
C
      TE          = .25 * (TEMP(I+1,J)  + TEMP(I,J)
     1                  +  TEMP(I,J-1)  + TEMP(I+1,J-1))
      TN          = TEMP(I,J)
      TW          = TEMP(I,J)
      TS          = TEMP(I,J-1)
C
C ---   YH2
C
      Y1E          = .25 * (YH2(I+1,J)  + YH2(I,J)
     1                  +  YH2(I,J-1)  + YH2(I+1,J-1))
      Y1N          = YH2(I,J)
      Y1W          = YH2(I,J)
      Y1S          = YH2(I,J-1)
C
C ---   YO2
C
      Y2E          = .25 * (YO2(I+1,J)  + YO2(I,J)
     1                  +  YO2(I,J-1)  + YO2(I+1,J-1))
      Y2N          = YO2(I,J)
      Y2W          = YO2(I,J)
      Y2S          = YO2(I,J-1)
C
C ---   YH2O
C
      Y3E          = .25 * (YH2O(I+1,J)  + YH2O(I,J)
     1                  +  YH2O(I,J-1)  + YH2O(I+1,J-1))
      Y3N          = YH2O(I,J)
      Y3W          = YH2O(I,J)
      Y3S          = YH2O(I,J-1)
C
C ---   YYN2
C
      Y4E          = .25 * (YYN2(I+1,J)  + YYN2(I,J)
     1                  +  YYN2(I,J-1)  + YYN2(I+1,J-1))
      Y4N          = YYN2(I,J)
```

```
      Y4W         = YYN2(I,J)
      Y4S         = YYN2(I,J-1)
C
C ---   CALCULATE THE GRADIENT TERMS
C
C
C ---   X  GRADIENTS
C
      DUDX        = (UE * DYE(4,I,J) + UN * DYN(4,I,J)
     1             + UW * DYW(4,I,J) + US * DYS(4,I,J)) / AV
      DVDX        = (VE * DYE(4,I,J) + VN * DYN(4,I,J)
     1             + VW * DYW(4,I,J) + VS * DYS(4,I,J)) / AV
      DTDX        = (TE * DYE(4,I,J) + TN * DYN(4,I,J)
     1             + TW * DYW(4,I,J) + TS * DYS(4,I,J)) / AV
      DY1DX       = (Y1E * DYE(4,I,J) + Y1N * DYN(4,I,J)
     1             + Y1W * DYW(4,I,J) + Y1S * DYS(4,I,J)) / AV
      DY2DX       = (Y2E * DYE(4,I,J) + Y2N * DYN(4,I,J)
     1             + Y2W * DYW(4,I,J) + Y2S * DYS(4,I,J)) / AV
      DY3DX       = (Y3E * DYE(4,I,J) + Y3N * DYN(4,I,J)
     1             + Y3W * DYW(4,I,J) + Y3S * DYS(4,I,J)) / AV
      DY4DX       = (Y4E * DYE(4,I,J) + Y4N * DYN(4,I,J)
     1             + Y4W * DYW(4,I,J) + Y4S * DYS(4,I,J)) / AV
      DY5DX       = -DY1DX - DY2DX - DY3DX - DY4DX
C
C ---   Y  GRADIENTS
C
      DUDY        = - (UE * DXE(4,I,J) + UN * DXN(4,I,J)
     1               + UW * DXW(4,I,J) + US * DXS(4,I,J)) / AV
      DVDY        = - (VE * DXE(4,I,J) + VN * DXN(4,I,J)
     1               + VW * DXW(4,I,J) + VS * DXS(4,I,J)) / AV
      DTDY        = - (TE * DXE(4,I,J) + TN * DXN(4,I,J)
     1               + TW * DXW(4,I,J) + TS * DXS(4,I,J)) / AV
      DY1DY       = - (Y1E * DXE(4,I,J) + Y1N * DXN(4,I,J)
     1               + Y1W * DXW(4,I,J) + Y1S * DXS(4,I,J)) / AV
      DY2DY       = - (Y2E * DXE(4,I,J) + Y2N * DXN(4,I,J)
     1               + Y2W * DXW(4,I,J) + Y2S * DXS(4,I,J)) / AV
      DY3DY       = - (Y3E * DXE(4,I,J) + Y3N * DXN(4,I,J)
     1               + Y3W * DXW(4,I,J) + Y3S * DXS(4,I,J)) / AV
      DY4DY       = - (Y4E * DXE(4,I,J) + Y4N * DXN(4,I,J)
     1               + Y4W * DXW(4,I,J) + Y4S * DXS(4,I,J)) / AV
      DY5DY       = - DY1DY - DY2DY - DY3DY - DY4DY
C
C ---   COMPUTE THE FULL SHEAR STRESS TERMS
C
      TXX         = - LAMB * VIS(I,J) * (DUDX + DVDY)
                   - 2.0  * VIS(I,J) * DUDX / REN
      TXY         = - VIS(I,J) / REN  * (DUDY + DVDX)
      TYY         = - LAMB * VIS(I,J) * (DUDX + DVDY)
                   - 2.0  * VIS(I,J) * DVDY / REN
C
```

```
C ---   COMPUTE THE VISCOUS CONTRIBUTIONS TO "F" AND "G"
C
      FV(4,1,I,J)  = 0.0
      FV(4,2,I,J)  = TXX
      FV(4,3,I,J)  = TXY
      FV(4,4,I,J)  =   .5 * (UVEL(I,J) + UVEL(I,J+1))*TXX
     1                + .5 * (VVEL(I,J) + VVEL(I,J+1))*TXY
     2                - SDIFF * FACT1 * DTDX
     3                - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2
     4                + CPH2 * T1 * TEMP(I,J)) *  DY1DX
     5                - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFO2
     6                + CPO2 * T1 * TEMP(I,J)) *  DY2DX
     7                - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2O
     8                + CPH2O * T1 * TEMP(I,J)) *  DY3DX
     9                - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFN2
     10               + CPN2 * T1 * TEMP(I,J)) *  DY4DX
     11               - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFOH
     12               + CPOH * T1 * TEMP(I,J)) *  DY5DX
      FV(4,5,I,J)  = - SDIFF * DY1DX
      FV(4,6,I,J)  = - SDIFF * DY2DX
      FV(4,7,I,J)  = - SDIFF * DY3DX
      FV(4,8,I,J)  = - SDIFF * DY4DX
C
      GV(4,1,I,J)  = 0.0
      GV(4,2,I,J)  = TXY
      GV(4,3,I,J)  = TYY
      GV(4,4,I,J)  =   .5 * (VVEL(I,J) + VVEL(I,J+1))*TYY
     1                + .5 * (UVEL(I,J) + UVEL(I,J+1))*TXY
     2                - SDIFF * FACT1 * DTDY
     3                - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2
     4                + CPH2 * T1 * TEMP(I,J)) *  DY1DY
     5                - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFO2
     6                + CPO2 * T1 * TEMP(I,J)) *  DY2DY
     7                - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2O
     8                + CPH2O * T1 * TEMP(I,J)) *  DY3DY
     9                - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2
     10               + CPN2 * T1 * TEMP(I,J)) *  DY4DY
     11               - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFOH
     12               + CPOH * T1 * TEMP(I,J)) *  DY5DY
      GV(4,5,I,J)  = - SDIFF * DY1DY
      GV(4,6,I,J)  = - SDIFF * DY2DY
      GV(4,7,I,J)  = - SDIFF * DY3DY
      GV(4,8,I,J)  = - SDIFF * DY4DY
C
C
C ---   STEP VERTICAL WALL CELL(S) EVALUATION I = NSX , J = 2 , NSYB
C
C
C                   ---------------
C ---               |  EAST FACE  |
```

```
C                ---------------
C
      I           = NSX
      DO 60  J    = 2 , NSYB
      AV          = .5 * (AREA(I,J) + AREA(I+1,J))
C
C ---   U-VELOCITY
C
      UE          = UVEL(I+1,J)
      UW          = UVEL(I,J)
      UN          = .25 * (UVEL(I+1,J) + UVEL(I+1,J+1)
     1                  +  UVEL(I,J+1) + UVEL(I,J))
      US          = .25 * (UVEL(I,J) + UVEL(I+1,J)
     1                  +  UVEL(I+1,J-1) + UVEL(I,J-1))
C
C ---   V-VELOCITY
C
      VE          = VVEL(I+1,J)
      VW          = VVEL(I,J)
      VN          = .25 * (VVEL(I+1,J) + VVEL(I+1,J+1)
     1                  +  VVEL(I,J+1) + VVEL(I+1,J))
      VS          = .25 * (VVEL(I,J)   + VVEL(I+1,J)
     1                  +  VVEL(I+1,J-1) + VVEL(I,J-1))
C
C ---   TEMPERATURE
C
      TE          = TEMP(I+1,J)
      TW          = TEMP(I,J)
      TN          = .25 * (TEMP(I+1,J)   + TEMP(I+1,J+1)
     1                  +  TEMP(I,J+1)   + TEMP(I+1,J))
      TS          = .25 * (TEMP(I,J)     + TEMP(I+1,J)
     1                  +  TEMP(I+1,J-1) + TEMP(I,J-1))
C
C ---   YH2
C
      Y1E          = YH2(I+1,J)
      Y1W          = YH2(I,J)
      Y1N          = .25 * (YH2(I+1,J)   + YH2(I+1,J+1)
     1                   +  YH2(I,J+1)   + YH2(I+1,J))
      Y1S          = .25 * (YH2(I,J)     + YH2(I+1,J)
     1                   +  YH2(I+1,J-1) + YH2(I,J-1))
C
C ---   YO2
C
      Y2E          = YO2(I+1,J)
      Y2W          = YO2(I,J)
      Y2N          = .25 * (YO2(I+1,J)   + YO2(I+1,J+1)
     1                   +  YO2(I,J+1)   + YO2(I+1,J))
      Y2S          = .25 * (YO2(I,J)     + YO2(I+1,J)
     1                   +  YO2(I+1,J-1) + YO2(I,J-1))
```

