

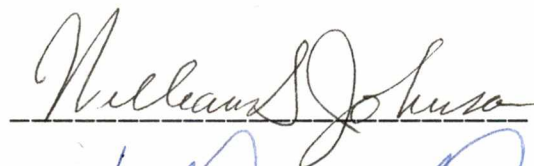
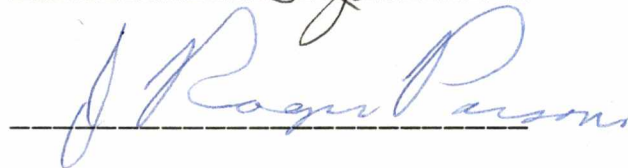
To the Graduate Council:

I am submitting herewith a thesis written by John P. Raulston Jr. entitled "A Numerical Study of Thermally-Driven Convective Wave Phenomenon". I have examined the final copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Mechanical Engineering.



M. Parang, Major Professor

We have read this thesis
and recommend its acceptance:

Accepted for the Council



Vice Chancellor and Dean of
The Graduate School

STATEMENT OF PERMISSION TO USE

In Presenting this thesis in partial fulfillment of the requirements for a Master's of Science degree at The University of Tennessee, Knoxville, I agree that the Library shall make it available to borrowers under rules of the Library. Brief quotations from this thesis are allowable without special permission, provided accurate acknowledgment of the source is made.

Permission for extensive quotation from or reproduction of this thesis may be granted by my major professor, or in his absence, by the Head of Interlibrary Services when, in the opinion of either, the proposed use of the material is for scholarly purposes. Any copying or use of the material in this thesis financial gain shall not be allowed without my written permission.

Signature John P. Raulston Jr.

Date November 2, 1990

**A NUMERICAL STUDY OF
THERMALLY-DRIVEN CONVECTIVE
WAVE PHENOMENON**

**A Thesis
Presented for the
Master of Science
Degree
The University of Tennessee, Knoxville**

**John P. Raulston Jr.
December 1990**

ACKNOWLEDGMENTS

The author wishes to express his thanks to NASA for the financial support provided for this work under contract number 958343. The assistance provided by Jet Propulsion Laboratory is also gratefully acknowledged.

The author also wishes to express his thanks to Dr. Masood Parang, advisor and chairman of his thesis committee, whose advice and encouragement have been invaluable in the completion of this work. Recognition is also due Dr. J. R. Parsons and Dr. W. S. Johnson who served as members of the author's thesis committee and have provided valuable support in the completion of the final text.

My sincerest gratitude is also due my wife Cindy for her patience and understanding during the completion of this work.

ABSTRACT

The subject of this study is the numerical modeling of a thermoacoustic convection (TAC) process. This investigation is carried out in two parts : (a) a review of earlier numerical models and the accuracy of their results, and (b) the development of a new numerical scheme for the solution of one and two-dimensional TAC processes.

The problem of determining the transient flow velocity, temperature, pressure, and density in a fluid confined between two large parallel plates with specified boundary and initial conditions (e.g., constant temperature at the walls, initially isothermal) has been addressed by earlier numerical studies. These previous models predicted very rapid transient behavior with temperature fields rapidly approaching steady state. The first part of the present investigation shows that these results may be in error due to selection of improperly large grid sizes. The rapid transient behavior reported in these earlier studies is shown to behave in a more physically reasonable way when a systematic grid refinement study is carried out and a more appropriate mesh size is selected.

A numerical model based on the SIMPLER algorithm, modified for flow compressibility , was developed for the solution of one and two-dimensional TAC processes. This model is shown to predict a much slower transient process than those predicted by other studies. That is, although the resulting thermally-induced fluid

motion is shown to enhance heat transfer above that predicted by a rigid conduction model, the present model shows significantly lower transient temperature solutions and smaller velocity amplitudes than seen in previous numerical models. Also, the solution of the one-dimensional TAC problem is shown to agree reasonably well with a solution developed by perturbation methods. The results of the one and two-dimensional models are also compared with available experimental data and discrepancies are discussed.

TABLE OF CONTENTS

CHAPTER	PAGE
1. INTRODUCTION.....	1
1.1 BACKGROUND.....	1
1.2 LITERATURE REVIEW.....	3
1.3 RESEARCH OBJECTIVES.....	15
2. ANALYSIS.....	18
2.1 MATHEMATICAL FORMULATION OF THE PROBLEM.....	18
2.2 PREVIOUS NUMERICAL SCHEMES.....	24
2.3 MODIFICATIONS:THE IMPLICIT SCHEME.....	34
3. NUMERICAL ANALYSIS AND DEVELOPMENT OF A TAC MODEL.....	45
3.1 FORMULATION OF THE FINITE-DIFFERENCE EQUATIONS.....	45
3.2 METHOD OF SOLUTION.....	56
3.3 ERROR ANALYSIS.....	60
4. RESULTS AND DISCUSSION.....	75
4.1 ONE-DIMENSIONAL RESULTS.....	75
4.2 TWO-DIMENSIONAL RESULTS.....	92
5. CONCLUSIONS	103
REFERENCES.....	106
APPENDICES.....	110

A. NORMALIZED AND DISCRETIZED EQUATIONS FOR THE TWO DIMENSIONAL MODEL.....	111
B. MODIPL.FOR COMPUTER CODE.....	118
C. TAC1D.FOR COMPUTER CODE.....	125
D. TAC2D.FOR COMPUTER CODE.....	139
E. CHART OF VARIOUS SPACE AND TIME INCREMENTS USED IN THE PROGRAMS MODIPL.FOR; TAC1D.FOR; TAC2D.FOR.....	200
VITA.....	204

LIST OF FIGURES

FIGURE	PAGE
1.1 Sound Production from a Soundhauss Tube.....	4
1.2 Sound Production from a Rijke Tube.....	5
1.3 Parallel Plate Model.....	9
1.4 Concentric Cylinder Model.....	10
1.5 Salah-Eddine's Experimental Apparatus.....	14
2.1 Continuity Equation in Discretized Form.....	26
2.2 Newton's Second Law in Discretized Form.....	27
2.3 Energy Equation in Discretized Form.....	28
2.4 Nodal System Sketch.....	29
2.5 Non-Dimensional Midpoint Velocity Versus Dimensionless Time from Ref [8].....	30
2.6 Temperature Profiles at 0.2 Seconds Ref. [8].....	31
2.7 Temperature Profiles at 0.1 Seconds.....	33
2.8 Midpoint Velocity Versus Time	37
2.9 Temperature Profiles at 0.1 Seconds ($\Delta t=0.001875$).....	38
2.10 Temperature Profiles at 0.1 Seconds ($\Delta t=0.015$).....	40
2.11 Temperature Profiles at 0.1 Seconds ($\Delta t=0.005$).....	41

2.12	Temperature Profiles at 0.1 Seconds ($\Delta x=1/44$).....	42
3.1	Nodal Control Volume Diagram.....	46
3.2	Continuity Equation in Discretized Form.....	51
3.3	Newton's Second Law in Discretized Form.....	53
3.4	Energy Equation in Discretized Form.....	54
3.5	Flow Diagram for the TAC Numerical Procedure.....	59
3.6	Unsteady Velocity at Midpoint for Various Time Steps ($\Delta x=1/88$).....	62
3.7	Unsteady Velocity at Midpoint for Various Mesh Sizes ($\Delta t=0.015$).....	64
3.8	Unsteady Velocity at Midpoint for Various Values of Convergence Criterion ($\Delta x=1/88; \Delta t=0.015$).....	65
3.9	Comparison of Numerical Results with Experimental Results.....	68
3.10	Unsteady Velocity at the Center for Various Time Steps (Mesh Size : 100×100).....	70
3.11	Unsteady Velocity at the Center for Various Mesh Sizes ($\Delta t=0.01$).....	72
3.12	Unsteady Velocity at the Center for Various Values of Convergence Criterion (Mesh : $100 \times 100; \Delta t=0.01$).....	73
4.1	Unsteady Velocity at Midpoint Between the Plates.....	76

4.2	Velocity Distribution at Various Times.....	78
4.3	Velocity Distribution at Various Times.....	79
4.4	Velocity Decay at Midpoint.....	80
4.5	The Effect of Viscosity on Midpoint Velocity.....	82
4.6	Temperature Distribution at Various Times.....	83
4.7	Comparison of Temperature Distributions.....	85
4.8	Comparison of Computed Temperature with Results from Ref. [12].....	87
4.9	Time Variation of Temperature Gradient at the Wall.....	88
4.10	Density Profile at Various Times.....	90
4.11	The Effect of Gravity on Midpoint Velocity.....	91
4.12	Unsteady Center Velocity.....	93
4.13	Comparison of Temperature Profiles at 0.1 Seconds.....	95
4.14	Unsteady Center Velocity.....	96
4.15	Temperature Map at $t=10.0$	98
4.16	Temperature Map at $t=20.0$	99
4.17	Temperature Map at $t=40.0$	100
4.18	Temperature Map from Ref. [9] at $t=20.0$	102

LIST OF SYMBOLS USED

a	general coefficient in the finite difference equations defined in the text
b	variable used in finite difference equations defined in Figures 3.2 to 3.4
c_v	specific heat at constant volume
c_p	specific heat at constant pressure
D	total derivative operator
F_B	body force
k	thermal conductivity
L	span of the model in the x-direction
H	span of the model in the y-direction
P	pressure
Pe	Peclet number (ratio of convection to diffusion)
P'	pressure correction (Figure 3.2 and Appendix A)
Pr	Prandtl number, $\mu c_p/k$
q	heat flux
R	gas constant
Re	Reynolds number based on the speed of sound, $\rho L(RT_o)^{0.5}/\mu$
T	Temperature
t	time
u	velocity in the x-direction
v	velocity in the y-direction
V	velocity vector

x	spatial coordinate
y	spatial coordinate
γ	ratio of specific heats
λ	second coefficient of viscosity
ρ	density
Φ	viscous dissipation
δ	finite-difference derivative operator
[,]	used to denote the greater of two quantities separated by the comma
Δ	small change in variable
∇	gradient operator

Subscripts :

E	node to the east
e	control volume interface to the east
ne	neighbor control volume boundary to the east
nw	neighbor control volume boundary to the west
k	indicates last node
i	node index
N	node to the north
n	control volume interface to the north
nn	neighbor control volume boundary to the north
nne	neighbor control volume boundary to the northeast
nwn	neighbor control volume boundary to the northwest
o	indicates initial value

- p having reference to pressure; also used to denote the point of interest
- S node to the south
- s control volume interface to the south
- W node to the west
- w control volume interface to the west; also wall temperature

Superscripts :

- n time step index
- o used to denote a past time quantity
- ' denotes dimensional quantities
- * used to denote an assumed pressure, velocity, or density computed based on the assumed pressure
- ^ used to denote a velocity calculated based on assumed values of velocity

CHAPTER 1

INTRODUCTION

1.1 BACKGROUND

In the past several decades, a great deal of attention has been focused on the importance and potential of material processing in space. Materials with special or desirable properties can be produced in a weightless or micro-gravity environment where buoyancy driven convective flows, sedimentation, and hydrostatic pressure forces are minimal. Growing high-quality crystals is an example of material processing that could benefit from manufacturing in a space environment.

In most material processing, various fluids are required to be heated or cooled, and a fundamental understanding of the basic heat transfer mechanisms is essential. In conditions of weightlessness, fluid motion can exist as a result of vibration, electromagnetic forces, surface tension, and thermal expansion. This study focuses on a convective heat transfer mechanism that results from thermal volumetric expansion.

Thermoconvection or thermoacoustic convection (TAC) heat transfer is a process driven by thermal volumetric expansion in a fluid. It is energy transport as a result of pressure waves that propagate throughout a medium. When a compressible fluid is

subjected to a rapid addition of heat, either by coming into contact with a high temperature boundary or generated from within the fluid itself (e.g., chemical reactions), the density of the fluid, near the heated boundary can decrease significantly. This decrease in density is associated with an expansion of the control volume occupied by that portion of the fluid. The volumetric expansion of the fluid results in propagation of pressure waves throughout the bulk of the fluid. These pressure waves are similar to those produced by a moving piston. As in that case the pressure waves produced by thermal expansion are acoustic in nature. The fluid expansion and ensuing acoustic waves that move away from the source of the thermal disturbance provide a convective mode for the transfer of heat. This convective motion is believed to allow for a greater transfer of heat to the surrounding fluid than would be achieved by the conduction mode alone.

A variety of studies have been performed on the TAC phenomenon in the past. Some of these studies have focused on the development of computer codes to numerically solve the governing mass, momentum, energy, and state equations. These previous numerical studies predicted very high heat transfer rates and rapid transients. Several experimental investigations were carried out subsequently to verify these numerical models. These experiments^{1,2,3} which were small in number suggest that the numerical studies of the TAC process may have overestimated the importance of this heat

transfer mechanism. The present study is an attempt to understand and rectify these earlier studies.

1.2 LITERATURE REVIEW

Soundhauss⁴ was the first to record that sound could be produced from the open end of a gas-filled tube. While observing glass blowers he found that the glass bulb and the blowing tube would "sing" when the bulb was heated. Soundhauss discovered that the tone was produced by the vibration of the air within the tube, and that the tone was also a characteristic of the geometry of the tube and the bulb. Soundhauss further noted that the strength of the tone grew more intense as the amount of heat transfer to the air increased. A sketch of a Soundhauss tube is shown in Figure 1.1.

Rijke⁵ also made some fundamental discoveries concerning the link between the transfer of heat and acoustic motion. He discovered that sound could be produced if a wire screen were placed in the lower half of a vertical tube, open at both ends, and the wire were subsequently heated. A diagram of a Rijke tube is shown in Figure 1.2. Rijke noted that the tone would cease if either end of the tube were closed, and that the tone would be encouraged if the wire screen were placed toward the lower quarter of the tube. Rijke theorized that the sound was produced by the vibration of the air passing through the tube. He believed that this vibration was caused by the rapid heating and cooling of the air as it first came into

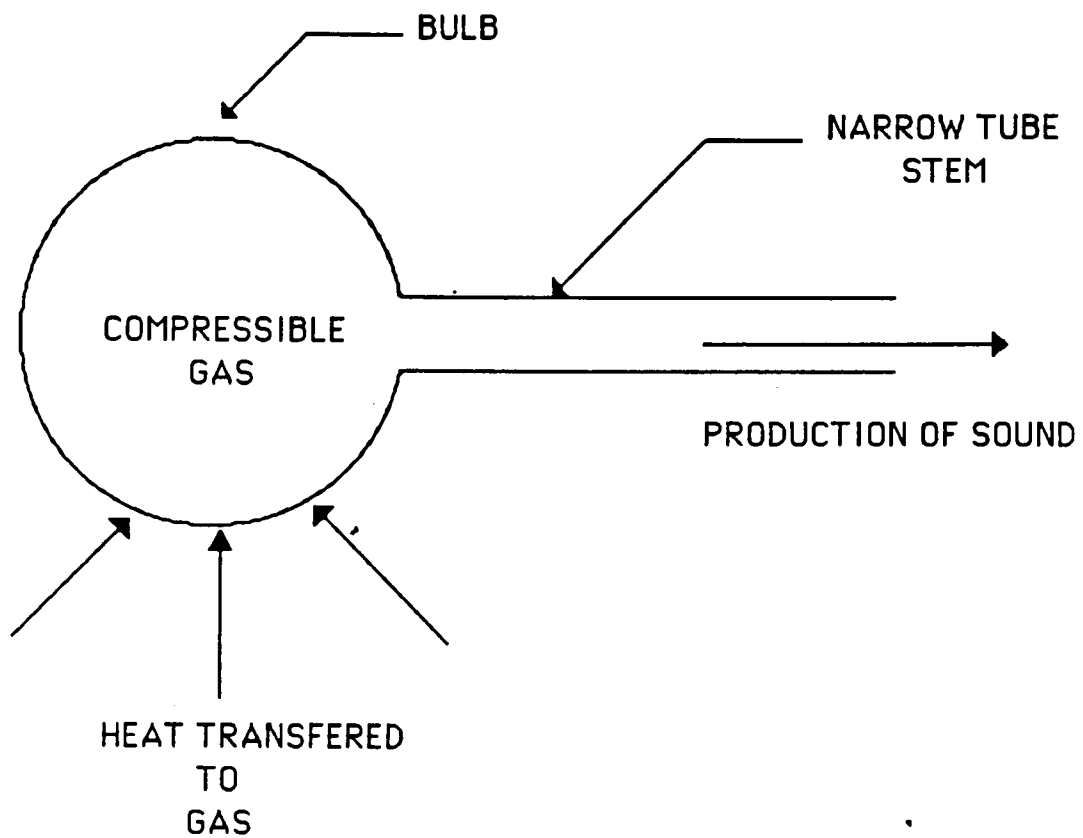


Figure 1.1 Sound Production From a Soundhauss Tube

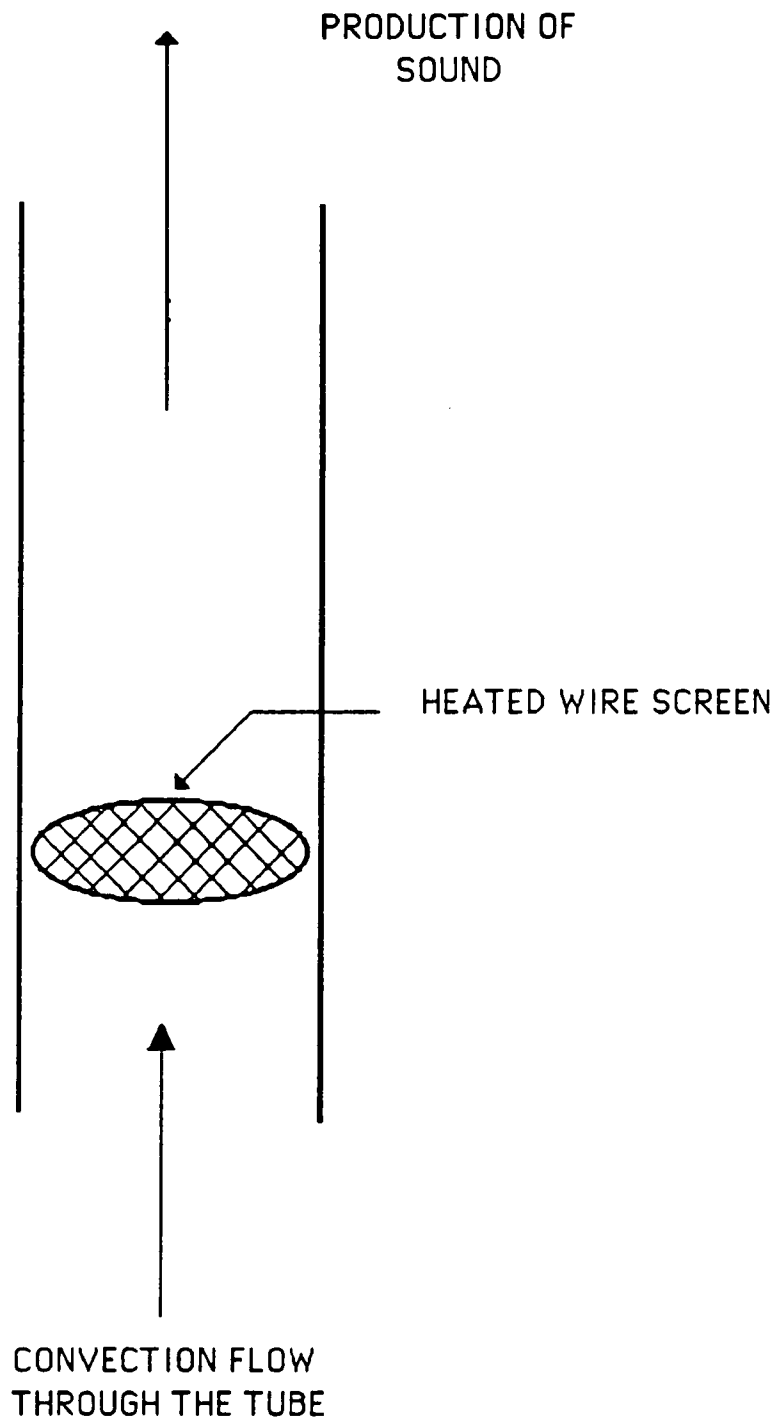


Figure 1.2 Sound Production From a Rijke Tube

contact with the heated wire screen and then with the cooler glass on the tube walls.