```
C
C ---   YH2O
C
      Y3E          = YH2O(I+1,J)
      Y3W          = YH2O(I,J)
      Y3N          = .25 * (YH2O(I+1,J)   + YH2O(I+1,J+1)
     1                    +  YH2O(I,J+1)   + YH2O(I+1,J))
      Y3S          = .25 * (YH2O(I,J)     + YH2O(I+1,J)
     1                    +  YH2O(I+1,J-1) + YH2O(I,J-1))
C
C ---   YYN2
C
      Y4E          = YYN2(I+1,J)
      Y4W          = YYN2(I,J)
      Y4N          = .25 * (YYN2(I+1,J)   + YYN2(I+1,J+1)
     1                    +  YYN2(I,J+1)   + YYN2(I+1,J))
      Y4S          = .25 * (YYN2(I,J)     + YYN2(I+1,J)
     1                    +  YYN2(I+1,J-1) + YYN2(I,J-1))
C
C ---   CALCULATE THE GRADIENT TERMS
C
C
C
C
C ---   X  GRADIENTS
C
      DUDX        = (UE * DYE(1,I,J) + UN * DYN(1,I,J)
     1             + UW * DYW(1,I,J) + US * DYS(1,I,J)) / AV
      DVDX        = (VE * DYE(1,I,J) + VN * DYN(1,I,J)
     1             + VW * DYW(1,I,J) + VS * DYS(1,I,J)) / AV
      DTDX        = (TE * DYE(1,I,J) + TN * DYN(1,I,J)
     1             + TW * DYW(1,I,J) + TS * DYS(1,I,J)) / AV
      DY1DX       = (Y1E * DYE(1,I,J) + Y1N * DYN(1,I,J)
     1             + Y1W * DYW(1,I,J) + Y1S * DYS(1,I,J)) / AV
      DY2DX       = (Y2E * DYE(1,I,J) + Y2N * DYN(1,I,J)
     1             + Y2W * DYW(1,I,J) + Y2S * DYS(1,I,J)) / AV
      DY3DX       = (Y3E * DYE(1,I,J) + Y3N * DYN(1,I,J)
     1             + Y3W * DYW(1,I,J) + Y3S * DYS(1,I,J)) / AV
      DY4DX       = (Y4E * DYE(1,I,J) + Y4N * DYN(1,I,J)
     1             + Y4W * DYW(1,I,J) + Y4S * DYS(1,I,J)) / AV
      DY5DX       = -DY1DX - DY2DX - DY3DX - DY4DX
C
C ---   Y  GRADIENTS
C
      DUDY        = - (UE * DXE(1,I,J) + UN * DXN(1,I,J)
     1               + UW * DXW(1,I,J) + US * DXS(1,I,J)) / AV
      DVDY        = - (VE * DXE(1,I,J) + VN * DXN(1,I,J)
     1               + VW * DXW(1,I,J) + VS * DXS(1,I,J)) / AV
      DTDY        = - (TE * DXE(1,I,J) + TN * DXN(1,I,J)
     1               + TW * DXW(1,I,J) + TS * DXS(1,I,J)) / AV
```

```
      DY1DY        = - (Y1E * DXE(1,I,J) + Y1N * DXN(1,I,J)
     1                 + Y1W * DXW(1,I,J) + Y1S * DXS(1,I,J)) / AV
      DY2DY        = - (Y2E * DXE(1,I,J) + Y2N * DXN(1,I,J)
     1                 + Y2W * DXW(1,I,J) + Y2S * DXS(1,I,J)) / AV
      DY3DY        = - (Y3E * DXE(1,I,J) + Y3N * DXN(1,I,J)
     1                 + Y3W * DXW(1,I,J) + Y3S * DXS(1,I,J)) / AV
      DY4DY        = - (Y4E * DXE(1,I,J) + Y4N * DXN(1,I,J)
     1                 + Y4W * DXW(1,I,J) + Y4S * DXS(1,I,J)) / AV
      DY5DY        = - DY1DY - DY2DY - DY3DY - DY4DY
C
C ---   COMPUTE THE FULL SHEAR STRESS TERMS
C
      TXX          = - LAMB * VIS(I,J) * (DUDX + DVDY)
                    - 2.0  * VIS(I,J) * DUDX / REN
      TXY          = - VIS(I,J) / REN  * (DUDY + DVDX)
      TYY          = - LAMB * VIS(I,J) * (DUDX + DVDY)
                    - 2.0  * VIS(I,J) * DVDY / REN
C
C ---   COMPUTE THE VISCOUS CONTRIBUTIONS TO "F" AND "G"
C
      FV(1,1,I,J)  = 0.0
      FV(1,2,I,J)  = TXX
      FV(1,3,I,J)  = TXY
      FV(1,4,I,J)  =   .5 * (UVEL(I,J) + UVEL(I,J+1))*TXX
     1                 + .5 * (VVEL(I,J) + VVEL(I,J+1))*TXY
     2                 - SDIFF * FACT1 * DTDX
     3                 - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2
     4                 + CPH2 * T1 * TEMP(I,J)) *  DY1DX
     5                 - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFO2
     6                 + CPO2 * T1 * TEMP(I,J)) *  DY2DX
     7                 - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2O
     8                 + CPH2O * T1 * TEMP(I,J)) *  DY3DX
     9                 - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2
     10                + CPN2 * T1 * TEMP(I,J)) *  DY4DX
     11                - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFOH
     12                + CPOH * T1 * TEMP(I,J)) *  DY5DX
      FV(1,5,I,J)  = - SDIFF * DY1DX
      FV(1,6,I,J)  = - SDIFF * DY2DX
      FV(1,7,I,J)  = - SDIFF * DY3DX
      FV(1,8,I,J)  = - SDIFF * DY4DX
C
      GV(1,1,I,J)  = 0.0
      GV(1,2,I,J)  = TXY
      GV(1,3,I,J)  = TYY
      GV(1,4,I,J)  =   .5 * (VVEL(I,J) + VVEL(I,J+1))*TYY
     1                 + .5 * (UVEL(I,J) + UVEL(I,J+1))*TXY
     2                 - SDIFF * FACT1 * DTDY
     3                 - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2
     4                 + CPH2 * T1 * TEMP(I,J)) *  DY1DY
     5                 - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFO2
```

```
     6                    + CPO2 * T1 * TEMP(I,J)) *  DY2DY
     7                    - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2O
     8                    + CPH2O * T1 * TEMP(I,J)) *  DY3DY
     9                    - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2
     10                   + CPN2 * T1 * TEMP(I,J)) *  DY4DY
     11                   - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFOH
     12                   + CPOH * T1 * TEMP(I,J)) *  DY5DY
       GV(1,5,I,J)  = - SDIFF * DY1DY
       GV(1,6,I,J)  = - SDIFF * DY2DY
       GV(1,7,I,J)  = - SDIFF * DY3DY
       GV(1,8,I,J)  = - SDIFF * DY4DY
60     CONTINUE
C
C                 ------------------
C ---             |   NORTH  FACE  |
C                 ------------------
C
       I          = NSX
       DO 70  J   = 2 , NSYB
       AV         = .5 * (AREA(I,J) + AREA(I,J+1))
C
C ---   U-VELOCITY
C
       UE         = .25 * (UVEL(I+1,J) + UVEL(I+1,J+1)
     1                +  UVEL(I,J+1) + UVEL(I,J))
       UN         = UVEL(I,J+1)
       UW         = 0.0
       US         = UVEL(I,J)
C
C ---   V-VELOCITY
C
       VE         = .25 * (VVEL(I+1,J) + VVEL(I+1,J+1)
     1                +  VVEL(I,J+1) + VVEL(I,J))
       VN         = VVEL(I,J+1)
       VW         = 0.0
       VS         = VVEL(I,J)
C
C ---   TEMPERATURE
C
       TE         = .25 * (TEMP(I+1,J) + TEMP(I+1,J+1)
     1                +  TEMP(I,J+1) + TEMP(I,J))
       TN         = TEMP(I,J+1)
       TW         = .5 * (TEMP(I,J) + TEMP(I,J+1))
       TS         = TEMP(I,J)
C
C ---   YH2
C
       Y1E        = .25 * (YH2(I+1,J) + YH2(I+1,J+1)
     1                +  YH2(I,J+1) + YH2(I,J))
       Y1N         = YH2(I,J+1)
```

```
      Y1W         = .5 * (YH2(I,J) + YH2(I,J+1))
      Y1S         = YH2(I,J)
C
C ---   YO2
C
      Y2E        = .25 * (YO2(I+1,J) + YO2(I+1,J+1)
     1                 +  YO2(I,J+1) + YO2(I,J))
      Y2N         = YO2(I,J+1)
      Y2W         = .5 * (YO2(I,J)   + YO2(I,J+1))
      Y2S         = YO2(I,J)
C
C ---   YH2O
C
      Y3E        = .25 * (YH2O(I+1,J) + YH2O(I+1,J+1)
     1                 +  YH2O(I,J+1) + YH2O(I,J))
      Y3N         = YH2O(I,J+1)
      Y3W         = .5 * (YH2O(I,J)   + YH2O(I,J+1))
      Y3S         = YH2O(I,J)
C
C ---   YYN2
C
      Y4E        = .25 * (YYN2(I+1,J) + YYN2(I+1,J+1)
     1                 +  YYN2(I,J+1) + YYN2(I,J))
      Y4N         = YYN2(I,J+1)
      Y4W         = .5 * (YYN2(I,J)   + YYN2(I,J+1))
      Y4S         = YYN2(I,J)
C
C ---   CALCULATE THE GRADIENT TERMS
C
C
C
C ---   X  GRADIENTS
C
      DUDX        = (UE * DYE(2,I,J) + UN * DYN(2,I,J)
     1             + UW * DYW(2,I,J) + US * DYS(2,I,J)) / AV
      DVDX        = (VE * DYE(2,I,J) + VN * DYN(2,I,J)
     1             + VW * DYW(2,I,J) + VS * DYS(2,I,J)) / AV
      DTDX        = (TE * DYE(2,I,J) + TN * DYN(2,I,J)
     1             + TW * DYW(2,I,J) + TS * DYS(2,I,J)) / AV
      DY1DX       = (Y1E * DYE(2,I,J) + Y1N * DYN(2,I,J)
     1             + Y1W * DYW(2,I,J) + Y1S * DYS(2,I,J)) / AV
      DY2DX       = (Y2E * DYE(2,I,J) + Y2N * DYN(2,I,J)
     1             + Y2W * DYW(2,I,J) + Y2S * DYS(2,I,J)) / AV
      DY3DX       = (Y3E * DYE(2,I,J) + Y3N * DYN(2,I,J)
     1             + Y3W * DYW(2,I,J) + Y3S * DYS(2,I,J)) / AV
      DY4DX       = (Y4E * DYE(2,I,J) + Y4N * DYN(2,I,J)
     1             + Y4W * DYW(2,I,J) + Y4S * DYS(2,I,J)) / AV
      DY5DX       = -DY1DX - DY2DX - DY3DX - DY4DX
C
C ---   Y  GRADIENTS
```