Lord Rayleigh provided further insight into the problem with the following criterion: "If heat be given to the air at the moment of the greatest condensation or taken from it at the moment of greatest rarefaction, the vibration is encouraged."⁶

In 1967, Larkin⁷ developed the first numerical model to solve the problem of one-dimensional heat flow to a confined ideal gas. Larkin analyzed helium confined between two infinite parallel plates to simulate one-dimensional heat flow. Initially the helium was assumed to be at rest with an initial temperature, T_0 (273 K), and pressure, P_0 (1 atm). For time $t > 0$ one boundary was instantly raised to $2T_0$ (546 K) while the other boundary was kept at its initial temperature. Larkin's model considered only a zero-gravity environment and thus the body force term was dropped from the momentum equation. Even in the absence of a gravitational field fluid motion was induced as the result of fluid compressibility. The oscillatory nature of the calculated velocity field indicated the acoustical nature of the problem. The calculated temperature field showed much higher values than those predicted by a conduction-only model. Larkin drew the following conclusions :

- a) The oscillatory nature of the velocity field indicates that it is acoustic in nature.

- b) Fluid motion was observed to greatly enhance the rate of heat transfer and pressure rise.
- c) Pressure rise rates were much higher than those predicted by a conduction-only model.

A subsequent study by Spradley, Bourgeois, Fan, and Grodzka⁸ also considered the problem of one-dimensional heat flow to a confined ideal gas in zero-gravity. This study was initiated to explain the behavior of data collected aboard the Apollo 14 spacecraft, and to assess the role of micro gravity convection in space processing. Spradley, et al. considered two problems. The first was that of helium confined between two infinite parallel plates. Rapid heat addition was provided by a step change in the temperature of one of the plates (twice its initial value) while the other plate was maintained at its initial temperature. The second problem considered was that of carbon dioxide confined between two infinite concentric cylinders. Again rapid heat addition was provided by a step change in the temperature of the inner cylinder to twice its initial value. The outer cylinder was maintained at its initial temperature.

Spradley, et al. chose an explicit finite-difference scheme (forward marching in time) to solve the thermal convection problem. Forward time differences were used on all unsteady terms, and space centered differences were used on all space derivatives. The exception to this was the convective terms in the momentum and energy equations. These convective terms were discretized using an

upwind, forward-backward looking scheme to prevent numerical instabilities. A sketch of the two problems considered by Spradley, et al. is shown in Figure 1.3 and 1.4.

As observed in these figures, the calculated transient fluid temperatures are very similar to those predicted by Larkin. The acoustic velocity field calculated by Spradley, et al. was, however, one order of magnitude lower than that predicted by Larkin. Spradley, attributed this to unspecified differences in the numerical techniques used in the studies. Spradley, et al. drew the following conclusions:

- a) Thermally induced wave motion is acoustic.
- b) TAC greatly enhances the rate of heat transfer,
- c) The mean pressure in the confined fluid rises more rapidly as a result of thermoacoustic effects,
- d) Pressure convection effects on Conservation of Energy can strongly influence fluid behavior, and
- e) The magnitude of TAC effects is a strong function of the severity of the thermal boundary condition.

A subsequent study presented by Spradley and Churchill⁹ considered the problem of TAC and mixed mode TAC-natural convection in a rectangular cavity. They considered three cases. The first case consisted of a one-dimensional simulation of the zero-gravity case to compare with the study by Larkin. The second case consisted of a steady-state, buoyancy-driven natural convection flow for comparison with the results obtained by Wilkes and

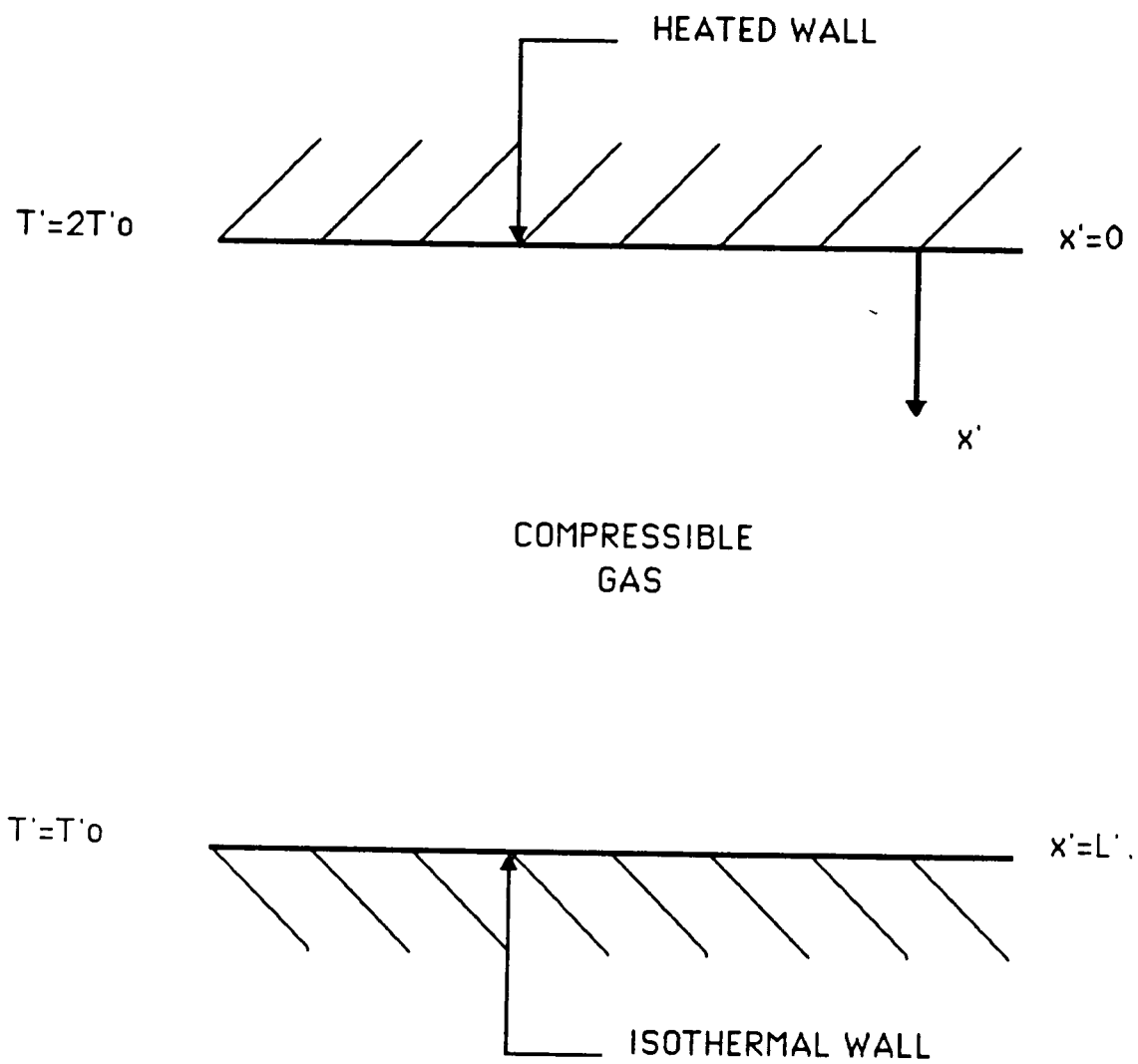


Figure 1.3 Parallel Plate Model

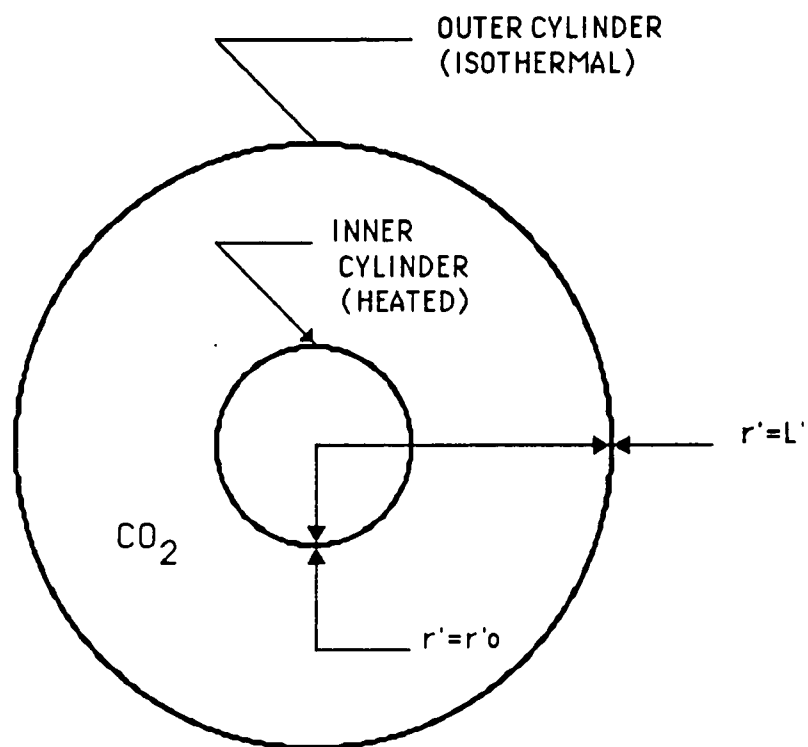


Figure 1.4 Concentric Cylinder Model

Churchill.¹⁰ The third case was that of unsteady convection in a container of helium, at varying levels of gravity.

Similar to the first study, an explicit finite-difference numerical scheme was employed. A cell-centered finite-difference grid was used for space derivatives, except for convection terms, with differences taken across the cell using known values in neighboring cells and/or interpolated values at the cell boundaries. The convection terms were evaluated again using an upwind scheme.

The results of the first case produced a temperature field that agreed with that calculated by Larkin. The second case produced results for a steady-state, buoyancy-driven convective flow in a rectangular region. This case was designed to compare with similar work conducted by Wilkes and Churchill. An exact simulation was not possible because Spradley and Churchill considered variable pressure and density effects that Wilkes and Churchill did not. Steady-state temperature and stream-function maps did agree qualitatively to those presented by Wilkes and Churchill. However, the latter was skewed somewhat toward the cooler regions of the fluid. Spradley and Churchill explained this effect as a consequence of the variable density and pressure effects considered in their study.

The third case solutions predicted rapidly decreasing velocities with initial magnitudes of Mach 0.2. The progression of time gave rise to density gradients and buoyancy-driven convection, depending upon the assumed value of gravity. In the zero-gravity case

temperature fields were found to be much higher than those predicted by a conduction only model.

In a series of papers published between 1979 and 1984 : Kasso^y;¹¹ Radwhan and Kasso^y;¹² Poland and Kasso^y;¹³ and Clark, Kasso^y, and Riley¹⁴ considered the problem of one-dimensional TAC. The specific problem of interest was that of a compressible medium enclosed between two infinite parallel plates. The initial plate and fluid temperatures were the same. The plate temperatures were then changed to a higher values and the resulting temperature and velocity fields analyzed. In each case a perturbation method was used to solve the temperature and velocity fields. In all cases the problem is shown to contain a small parameter formed by the ratio of two time scales: a slow conduction time and a fast acoustic time. The problem is then cast in a form suitable for singular perturbation methods with appropriate expansions made by employing this small parameter.

Salah-Eddine¹ conducted a series of TAC experiments. His experimental apparatus consisted of an insulated cylinder, closed at both ends, with a heater located at one end. A series of thermocouples were placed along the axis of the cylinder in order to measure the temperature rise in the air. Experiments were conducted in both gravity and micro-gravity environments. The micro-gravity experiments were made possible by using the Drop Tower Facility at NASA-Lewis Research Center. Experimental results were compared with a numerical conduction model. It was

found that the temperatures measured by the thermocouples were much higher than those predicted by a conduction-only model, and that for the particular geometry considered where heating was applied from the top, no substantial difference in the measurement of the surrounding fluid temperatures between gravity and zero-gravity environments was found. It was also found that radiation absorption by the thermocouples can produce substantial error in the measurement of the surrounding fluid temperature. This required an adjustment in the experimental results by calculating the radiation-induced error numerically. A sketch of Salah-Eddine's experimental apparatus is shown in Figure 1.5.

In a subsequent study, Catt³ experimentally measured the radiation-induced error in the fluid temperature by the thermocouples. The apparatus consisted of a rectangular cavity with a heater located at one end. The end plate opposite the heater was removable in order to allow for open air testing. Thermocouples were placed along the vertical axis of the apparatus. The effects of radiation heat transfer were accounted for by evacuating the cavity and measuring the radiation absorbed by the thermocouples. The resulting effect of TAC was estimated by calculating an experimental Nusselt number based on convective cooling of the thermocouple to the air. Some conclusions reached by Catt were:

- a) Relatively large initial air movement is generated which subsides with time.

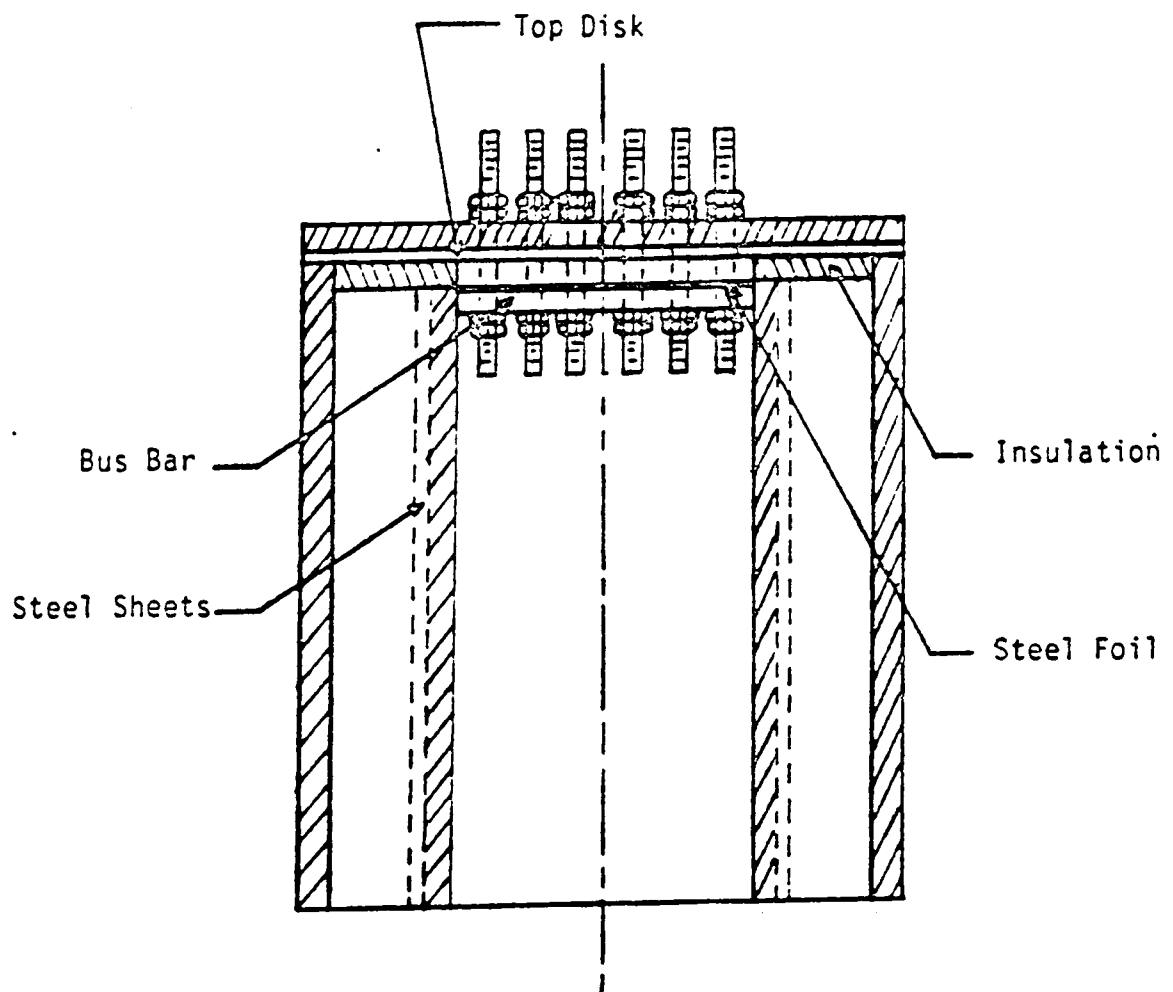


Figure 1.5 Salah-Eddine's Experimental Apparatus

- b) The presence of convective cooling of the thermocouples is evidence of the existence of TAC.
- c) The amount of heat transfer from the thermocouple to the air exceeded that predicted by a conduction-only model or a combined conduction-natural convection model.

1.3 RESEARCH OBJECTIVES

A careful review of the relevant literature indicates that previous TAC numerical models are inconsistent with other analytical results and the available experimental data. These numerical schemes predict unreasonably high heat transfer rates. For example the analytical work conducted by Spradley, et al., for helium confined between two infinite parallel plates 15.3 cm apart, predicts a near steady-state temperature and density profile after only 0.2 seconds. However, Radhwan and Kassoy suggest that a near steady-state temperature profile is not reached until 16 minutes after initiation of heating.

Furthermore past analytical and numerical works have predicted vastly different acoustic velocities. The numerical code developed by Larkin predicted acoustic velocities with a maximum Mach number of 0.12. Spradley, et al. on the other hand, predicted maximum acoustic Mach numbers of 0.0175. The perturbation scheme developed by Radhwan and Kassoy predicted acoustic wave Mach numbers of approximately 10^{-3} .

Additional investigation of the numerical scheme presented by Spradley, et al. indicates a significant dependence upon the selected mesh size. Also the choice of an explicit finite-difference scheme for the solution of a transient non-linear problem seems inappropriate given the potential numerical instabilities characteristic of explicit schemes. The above inconsistencies and problems that exist in previous numerical schemes provide the key motivation for this research.

The primary purpose of this research, therefore, is the development of an accurate numerical model for the study of thermoacoustic oscillations and the resulting heat transfer. To accomplish this the following research objectives are pursued in this study:

- a) The development of an implicit version of the Spradley, et al. model and the evaluation of its accuracy through successive grid and time step refinement studies, and comparison with available experimental data.
- b) The development of a new numerical scheme for the solution of a one-dimensional TAC problem based on a modified (compressible) SIMPLER (Semi-Implicit Method for Pressure Linked Equations Revised) algorithm .¹⁵
- c) Evaluation of the effectiveness of this new model through systematic mesh and time step refinement studies, and comparison with available experimental data.

- d) Expansion of this new numerical model to solve the two-dimensional TAC and mixed mode (TAC and natural convection) problem, and compare the results with the two-dimensional solution of Spradley and Churchill, and available experimental data.

CHAPTER 2

ANALYSIS

In this chapter a one-dimensional TAC problem is examined in detail. The problem is defined and the fundamental equations governing the flow are reduced from their most general form, and non-dimensionalized. Then the equations are discretized in much the same fashion as that presented by Spradley, et al. A more detailed discussion of the model used in Reference [8] follows.

2.1 MATHEMATICAL FORMULATION OF THE PROBLEM

The problem under consideration is that of a compressible fluid enclosed between two infinite parallel plates, a distance L' apart, in a zero-gravity environment. Initially the fluid is at rest and isothermal at temperature T_o' . At time $t'=0$ the temperature of one of the plates is rapidly increased to a high and constant temperature T_w' . A sketch of the problem is shown in Figure 1.3.

The general dimensional equations governing the resulting flow are:

Continuity:

$$\frac{D\rho}{Dt} + \rho(\nabla \cdot \vec{V}) = 0 \quad (2.1)$$

Newton's Second Law:

$$\rho \frac{D\vec{V}}{Dt} = \vec{F}_B - \nabla P - \nabla x [\mu (\nabla x \vec{V})] + \nabla [(\lambda - 2\mu) \nabla \cdot \vec{V}] \quad (2.2)$$

Energy:

$$\rho \frac{De}{Dt} + e \left[\frac{D\rho}{Dt} + \rho \nabla \cdot \vec{V} \right] = -\nabla q'' + q''' + P \nabla \cdot \vec{V} + \mu \Phi \quad (2.3)$$

State:

$$P = P(\rho, T) \quad (2.4)$$

The following assumptions are made :

- (a) The fluid is Newtonian and obeys Stokes's viscosity hypothesis ($\lambda = -2\mu$).
- (b) Thermal properties k, c_v, μ, γ are constant.
- (c) There are no radiation or internal heat sources.
- (d) Viscous dissipation is negligible.
- (e) Body Forces are Negligible.
- (f) Ideal gas equation of state.
- (g) Only changes along one space dimension (x) are important.