```
C
      DUDY        = - (UE * DXE(2,I,J) + UN * DXN(2,I,J)
     1                + UW * DXW(2,I,J) + US * DXS(2,I,J)) / AV
      DVDY        = - (VE * DXE(2,I,J) + VN * DXN(2,I,J)
     1                + VW * DXW(2,I,J) + VS * DXS(2,I,J)) / AV
      DTDY        = - (TE * DXE(2,I,J) + TN * DXN(2,I,J)
     1                + TW * DXW(2,I,J) + TS * DXS(2,I,J)) / AV
      DY1DY       = - (Y1E * DXE(2,I,J) + Y1N * DXN(2,I,J)
     1                + Y1W * DXW(2,I,J) + Y1S * DXS(2,I,J)) / AV
      DY2DY       = - (Y2E * DXE(2,I,J) + Y2N * DXN(2,I,J)
     1                + Y2W * DXW(2,I,J) + Y2S * DXS(2,I,J)) / AV
      DY3DY       = - (Y3E * DXE(2,I,J) + Y3N * DXN(2,I,J)
     1                + Y3W * DXW(2,I,J) + Y3S * DXS(2,I,J)) / AV
      DY4DY       = - (Y4E * DXE(2,I,J) + Y4N * DXN(2,I,J)
     1                + Y4W * DXW(2,I,J) + Y4S * DXS(2,I,J)) / AV
      DY5DY       = - DY1DY - DY2DY - DY3DY - DY4DY
C
C ---  COMPUTE THE FULL SHEAR STRESS TERMS
C
      TXX         = - LAMB * VIS(I,J) * (DUDX + DVDY)
                    - 2.0  * VIS(I,J) * DUDX / REN
      TXY         = - VIS(I,J) / REN  * (DUDY + DVDX)
      TYY         = - LAMB * VIS(I,J) * (DUDX + DVDY)
                    - 2.0  * VIS(I,J) * DVDY / REN
C
C ---  COMPUTE THE VISCOUS CONTRIBUTIONS TO "F" AND "G"
C
      FV(2,1,I,J)  = 0.0
      FV(2,3,I,J)  = TXX
      FV(2,3,I,J)  = TXY
      FV(2,4,I,J)  =   .5 * (UVEL(I,J) + UVEL(I,J+1))*TXX
     1                + .5 * (VVEL(I,J) + VVEL(I,J+1))*TXY
     2                - SDIFF * FACT1 * DTDX
     3                - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2
     4                + CPH2 * T1 * TEMP(I,J)) *  DY1DX
     5                - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFO2
     6                + CPO2 * T1 * TEMP(I,J)) *  DY2DX
     7                - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2O
     8                + CPH2O * T1 * TEMP(I,J)) *  DY3DX
     9                - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2
     10               + CPN2 * T1 * TEMP(I,J)) *  DY4DX
     11               - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFOH
     12               + CPOH * T1 * TEMP(I,J)) *  DY5DX
      FV(2,5,I,J)  = - SDIFF * DY1DX
      FV(2,6,I,J)  = - SDIFF * DY2DX
      FV(2,7,I,J)  = - SDIFF * DY3DX
      FV(2,8,I,J)  = - SDIFF * DY4DX
C
      GV(2,1,I,J)  = 0.0
      GV(2,2,I,J)  = TXY
```

```
      GV(2,3,I,J)  = TYY
      GV(2,4,I,J)  =   .5 * (VVEL(I,J) + VVEL(I,J+1))*TYY
     1                 + .5 * (UVEL(I,J) + UVEL(I,J+1))*TXY
     2                 - SDIFF * FACT1 * DTDY
     3                 - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2
     4                 + CPH2 * T1 * TEMP(I,J)) *  DY1DY
     5                 - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFO2
     6                 + CPO2 * T1 * TEMP(I,J)) *  DY2DY
     7                 - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2O
     8                 + CPH2O * T1 * TEMP(I,J)) *  DY3DY
     9                 - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2
     10                + CPN2 * T1 * TEMP(I,J)) *  DY4DY
     11                - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFOH
     12                + CPOH * T1 * TEMP(I,J)) *  DY5DY
      GV(2,5,I,J)  = - SDIFF * DY1DY
      GV(2,6,I,J)  = - SDIFF * DY2DY
      GV(2,7,I,J)  = - SDIFF * DY3DY
      GV(2,8,I,J)  = - SDIFF * DY4DY
70    CONTINUE
C
C
C                    ---------------
C ---                |  WEST FACE  |
C                    ---------------
C
      I           = NSX
      DO 80  J    = 2 , NSYB
      AV          = .5 * (AREA(I,J) + AREA(I-1,J))
C
C ---  U-VELOCITY
C
      UE          =  UVEL(I,J)
      UW          = -UVEL(I,J)
      UN          =  0.0
      US          =  0.0
C
C ---  V-VELOCITY
C
      VE          = VVEL(I,J)
      VW          = -VVEL(I,J)
      VN          = 0.0
      VS          = 0.0
C
C ---  TEMPERATURE
C
      TE          = TEMP(I,J)
      TW          = TEMP(I,J)
      TN          = .5 * (TEMP(I,J)   + TEMP(I,J+1))
      TS          = .5 * (TEMP(I,J)   + TEMP(I,J-1))
C
```

```
C ---   YH2
C
      Y1E          = YH2(I,J)
      Y1W          = YH2(I,J)
      Y1N          = .5 * (YH2(I,J)   + YH2(I,J+1))
      Y1S          = .5 * (YH2(I,J)   + YH2(I,J-1))
C
C ---   YO2
C
      Y2E          = YO2(I,J)
      Y2W          = YO2(I,J)
      Y2N          = .5 * (YO2(I,J)   + YO2(I,J+1))
      Y2S          = .5 * (YO2(I,J)   + YO2(I,J-1))
C
C ---   YH2O
C
      Y3E          = YH2O(I,J)
      Y3W          = YH2O(I,J)
      Y3N          = .5 * (YH2O(I,J)   + YH2O(I,J+1))
      Y3S          = .5 * (YH2O(I,J)   + YH2O(I,J-1))
C
C ---   YYN2
C
      Y4E          = YYN2(I,J)
      Y4W          = YYN2(I,J)
      Y4N          = .5 * (YYN2(I,J)   + YYN2(I,J+1))
      Y4S          = .5 * (YYN2(I,J)   + YYN2(I,J-1))
C
C ---   CALCULATE THE GRADIENT TERMS
C
C
C ---   X  GRADIENTS
C
      DUDX        = (UE * DYE(3,I,J) + UN * DYN(3,I,J)
     1             + UW * DYW(3,I,J) + US * DYS(3,I,J)) / AV
      DVDX        = (VE * DYE(3,I,J) + VN * DYN(3,I,J)
     1             + VW * DYW(3,I,J) + VS * DYS(3,I,J)) / AV
      DTDX        = (TE * DYE(3,I,J) + TN * DYN(3,I,J)
     1             + TW * DYW(3,I,J) + TS * DYS(3,I,J)) / AV
      DY1DX       = (Y1E * DYE(3,I,J) + Y1N * DYN(3,I,J)
     1             + Y1W * DYW(3,I,J) + Y1S * DYS(3,I,J)) / AV
      DY2DX       = (Y2E * DYE(3,I,J) + Y2N * DYN(3,I,J)
     1             + Y2W * DYW(3,I,J) + Y2S * DYS(3,I,J)) / AV
      DY3DX       = (Y3E * DYE(3,I,J) + Y3N * DYN(3,I,J)
     1             + Y3W * DYW(3,I,J) + Y3S * DYS(3,I,J)) / AV
      DY4DX       = (Y4E * DYE(3,I,J) + Y4N * DYN(3,I,J)
     1             + Y4W * DYW(3,I,J) + Y4S * DYS(3,I,J)) / AV
      DY5DX       = -DY1DX - DY2DX - DY3DX - DY4DX
C
C ---   Y  GRADIENTS
```

```
C
      DUDY        = - (UE * DXE(3,I,J) + UN * DXN(3,I,J)
     1                + UW * DXW(3,I,J) + US * DXS(3,I,J)) / AV
      DVDY        = - (VE * DXE(3,I,J) + VN * DXN(3,I,J)
     1                + VW * DXW(3,I,J) + VS * DXS(3,I,J)) / AV
      DTDY        = - (TE * DXE(3,I,J) + TN * DXN(3,I,J)
     1                + TW * DXW(3,I,J) + TS * DXS(3,I,J)) / AV
      DY1DY       = - (Y1E * DXE(3,I,J) + Y1N * DXN(3,I,J)
     1                + Y1W * DXW(3,I,J) + Y1S * DXS(3,I,J)) / AV
      DY2DY       = - (Y2E * DXE(3,I,J) + Y2N * DXN(3,I,J)
     1                + Y2W * DXW(3,I,J) + Y2S * DXS(3,I,J)) / AV
      DY3DY       = - (Y3E * DXE(3,I,J) + Y3N * DXN(3,I,J)
     1                + Y3W * DXW(3,I,J) + Y3S * DXS(3,I,J)) / AV
      DY4DY       = - (Y4E * DXE(3,I,J) + Y4N * DXN(3,I,J)
     1                + Y4W * DXW(3,I,J) + Y4S * DXS(3,I,J)) / AV
      DY5DY       = - DY1DY - DY2DY - DY3DY - DY4DY
C
C ---   COMPUTE THE FULL SHEAR STRESS TERMS
C
      TXX         = - LAMB * VIS(I,J) * (DUDX + DVDY)
                     - 2.0  * VIS(I,J) * DUDX / REN
      TXY         = - VIS(I,J) / REN  * (DUDY + DVDX)
      TYY         = - LAMB * VIS(I,J) * (DUDX + DVDY)
                     - 2.0  * VIS(I,J) * DVDY / REN
C
C ---   COMPUTE THE VISCOUS CONTRIBUTIONS TO "F" AND "G"
C
      FV(3,1,I,J)  = 0.0
      FV(3,2,I,J)  = TXX
      FV(3,3,I,J)  = TXY
      FV(3,4,I,J)  =   .5 * (UVEL(I,J) + UVEL(I+1,J))*TXX
     1                 + .5 * (VVEL(I,J) + VVEL(I+1,J))*TXY
     2                 - SDIFF * FACT1 * DTDX
     3                 - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2
     4                 + CPH2 * T1 * TEMP(I,J)) *  DY1DX
     5                 - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFO2
     6                 + CPO2 * T1 * TEMP(I,J)) *  DY2DX
     7                 - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2O
     8                 + CPH2O * T1 * TEMP(I,J)) *  DY3DX
     9                 - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2
     10                + CPN2 * T1 * TEMP(I,J)) *  DY4DX
     11                - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFOH
     12                + CPOH * T1 * TEMP(I,J)) *  DY5DX
      FV(3,5,I,J)  = - SDIFF * DY1DX
      FV(3,6,I,J)  = - SDIFF * DY2DX
      FV(3,7,I,J)  = - SDIFF * DY3DX
      FV(3,8,I,J)  = - SDIFF * DY4DX
C
      GV(3,1,I,J)  = 0.0
      GV(3,2,I,J)  = TXY
```