Equations (2.1) through (2.4) can now be written as:

Continuity:

$$\frac{\partial}{\partial t'}(\rho') + \frac{\partial}{\partial x'}(\rho' u') = 0 \quad (2.5)$$

Newton's Second Law:

$$\frac{\partial}{\partial t'}(\rho' u') + \frac{\partial}{\partial x'}(\rho' u' u') = -\frac{\partial P'}{\partial x'} + \frac{4}{3}\mu' \frac{\partial^2 u'}{\partial x'^2} \quad (2.6)$$

Energy:

$$\rho' c' \frac{\partial T'}{\partial t'} + \rho' c' u' \frac{\partial T'}{\partial x'} = -P' \frac{\partial u'}{\partial x'} + k' \frac{\partial^2 T'}{\partial x'^2} \quad (2.7)$$

State:

$$P' = \rho' R' T' \quad (2.8)$$

The appropriate boundary and initial conditions are:

Velocity:

$$u'(x', t'=0) = 0 \quad \text{Fluid initially at rest} \quad (2.9)$$

$$u'(x'=0, t')=0=u'(x=L', t') \quad \text{No slip at the wall} \quad (2.10)$$

Temperature:

$$T'(x', t'=0)=T'_o \quad \text{Initially isothermal} \quad (2.11)$$

$$T'(x'=0,t')=T'_w \quad \text{Heated wall} \quad (2.12)$$

$$T'(x'=L',t')=T'_o \quad \text{Isothermal wall} \quad (2.13)$$

Pressure:

$$\frac{\partial P'}{\partial x'}(x',t'=0) = 0 \quad \text{No body forces} \quad (2.14)$$

Density:

$$\rho'_o(x',t'=0) = \frac{P'_o}{R'T'_o} \quad (2.15)$$

Equations (2.5) through (2.15) define the mathematical model for the analysis of a one-dimensional TAC in zero-gravity. Because the problem involves pressure waves that propagate at acoustic velocities, the initial speed of sound (precisely $\sqrt{R'T'_o}$) is used as the characteristic velocity of the problem. The characteristic time of the problem is the time required for a pressure wave to traverse the length of the container. The temperature, density, and pressure are non-dimensionalized with their initial values T'_o , ρ'_o , P'_o respectively. The space coordinate is non-dimensionalized using the distance between the plates L' . The normalized form of the governing equations, (2.5) to (2.15), can be written as:

Continuity:

$$\frac{\partial}{\partial t}(\rho) + \frac{\partial}{\partial x}(\rho \cdot u) = 0 \quad (2.16)$$

Newton's Second Law:

$$\frac{\partial}{\partial t}(\rho u) + \frac{\partial}{\partial x}(\rho \cdot u u) = -\frac{\partial P}{\partial x} + \frac{4}{3} \left(\frac{1}{\text{Re}} \right) \frac{\partial^2 u}{\partial x^2} \quad (2.17)$$

Energy:

$$\rho \frac{\partial T}{\partial t} + \rho \cdot u \frac{\partial T}{\partial x} = -(\gamma - 1) P \frac{\partial u}{\partial x} + \left(\frac{\gamma}{\text{RePr}} \right) \frac{\partial^2 T}{\partial x^2} \quad (2.18)$$

State:

$$P = \rho \cdot T \quad (2.19)$$

The dimensionless boundary and initial conditions, (2.9) to (2.15) can now be recast as:

Velocity:

$$u(x=0,t)=0=u(x=1,t) \quad (2.20)$$

$$u(x,t=0)=0 \quad (2.21)$$

Temperature:

$$T(x,t=0)=1.0 \quad (2.22)$$

$$T(x=L,t)=1.0 \quad (2.23)$$

$$T(x=0,t)=T'_w/T'_o \quad (2.24)$$

Density:

$$\rho(x,t=0)=1.0 \quad (2.25)$$

Pressure:

$$P(x,t=0)=1.0 \quad (2.26)$$

The dimensionless groups which appear in the equations are:

Prandtl Number $Pr = \frac{\mu' c_p'}{k'}$

Reynolds Number $Re = \frac{\rho' \cdot L' \sqrt{R' T'_o}}{\mu'}$

Peclet Number $Pe = Re \cdot Pr$

Ratio of Specific Heats $\gamma = \frac{c'_p}{c'_v}$

2.2 PREVIOUS NUMERICAL SCHEMES

As previously mentioned, the above normalized equations have been solved numerically, by Spradley, et al., to obtain the temperature, density, pressure, and velocity fields. The discretized equations and method of solution reported in that study will be considered here in some detail as a key reference for comparison with the methods employed in this study.

Spradley, et al. applied an explicit, finite-difference scheme for the solution of the above equations. Forward time differences were applied to all unsteady terms, and space derivatives were evaluated by means of a central difference scheme. The exception lies in the convection terms which were modeled by an upwind, forward-backward looking scheme.

The use of this upwind scheme, first put forth by Courant, Isaacson, and Rees¹⁶ requires that the space derivative of interest be evaluated with full knowledge of the direction of the flow. This eliminates certain numerical instabilities present with central differencing techniques. The upwind scheme presented by Spradley, et al. for the convection term at the i^{th} node and the n^{th} time step is given by equation (2.27).

$$\begin{aligned}
u \frac{\partial T}{\partial x} &= u^n_i \left(\frac{T^n_{i+1} - T^n_i}{\Delta x} \right) & \text{if } u^n_i < 0 \\
&\text{or} \\
u \frac{\partial T}{\partial x} &= u^n_i \left(\frac{T^n_i - T^n_{i-1}}{\Delta x} \right) & \text{if } u^n_i > 0
\end{aligned} \tag{2.27}$$

The physical significance of equation (2.27) is that in a strong convective flow the flow conditions in a node are primarily determined by the conditions upstream of the node. The discretized forms of the Continuity, Newton's Second Law, and Energy equations are shown in Figures 2.1 to 2.3. A sketch of the nodal system used in this model is shown in Figure 2.4.

The method of solution employed by Spradley, et al. first involved the solution of the momentum equation to obtain the $(\rho u)_i^{n+1}$ term. Using, this term the spatial first derivative was solved and then substituted into the continuity equation to obtain the new density profile. Solving the energy equation provided the new temperature field. Finally, the state equation was used to calculate the new pressure field.

Results of the numerical model developed in Reference [8] are shown in Figures 2.5 and 2.6. Figure 2.5 shows the non-dimensional midpoint velocity versus dimensionless time. The oscillatory nature of the velocity, characteristic of the problem, is clearly seen in this figure. The period of oscillation is indicative of the acoustic nature of the problem. That is, the period of oscillation of the velocity wave should be the time required for the acoustic wave to

Continuity Equation:

$$\frac{\partial}{\partial t} \rho = - \frac{\partial}{\partial x} (\rho u)$$

Finte Difference Equations:

$$\rho_i^{n+1} = \rho_i^n - \Delta t \left| \delta_x (\rho u)_i^{n+1} \right|$$

$$\delta_x (\rho u)_i^{n+1} = \frac{(\rho u)_{i+1}^{n+1} - (\rho u)_{i-1}^{n+1}}{2\Delta x}$$

$$\delta_x (\rho u)_1^{n+1} = \frac{(\rho u)_1^{n+1} + (\rho u)_2^{n+1}}{2\Delta x}$$

$$\delta_x (\rho u)_k^n = - \frac{(\rho u)_k^{n+1} + (\rho u)_{k-1}^{n+1}}{2\Delta x}$$

$$u_i^{n+1} = \frac{\rho u_i^{n+1}}{\rho_i^{n+1}}$$

Figure 2.1 Continuity Equation In Discretized Form

Newton's 2nd Law:

$$\frac{\partial}{\partial t} (\rho u) + \frac{\partial}{\partial x} (\rho u u) = -\frac{\partial P}{\partial x} + a \frac{Lg'}{RT_o} \rho + \frac{4}{3} \left(\frac{1}{Re} \right) \frac{\partial^2 u}{\partial x^2}$$

Finite Difference Equations:

$$(\rho u)_i^{n+1} = (\rho u)_i^n - \Delta t \left[\delta_x (\rho u u)_i^n + \delta_x (P^n)_i - a \frac{Lg'}{RT_o} (\rho)_i^n - \frac{4}{3} \frac{1}{Re} \delta_x^2 (u_i^n) \right]$$

$$\delta_x (\rho u u)_i^n = \frac{(\rho u)_{i+1}^n u_{i+1}^n - (\rho u)_i^n u_i^n}{\Delta x} \quad \text{for } u_i^n < 0$$

$$\delta_x (\rho u u)_i^n = \frac{(\rho u)_i^n u_i^n - (\rho u)_{i-1}^n u_{i-1}^n}{\Delta x} \quad \text{for } u_i^n \geq 0$$

$$\delta_x (\rho u u)_1^n = \frac{(\rho u)_1^n u_1^n + (\rho u)_2^n u_2^n}{2\Delta x}$$

$$\delta_x (\rho u u)_k^n = -\frac{(\rho u)_k^n u_k^n + (\rho u)_{k-1}^n u_{k-1}^n}{2\Delta x}$$

$$\delta_x (P^n)_1 = \frac{P_2^n - P_1^n}{\Delta x}, \quad \delta_x (P^n)_k = \frac{P_k^n - P_{k-1}^n}{\Delta x}$$

$$\delta_x (P^n)_i = \frac{P_{i+1}^n - P_{i-1}^n}{2\Delta x}$$

$$\delta_x^2 (u_1^n) = \frac{u_2^n - 3u_1^n}{\Delta x^2}, \quad \delta_x^2 (u_k^n) = \frac{u_{k-1}^n - 3u_k^n}{\Delta x^2}$$

$$\delta_x^2 (u_i^n) = \frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{\Delta x^2}$$

Figure 2.2 Newton's Second Law In Discretized Form

Energy Equation:

$$\frac{\partial T}{\partial t} + u \frac{\partial T}{\partial x} = - \frac{(\gamma - 1)}{\rho} P \frac{\partial u}{\partial x} + \frac{\gamma}{\rho} \left(\frac{1}{RePr} \right) \frac{\partial^2 T}{\partial x^2}$$

Finite Difference Equations:

$$T_i^{n+1} = T_i^n - \Delta t \left[u_i^{n+1} \delta_x (T_i^n) + \frac{\gamma - 1}{\rho_i^{n+1}} P_i^n \delta_x (u_i^{n+1}) - \frac{\gamma}{\rho_i^{n+1}} \left(\frac{1}{RePr} \right) \delta_x^2 (T_i^n) \right]$$

$$\delta_x (u_i^{n+1}) = \frac{u_{i+1}^{n+1} - u_{i-1}^{n+1}}{2 \Delta x}$$

$$\delta_x (u)_1^{n+1} = \frac{u_1^{n+1} + u_2^{n+1}}{2 \Delta x} \quad \delta_x (u)_k^{n+1} = - \frac{u_k^{n+1} + u_{k-1}^{n+1}}{2 \Delta x}$$

$$\delta_x (T_i^n) = \begin{cases} (T_{i+1}^n - T_i^n) / \Delta x & u_i^n < 0 \\ (T_i^n - T_{i-1}^n) / \Delta x & u_i^n \geq 0 \end{cases}$$

$$\delta_x (T_1^n) = \begin{cases} (T_2^n - T_1^n) / \Delta x & u_1^n < 0 \\ (T_1^n - T_w) / (0.5 \Delta x) & u_1^n \geq 0 \end{cases}$$

$$\delta_x (T_k^n) = \begin{cases} (T_k^n - T_{k-1}^n) / \Delta x & u_k^n \geq 0 \\ (T_k^n - T_w) / (0.5 \Delta x) & u_k^n < 0 \end{cases}$$

$$\delta_x^2 (T_i^n) = \frac{T_{i+1}^n - 2T_i^n + T_{i-1}^n}{\Delta x^2}$$

$$\delta_x^2 (T_1^n) = \frac{4}{3} \left(\frac{T_2^n - 3T_1^n + 2T_w}{\Delta x^2} \right)$$

$$\delta_x^2 (T_k^n) = \frac{4}{3} \left(\frac{T_{k-1}^n - 3T_k^n + 2T_w}{\Delta x^2} \right)$$

Figure 2.3 Energy Equation In Discretized Form

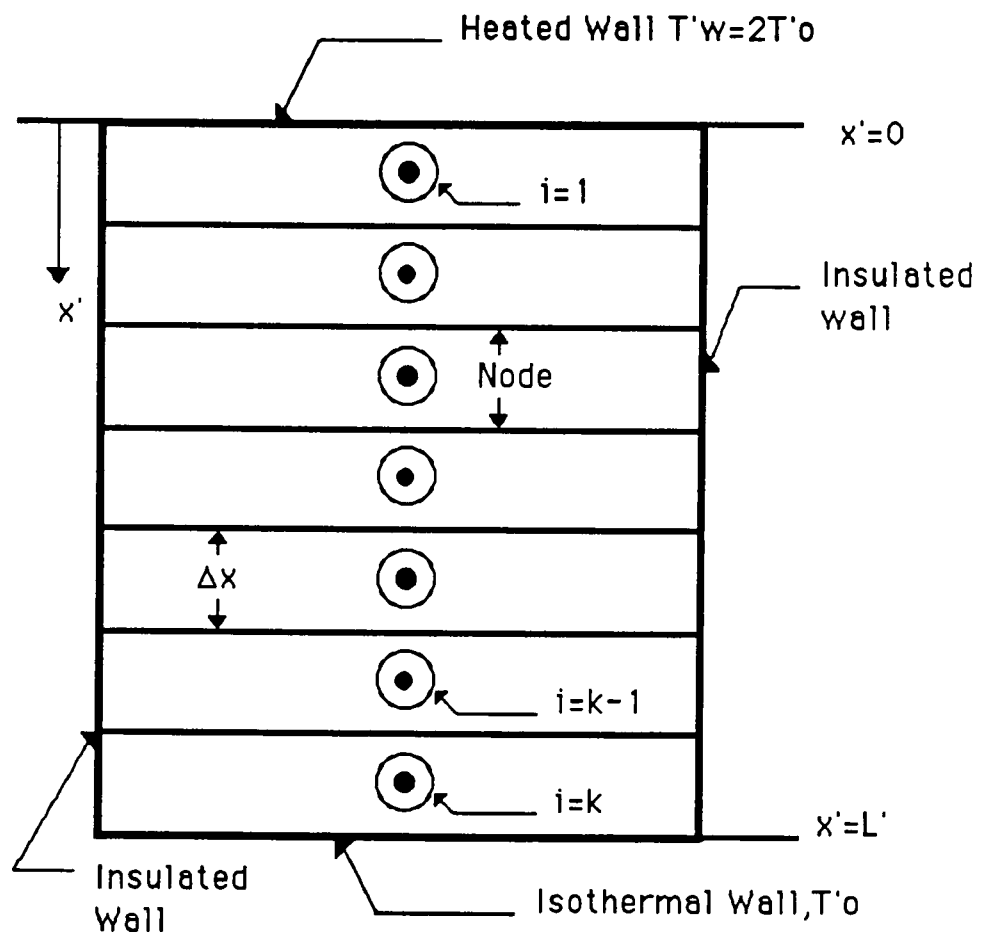


Figure 2.4 Nodal System Sketch

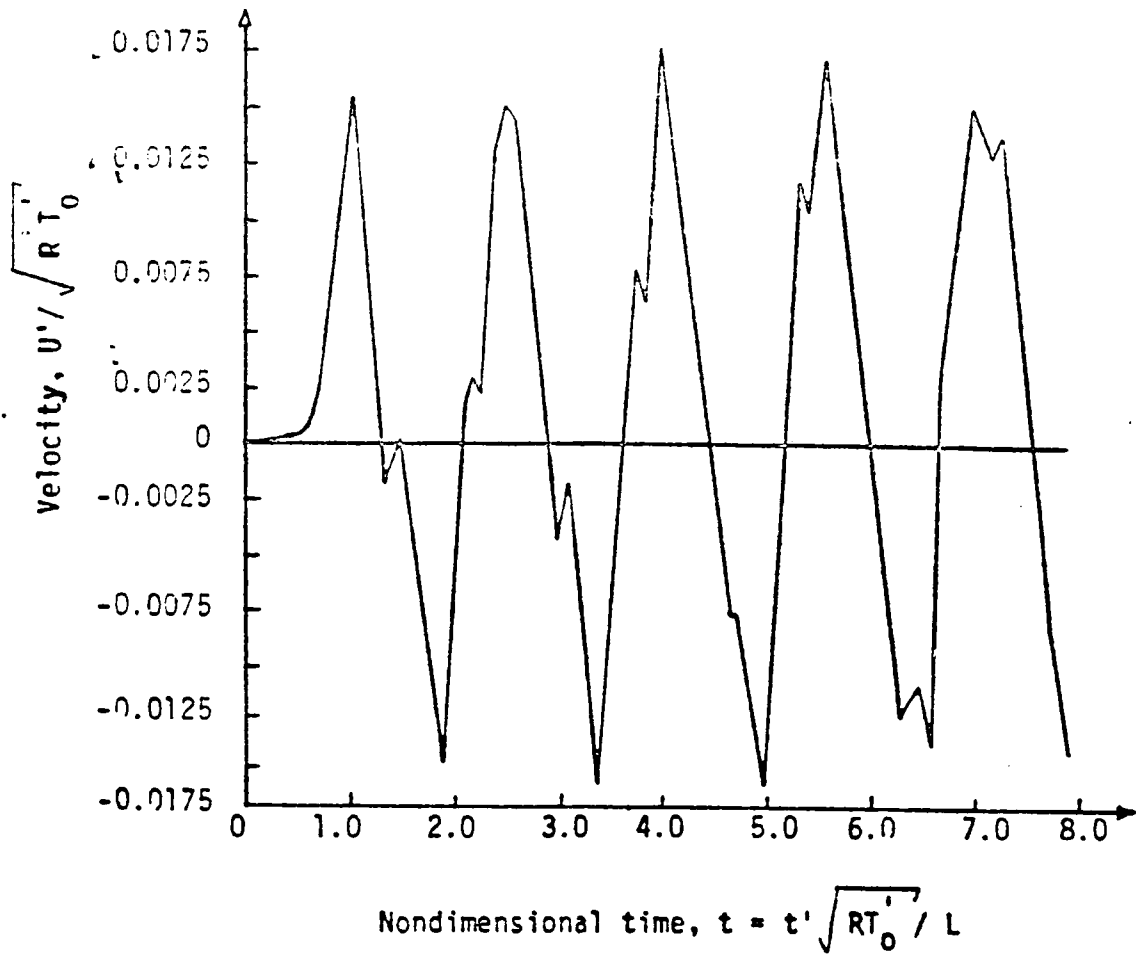


Figure 2.5 Non-Dimensional Midpoint Velocity
Versus Dimensionless Time from Ref [8]

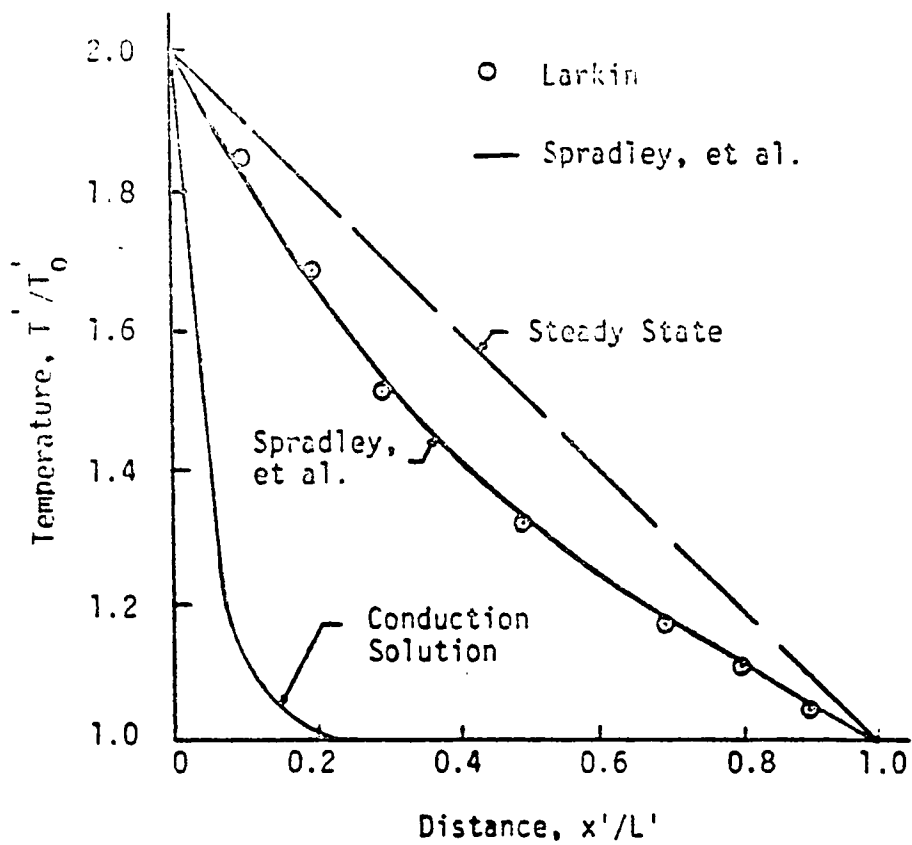


Figure 2.6 Temperature Profiles at 0.2 Seconds from Ref [8]

travel the distance between the plates and back (in dimensionless terms a distance of 2.0) divided by the dimensionless speed of sound $\gamma^{1/2}$. The measured period of oscillation of the velocity wave shown in Figure 2.5 is approximately 1.5. The calculated period for this wave is 1.55.