```
      GV(3,3,I,J)  = TYY
      GV(3,4,I,J)  =   .5 * (VVEL(I,J) + VVEL(I+1,J))*TYY
     1                 + .5 * (UVEL(I,J) + UVEL(I+1,J))*TXY
     2                 - SDIFF * FACT1 * DTDY
     3                 - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2
     4                 + CPH2 * T1 * TEMP(I,J)) *  DY1DY
     5                 - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFO2
     6                 + CPO2 * T1 * TEMP(I,J)) *  DY2DY
     7                 - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2O
     8                 + CPH2O * T1 * TEMP(I,J)) *  DY3DY
     9                 - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2
     10                + CPN2 * T1 * TEMP(I,J)) *  DY4DY
     11                - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFOH
     12                + CPOH * T1 * TEMP(I,J)) *  DY5DY
      GV(3,5,I,J)  = - SDIFF * DY1DY
      GV(3,6,I,J)  = - SDIFF * DY2DY
      GV(3,7,I,J)  = - SDIFF * DY3DY
      GV(3,8,I,J)  = - SDIFF * DY4DY
80    CONTINUE
C
C
C                   ----------------
C ---               | SOUTH FACE   |
C                   ----------------
C
      I            = NSX
      DO 90  J     = 2 , NSXB
      AV           = .5 * (AREA(I,J) + AREA(I,J-1))
C
C ---  U-VELOCITY
C
      UE           = .25 * (UVEL(I,J)     + UVEL(I+1,J)
     1                   +  UVEL(I+1,J-1) + UVEL(I,J-1))
      UN           = UVEL(I,J)
      UW           = 0.0
      US           = UVEL(I,J-1)
C
C ---  V-VELOCITY
C
      VE           = .25 * (VVEL(I,J)     + VVEL(I+1,J)
     1                   +  VVEL(I+1,J-1) + VVEL(I,J-1))
      VN           = VVEL(I,J)
      VW           = 0.0
      VS           = VVEL(I,J-1)
C
C ---  TEMPERATURE
C
      TE           = .25 * (TEMP(I,J)     + TEMP(I+1,J)
     1                   +  TEMP(I+1,J-1) + TEMP(I,J-1))
      TN           = TEMP(I,J)
```

```
      TW          = .5 * (TEMP(I,J) + TEMP(I,J-1))
      TS          = TEMP(I,J-1)
C
C ---   YH2
C
      Y1E          = .25 * (YH2(I,J)     + YH2(I+1,J)
     1                   +  YH2(I+1,J-1) + YH2(I,J-1))
      Y1N          = YH2(I,J)
      Y1W          = .5 * (YH2(I,J) + YH2(I,J-1))
      Y1S          = YH2(I,J-1)
C
C ---   YO2
C
      Y2E          = .25 * (YO2(I,J)     + YO2(I+1,J)
     1                   +  YO2(I+1,J-1) + YO2(I,J-1))
      Y2N          = YO2(I,J)
      Y2W          = .5 * (YO2(I,J) + YO2(I,J-1))
      Y2S          = YO2(I,J-1)
C
C ---   YH2O
C
      Y3E          = .25 * (YH2O(I,J)     + YH2O(I+1,J)
     1                   +  YH2O(I+1,J-1) + YH2O(I,J-1))
      Y3N          = YH2O(I,J)
      Y3W          = .5 * (YH2O(I,J) + YH2O(I,J-1))
      Y3S          = YH2O(I,J-1)
C
C ---   YYN2
C
      Y4E          = .25 * (YYN2(I,J)     + YYN2(I+1,J)
     1                   +  YYN2(I+1,J-1) + YYN2(I,J-1))
      Y4N          = YYN2(I,J)
      Y4W          = .5 * (YYN2(I,J) + YYN2(I,J-1))
      Y4S          = YYN2(I,J-1)
C
C ---   CALCULATE THE GRADIENT TERMS
C
C
C ---   X  GRADIENTS
C
      DUDX        = (UE * DYE(4,I,J) + UN * DYN(4,I,J)
     1             + UW * DYW(4,I,J) + US * DYS(4,I,J)) / AV
      DVDX        = (VE * DYE(4,I,J) + VN * DYN(4,I,J)
     1             + VW * DYW(4,I,J) + VS * DYS(4,I,J)) / AV
      DTDX        = (TE * DYE(4,I,J) + TN * DYN(4,I,J)
     1             + TW * DYW(4,I,J) + TS * DYS(4,I,J)) / AV
      DY1DX       = (Y1E * DYE(4,I,J) + Y1N * DYN(4,I,J)
     1             + Y1W * DYW(4,I,J) + Y1S * DYS(4,I,J)) / AV
      DY2DX       = (Y2E * DYE(4,I,J) + Y2N * DYN(4,I,J)
     1             + Y2W * DYW(4,I,J) + Y2S * DYS(4,I,J)) / AV
```

```
      DY3DX      = (Y3E * DYE(4,I,J) + Y3N * DYN(4,I,J)
     1            + Y3W * DYW(4,I,J) + Y3S * DYS(4,I,J)) / AV
      DY4DX      = (Y4E * DYE(4,I,J) + Y4N * DYN(4,I,J)
     1            + Y4W * DYW(4,I,J) + Y4S * DYS(4,I,J)) / AV
      DY5DX      = -DY1DX - DY2DX - DY3DX - DY4DX
C
C ---   Y  GRADIENTS
C
      DUDY       = - (UE * DXE(4,I,J) + UN * DXN(4,I,J)
     1              + UW * DXW(4,I,J) + US * DXS(4,I,J)) / AV
      DVDY       = - (VE * DXE(4,I,J) + VN * DXN(4,I,J)
     1              + VW * DXW(4,I,J) + VS * DXS(4,I,J)) / AV
      DTDY       = - (TE * DXE(4,I,J) + TN * DXN(4,I,J)
     1              + TW * DXW(4,I,J) + TS * DXS(4,I,J)) / AV
      DY1DY      = - (Y1E * DXE(4,I,J) + Y1N * DXN(4,I,J)
     1              + Y1W * DXW(4,I,J) + Y1S * DXS(4,I,J)) / AV
      DY2DY      = - (Y2E * DXE(4,I,J) + Y2N * DXN(4,I,J)
     1              + Y2W * DXW(4,I,J) + Y2S * DXS(4,I,J)) / AV
      DY3DY      = - (Y3E * DXE(4,I,J) + Y3N * DXN(4,I,J)
     1              + Y3W * DXW(4,I,J) + Y3S * DXS(4,I,J)) / AV
      DY4DY      = - (Y4E * DXE(4,I,J) + Y4N * DXN(4,I,J)
     1              + Y4W * DXW(4,I,J) + Y4S * DXS(4,I,J)) / AV
      DY5DY      = - DY1DY - DY2DY - DY3DY - DY4DY
C
C ---   COMPUTE THE FULL SHEAR STRESS TERMS
C
      TXX        = - LAMB * VIS(I,J) * (DUDX + DVDY)
                   - 2.0  * VIS(I,J) * DUDX / REN
      TXY        = - VIS(I,J) / REN  * (DUDY + DVDX)
      TYY        = - LAMB * VIS(I,J) * (DUDX + DVDY)
                   - 2.0  * VIS(I,J) * DVDY / REN
C
C ---   COMPUTE THE VISCOUS CONTRIBUTIONS TO "F" AND "G"
C
      FV(4,1,I,J)  = 0.0
      FV(4,2,I,J)  = TXX
      FV(4,3,I,J)  = TXY
      FV(4,4,I,J)  =   .5 * (UVEL(I,J) + UVEL(I+1,J))*TXX
     1                + .5 * (VVEL(I,J) + VVEL(I+1,J))*TXY
     2                - SDIFF * FACT1 * DTDX
     3                - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2
     4                + CPH2 * T1 * TEMP(I,J)) *  DY1DX
     5                - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFO2
     6                + CPO2 * T1 * TEMP(I,J)) *  DY2DX
     7                - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2O
     8                + CPH2O * T1 * TEMP(I,J)) *  DY3DX
     9                - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2
     10               + CPN2 * T1 * TEMP(I,J)) *  DY4DX
     11               - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFOH
     12               + CPOH * T1 * TEMP(I,J)) *  DY5DX
```

```
      FV(4,5,I,J)  = - SDIFF * DY1DX
      FV(4,6,I,J)  = - SDIFF * DY2DX
      FV(4,7,I,J)  = - SDIFF * DY3DX
      FV(4,8,I,J)  = - SDIFF * DY4DX
C
      GV(4,1,I,J)  = 0.0
      GV(4,2,I,J)  = TXY
      GV(4,3,I,J)  = TYY
      GV(4,4,I,J)  =   .5 * (VVEL(I,J) + VVEL(I+1,J))*TYY
     1                + .5 * (UVEL(I,J) + UVEL(I+1,J))*TXY
     2                - SDIFF * FACT1 * DTDY
     3                - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2
     4                + CPH2 * T1 * TEMP(I,J)) *  DY1DY
     5                - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFO2
     6                + CPO2 * T1 * TEMP(I,J)) *  DY2DY
     7                - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2O
     8                + CPH2O * T1 * TEMP(I,J)) *  DY3DY
     9                - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2
     10               + CPN2 * T1 * TEMP(I,J)) *  DY4DY
     11               - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFOH
     12               + CPOH * T1 * TEMP(I,J)) *  DY5DY
      GV(4,5,I,J)  = - SDIFF * DY1DY
      GV(4,6,I,J)  = - SDIFF * DY2DY
      GV(4,7,I,J)  = - SDIFF * DY3DY
      GV(4,8,I,J)  = - SDIFF * DY4DY
90    CONTINUE