A spatial temperature distribution is shown in Figure 2.6 at a time of 0.2 seconds. This solution is compared with a steady-state solution, a rigid conduction solution at 0.2 seconds, and the solution presented by Larkin at 0.2 seconds. The rapid transient behavior exhibited by the Spradley, et al. model is clearly seen in this figure. Although it is believed that TAC aids in the heat transfer process, available experimental data does not indicate that such rapid transient behavior exists.

In order to verify the original model developed in Reference [8] a grid refinement study was carried out by Parang.³ The original work used 22 nodes and a non-dimensional time increment of 0.015. It was shown that the results were significantly dependent on the selected mesh size. For example Figure 2.7 shows temperature profiles at 0.1 seconds for 22, 44, and 88 nodes, respectively, compared to a rigid conduction solution. It is clearly seen that as the number of nodes increases, the temperature profile approaches that of a conduction solution. Thus giving further evidence that the original numerical study may have overestimated the fluid temperature distribution.

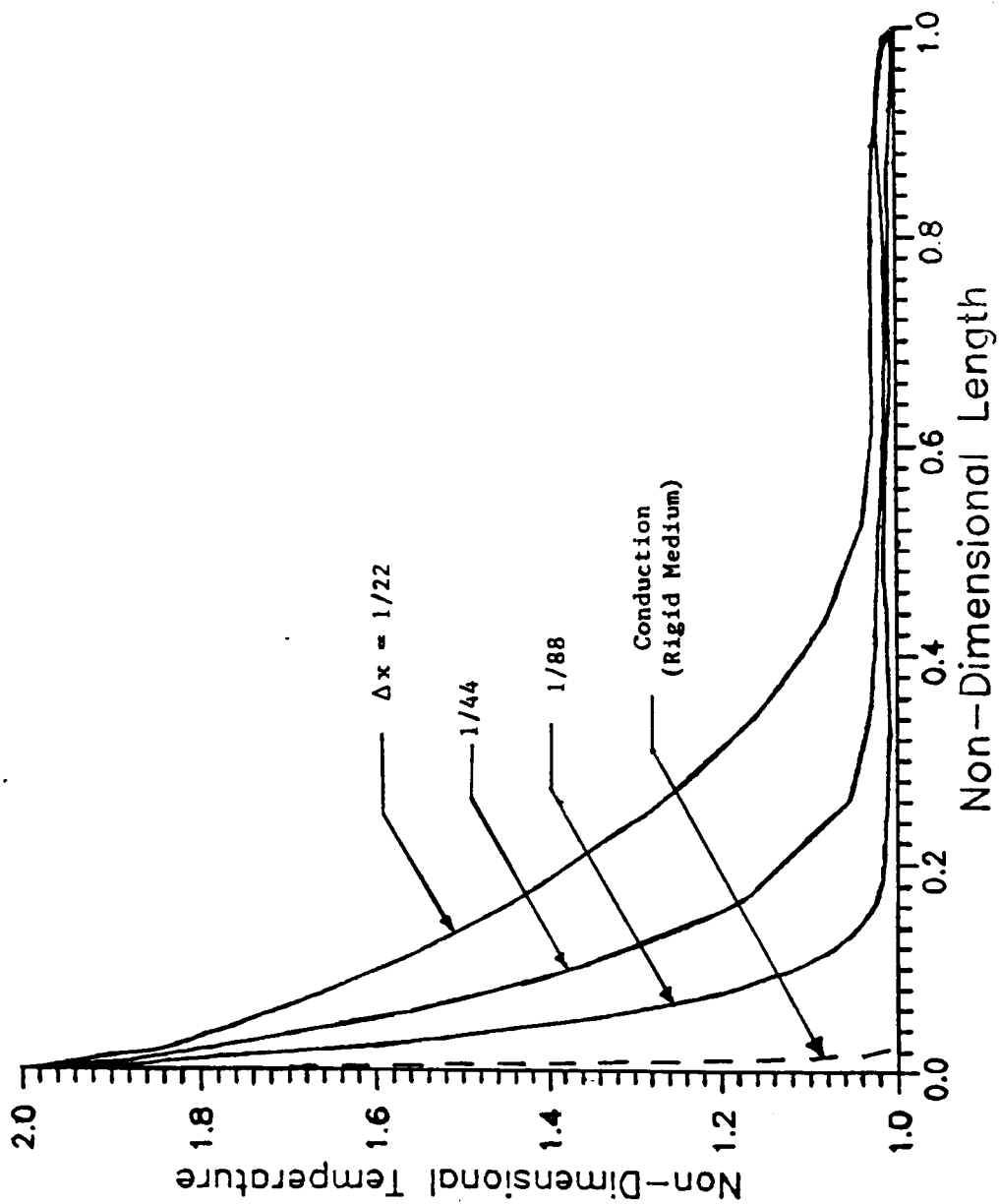


Figure 2.7 Temperature Profiles At 0.1 Seconds from Ref [3]

2.3 MODIFICATIONS: THE IMPLICIT SCHEME

In this study the numerical scheme used in Reference [8] was first reproduced and then modified to assess the accuracy of the TAC model. The resulting program, MODIPL.FOR, is listed in Appendix B. The primary modification was to replace the forward-marching, explicit scheme with an implicit scheme in time. One of the major benefits of the implicit method is that there are no restrictions on the size of the mesh or the time increment for stability purposes. Thus a whole range of space and time increments are available, in principle, for use.

The physics of the problem, however, require a certain limitation on the choice of the space and time increment. If the choice of the time step is too large, then the acoustic wave might be missed entirely. Likewise a sufficiently small space step is required for reasonable numerical accuracy. While no strict requirements were placed on the space and time steps, larger values of space and time steps were not selected in the implicit scheme. This was done to satisfy two purposes. The first was to make it easier to compare with the original work, and the second was to produce answers with improved accuracy. Thus no significant savings in computer run time was achieved through the use of the implicit scheme.

The discretized equations used in the implicit scheme are the same as those shown in Figures 2.1 through 2.3 with the exception

that all space derivatives were evaluated at a future time. Furthermore, the solution method involves an iterative scheme for the solution of fluid temperature, density, pressure, and velocity.

The actual procedure for solution of the discretized equations is similar to that of the explicit scheme. Initially the momentum equation is solved to obtain $(\rho u)_i^{n+1}$. This term is then used to evaluate the spatial first derivative used in the continuity equation. Thus the continuity equation can be solved to obtain the new density field. The new velocity field is then calculated by substituting the newly found density and the $(\rho u)_i^{n+1}$ back into the spatial first derivative term from the continuity equation and solving for the new velocity. The energy and state equations yield the new temperature and pressure fields, respectively. It should be noted, however, that the new velocities, densities, temperatures, and pressures, for each node, were evaluated based on assumed values at neighboring nodes. These newly calculated values form the basis for the next set of assumed values in the next iteration. Iteration is continued until the calculated value of velocity, for each node differs from the previously assumed value of velocity for the same node by only a small number. The velocity was chosen as the variable upon which convergence should be based for two reasons. First of all, the velocity is numerically the most sensitive variable. Basing convergence upon computed values of velocity adds a greater degree of restriction on numerical computation and presumably improves accuracy. Secondly, the very nature of

thermally induced convection depends upon the acoustic velocity field. Calculating this field correctly helps to ensure that the proper temperature field is obtained and thus the proper effect of TAC on the heat transfer process is determined.

For the initial investigation of the implicit scheme, the same time step and mesh size ($\Delta t=0.015$, $\Delta x=1/22$) as used by Spradley, et al. were utilized. The results agreed very well with those presented by Parang. An initial Mach number of 0.38 was calculated which quickly damped out as time progressed. This velocity field does, however, represent a slight increase from that presented by Spradley, et al. These results are shown in Figure 2.8. Temperature profiles at 0.1 seconds were produced and the results are in agreement with those in Reference [3]. Using 22 nodes and a time step of 0.015 non-dimensional time units the progress of the temperature field toward the steady-state is clearly evident, as observed in Figure 2.9.

A grid and time step refinement study was undertaken to evaluate the accuracy of the implicit scheme. Time steps and mesh sizes were systematically decreased. The time step was decreased from 0.015 to 0.001875 non-dimensional units. Likewise the number of nodes was increased from 22 to 88. The results of this investigation revealed a significant dependence on the mesh size chosen. However, the dependence on the time step was found to be minimal. Increasing the number of nodes beyond 88 was found to increase computer time substantially and make convergence

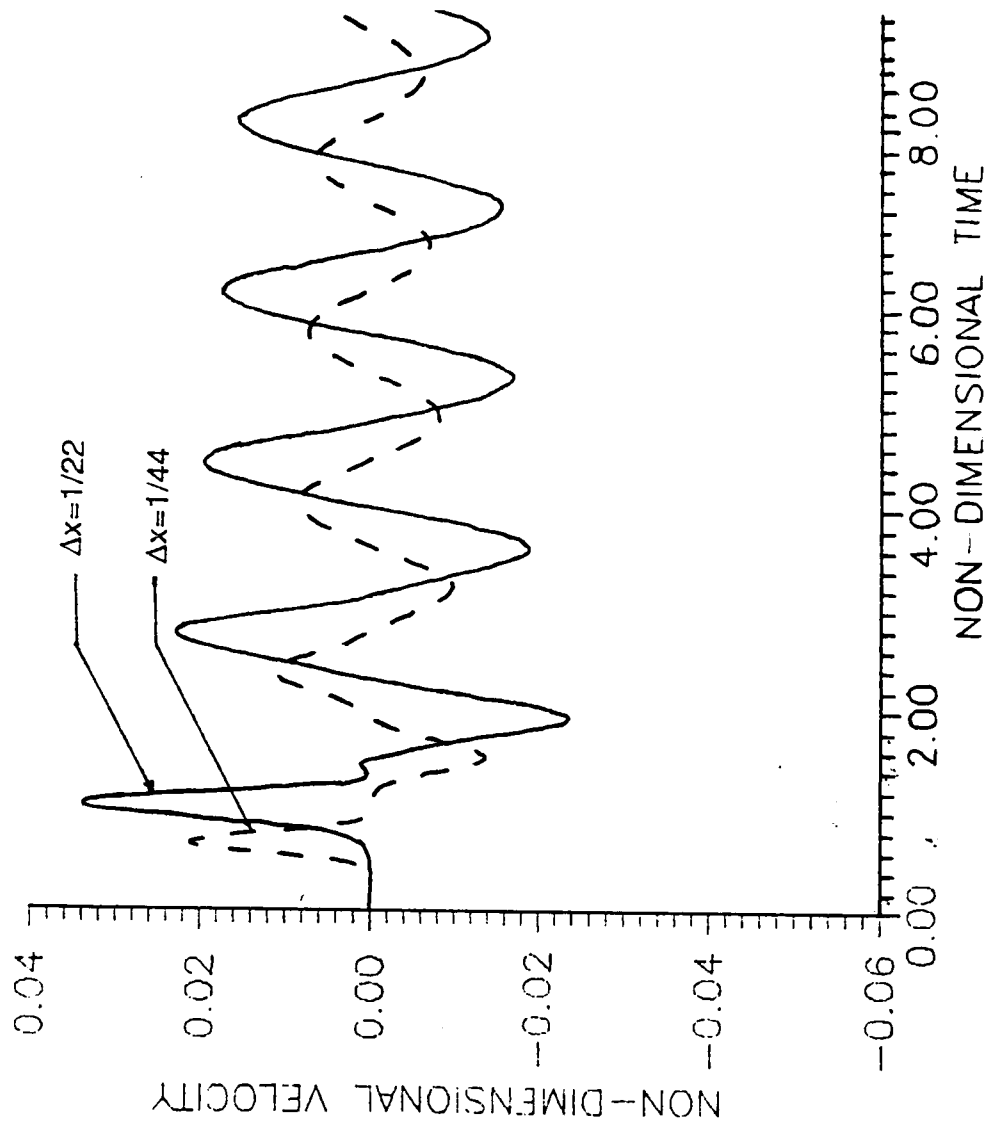


Figure 2.8 Midpoint Velocity Versus Time

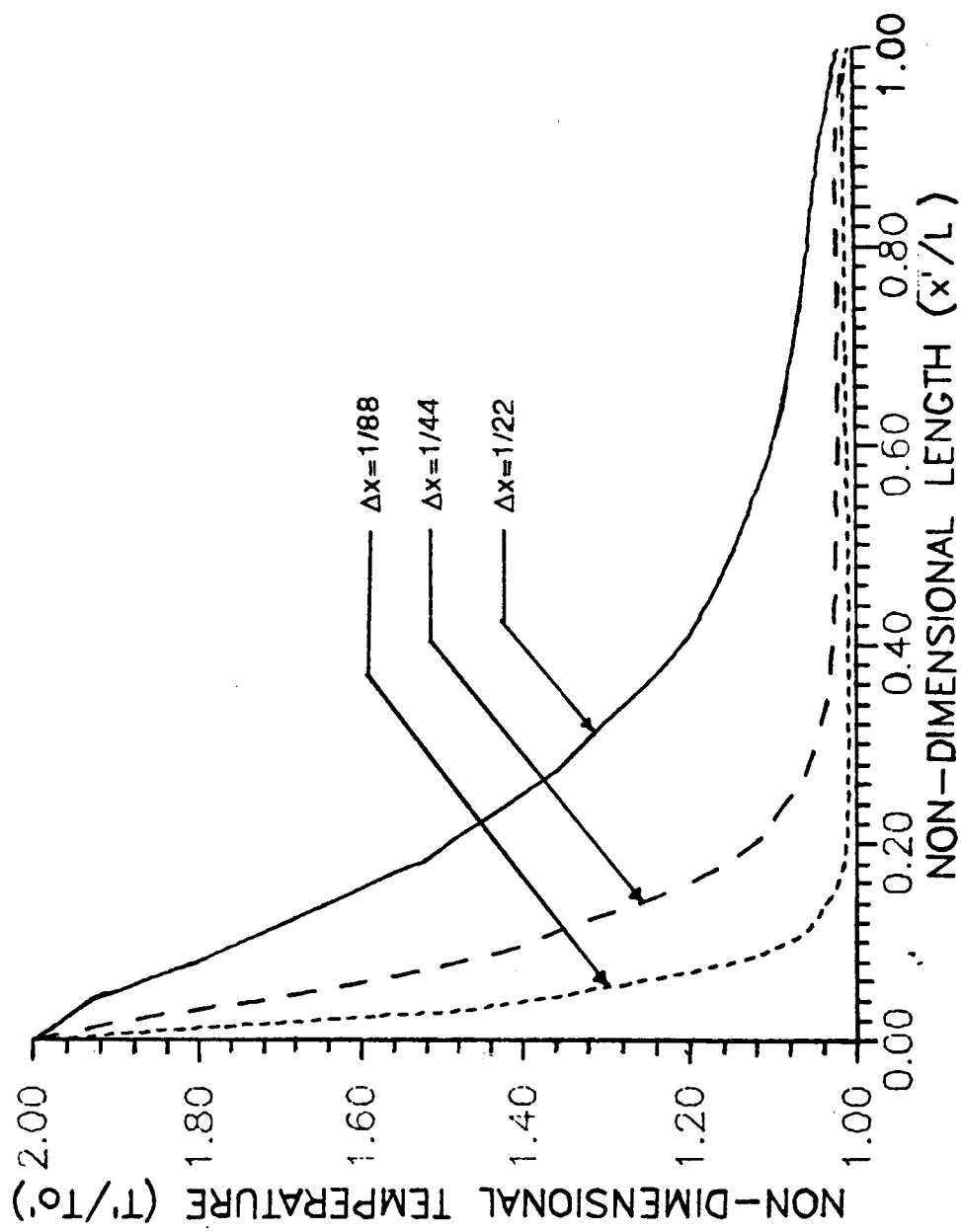


Figure 2.9 Temperature Profiles At 0.1 Seconds ($\Delta t = 0.001875$)

difficult. Thus the a $\Delta x=1/88$ was selected for a large number of the additional computations.

Figure 2.8 shows a comparison of the velocity at the center node versus time for 22 and 44 nodes. The increased smoothness of the velocity field in this implicit scheme can be readily observed. The decrease of the magnitude of the velocity, as the result of doubling the number of nodes, is also apparent.

Figures 2.9 through 2.11 show temperature profiles at 0.1 seconds produced by using time steps of 0.015, 0.005, and 0.001875 units, respectively. In each figure the results for 22, 44, and 88 nodes are included for comparison purposes. The temperature profiles which were seen to rapidly approach steady-state at 22 nodes resemble those of a conduction dominated process for 88 nodes.

Figure 2.12 compares temperature profiles at 0.1 seconds using 44 nodes and time steps of 0.015, 0.005, and 0.001875 units respectively. Figure 2.12 clearly shows a greatly reduced dependence on the size of the time step compared to that of the space step. Thus indicating the validity of the time scaling such that the acoustic wave is not "missed" because of too large of a time step.

Results of the implicit scheme clearly indicate the dependence of the results on the size of the mesh. However, as the mesh size is further reduced the changes between any two temperature profiles produced with two successively smaller mesh sizes decreases.

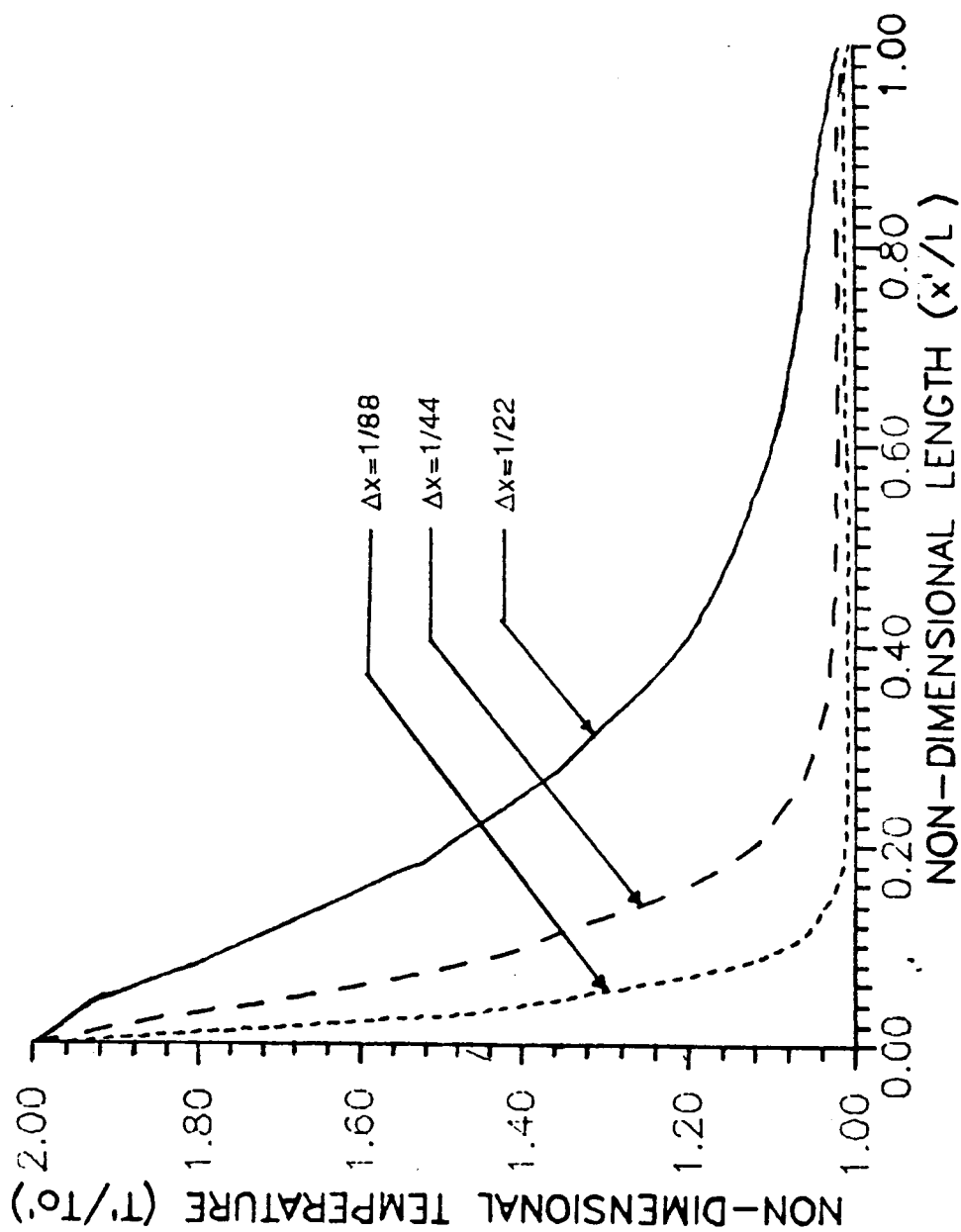


Figure 2.10 Temperature Profiles At 0.1 Seconds ($\Delta t = 0.015$)