C
C ---   LOWER STEP CORNER CELL EVALUATION I = NSX , J = 1
C
C
C                    ---------------
C ---                |  EAST FACE  |
C                    ---------------
C
      I           = NSX
      J           = 1
      AV          = .5 * (AREA(I,J) + AREA(I+1,J))
C
C ---   U-VELOCITY
C
      UE          = UVEL(I+1,J)
      UW          = UVEL(I,J)
      UN          = .25 * (UVEL(I+1,J) + UVEL(I+1,J+1)
     1                  +  UVEL(I,J+1) + UVEL(I,J))
      US          = 0.0
C
C ---   V-VELOCITY
C
      VE          = VVEL(I+1,J)
```

```
      VW          = VVEL(I,J)
      VN          = .25 * (VVEL(I+1,J) + VVEL(I+1,J+1)
     1                 +  VVEL(I,J+1) + VVEL(I+1,J))
      VS          = 0.0
C
C ---   TEMPERATURE
C
      TE          = TEMP(I+1,J)
      TW          = TEMP(I,J)
      TN          = .25 * (TEMP(I+1,J) + TEMP(I+1,J+1)
     1                 +  TEMP(I,J+1) + TEMP(I+1,J))
      TS          = .5 * (TEMP(I,J) + TEMP(I+1,J))
C
C ---   YH2
C
      Y1E          = YH2(I+1,J)
      Y1W          = YH2(I,J)
      Y1N          = .25 * (YH2(I+1,J) + YH2(I+1,J+1)
     1                  +  YH2(I,J+1) + YH2(I+1,J))
      Y1S          = .5 * (YH2(I,J) + YH2(I+1,J))
C
C ---   YO2
C
      Y2E          = YO2(I+1,J)
      Y2W          = YO2(I,J)
      Y2N          = .25 * (YO2(I+1,J) + YO2(I+1,J+1)
     1                  +  YO2(I,J+1) + YO2(I+1,J))
      Y2S          = .5 * (YO2(I,J) + YO2(I+1,J))
C
C ---   YH2O
C
      Y3E          = YH2O(I+1,J)
      Y3W          = YH2O(I,J)
      Y3N          = .25 * (YH2O(I+1,J) + YH2O(I+1,J+1)
     1                  +  YH2O(I,J+1) + YH2O(I+1,J))
      Y3S          = .5 * (YH2O(I,J) + YH2O(I+1,J))
C
C ---   YYN2
C
      Y4E          = YYN2(I+1,J)
      Y4W          = YYN2(I,J)
      Y4N          = .25 * (YYN2(I+1,J) + YYN2(I+1,J+1)
     1                  +  YYN2(I,J+1) + YYN2(I+1,J))
      Y4S          = .5 * (YYN2(I,J) + YYN2(I+1,J))
C
C ---   CALCULATE THE GRADIENT TERMS
C
C
C
C ---   X  GRADIENTS
```

```
C
      DUDX        = (UE * DYE(1,I,J) + UN * DYN(1,I,J)
     1             + UW * DYW(1,I,J) + US * DYS(1,I,J)) / AV
      DVDX        = (VE * DYE(1,I,J) + VN * DYN(1,I,J)
     1             + VW * DYW(1,I,J) + VS * DYS(1,I,J)) / AV
      DTDX        = (TE * DYE(1,I,J) + TN * DYN(1,I,J)
     1             + TW * DYW(1,I,J) + TS * DYS(1,I,J)) / AV
      DY1DX       = (Y1E * DYE(1,I,J) + Y1N * DYN(1,I,J)
     1             + Y1W * DYW(1,I,J) + Y1S * DYS(1,I,J)) / AV
      DY2DX       = (Y2E * DYE(1,I,J) + Y2N * DYN(1,I,J)
     1             + Y2W * DYW(1,I,J) + Y2S * DYS(1,I,J)) / AV
      DY3DX       = (Y3E * DYE(1,I,J) + Y3N * DYN(1,I,J)
     1             + Y3W * DYW(1,I,J) + Y3S * DYS(1,I,J)) / AV
      DY4DX       = (Y4E * DYE(1,I,J) + Y4N * DYN(1,I,J)
     1             + Y4W * DYW(1,I,J) + Y4S * DYS(1,I,J)) / AV
      DY5DX       = -DY1DX - DY2DX - DY3DX - DY4DX
C
C ---   Y  GRADIENTS
C
      DUDY        = - (UE * DXE(1,I,J) + UN * DXN(1,I,J)
     1               + UW * DXW(1,I,J) + US * DXS(1,I,J)) / AV
      DVDY        = - (VE * DXE(1,I,J) + VN * DXN(1,I,J)
     1               + VW * DXW(1,I,J) + VS * DXS(1,I,J)) / AV
      DTDY        = - (TE * DXE(1,I,J) + TN * DXN(1,I,J)
     1               + TW * DXW(1,I,J) + TS * DXS(1,I,J)) / AV
      DY1DY       = - (Y1E * DXE(1,I,J) + Y1N * DXN(1,I,J)
     1               + Y1W * DXW(1,I,J) + Y1S * DXS(1,I,J)) / AV
      DY2DY       = - (Y2E * DXE(1,I,J) + Y2N * DXN(1,I,J)
     1               + Y2W * DXW(1,I,J) + Y2S * DXS(1,I,J)) / AV
      DY3DY       = - (Y3E * DXE(1,I,J) + Y3N * DXN(1,I,J)
     1               + Y3W * DXW(1,I,J) + Y3S * DXS(1,I,J)) / AV
      DY4DY       = - (Y4E * DXE(1,I,J) + Y4N * DXN(1,I,J)
     1               + Y4W * DXW(1,I,J) + Y4S * DXS(1,I,J)) / AV
      DY5DY       = - DY1DY - DY2DY - DY3DY - DY4DY
C
C ---   COMPUTE THE FULL SHEAR STRESS TERMS
C
      TXX         = - LAMB * VIS(I,J) * (DUDX + DVDY)
                    - 2.0  * VIS(I,J) * DUDX / REN
      TXY         = - VIS(I,J) / REN  * (DUDY + DVDX)
      TYY         = - LAMB * VIS(I,J) * (DUDX + DVDY)
                    - 2.0  * VIS(I,J) * DVDY / REN
C
C ---   COMPUTE THE VISCOUS CONTRIBUTIONS TO "F" AND "G"
C
      FV(1,1,I,J)  = 0.0
      FV(1,2,I,J)  = TXX
      FV(1,3,I,J)  = TXY
      FV(1,4,I,J)  =   .5 * (UVEL(I,J) + UVEL(I,J+1))*TXX
     1                + .5 * (VVEL(I,J) + VVEL(I,J+1))*TXY
```

```
     2                   - SDIFF * FACT1 * DTDX
     3                   - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2
     4                   + CPH2 * T1 * TEMP(I,J)) *  DY1DX
     5                   - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFO2
     6                   + CPO2 * T1 * TEMP(I,J)) *  DY2DX
     7                   - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2O
     8                   + CPH2O * T1 * TEMP(I,J)) *  DY3DX
     9                   - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2
     10                  + CPN2 * T1 * TEMP(I,J)) *  DY4DX
     11                  - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFOH
     12                  + CPOH * T1 * TEMP(I,J)) *  DY5DX
      FV(1,5,I,J)  = - SDIFF * DY1DX
      FV(1,6,I,J)  = - SDIFF * DY2DX
      FV(1,7,I,J)  = - SDIFF * DY3DX
      FV(1,8,I,J)  = - SDIFF * DY4DX
C
      GV(1,1,I,J)  = 0.0
      GV(1,2,I,J)  = TXY
      GV(1,3,I,J)  = TYY
      GV(1,4,I,J)  =   .5 * (VVEL(I,J) + VVEL(I,J+1))*TYY
     1                   + .5 * (UVEL(I,J) + UVEL(I,J+1))*TXY
     2                   - SDIFF * FACT1 * DTDY
     3                   - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2
     4                   + CPH2 * T1 * TEMP(I,J)) *  DY1DY
     5                   - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFO2
     6                   + CPO2 * T1 * TEMP(I,J)) *  DY2DY
     7                   - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2O
     8                   + CPH2O * T1 * TEMP(I,J)) *  DY3DY
     9                   - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2
     10                  + CPN2 * T1 * TEMP(I,J)) *  DY4DY
     11                  - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFOH
     12                  + CPOH * T1 * TEMP(I,J)) *  DY5DY
      GV(1,5,I,J)  = - SDIFF * DY1DY
      GV(1,6,I,J)  = - SDIFF * DY2DY
      GV(1,7,I,J)  = - SDIFF * DY3DY
      GV(1,8,I,J)  = - SDIFF * DY4DY
C
C                 -----------------
C ---             !   NORTH  FACE  !
C                 -----------------
C
      I           = NSX
      J           = 1
      AV          = .5 * (AREA(I,J) + AREA(I,J+1))
C
C ---   U-VELOCITY
C
      UE          = .25 * (UVEL(I+1,J) + UVEL(I+1,J+1)
     1                 +  UVEL(I,J+1) + UVEL(I,J))
      UN          = UVEL(I,J+1)
```

```
      UW         = 0.0
      US         = UVEL(I,J)
C
C ---   V-VELOCITY
C
      VE         = .25 * (VVEL(I+1,J) + VVEL(I+1,J+1)
     1                 +  VVEL(I,J+1) + VVEL(I,J))
      VN         = VVEL(I,J+1)
      VW         = 0.0
      VS         = VVEL(I,J)
C
C ---   TEMPERATURE
C
      TE         = .25 * (TEMP(I+1,J) + TEMP(I+1,J+1)
     1                 +  TEMP(I,J+1) + TEMP(I,J))
      TN         = TEMP(I,J+1)
      TW         = .5 * (TEMP(I,J) + TEMP(I,J+1))
      TS         = TEMP(I,J)
C
C ---    YH2
C
      Y1E         = .25 * (YH2(I+1,J) + YH2(I+1,J+1)
     1                  +  YH2(I,J+1) + YH2(I,J))
      Y1N         = YH2(I,J+1)
      Y1W         = .5 * (YH2(I,J) + YH2(I,J+1))
      Y1S         = YH2(I,J)
C
C ---    YO2
C
      Y2E         = .25 * (YO2(I+1,J) + YO2(I+1,J+1)
     1                  +  YO2(I,J+1) + YO2(I,J))
      Y2N         = YO2(I,J+1)
      Y2W         = .5 * (YO2(I,J) + YO2(I,J+1))
      Y2S         = YO2(I,J)
C
C ---    YH2O
C
      Y3E         = .25 * (YH2O(I+1,J) + YH2O(I+1,J+1)
     1                  +  YH2O(I,J+1) + YH2O(I,J))
      Y3N         = YH2O(I,J+1)
      Y3W         = .5 * (YH2O(I,J) + YH2O(I,J+1))
      Y3S         = YH2O(I,J)
C
C ---    YYN2
C
      Y4E         = .25 * (YYN2(I+1,J) + YYN2(I+1,J+1)
     1                  +  YYN2(I,J+1) + YYN2(I,J))
      Y4N         = YYN2(I,J+1)
      Y4W         = .5 * (YYN2(I,J) + YYN2(I,J+1))
      Y4S         = YYN2(I,J)
```

```
C
C ---   CALCULATE THE GRADIENT TERMS
C
C
C
C ---   X  GRADIENTS
C
      DUDX       = (UE * DYE(2,I,J) + UN * DYN(2,I,J)
     1            + UW * DYW(2,I,J) + US * DYS(2,I,J)) / AV
      DVDX       = (VE * DYE(2,I,J) + VN * DYN(2,I,J)
     1            + VW * DYW(2,I,J) + VS * DYS(2,I,J)) / AV
      DTDX       = (TE * DYE(2,I,J) + TN * DYN(2,I,J)
     1            + TW * DYW(2,I,J) + TS * DYS(2,I,J)) / AV
      DY1DX      = (Y1E * DYE(2,I,J) + Y1N * DYN(2,I,J)
     1            + Y1W * DYW(2,I,J) + Y1S * DYS(2,I,J)) / AV
      DY2DX      = (Y2E * DYE(2,I,J) + Y2N * DYN(2,I,J)
     1            + Y2W * DYW(2,I,J) + Y2S * DYS(2,I,J)) / AV
      DY3DX      = (Y3E * DYE(2,I,J) + Y3N * DYN(2,I,J)
     1            + Y3W * DYW(2,I,J) + Y3S * DYS(2,I,J)) / AV
      DY4DX      = (Y4E * DYE(2,I,J) + Y4N * DYN(2,I,J)
     1            + Y4W * DYW(2,I,J) + Y4S * DYS(2,I,J)) / AV
      DY5DX      = -DY1DX - DY2DX - DY3DX - DY4DX
C
C ---   Y  GRADIENTS
C
      DUDY       = - (UE * DXE(2,I,J) + UN * DXN(2,I,J)
     1              + UW * DXW(2,I,J) + US * DXS(2,I,J)) / AV
      DVDY       = - (VE * DXE(2,I,J) + VN * DXN(2,I,J)
     1              + VW * DXW(2,I,J) + VS * DXS(2,I,J)) / AV
      DTDY       = - (TE * DXE(2,I,J) + TN * DXN(2,I,J)
     1              + TW * DXW(2,I,J) + TS * DXS(2,I,J)) / AV
      DY1DY      = - (Y1E * DXE(2,I,J) + Y1N * DXN(2,I,J)
     1              + Y1W * DXW(2,I,J) + Y1S * DXS(2,I,J)) / AV
      DY2DY      = - (Y2E * DXE(2,I,J) + Y2N * DXN(2,I,J)
     1              + Y2W * DXW(2,I,J) + Y2S * DXS(2,I,J)) / AV
      DY3DY      = - (Y3E * DXE(2,I,J) + Y3N * DXN(2,I,J)
     1              + Y3W * DXW(2,I,J) + Y3S * DXS(2,I,J)) / AV
      DY4DY      = - (Y4E * DXE(2,I,J) + Y4N * DXN(2,I,J)
     1              + Y4W * DXW(2,I,J) + Y4S * DXS(2,I,J)) / AV
      DY5DY      = - DY1DY - DY2DY - DY3DY - DY4DY
C
C ---   COMPUTE THE FULL SHEAR STRESS TERMS
C
      TXX        = - LAMB * VIS(I,J) * (DUDX + DVDY)
                  - 2.0  * VIS(I,J) * DUDX / REN
      TXY        = - VIS(I,J) / REN  * (DUDY + DVDX)
      TYY        = - LAMB * VIS(I,J) * (DUDX + DVDY)
                  - 2.0  * VIS(I,J) * DVDY / REN
C
C ---   COMPUTE THE VISCOUS CONTRIBUTIONS TO "F" AND "G"
```

```
C
      FV(2,1,I,J)  = 0.0
      FV(2,2,I,J)  = TXX
      FV(2,3,I,J)  = TXY
      FV(2,4,I,J)  =   .5 * (UVEL(I,J) + UVEL(I,J+1))*TXX
     1                + .5 * (VVEL(I,J) + VVEL(I,J+1))*TXY
     2                - SDIFF * FACT1 * DTDX
     3                - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2
     4                + CPH2 * T1 * TEMP(I,J)) *  DY1DX
     5                - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFO2
     6                + CPO2 * T1 * TEMP(I,J)) *  DY2DX
     7                - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2O
     8                + CPH2O * T1 * TEMP(I,J)) *  DY3DX
     9                - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2
     10               + CPN2 * T1 * TEMP(I,J)) *  DY4DX
     11               - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFOH
     12               + CPOH * T1 * TEMP(I,J)) *  DY5DX
      FV(2,5,I,J)  = - SDIFF * DY1DX
      FV(2,6,I,J)  = - SDIFF * DY2DX
      FV(2,7,I,J)  = - SDIFF * DY3DX
      FV(2,8,I,J)  = - SDIFF * DY4DX
C
      GV(2,1,I,J)  = 0.0
      GV(2,2,I,J)  = TXY
      GV(2,3,I,J)  = TYY
      GV(2,4,I,J)  =   .5 * (VVEL(I,J) + VVEL(I,J+1))*TYY
     1                + .5 * (UVEL(I,J) + UVEL(I,J+1))*TXY
     2                - SDIFF * FACT1 * DTDY
     3                - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2
     4                + CPH2 * T1 * TEMP(I,J)) *  DY1DY
     5                - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFO2
     6                + CPO2 * T1 * TEMP(I,J)) *  DY2DY
     7                - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2O
     8                + CPH2O * T1 * TEMP(I,J)) *  DY3DY
     9                - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2
     10               + CPN2 * T1 * TEMP(I,J)) *  DY4DY
     11               - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFOH
     12               + CPOH * T1 * TEMP(I,J)) *  DY5DY
      GV(2,5,I,J)  = - SDIFF * DY1DY
      GV(2,6,I,J)  = - SDIFF * DY2DY
      GV(2,7,I,J)  = - SDIFF * DY3DY
      GV(2,8,I,J)  = - SDIFF * DY4DY
C
C
C                   ---------------
C ---               | WEST FACE |
C                   ---------------
C
      I            = NSX
      J            = 1
```

```
      AV          = .5 * (AREA(I,J) + AREA(I-1,J))
C
C ---   U-VELOCITY
C
      UE          =  UVEL(I,J)
      UW          = -UVEL(I,J)
      UN          =  0.0
      US          =  0.0
C
C ---   V-VELOCITY
C
      VE          =  VVEL(I,J)
      VW          =  -VVEL(I,J)
      VN          =  0.0
      VS          =  0.0
C
C ---   TEMPERATURE
C
      TE          = TEMP(I,J)
      TW          = TEMP(I,J)
      TN          = .5 * (TEMP(I,J)  + TEMP(I,J+1))
      TS          = TEMP(I,J)
C
C ---   YH2
C
      Y1E          = YH2(I,J)
      Y1W          = YH2(I,J)
      Y1N          = .5 * (YH2(I,J)  + YH2(I,J+1))
      Y1S          = YH2(I,J)
C
C ---   YO2
C
      Y2E          = YO2(I,J)
      Y2W          = YO2(I,J)
      Y2N          = .5 * (YO2(I,J)  + YO2(I,J+1))
      Y2S          = YO2(I,J)
C
C ---   YH2O
C
      Y3E          = YH2O(I,J)
      Y3W          = YH2O(I,J)
      Y3N          = .5 * (YH2O(I,J)  + YH2O(I,J+1))
      Y3S          = YH2O(I,J)
C
C ---   YYN2
C
      Y4E          = YYN2(I,J)
      Y4W          = YYN2(I,J)
      Y4N          = .5 * (YYN2(I,J)  + YYN2(I,J+1))
      Y4S          = YYN2(I,J)
```

```
C
C ---   CALCULATE THE GRADIENT TERMS
C
C
C
C ---   X  GRADIENTS
C
      DUDX       = (UE * DYE(3,I,J) + UN * DYN(3,I,J)
     1             + UW * DYW(3,I,J) + US * DYS(3,I,J)) / AV
      DVDX       = (VE * DYE(3,I,J) + VN * DYN(3,I,J)
     1             + VW * DYW(3,I,J) + VS * DYS(3,I,J)) / AV
      DTDX       = (TE * DYE(3,I,J) + TN * DYN(3,I,J)
     1             + TW * DYW(3,I,J) + TS * DYS(3,I,J)) / AV
      DY1DX      = (Y1E * DYE(3,I,J) + Y1N * DYN(3,I,J)
     1             + Y1W * DYW(3,I,J) + Y1S * DYS(3,I,J)) / AV
      DY2DX      = (Y2E * DYE(3,I,J) + Y2N * DYN(3,I,J)
     1             + Y2W * DYW(3,I,J) + Y2S * DYS(3,I,J)) / AV
      DY3DX      = (Y3E * DYE(3,I,J) + Y3N * DYN(3,I,J)
     1             + Y3W * DYW(3,I,J) + Y3S * DYS(3,I,J)) / AV
      DY4DX      = (Y4E * DYE(3,I,J) + Y4N * DYN(3,I,J)
     1             + Y4W * DYW(3,I,J) + Y4S * DYS(3,I,J)) / AV
      DY5DX      = -DY1DX - DY2DX - DY3DX - DY4DX
C
C ---   Y  GRADIENTS
C
      DUDY       = - (UE * DXE(3,I,J) + UN * DXN(3,I,J)
     1               + UW * DXW(3,I,J) + US * DXS(3,I,J)) / AV
      DVDY       = - (VE * DXE(3,I,J) + VN * DXN(3,I,J)
     1               + VW * DXW(3,I,J) + VS * DXS(3,I,J)) / AV
      DTDY       = - (TE * DXE(3,I,J) + TN * DXN(3,I,J)
     1               + TW * DXW(3,I,J) + TS * DXS(3,I,J)) / AV
      DY1DY      = - (Y1E * DXE(3,I,J) + Y1N * DXN(3,I,J)
     1               + Y1W * DXW(3,I,J) + Y1S * DXS(3,I,J)) / AV
      DY2DY      = - (Y2E * DXE(3,I,J) + Y2N * DXN(3,I,J)
     1               + Y2W * DXW(3,I,J) + Y2S * DXS(3,I,J)) / AV
      DY3DY      = - (Y3E * DXE(3,I,J) + Y3N * DXN(3,I,J)
     1               + Y3W * DXW(3,I,J) + Y3S * DXS(3,I,J)) / AV
      DY4DY      = - (Y4E * DXE(3,I,J) + Y4N * DXN(3,I,J)
     1               + Y4W * DXW(3,I,J) + Y4S * DXS(3,I,J)) / AV
      DY5DY      = - DY1DY - DY2DY - DY3DY - DY4DY
C
C ---   COMPUTE THE FULL SHEAR STRESS TERMS
C
      TXX        = - LAMB * VIS(I,J) * (DUDX + DVDY)
                   - 2.0  * VIS(I,J) * DUDX / REN
      TXY        = - VIS(I,J) / REN  * (DUDY + DVDX)
      TYY        = - LAMB * VIS(I,J) * (DUDX + DVDY)
                   - 2.0  * VIS(I,J) * DVDY / REN
C
C ---   COMPUTE THE VISCOUS CONTRIBUTIONS TO "F" AND "G"
```