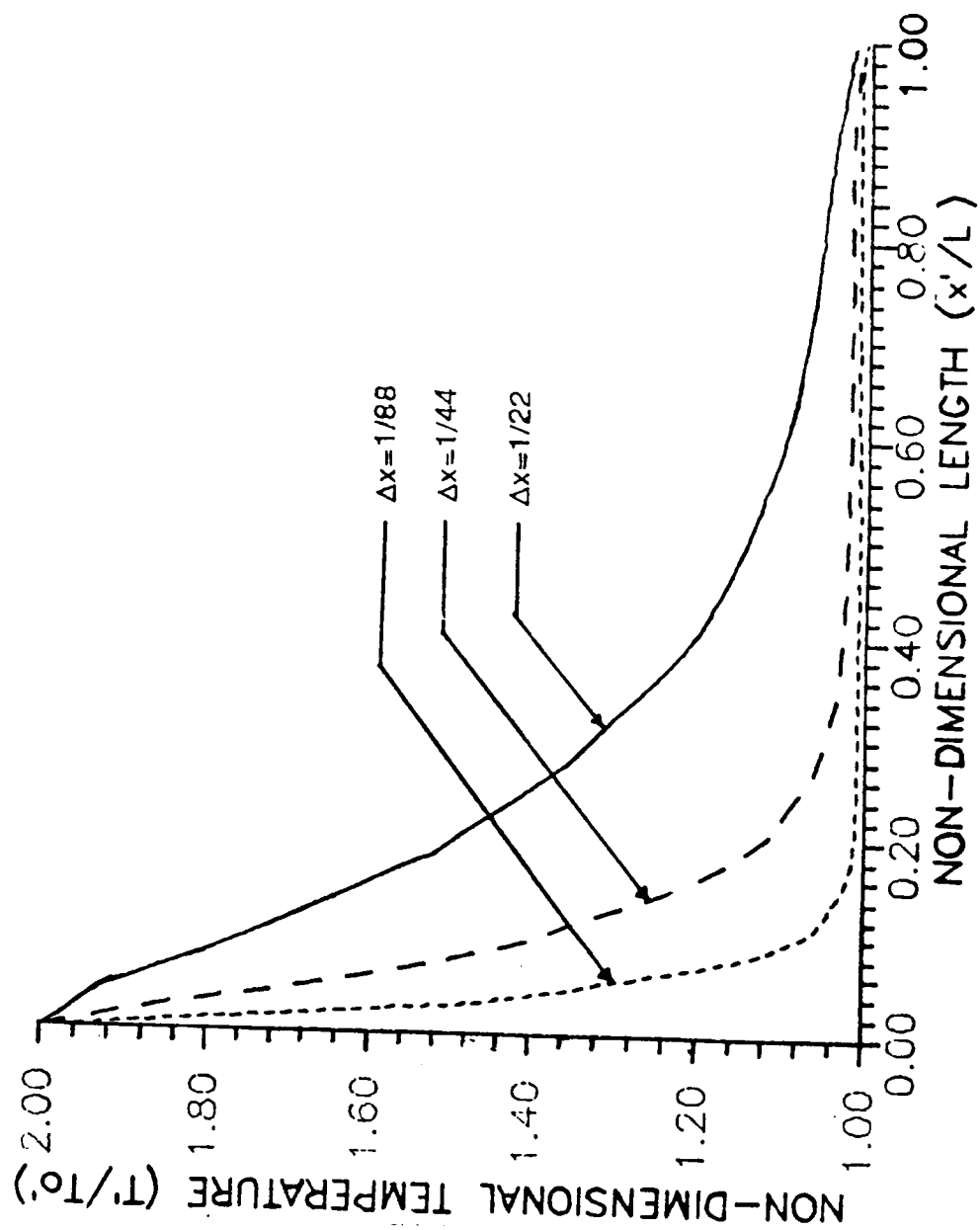


Figure 2.11 Temperature Profiles At 0.1 Seconds ($\Delta t = 0.005$)

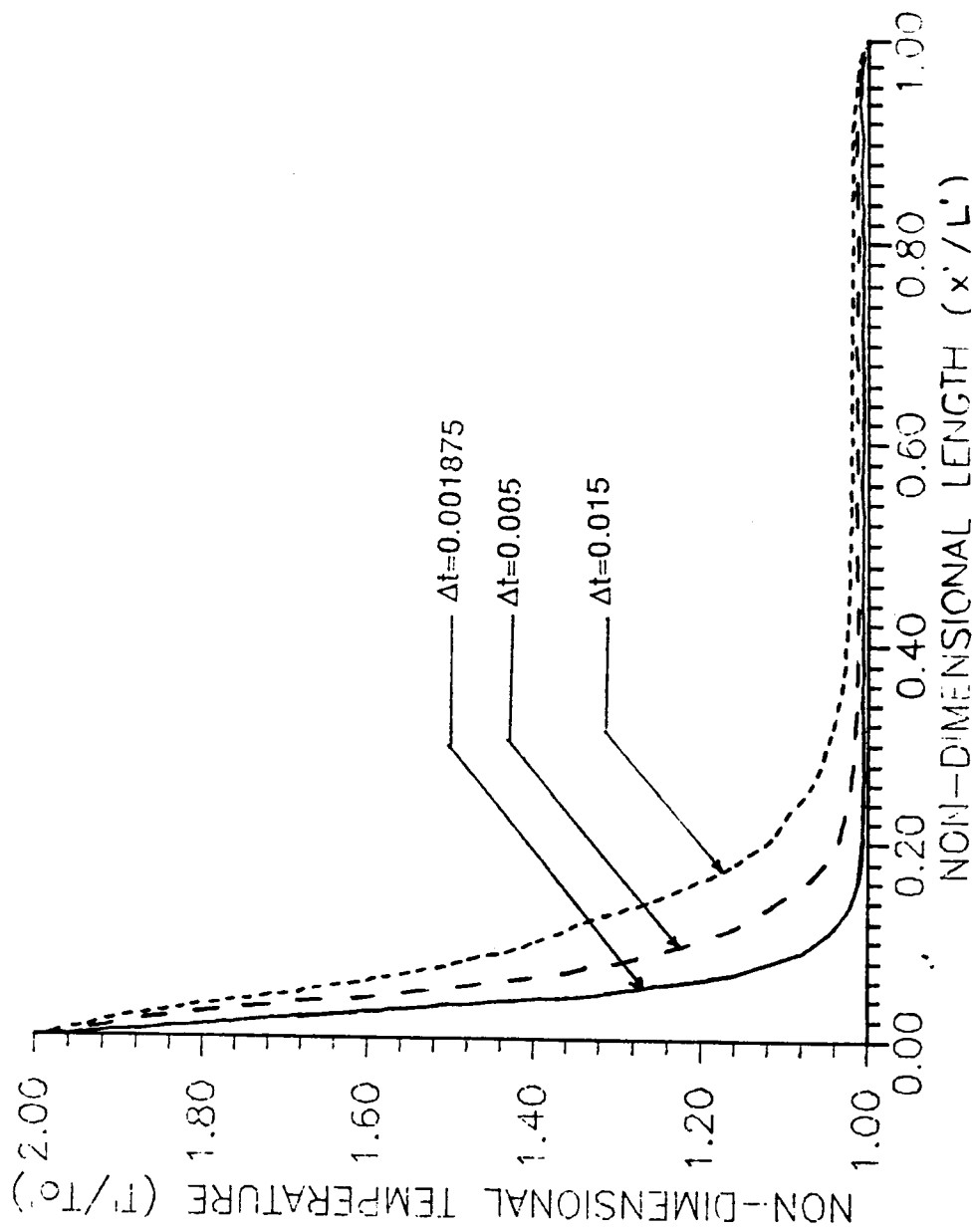


Figure 2.12 Temperature Profiles At 0.1 Seconds ($\Delta x = 1/44$)

Successively smaller meshes were investigated the smallest of which used a $\Delta x=1/156$ and this trend was shown to continue.

Although the use of successively smaller meshes causes the resulting temperature profiles to resemble those of a conduction dominated process, the lower limit of the process is not a rigid conduction solution. The lower limit of the process is one in which the oscillatory component of fluid velocity is zero, but a redistribution of the mass of fluid occurs through a bulk motion which comes as a result of density changes from the State equation.

The changes that occur in the fluid very close to the heated surface during the early stages of the process have an important effect upon the successive heating of other fluid regions. A finer mesh certainly improves the accuracy of the numerical solution near the heated surface. It is therefore assumed that a better result is obtained when a finer mesh is used in the numerical computation.

The nature of the solution process of the implicit model requires that the momentum, continuity, energy equations be solved first for the velocity, density and temperature fields, respectively, and then the state equation be solved for the pressure. The pressure field has a uniquely important role in the calculation of the spatial pressure derivative used in the momentum equation. The computation of this derivative requires the subtraction of two sometimes very similar numbers. Thus the possibility exists that this derivative can become numerically lost. The acoustic nature of the problem makes this possibility very troublesome. The lack of available

experimental data and other reliable numerical models for comparison purposes makes verification of the implicit model difficult. Thus a new numerical scheme is proposed for the solution of the TAC process.

CHAPTER 3

NUMERICAL ANALYSIS AND DEVELOPMENT OF A TAC MODEL

In this chapter a numerical scheme based on the SIMPLER algorithm¹⁵ is developed to model a one-dimensional and a two-dimensional thermal convection process. This algorithm is modified for a fully compressible medium. The accuracy of the results is evaluated by a systematic grid refinement study and the computed flow properties are compared with available experimental data. In the following sections the development and derivation of the finite-difference equations will first be discussed. The method of solution of these equations, the advantages of the algorithm, and the modifications for compressible effects are then examined in the next section. Finally, the accuracy of the model is discussed through a grid refinement study and comparison with experimental data.

3.1 FORMULATION OF THE FINITE-DIFFERENCE EQUATIONS

One-Dimensional Model

The present model is a finite-volume numerical scheme, and the discretization of the governing equations of the TAC process is based on a finite control volume similar to the one shown in Figure 3.1. The finite nodal control volume shown in Figure 3.1 shows the

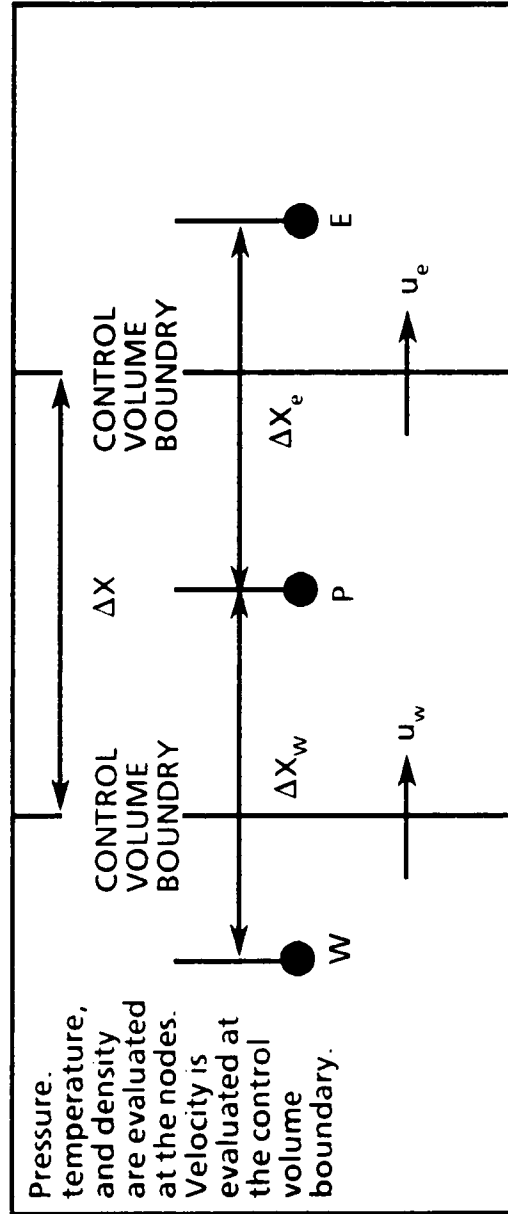


Figure 3.1 Nodal Control Volume Diagram

distribution of major nodes, located half-way between the control volume interfaces, and the distribution of minor nodes. These minor nodes are located at the control volume interfaces and used for the computation of the velocity field.

The mathematical model governing the one-dimensional TAC process was defined in Equations (2.16) to (2.26). In order to illustrate the discretization process the derivation of the discretized energy equation is carried to its final form. The derivation of the discretized Newton's second law, continuity, and the resulting pressure equations is briefly discussed. More detailed discussion of these equations may be found in Ref. [17].

The proper form of the energy equation used in the one-dimensional TAC problem is given by Equation (2.18). Using the continuity Equation (2.16), the energy Equation (2.18) may be cast in the following form:

$$\frac{\partial(\rho T)}{\partial \cdot t} + \frac{\partial(\rho u T)}{\partial \cdot x} = -(\gamma - 1)P \frac{\partial u}{\partial \cdot x} + \left(\frac{\gamma}{\text{RePr}} \right) \frac{\partial^2 T}{\partial \cdot x^2} \quad (3.1)$$

The discretized form of Equation (3.1) is achieved through integration of each of the individual terms across the finite control volume and through a unit time step. This is illustrated in the integration of the unsteady term in the energy equation with Equation (3.2).

$$\int_w^e \int_t^{t+\Delta t} \frac{\partial(\rho T)}{\partial \cdot t} dt \cdot dx = [(\rho T)_p^1 - (\rho T)_p^0] \Delta x \quad (3.2)$$

Focusing attention on the convective term (the last term on the left side of Equation (3.1)) the integration proceeds as follows:

$$\int_t^{t+\Delta t} \int_w^e \frac{\partial(\rho u T)}{\partial \cdot x} dx \cdot dt = [(\rho u T)_e - (\rho u T)_w] \Delta t \quad (3.3)$$

Employing an upwind scheme for the determination of the unknown values of T_e and T_w , we select:

$$\text{if } (\rho u)_e > 0, \quad \text{select } T_e = T_P \quad (3.4)$$

$$\text{if } (\rho u)_e < 0, \quad \text{select } T_e = T_E \quad (3.5)$$

Applying the upwind scheme to Equation (3.3) and introducing the notation $[A, B]$ to denote the greater of quantities A or B, the convective terms become:

$$\{[(\rho u)_e, 0] T_P - [-(\rho u)_e, 0] T_E\} \Delta t - \{[(\rho u)_w, 0] T_w + [-(\rho u)_w, 0] T_P\} \Delta t \quad (3.6)$$

Integrating the conduction term (the last term on the right side of Equation (3.1)) yields:

$$\left(\frac{\gamma}{\text{RePr}} \right) \int_t^{t+\Delta t} \int_w^e \frac{\partial^2 T}{\partial \cdot x^2} dx \cdot dt = \left(\frac{\gamma}{\text{RePr}} \right) \left\{ \left(\frac{\partial T}{\partial \cdot x} \right)_e - \left(\frac{\partial T}{\partial \cdot x} \right)_w \right\} \Delta t \quad (3.7)$$

Linearizing the derivatives in Equation (3.7) gives:

$$\left(\frac{\gamma}{\text{RePr}}\right) \int_t^{t+\Delta t} \int_w^e \frac{\partial^2 T}{\partial x^2} dx dt = \left(\frac{\gamma}{\text{RePr}}\right) \left\{ \frac{T_E - T_P}{\Delta x_e} - \frac{T_P - T_W}{\Delta x_w} \right\} \Delta t \quad (3.8)$$

A simple diffusive-convective balance in a node similar to the one shown in Figure 3.1 can be shown to yield an exact solution in terms of an exponential of a nodal Peclet number¹⁷ defined by Equation (3.9):

$$\text{Pe} = \frac{(\rho u) \Delta x}{\frac{\gamma}{\text{RePr}}} \quad (3.9)$$

An accurate curve fit of this exact solution can be developed in the form of a power law with a nodal Peclet number as its argument:

$$[0, (1 - 0.1|\text{Pe}|)^5] \quad (3.10)$$

This formulation can be shown to yield a good representation of the nodal temperature profile throughout a wide range of nodal Peclet numbers.¹⁷

The discretization of the pressure work term (the first term on the right hand side of Equation (3.1)) proceeds as follows:

$$-(\gamma - 1) \int_t^{t+\Delta t} \int_w^e P \frac{\partial u}{\partial x} dx dt = -(\gamma - 1) P_P (u_e - u_w) \Delta t \quad (3.11)$$

Using the ideal gas equation of state, the nodal pressure is recast in terms of density and temperature. The purpose of this formulation is to treat the pressure work term as a source term.

Collecting terms that multiply each of the variable temperatures and dividing by the time step Δt , the energy equation is recast in the convenient form :

$$a_P T_P = a_E T_E + a_W T_W + b \quad (3.12)$$

with

$$a_P = a_E + a_W + a_P^0 \quad (3.13)$$

and

$$b = -(\gamma - 1) \rho_P (u_e - u_w) \quad (3.14)$$

All of the coefficients in the above equations are defined in Figure 3.4.

The continuity and momentum equations are discretized in a similar fashion as the energy equation. Figures 3.2 through 3.4 show the complete set of discretized continuity (and resulting pressure equations) , momentum, and energy equations, respectively. In the discretization of both the continuity and momentum equations a

Continuity Equation:

$$\frac{\partial}{\partial t} \rho = - \frac{\partial}{\partial x} (\rho u)$$

Finite Difference Equation:

$$(\rho_p^o - \rho_p) \Delta x / \Delta t + [(\rho u)_e] = 0$$

Pressure Equation (from continuity)

$$a_{pp} P_p = a_{ep} P_e + a_{wp} P_w + b_p$$

where:

$$a_{ep} = \rho_e / a_e; \quad a_{wp} = \rho_w / a_{w(t-1)}$$

$$b_p = (\rho_p^o - \rho_p) \Delta x / \Delta t + [(\rho u)_w - (\rho u)_e]$$

$$a_{pp} = a_{ep} + a_{wp}$$

Figure 3.2 Continuity Equation In Discretized Form

Pressure Correction Equation, P' (also from continuity)

$$a_{Pp} P'_p = a_{Ep} P'_E + a_{Wp} P'_W + b_{P'}$$

where

$$a_{Ep} = \rho_e^* / a_e - u_e^* / (T_p + T_E)$$

$$a_{Wp} = \rho_w^* / a_{e(t-1)} + u_w^* / (T_W + T_p)$$

$$b_{P'} = (\rho_p^0 - \rho_p) \Delta x / \Delta t + [(\rho^* u^*)_w - (\rho^* u^*)_e]$$

Figure 3.2 Continued

Newton's 2nd Law:

$$\frac{\partial}{\partial t} (\rho u) + \frac{\partial}{\partial x} (\rho u u) = - \frac{\partial P}{\partial x} + a \frac{L g'}{R' T_o} \rho + \frac{4}{3} \left(\frac{1}{Re} \right) \frac{\partial^2 u}{\partial x^2}$$

Finite Difference Equations:

u_e - Equation:

$$u_e = (a_{ne} u_{ne} + a_{nw} u_{nw} + b) / a_e$$

where:

$$a_{ne} = \frac{|(4/3)(1/Re)|}{\Delta x_e} \{1 - 0.1 |Pe_e|^5\} + || - (\rho u)_e, 0 ||$$

$$a_{nw} = \frac{|(4/3)(1/Re)|}{\Delta x_w} \{1 - 0.1 |Pe_w|^5\} + || - (\rho u)_w, 0 ||$$

$$a_e^o = p_e^o \Delta x / \Delta t; \quad b = a_e^o u_e^o; \quad Pe_e = (\rho u)_e \Delta x / [(4/3)(1/Re)]$$

$$Pe_w = (\rho u)_w \Delta x / [(4/3)(1/Re)]$$

$$a_e = a_{ne} + a_{nw} + a_e^o$$

u* Equation:

$$a_e u_e^* = \Sigma a_{nb} u_{nb} + b + (P_p - P_E)$$

Velocity Correction Equation:

$$u_e = u_e^* + (1/a_e) [P'_p - P'_E]$$

Figure 3.3 Newton's Second Law In Discretized Form

Energy Equation

$$\frac{\partial T}{\partial t} + u \frac{\partial T}{\partial x} = - \frac{(\gamma - 1)}{\rho} p \frac{\partial u}{\partial x} + \frac{\gamma}{\rho} \left(\frac{1}{RePr} \right) \frac{\partial^2 T}{\partial x^2}$$

Finite Difference Equation:

$$a_P T_P = a_E T_E + a_W T_W + b_T$$

where:

$$a_E = \frac{|1/RePr|}{\Delta x_e} \{1 - 0.1 |Pe_{Te}|^5\} + ||-(\rho u)_e, 0||$$

$$a_W = \frac{|1/RePr|}{\Delta x_w} \{1 - 0.1 |Pe_{Tw}|^5\} + ||(\rho u)_w, 0||$$

$$a_P^o = p_p^o \Delta x / \Delta t; \quad b_T = a_P^o T_p^o$$

$$Pe_{Te} = \frac{(\rho u)_e \Delta x}{|1/RePr|}; \quad Pe_{Tw} = \frac{(\rho u)_w \Delta x}{|1/RePr|}$$

$$a_P = a_E + a_W + a_P^o + (\gamma - 1) \rho_P (u_e - u_w)$$

Figure 3.4 Energy Equation In Discretized Form

number of first derivatives are integrated across a nodal control volume, as is the case with the energy equation. In the momentum and continuity equations it becomes necessary to know the value of the velocity at the control volume interfaces. Because of this, it is convenient to utilize a staggered grid system in which the temperature, density, and pressure are calculated at major nodes located in the center of the finite control volumes, and the velocity is calculated at minor nodes located at the control volumes' interfaces.