```
C
      FV(3,1,I,J)  = 0.0
      FV(3,2,I,J)  = TXX
      FV(3,3,I,J)  = TXY
      FV(3,4,I,J)  =   .5 * (UVEL(I,J) + UVEL(I,J+1))*TXX
     1                + .5 * (VVEL(I,J) + VVEL(I,J+1))*TXY
     2                - SDIFF * FACT1 * DTDX
     3                - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2
     4                + CPH2 * T1 * TEMP(I,J)) *  DY1DX
     5                - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFO2
     6                + CPO2 * T1 * TEMP(I,J)) *  DY2DX
     7                - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2O
     8                + CPH2O * T1 * TEMP(I,J)) *  DY3DX
     9                - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2
     10               + CPN2 * T1 * TEMP(I,J)) *  DY4DX
     11               - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFOH
     12               + CPOH * T1 * TEMP(I,J)) *  DY5DX
      FV(3,5,I,J)  = - SDIFF * DY1DX
      FV(3,6,I,J)  = - SDIFF * DY2DX
      FV(3,7,I,J)  = - SDIFF * DY3DX
      FV(3,8,I,J)  = - SDIFF * DY4DX
C
      GV(3,1,I,J)  = 0.0
      GV(3,2,I,J)  = TXY
      GV(3,3,I,J)  = TYY
      GV(3,4,I,J)  =   .5 * (VVEL(I,J) + VVEL(I,J+1))*TYY
     1                + .5 * (UVEL(I,J) + UVEL(I,J+1))*TXY
     2                - SDIFF * FACT1 * DTDY
     3                - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2
     4                + CPH2 * T1 * TEMP(I,J)) *  DY1DY
     5                - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFO2
     6                + CPO2 * T1 * TEMP(I,J)) *  DY2DY
     7                - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2O
     8                + CPH2O * T1 * TEMP(I,J)) *  DY3DY
     9                - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2
     10               + CPN2 * T1 * TEMP(I,J)) *  DY4DY
     11               - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFOH
     12               + CPOH * T1 * TEMP(I,J)) *  DY5DY
      GV(3,5,I,J)  = - SDIFF * DY1DY
      GV(3,6,I,J)  = - SDIFF * DY2DY
      GV(3,7,I,J)  = - SDIFF * DY3DY
      GV(3,8,I,J)  = - SDIFF * DY4DY
C
C
C             ______________
C             | SOUTH FACE |
C             --------------
C
C     I = NSX
      J = 1
```

```
C
      AV          = AREA(I,J)
C
C ---   U-VELOCITY
C
      UE          = 0.0
      UN          = UVEL(I,J)
      UW          = 0.0
      US          = -UVEL(I,J)
C
C ---   V-VELOCITY
C
      VE          =  0.0
      VN          =  VVEL(I,J)
      VW          =  0.0
      VS          = -VVEL(I,J)
C
C ---   TEMPERATURE
C
      TE          = .5 * (TEMP(I+1,J)  + TEMP(I,J))
      TN          = TEMP(I,J)
      TW          = TEMP(I,J)
      TS          = TEMP(I,J)
C
C ---   YH2
C
      Y1E          = .5 * (YH2(I+1,J)  + YH2(I,J))
      Y1N          = YH2(I,J)
      Y1W          = YH2(I,J)
      Y1S          = YH2(I,J)
C
C ---   YO2
C
      Y2E          = .5 * (YO2(I+1,J)  + YO2(I,J))
      Y2N          = YO2(I,J)
      Y2W          = YO2(I,J)
      Y2S          = YO2(I,J)
C
C ---   YH2O
C
      Y3E          = .5 * (YH2O(I+1,J)  + YH2O(I,J))
      Y3N          = YH2O(I,J)
      Y3W          = YH2O(I,J)
      Y3S          = YH2O(I,J)
C
C ---   YYN2
C
      Y4E          = .5 * (YYN2(I+1,J)  + YYN2(I,J))
      Y4N          = YYN2(I,J)
      Y4W          = YYN2(I,J)
```

```
      Y4S          = YYN2(I,J)
C
C ---   CALCULATE THE GRADIENT TERMS
C
C
C
C ---   X  GRADIENTS
C
      DUDX        = (UE * DYE(4,I,J) + UN * DYN(4,I,J)
     1             + UW * DYW(4,I,J) + US * DYS(4,I,J)) / AV
      DVDX        = (VE * DYE(4,I,J) + VN * DYN(4,I,J)
     1             + VW * DYW(4,I,J) + VS * DYS(4,I,J)) / AV
      DTDX        = (TE * DYE(4,I,J) + TN * DYN(4,I,J)
     1             + TW * DYW(4,I,J) + TS * DYS(4,I,J)) / AV
      DY1DX       = (Y1E * DYE(4,I,J) + Y1N * DYN(4,I,J)
     1             + Y1W * DYW(4,I,J) + Y1S * DYS(4,I,J)) / AV
      DY2DX       = (Y2E * DYE(4,I,J) + Y2N * DYN(4,I,J)
     1             + Y2W * DYW(4,I,J) + Y2S * DYS(4,I,J)) / AV
      DY3DX       = (Y3E * DYE(4,I,J) + Y3N * DYN(4,I,J)
     1             + Y3W * DYW(4,I,J) + Y3S * DYS(4,I,J)) / AV
      DY4DX       = (Y4E * DYE(4,I,J) + Y4N * DYN(4,I,J)
     1             + Y4W * DYW(4,I,J) + Y4S * DYS(4,I,J)) / AV
      DY5DX       = -DY1DX - DY2DX - DY3DX - DY4DX
C
C ---   Y  GRADIENTS
C
      DUDY        = - (UE * DXE(4,I,J) + UN * DXN(4,I,J)
     1               + UW * DXW(4,I,J) + US * DXS(4,I,J)) / AV
      DVDY        = - (VE * DXE(4,I,J) + VN * DXN(4,I,J)
     1               + VW * DXW(4,I,J) + VS * DXS(4,I,J)) / AV
      DTDY        = - (TE * DXE(4,I,J) + TN * DXN(4,I,J)
     1               + TW * DXW(4,I,J) + TS * DXS(4,I,J)) / AV
      DY1DY       = - (Y1E * DXE(4,I,J) + Y1N * DXN(4,I,J)
     1               + Y1W * DXW(4,I,J) + Y1S * DXS(4,I,J)) / AV
      DY2DY       = - (Y2E * DXE(4,I,J) + Y2N * DXN(4,I,J)
     1               + Y2W * DXW(4,I,J) + Y2S * DXS(4,I,J)) / AV
      DY3DY       = - (Y3E * DXE(4,I,J) + Y3N * DXN(4,I,J)
     1               + Y3W * DXW(4,I,J) + Y3S * DXS(4,I,J)) / AV
      DY4DY       = - (Y4E * DXE(4,I,J) + Y4N * DXN(4,I,J)
     1               + Y4W * DXW(4,I,J) + Y4S * DXS(4,I,J)) / AV
      DY5DY       = - DY1DY - DY2DY - DY3DY - DY4DY
C
C ---   COMPUTE THE FULL SHEAR STRESS TERMS
C
      TXX         = - LAMB * VIS(I,J) * (DUDX + DVDY)
                    - 2.0  * VIS(I,J) * DUDX / REN
      TXY         = - VIS(I,J) / REN  * (DUDY + DVDX)
      TYY         = - LAMB * VIS(I,J) * (DUDX + DVDY)
                    - 2.0  * VIS(I,J) * DVDY / REN
C
```