The continuity equation is further manipulated to develop a pressure equation (for the calculation of the pressure field) and a pressure correction equation (for the correction of the density and velocity). To begin, the pressure equation is developed by dividing the velocity into two distinct parts. The first term $\hat{u}$ is a velocity based on an assumed velocity field. The second term contains the discretized spatial pressure derivative from the momentum equation. These two terms are substituted into the discretized continuity equation. Terms are separated and the source term that appears in the equation has the same form as the discretized continuity equation based on $\hat{u}$. It is this pressure equation that is used to calculate the pressure field.

The pressure correction equation is derived in a similar fashion. Again the velocity is divided into two terms. The first term u^* is the result of adding the $\hat{u}$ velocity and the discretized spatial pressure derivative based on P^* (from the pressure equation). The

second term contains the spatial pressure derivative from the momentum equation in terms of a pressure correction P' . For the pressure correction equation, however, the density is also divided into two terms. The first term ρ^* is based on the P^* pressure field. The remaining term comes from using the pressure correction P' in the State equation. All four of these terms are substituted into the discretized continuity equation in order to solve for the pressure correction P' . The source term in this equation is similar to the discretized continuity equation except here it is based on ρ^* and u^* .

Two-Dimensional Model

The two-dimensional model consists of a rectangular cavity spanning a distance L in the x -direction and a distance H in the y -direction. The top and bottom walls of the cavity are insulated. The left and right vertical walls have constant temperature boundary conditions with the left wall being the source of the heating. This problem is investigated both with and without a gravity field.

The discretization of the equations governing the two-dimensional thermal convection process proceeds in the same fashion as is done for the one-dimensional case. The reader is referred to Appendix A for the mathematical model governing the two-dimensional TAC phenomenon and the full set of discretized equations. The computer code developed for the solution of these equations, 2DTAC.FOR, is listed in Appendix D.

3.2 METHOD OF SOLUTION

The discretized equations shown in Figures 3.2 to 3.4 are used in a computer code, 1DTAC.FOR, for the solution of the one-dimensional TAC problem. This program is listed in Appendix C. The method of solution of these equations is carried out as follows. First a velocity field is assumed. This assumed velocity field is corrected by solving the $\hat{u}$ velocity equation. The $\hat{u}$ velocity equation is the same as the discretized momentum equation except that the spatial pressure derivative has been dropped. The purpose of the $\hat{u}$ equation is to smooth out the assumed velocity field by recalculating the velocity based on its neighboring values. The spatial pressure derivative is then reintroduced as a further correction in the calculation of the proper velocity field.

After the solution of the $\hat{u}$ equation, the pressure equation (that is a modified form of the continuity equation) is solved to obtain the pressure field. This pressure field is treated as an assumed pressure, P^* , and is used to correct the $\hat{u}$ velocity to obtain u^* . The P^* pressure is also used to calculate the assumed density ρ^* using the state equation. If the correct velocity field is assumed, the pressure equation would immediately yield the correct pressure field which would result in a zero value correction.

The values of u^* and ρ^* are used to determine the coefficients and the source term in the pressure correction equation, P' . Then the

values of P' are used to correct the velocity u^* by calculating and adding the spatial pressure derivative to the u^* term. The pressure correction P' is also used to update the pressure P^* which is then used in the state equation to correct the density field. The pressure obtained from the pressure equation, P^* , is not corrected, for the purpose of obtaining the correct pressure field. The energy equation is solved to obtain the temperature field and the discretized continuity equation is used once again to check for a converged flow field. If the calculated flow field satisfies the discretized continuity equation within some small range, then the program is considered to have converged. That is, if the proper flow field is numerically computed then the energy equation will automatically yield the correct temperature field. A flow diagram of the algorithm is shown in Figure 3.5.

The described algorithm is semi-implicit. The values of density, temperature, pressure, and velocity are all calculated based on neighboring values at a future time which requires the method to include some iteration. However, the method of correcting the various velocities (e.g., $\hat{u}$ and u^*) and the assumed density ρ^* is based upon known values from previously solved equations. In this sense the method is a semi-implicit method.

The described algorithm has several advantages over a more straight-forward solution of the momentum, continuity, energy, and state equations. The staggered grid system allows for the spatial pressure derivative from the momentum equation to become the

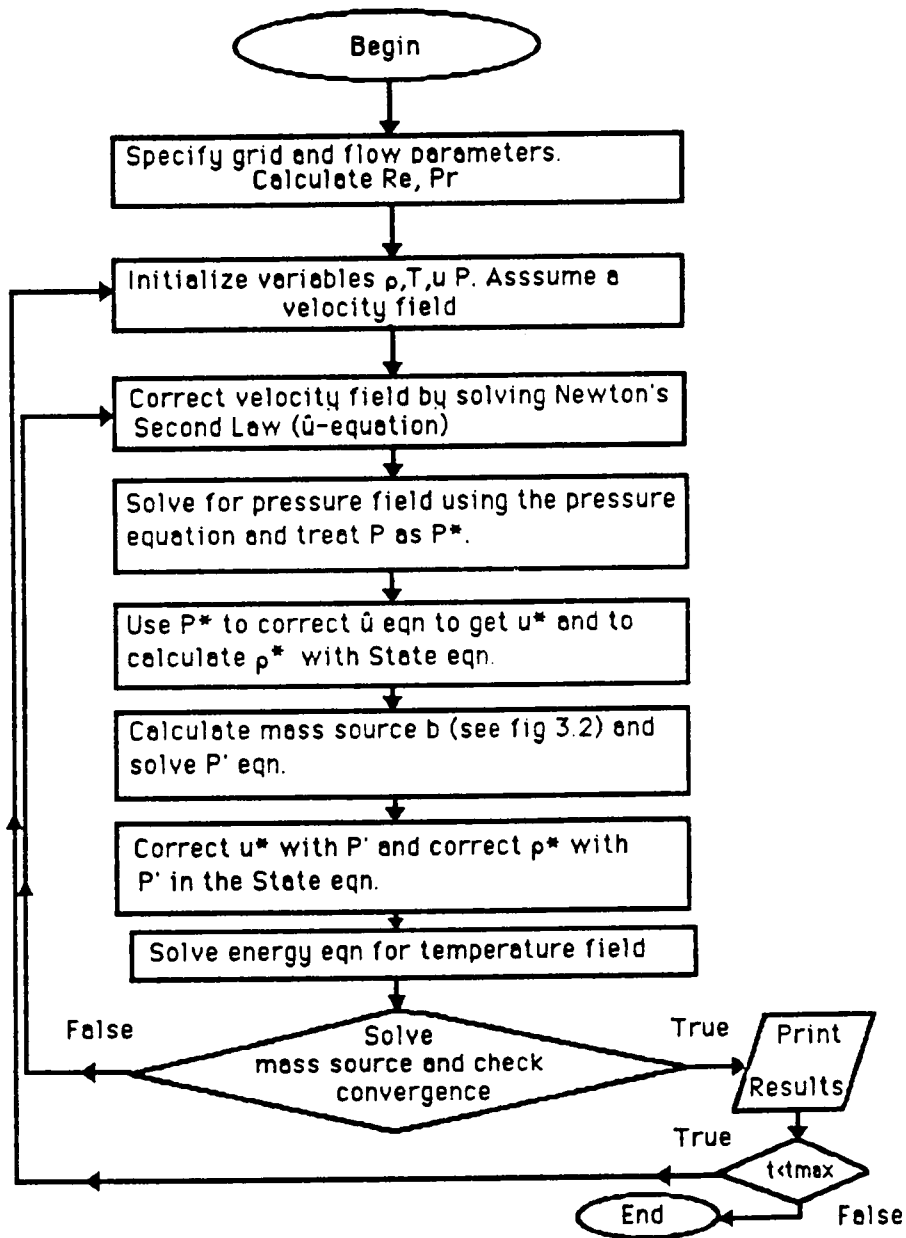


Figure 3.5 Flow Diagram For The TAC Numerical Procedure

natural driving force for the velocity. Secondly, the use of a pressure correction provides a systematic method for the adjustment of the velocity and density. This pressure correction, which contains only a small change in the pressure resulting from heat addition, helps to ensure that the spatial pressure derivative will not become numerically lost in the computation.

The present model contains several modifications and additions to the SIMPLER algorithm developed by Patankar. The most significant is the inclusion of variable density terms in the pressure correction equation. Another major modification is the computation of an assumed density ρ^* using the state equation and the assumed pressure P^* . The use of a corrected pressure to calculate a corrected density is also of importance. It is believed that these modifications make it possible to correctly include the effects of compressibility, which is a key feature in the TAC problem. During early stages of the research the algorithm was shown to exhibit convergence problems when either the assumed pressure P^* was not used to calculate the assumed density ρ^* or the corrected pressure was not used to correct the ρ^* term. The sensitivity of the program to these modifications is presumably an evidence of the strong link between pressure and density that is an important part in the TAC process.

The solution of the discretized equations comprising the two-dimensional TAC model is the same as for the one-dimensional case with one exception. The introduction of a second space dimension into the problem requires an additional momentum equation. Thus

whenever the $\hat{u}$ or u^* velocity is calculated a corresponding $\hat{v}$ and v^* velocity is calculated, and whenever the u^* equation is corrected the v^* velocity is also corrected. Thus there is no actual change in the algorithm, however the additional velocity equations and the additional terms present in the continuity, and energy equations result in increased computer run times.

3.3 ERROR ANALYSIS

The accuracy of the present numerical model and its dependence on the space and time mesh size is examined through a grid refinement study similar to the previous case discussed in Chapter 2. Furthermore, a comparison with available experimental data is also attempted to verify the accuracy of the present numerical model.

One-Dimensional Model

Figure 3.6 shows a plot of midpoint velocity as a function of time for a fixed spatial step, $\Delta x = 1/88$, using several values for time increment. The physical occurrence of the velocity wave is indicated by examining its period of the oscillation. The period of oscillation for the velocity wave computed by $2.0/(\gamma)^{1/2}$ is 1.69. The period of oscillation of the velocity wave shown in Figure 3.6 is approximately 1.70. This confirms the physical basis of the numerically computed velocity field. The results of Figure 3.6

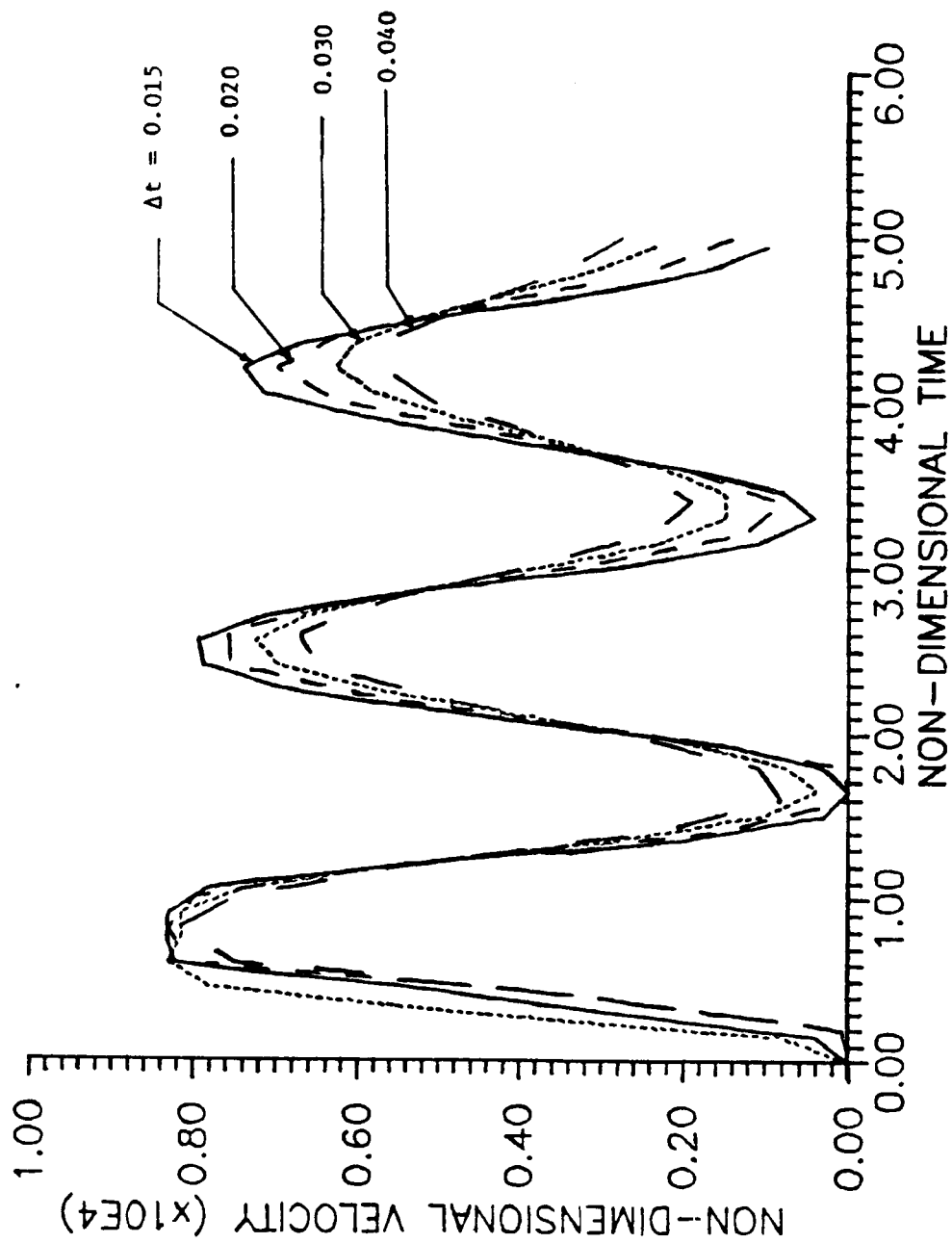


Figure 3.6 Unsteady Velocity At Midpoint For Various Time Steps ($\Delta x=1/88$)

indicate that as the time increment decreases the amplitude of the velocity increases. Decreasing the time increment below $\Delta t=0.015$ proved difficult because of problems with numerical convergence. However, Figure 3.6 indicates increasingly smaller changes in velocity amplitude with increasingly smaller time increments. Thus a time increment of $\Delta t=0.015$ was chosen as a convenient and acceptable value for all subsequent calculations.

A similar approach was used in selecting the proper space increment. Figure 3.7 shows midpoint velocity versus time for a constant time increment, $\Delta t=0.015$, and various space increments. Decreasing the mesh size initially showed significant change in the velocity amplitude. However, as the mesh size was further decreased, the changes in the velocity amplitude decreased. Space increments smaller than $\Delta x \leq 1/92$ were not used because of numerical convergence problems. Thus a space increment of $\Delta x=1/88$ was chosen as a reasonable value and used in all subsequent computations except when stated otherwise. Please refer to Appendix E for a chart which shows the various space and time increments used in the evaluation of the one-dimensional program.

The semi-implicit method discussed requires a small parameter, ϵ , to be used in the convergence criterion for the iterative process. Figure 3.8 shows midpoint velocity as a function of time with varying values of the convergence criterion parameter. A wide range of values of ϵ were used and the results indicated no

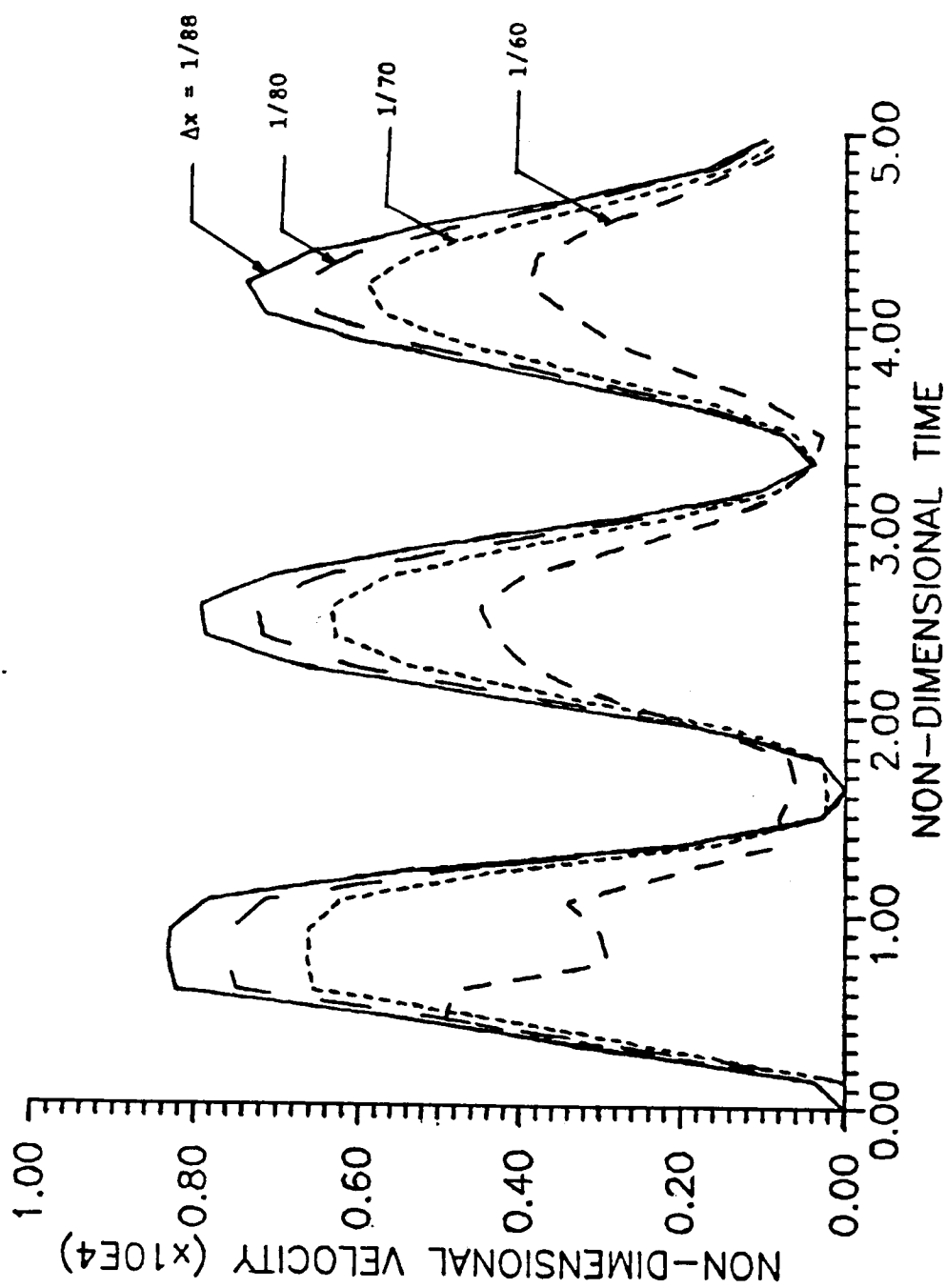


Figure 3.7 Unsteady Velocity At Midpoint For Various Mesh Sizes ($\Delta t=0.015$)

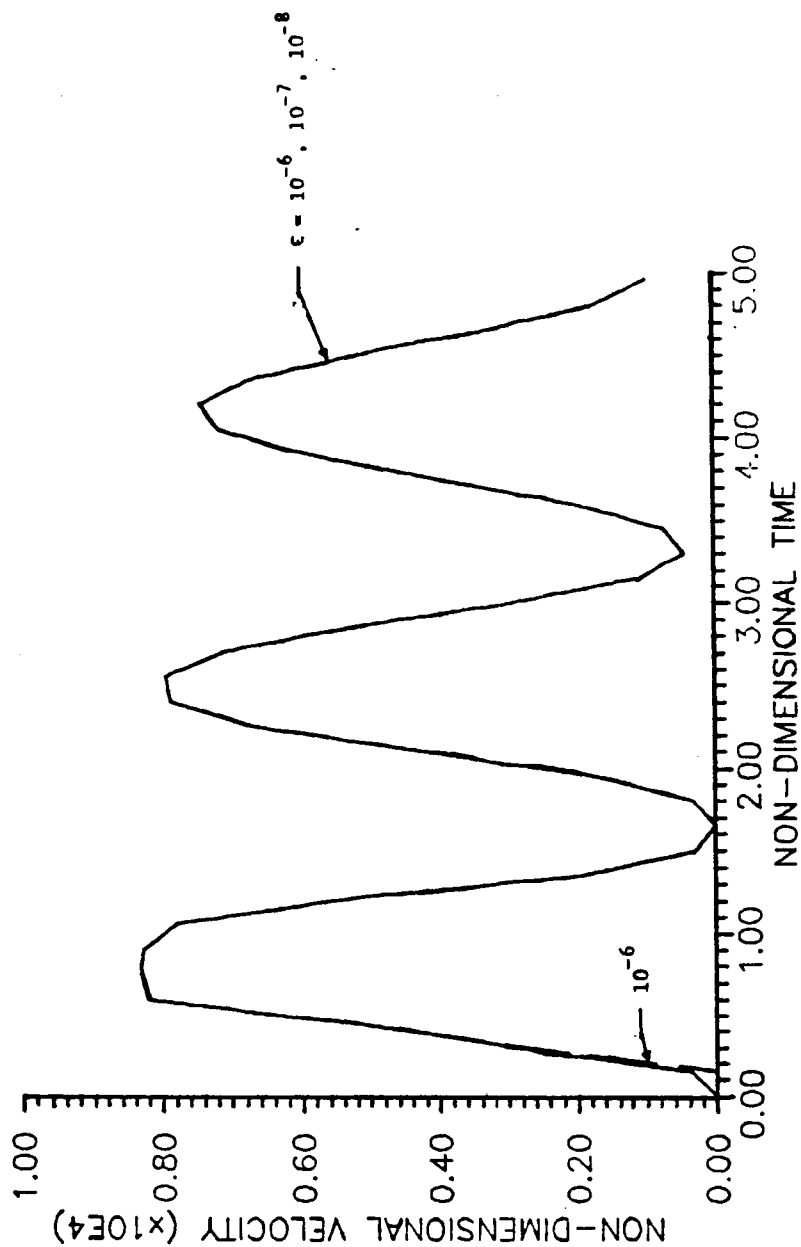


Figure 3.8 Unsteady Velocity At Midpoint For Various Values Of Convergence Criterion
 $(\Delta x = 1/88; \Delta t = 0.015)$

significant dependence on the size of ϵ as shown in Figure 3.8. Thus a value of $\epsilon=10^{-7}$ was chosen for subsequent computations.

Evaluation of the accuracy of the numerical results with experimental data proved somewhat difficult because of the lack of good and accurate experimental data. One of the few available sources of experimental data available is in a study by Salah-Eddine¹. In this study air temperature profiles were measured in a cylindrical configuration using thermocouples. Experiments were conducted in gravity and zero-gravity environments. Zero-gravity experiments were made possible by drop testing the apparatus and limited to experimental times of 0.2 seconds. The experimental conditions also limited the measurements of temperature at a few select locations along the axis of the cylinder.

The nature of temperature measurement by thermocouples in this experiment involved radiation absorption by the thermocouples. This radiation-induced error was computed numerically and the resulting experimental data adjusted accordingly. The result is an uncertainty which would not be present if the radiation-induced error had been determined experimentally. The hot wall temperature in the experiment ranged from a non-dimensional value of 1.0 to a high value of 3.3. The time-averaged wall temperature was also larger than that used in the numerical model. Also, the temperature was measured slightly off the axis of the cylinder. Thus the geometry and boundary conditions of the experiment do not match those of the presented numerical model.

A set of experimental data, which is still under an ongoing study is available for qualitative comparison.¹⁹ The experiments were performed in a gravitational environment with air as the fluid medium. Boundary temperatures in the experiment varied from a non-dimensional value of 1.0 to a high of 1.56. Thus the time-averaged value remained lower than that of the present numerical study. A Schmidt-schlieren system using a He-Ne laser was shown to eliminate radiation-induced errors in the temperature measurement.

Even with the uncertainties already discussed, it is still of some value to compare the computed results with those in Refs. [1] and [19]. Figure 3.9 shows a comparison of the numerically-computed temperature profiles with those two sets of experimental data. The results show large differences in temperature at longer times in the case of Ref. [1]. These differences can probably be attributed to various differences between the numerical and experimental conditions stated earlier. Comparison with Ref. [19] show discrepancies at shorter times which are probably due to the much slower rise in the wall temperature in the experiments.

Two-Dimensional Model

Two cases of the two-dimensional model were evaluated. The first was a simulation of a one-dimensional TAC process in a square cavity 6.0"x6.0" filled with air. The one-dimensional simulation was

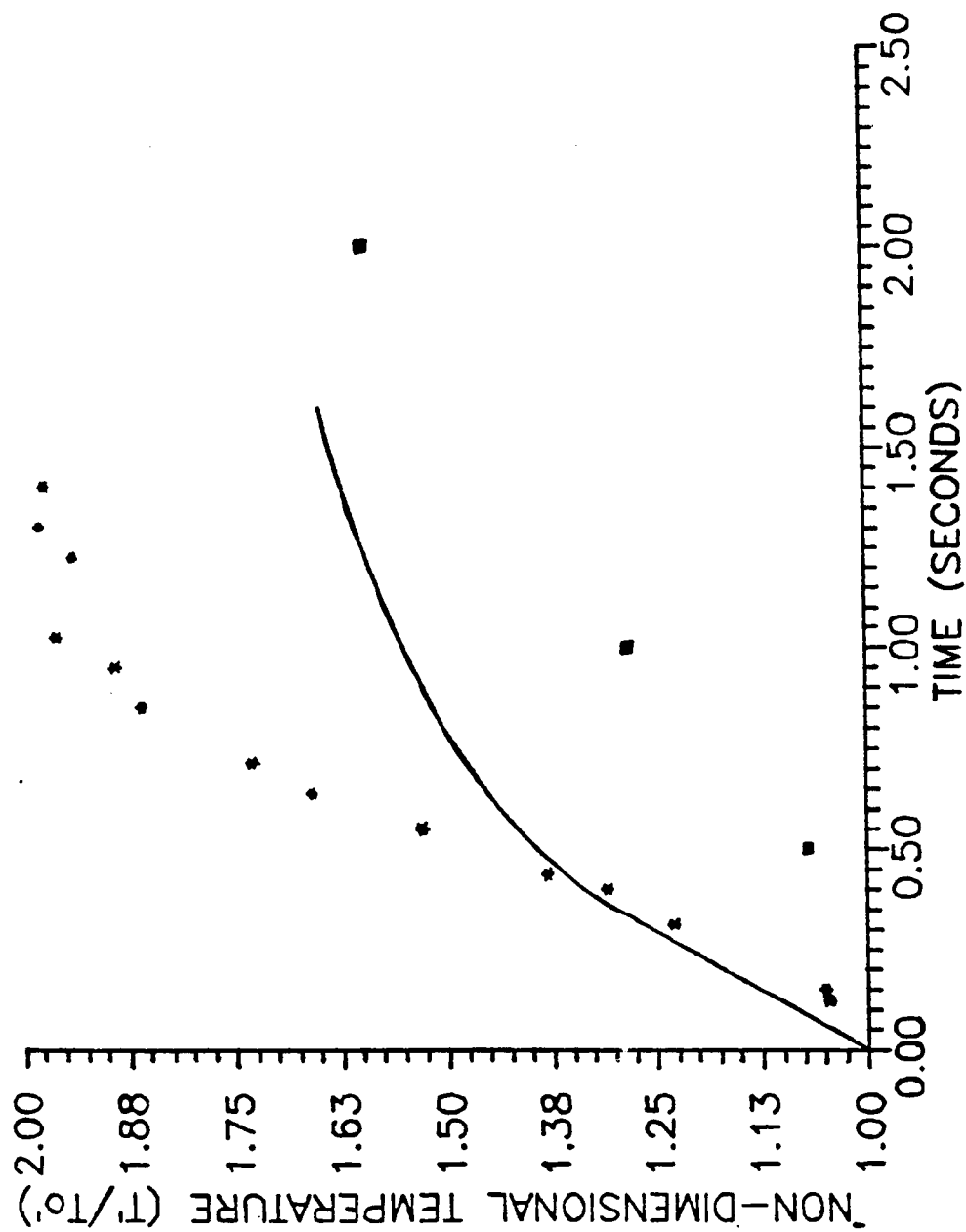


Figure 3.9 Comparison Of Numerical Results (—) With Experimental

Results (Ref [1]:*,Ref [18]:■)

accomplished by insulating the top and bottom walls of the container and evaluating the process in a zero-gravity environment. Constant temperature boundary conditions were used on the left and right vertical walls with the source of the heating coming from the left wall.

The second case was a simulation of the two-dimensional model presented by Spradley and Churchill⁹. This model consisted of a square cavity 1.0"x1.0" filled with helium. The top and bottom walls of the container were insulated, and constant temperature boundary conditions were used on the left and right vertical walls. Again the source of the heating came from a high constant temperature ($2T_0$) on the left wall. The inclusion of a gravity field in this problem resulted in the evaluation of both TAC and natural convection effects. Mesh refinement studies were performed on both cases. A representative discussion of the mesh refinement study performed on the first case follows.

Figure 3.10 shows the midpoint, x-direction, velocity as a function of time for a fixed mesh, $\Delta x=1/100$; $\Delta y=1/100$, using several values of time increment. The results indicate no significant dependence on the time increment below a value of $\Delta t=0.02$. However further research indicated that using a time step on the order of $\Delta t \approx 0.001$ results in an unrealistic velocity field with an increasing amplitude as time progresses. This is not surprising behavior since a time step, $\Delta t \approx 0.001$, requires approximately 1700 steps to reach a dimensionless time of 1.7 (roughly the period of oscillation for the

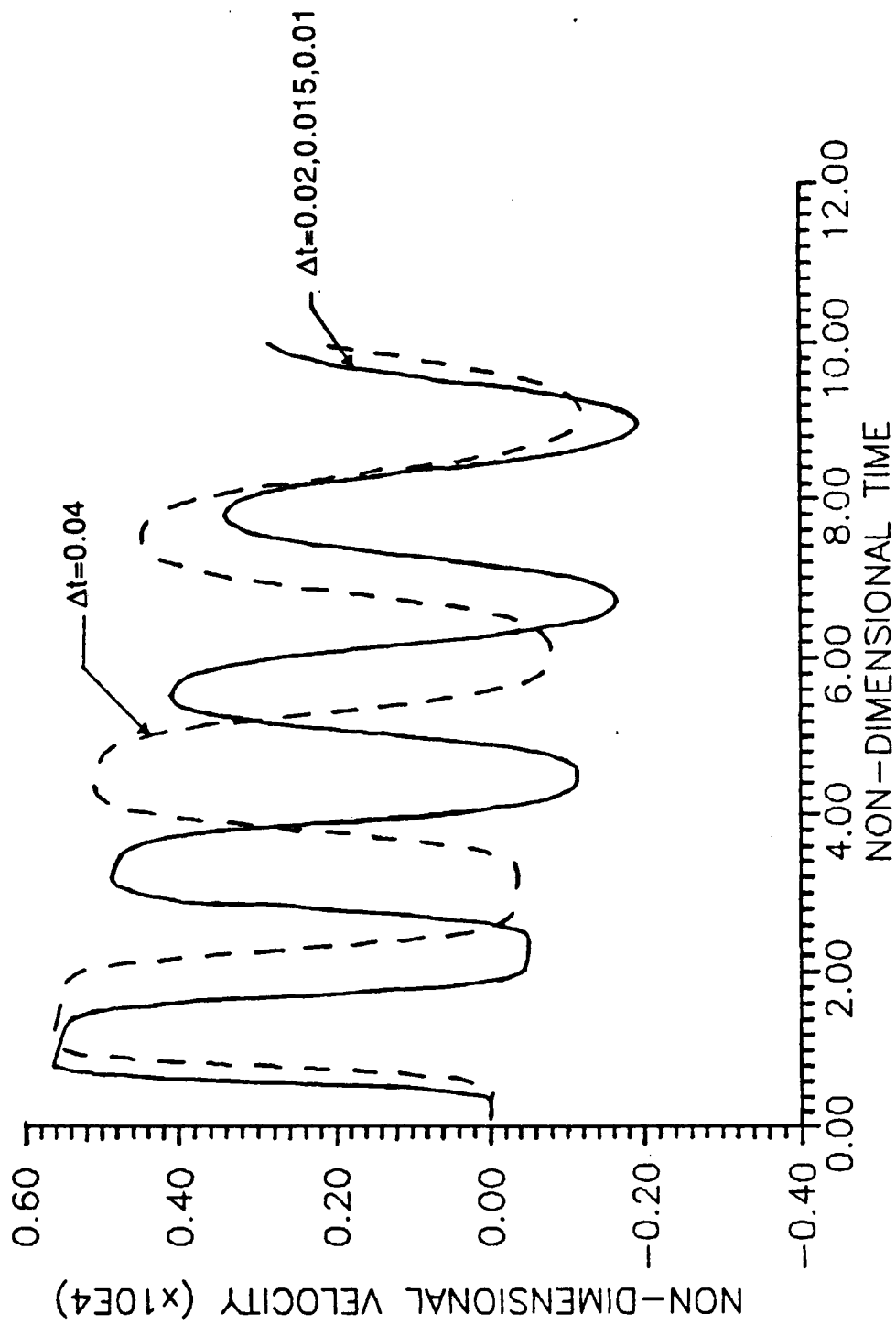


Figure 3.10 Unsteady Velocity At The Center For Various Time Steps (Mesh Size 100x100)

acoustic wave). Thus numerical errors have ample opportunity to develop and overshadow any meaningful computations. On the other hand as the time increment is increased, $\Delta t \approx 0.1$, no acoustic velocity wave is observed. Thus indicating that important information concerning the acoustic velocity is being "missed" by such a large time increment. A time increment of $\Delta t = 0.01$ was chosen as a reasonable and accurate value for further computations.

Figure 3.11 shows the dependence of the acoustic, x-direction, velocity on the selected mesh size. Using a constant time increment, $\Delta t = 0.01$, the number of nodes was increased from 50×50 to 100×100 . Decreasing the mesh size shows an increase in the velocity amplitude similar to the behavior of the one-dimensional model. The magnitude of this velocity increase decreases with successive decreases in mesh size. The tremendous increase in computer run times for meshes smaller than 100×100 proved too costly and time consuming for any practical value. Thus meshes larger than 200×200 were not examined and a mesh of 100×100 was chosen for subsequent calculations.

The sensitivity of the numerical results of the two-dimensional model to convergence criterion, ϵ , is examined in Figure 3.12. The plot of midpoint, x-direction, velocity versus time for a mesh of 100×100 and a time step of $\Delta t = 0.01$ is shown for various values of convergence criterion. The results clearly indicate that there is no significant sensitivity on the computed results to the convergence

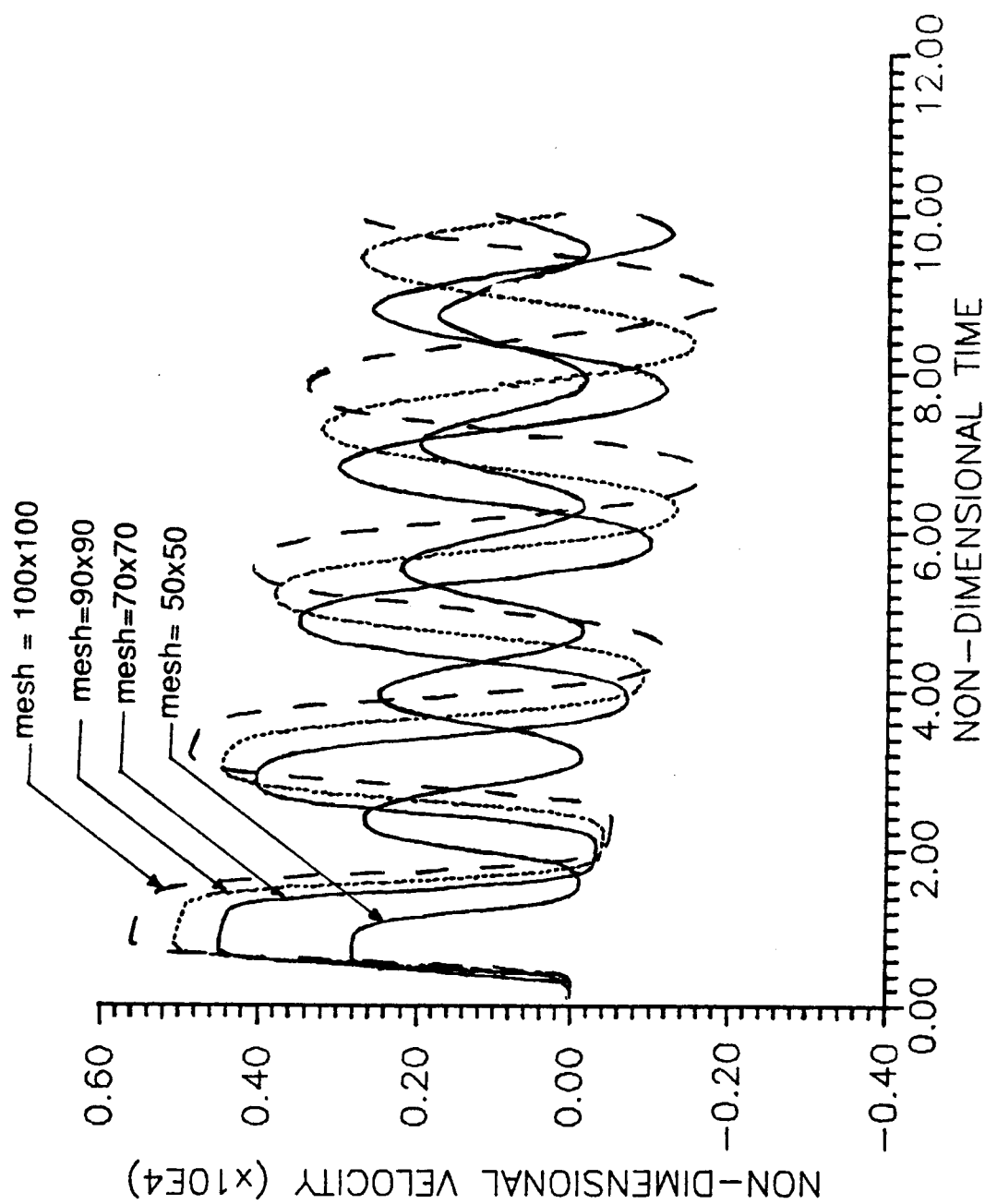


Figure 3.11 Unsteady Velocity At The Center For Various Mesh Sizes ($\Delta t=0.01$)

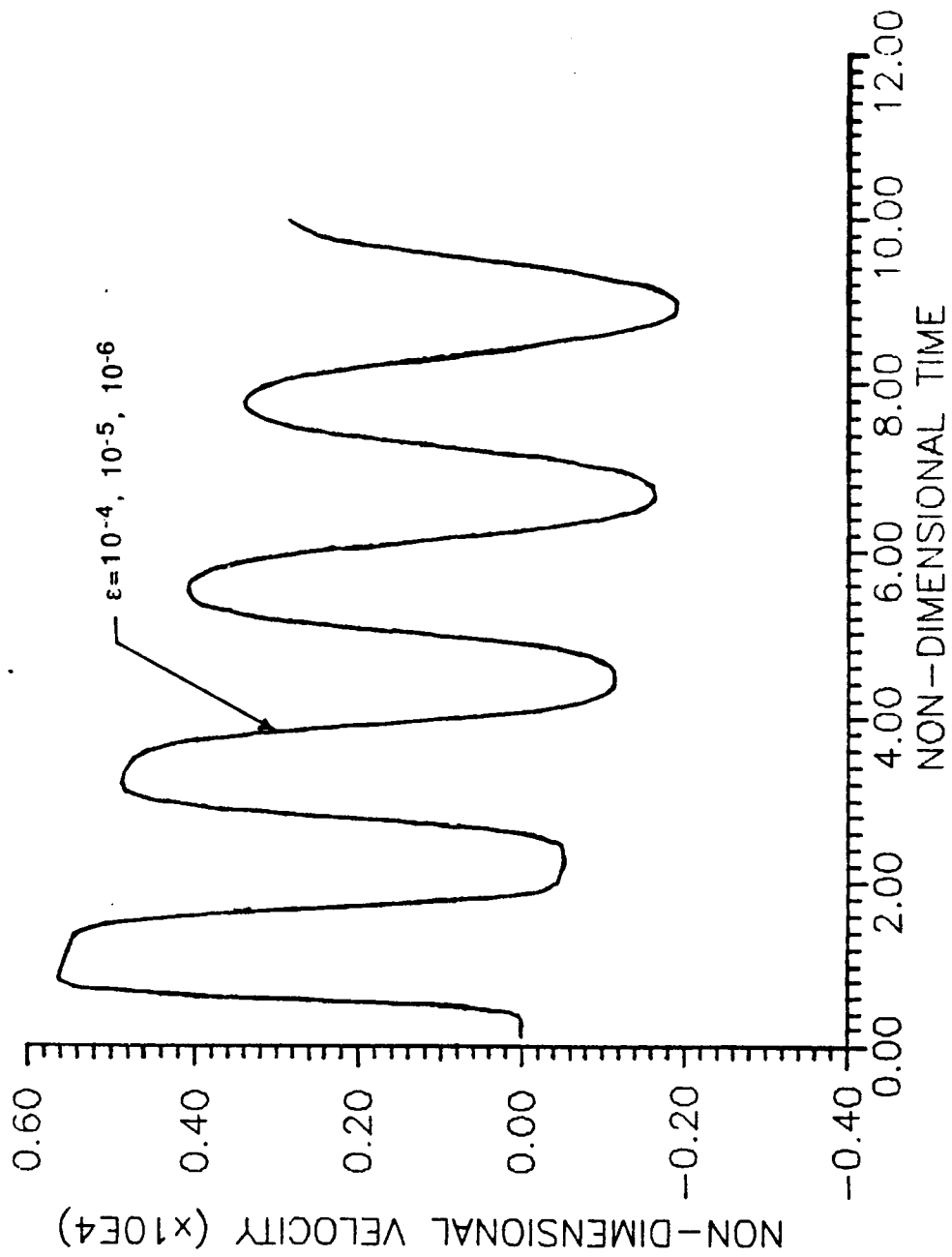


Figure 3.12 Unsteady Velocity At The Center For Various Values Of Convergence Criterion
 (Mesh : 100x100; $\Delta t = 0.01$)

criterion, and as a result a value of 10^{-4} was chosen as a reasonable value which would speed computations.