```
C ---   COMPUTE THE VISCOUS CONTRIBUTIONS TO "F" AND "G"
C
        FV(4,1,I,J)  = 0.0
        FV(4,2,I,J)  = TXX
        FV(4,3,I,J)  = TXY
        FV(4,4,I,J)  =   - SDIFF * FACT1 * DTDX
     3                   - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2
     4                   + CPH2 * T1 * TEMP(I,J)) *  DY1DX
     5                   - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFO2
     6                   + CPO2 * T1 * TEMP(I,J)) *  DY2DX
     7                   - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2O
     8                   + CPH2O * T1 * TEMP(I,J)) *  DY3DX
     9                   - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2
     10                  + CPN2 * T1 * TEMP(I,J)) *  DY4DX
     11                  - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFOH
     12                  + CPOH * T1 * TEMP(I,J)) *  DY5DX
        FV(4,5,I,J)  = - SDIFF * DY1DX
        FV(4,6,I,J)  = - SDIFF * DY2DX
        FV(4,7,I,J)  = - SDIFF * DY3DX
        FV(4,8,I,J)  = - SDIFF * DY4DX
C
        GV(4,1,I,J)  = 0.0
        GV(4,2,I,J)  = TXY
        GV(4,3,I,J)  = TYY
        GV(4,4,I,J)  =   - SDIFF * FACT1 * DTDY
     3                   - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2
     4                   + CPH2 * T1 * TEMP(I,J)) *  DY1DY
     5                   - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFO2
     6                   + CPO2 * T1 * TEMP(I,J)) *  DY2DY
     7                   - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2O
     8                   + CPH2O * T1 * TEMP(I,J)) *  DY3DY
     9                   - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFH2
     10                  + CPN2 * T1 * TEMP(I,J)) *  DY4DY
     11                  - SDIFF/VELO1**2 * 1.0 / CPND(I,J) * (DFOH
     12                  + CPOH * T1 * TEMP(I,J)) *  DY5DY
        GV(4,5,I,J)  = - SDIFF * DY1DY
        GV(4,6,I,J)  = - SDIFF * DY2DY
        GV(4,7,I,J)  = - SDIFF * DY3DY
        GV(4,8,I,J)  = - SDIFF * DY4DY
        RETURN
        END
```

References

[1] Beach, L.
Aeropropulsion 1979.
In *NASA Conference Publication 2092.* 1979.

[2] Blottner, F.G.
Finite Difference Methods Of Solution Of The Boundary Layer Equations.
AIAA Journal 8, 1970.

[3] Brandt. A.
Multi-Level Adaptive Computations In Fluid Dynamics.
In *AIAA 79-1455.* 1979.

[4] Bussing, T.R.A., Murman, E.M.,
A Finite Volume Method For The Calculation Of Compressible Chemically Reacting Flows.
In *AIAA 85-0331.* January, 1985.

[5] Bussing,T.R.A.,Murman,E.M.
A One-Dimensional Unsteady Model Of Dual Mode Scramjet Operation.
In *AIAA 83-0422.* January, 1983.

[6] Bussing, T.R.A., Murman, E.M.
Numerical Investigation Of 2-Dimensional H2-Air Flame Holding Over Ramps And Rearward Facing Steps.
In *AIAA 85-1250.* July, 1985.

[7] Candel, S.M., Darabiha, N., Esposito, E.
Models For A Turbulent, Premixed Dump Combustor.
In *AIAA Paper 82-1261.* June, 1982.

[8] Carter,J.E.
Numerical Solution Of The Navier-Stokes Equations For The Supersonic Laminar Flow Over A Two Dimensional Compression Corner.
In *NASA TR R-385.* July, 1972.

[9] Chima. R.V.
Analysis Of Inviscid And Viscous Flows In Cascades With An Explicit Multiple-Grid Algorithm.
In *AIAA 84-1663.* June, 1984.

[10] Curtiss,C.F.,Hirschfelder,J.O.
Integration Of Stiff Equations.
Proc. Natl. Acad. Sci. 38, 1952.

[11] Dannenhoffer, J.F., Baron, J.R.,
Grid Adapation For The 2-D Euler Equations.
In *AIAA 85-0484.* January, 1985.

[12] Dannenhoffer, J.F., Baron, J.R.,
Adaptive Procedure For Steady State Solution FOf Hyperbolic Equations.
In *AIAA P-84-0005.* January 9-12, 1970.

[13] Drummond, J.P, Weidner, E.H.
A Numerical Study Of Candidate Transverse Fuel Injector Configurations In The Langley Scramjet Engine.
In *17th JANNAF Combustion Meeting.* September, 1980.

[14] Drummond, J.P.
Numerical Investigation Of The Perpendicular Injector Flow Field In A Hydrogen Fueled Scramjet.
In *AIAA Paper 79-1482.* July, 1979.

[15] Drummond, J.P.,Hussaini,M.Y.,Zang,T.A.
Spectral Methods For Modeling Supersonic Chemically Reacting Flow.
In *AIAA 85-0302.* January, 1985.

[16] Drummond , P.
.
July, 1984.
Personal Communication.

[17] Eriksson, L.E., Rizzi ,A.
Analysis By Computer Of The Convergence To Steady State Of Discrete Approximations To The Euler Equations.
In *AIAA CFD Conf AIAA-1951.* 1983.

[18] Ferri, A.
Review Of Problems In Aplications Of Supersonic Combustion.
Journal Of The Royal Aeronautical Society , September, 1964.

[19] Gear,C.W.
Numerical Initial Value Problems In Ordinary Differential Equations.
Prentice-Hall, Inc, 1971.

[20] Hindmarsh,A.C.
Ordinary Differential Equation Solver.
In *UCID-30001 Lawrence Livermore Laboratory.* 1974.

[21] Jakubowski, A.K.,Lewis, C.H.
Experimental Study Of Supersonic Laminar Base Flow With And Without Suction.
AIAA Journal 12, December, 1973.

[22] Jameson, A., Schmidt, W., Turkel, E.
Numerical Solution Of The Euler Equations By Finite Volume Methods Using Runga-Kutta Time-Steping Scheme.
In *AIAA 81-1259.* February, 1978.

[23] Johnson, G.M.
Multiple-Grid Acceleration of Lax-Wendroff Algorithms.
In *NASA TM-82843.* 1982.

[24] Jones,R.A., Huber,P.W.
Toward Scramjet Aircraft.
In *Aeronautics And Astronautics.* February, 1978.

[25] Kee, R.J., Dwyer, H.A.
Review Of Stiffness And Implicit Finite-Difference Methods In Combustion Modeling.
In *Progress in Astronautics and Aeronautics.* August, 1979.

[26] Kumar, A.
Three Dimensional Inviscid Analysis Of The Scramjet Inlet Flow Field.
In *AIAA Paper 82-0060.* January, 1982.

[27] Kumar,A.
Numerical Analysis Of The Scramjet-Inlet Flow Field By Using The Three-Dimensional Navier-Stokes Equations.
In *JANNAF Propulsion Meeting.* Febuary, 1983.

[28] Lapidus,L.
Numerical Solution Of Stiff Ordinary Differential Equations.
In *Dept Of Chemical Engineering Princeton University.* January, 1977.

[29] .
GIM Code.
Technical Report D867354, LockHeed-Huntsville Research & Engineering Center, 1985.

[30] Lomax, H., Bailey, H.E.
A Critical Analysis Of Various Numerical Integration Methods For ComputingTthe Flow Of A Gas In Chemical Nonequilibrium.
In *NASA Technical Report TN D-4109.* August, 1967.

[31] Lomax, H.
Stable Implict And Explicit Numerical Methods For Integrating Quasi-Linear Differential Equations With Parasitic-Stiff And Parasitic-Saddle Eigenvalues.
In *NASA Technical Report TN D-4703.* July, 1968.

[32] MacCormack, R.W.
The Effects Of Viscosity In Hypervelocity Impact Createring.
In *AIAA Paper 60-345.* May, 1969.

[33] Murman, E.M.
.
Febuary, 1983.
MIT 16.026 Class Notes.

[34] McBride,B.J.,Heimel,S.,Ehlers,J.G.,Gorden,S.
Thermodynamic Properties To 6000' For 210 Substances Involving The First 18 Elements.
In *NASA SP-3001.* 1963.

[35] Ni, R.
A Multiple-Grid Scheme For Solving The Euler Equations.
AIAA Journal (20), Nov, 1982.

[36] Otey, G.R., Dwyer, H.A.
Numerical Study Of The Interaction Of Fast Chemistry And Diffusion.
AIAA Journal (17), June, 1983.

[37] Park,C.
On Convergence Of Computation Of Chemically Reacting Flows.
In *AIAA 85-0247*. January, 1985.

[38] Peyret, R, Taylor, T.D.
Computational Methods For Fluid Flow.
Springer-Verlag, 1983.

[39] Powell, K.G.
The Effect Of Artificial Viscosity Models On Conically Self-Similar Solutions To The Euler Equations.
Master's thesis, MIT - Dept Aero & Astro, August, 1984.

[40] Pratt, D.T.
CREK-1D: A Computer Code For Transient, Gas-Phase Combustion Kinetics.
In *The Western States Section, The Combustion Institue.* April, 1983.

[41] Jameson, A.
Solution Of The Euler Equations For Two Dimensional Transonic Flow By A Multigrid Method.
In *Princeton University MAE Report No. 1613.* June, 1983.

[42] Rizzi,A.W.,Bailey,H.E.
Split Space-Marching Finite-Volume Method For Chemically Reacting Supersonic Flow.
AIAA Journal 14, 1976.

[43] Rogers, R.C.
A Model Of Transverse Fuel Injection Applied To The Computation Of Supersonic Combustion Flow.
In *AIAA Paper 79-0359.* January, 1979.

[44] Rogers, R.C., Chintz, W.
On The Use Of A Global Hyrodgen-Air Combustion Model In The Calculation Of Turbulent Reacting Flows.
In *AIAA Paper 82-0112.* 1982.

[45] Scott,J.N.,Hankey,W.L.
Numerical Simulation Of Cold Flow in An Axisymmetric Centerbody Combustor.
In *AIAA 83-1741.* July, 1983.

[46] Seinfeld, J.H., Lapidus, L., Hwang, M.
Review Of Numerical Integration Techniques For Ordinary Differential Equations.
In *Ind. Eng. Chem. Fundam..* 1970.

[47] Shapiro, A.H.
Compressible Fluid Flow.
The Ronald Press Company, 1953.

[48] Markatos,N.C.,Spalding,D.B.,Tatchell,D.G.
Combustion Of Hydrogen Injected Into A Supersonic Airstream (The Ship Computer Program).
In *NASA CR-2802.* April, 1977.

[49] Sindir, M.M, Harsha, P.T.
Assesment Of Turbulence Models For Scramjet Flowfields.
In *NASA Contractor Report 3643.* February, 1979.

[50] Smoot,L.D., Hecker,W.C., Williams,G.A.
Prediction Of Propagating Methane-Air Flames.
Combustion And Flame 26, 1976.

[51] Stiles,R.J.
Analysis Of Steady, Two-Dimensional, Chemically Reacting, Nonequilibrium, Inviscid Flow In Nozzles.
AIAA Journal 23, 1984.

[52] Swanson, R.C.,Turkel, E.
A Multistage Scheme For The Navier Stokes Equations.
In *AIAA 85-0035.* January, 1985.

[53] Thomas,P.D.,Wilson,K.H.
Efficient Computation of Stiff Chemically Reacting Flow In Turbulent Free Jets.
AIAA Journal 14, 1976.

[54] Tong, S.S.
Proceedures For Accurate Inviscid Flow Simulations And Profile Refinement Of Turbomachinary Cascades.
PhD thesis, MIT - Dept Aero & Astro, January, 1984.

[55] Usab,W.J., Murman, E.M.
Embedded Mesh Solution Of The Euler Equations Using a Multiple-Grid Method.
In *AIAA 83-1946-CP.* July, 1970.

[56] Warming, R.F., Beam, R.M.
On The Construction Of Implicit Factored Schemes For Conservation Laws.
In *SIAM-AMS Proceedings-Sympsium On Computational Flow Dynamics,II.* 1977.

[57] Weider, E.H., Drummond, J.P.
A Parametic Study Of Staged Fuel Injector Configurations For Scramjet Applications.
In *AIAA Paper 81-1468.* July, 1981.

[58] Williams, F.A.
Combustion Theory.
Addison Wesley, 1965.