CHAPTER 4

RESULTS AND DISCUSSION

In this chapter the computed results of both the one-dimensional and two-dimensional numerical schemes are presented for the unsteady flow velocity, fluid temperature and density. The one-dimensional model is evaluated for the effect of fluid viscosity on the velocity field. Also the computed temperature fields are compared with the results of Spradley, et al.⁸ and the perturbation scheme of Rhadwan and Kassoy¹².

The results of the two-dimensional model are presented the two-cases previously discussed. The first case is designed to simulate a one-dimensional heat flow and is used for comparison purposes and verification of the two-dimensional model. The second case uses the same boundary conditions and fluid as used by Spradley and Churchill⁹ the two results are compared and discussed.

4.1 ONE-DIMENSIONAL RESULTS

The fluid velocity field is examined at various locations from the heated wall. The time dependance of the midpoint velocity is shown in Figure 4.1, and the oscillatory nature of the fluid velocity is clearly shown. The period of the oscillations is approximately 1.7 dimensionless time units which compares well with the ideal value

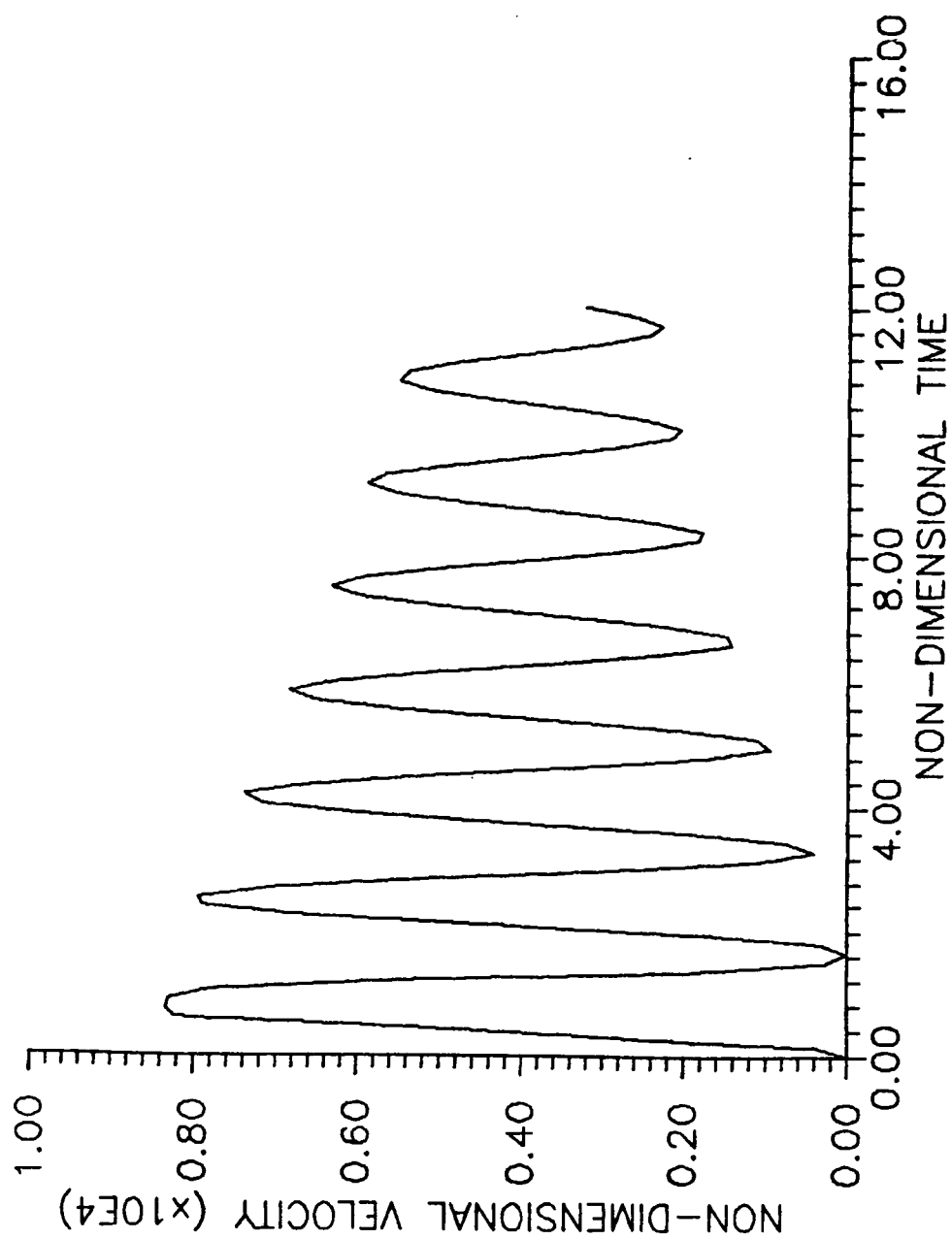


Figure 4.1 Unsteady Velocity At Midpoint Between the Plates

of $1.69 (2.0/\gamma^{1/2})$, and indicates the acoustic nature of the problem. The driving mechanism for the velocity field is a step change in the wall temperature to twice its initial value.

The time dependence of the velocity field is also shown in Figures 4.2 and 4.3. In figure 4.2 velocity profiles are shown at dimensionless times of 0.3, 0.5, 0.6 and 1.0. A wave front is observed to traverse the length between the heated boundary and the opposing wall. This rather sharp wave front is already well established by the dimensionless time of 0.3 and is observed to have an amplitude slightly higher than 8×10^{-5} . Figure 4.3 shows the continuation of this wave at times of 1.0, 1.5 and 3.0. While there exists a net positive flow away from the heated surface the wave front begins to decay. By a dimensionless time of 3.0 a flow reversal has occurred which covers roughly 40% of the flow field and continues to grow in the region away from the hot wall. Only in the region very near the heated surface does the velocity remain large.

The decay of the flow field is clearly shown in Figure 4.4 which focuses attention on the midpoint velocity over a larger time span. The substantial decay in the velocity amplitude is illustrated by a 54% decrease from its initial value in a period of $\Delta t = 25$. This represents an average deceleration of 1.8×10^{-3} . For air initially at atmospheric pressure and room temperature this represents a deceleration of 1.72 cm/sec^2 .

We like to investigate now the importance of fluid viscosity in the attenuation of the velocity field. The role of fluid viscosity in

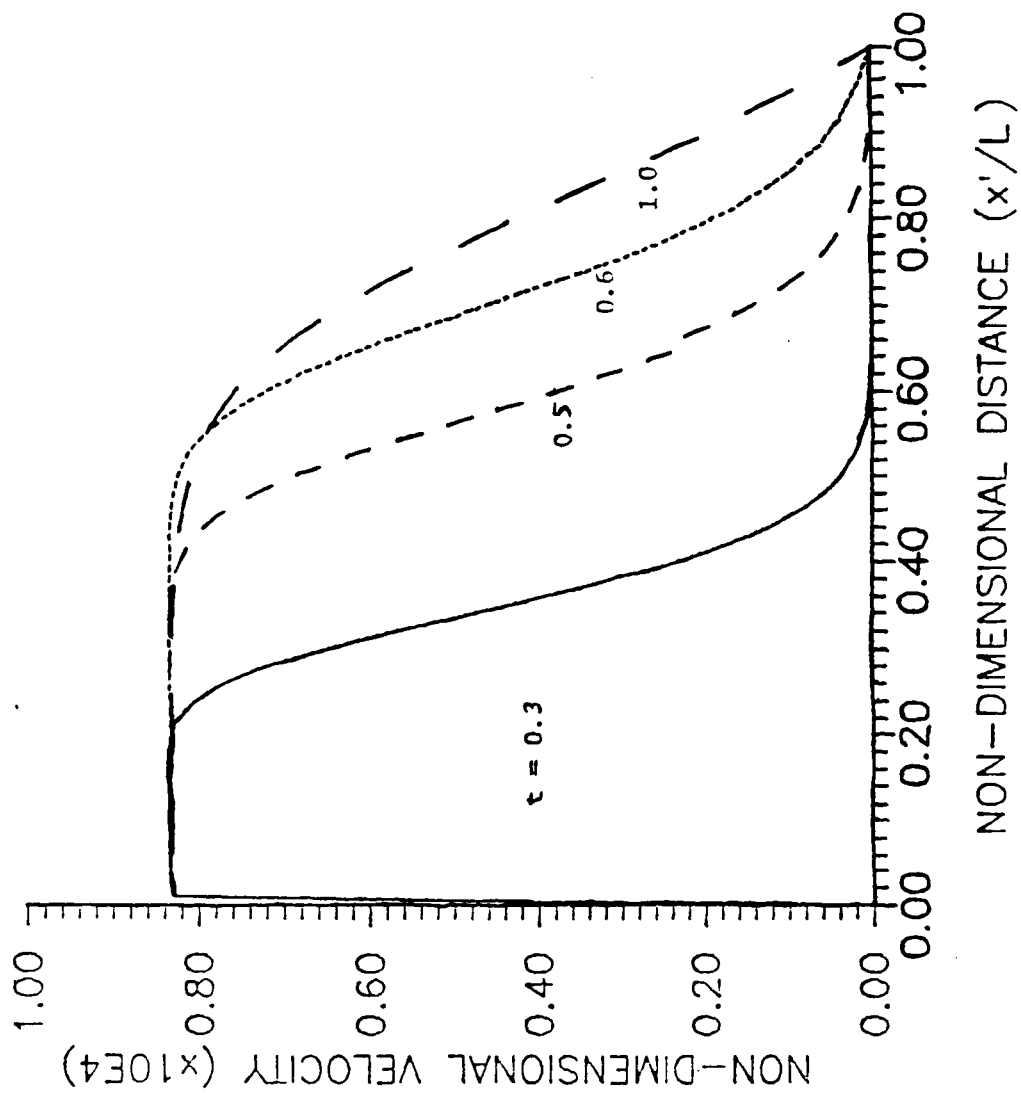


Figure 4.2 Velocity Distribution At Various Times

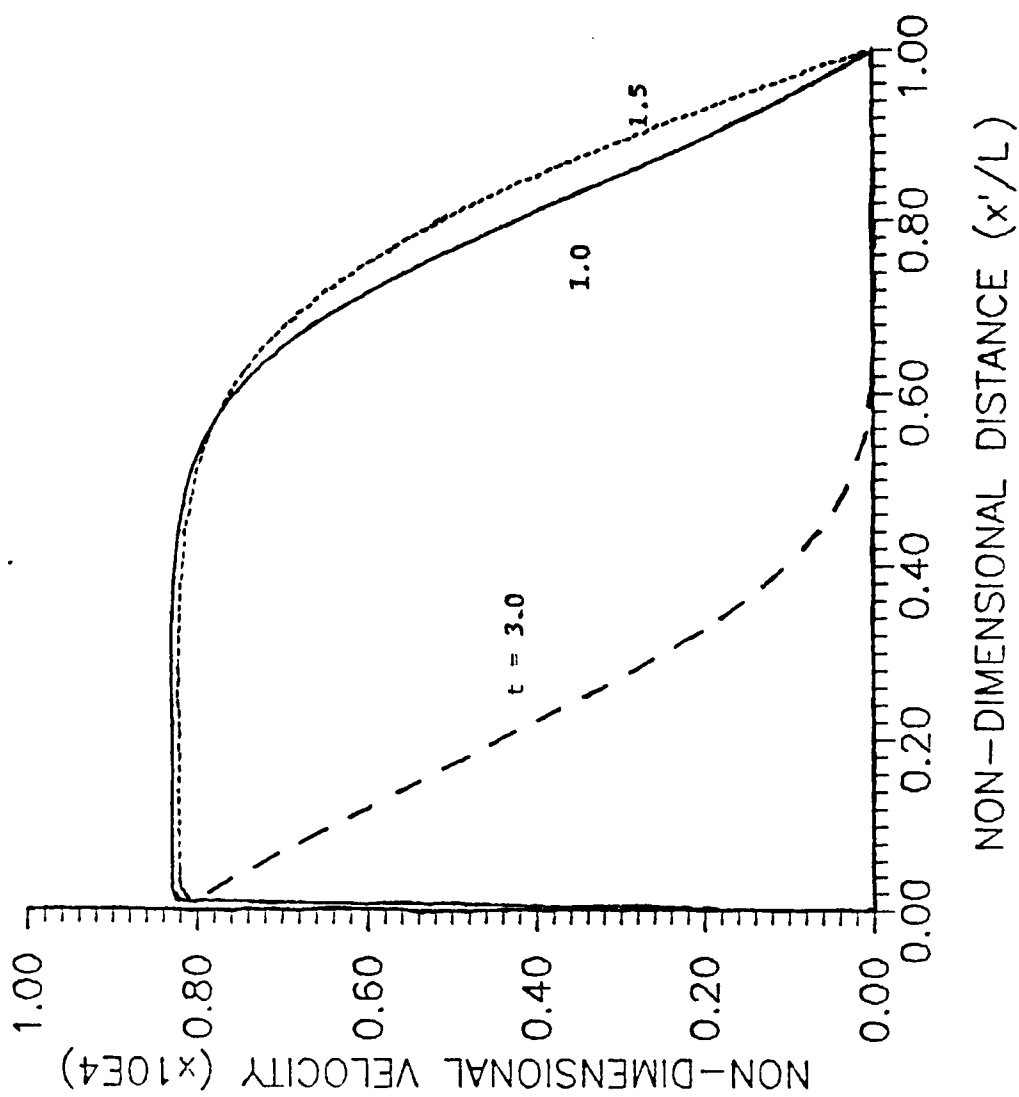


Figure 4.3 Velocity Distribution At Various Times

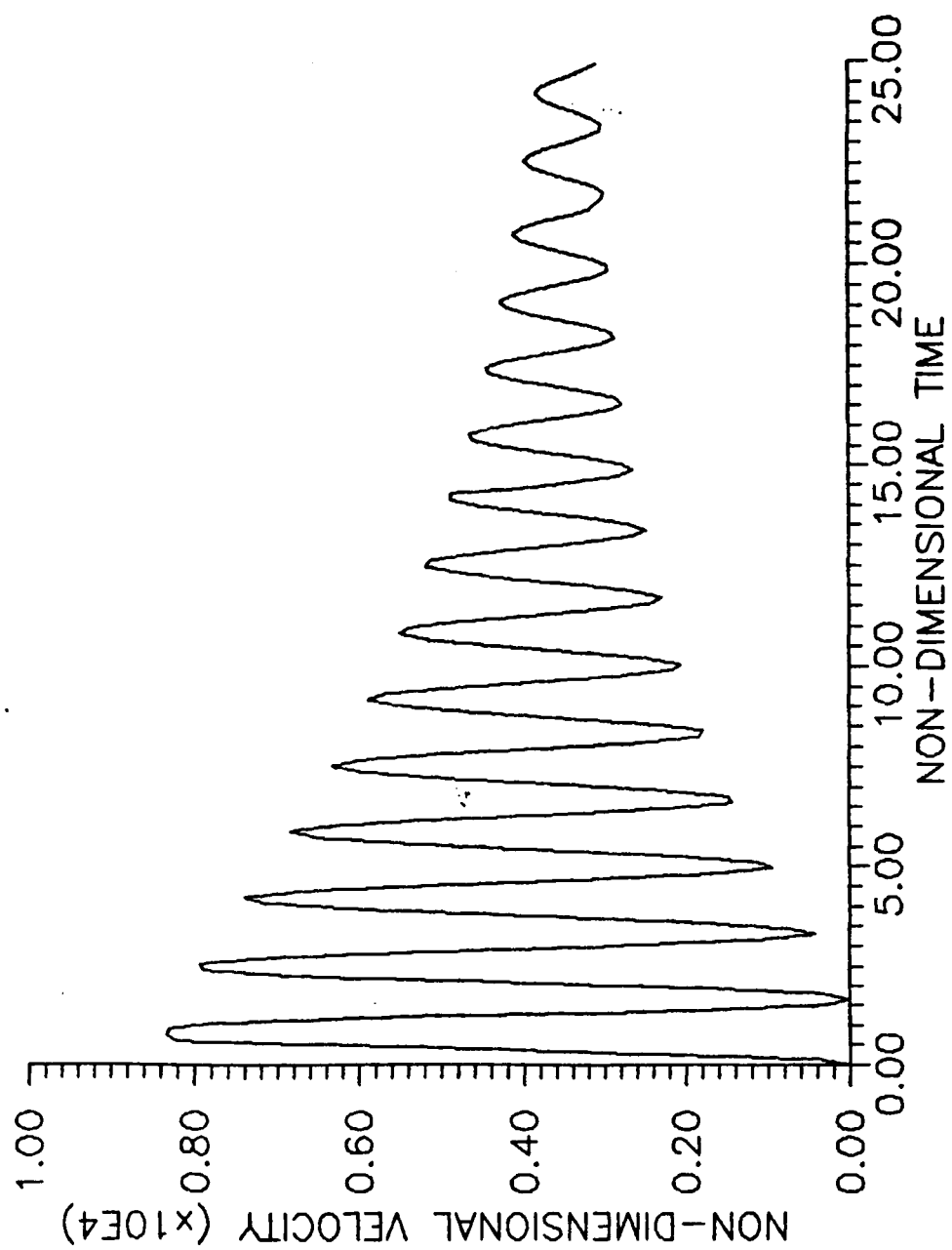


Figure 4.4 Velocity Decay At The Midpoint

the damping of the velocity field is not expected to be significant, and for the purpose of verification the velocity field was computed for various values of Reynolds number corresponding to a 6 order of magnitude change in the kinematic viscosity coefficient. The results of this investigation, which are shown in Figure 4.5, indicate only small changes in the wave profile even with these large decreases in the fluid viscosity.

The rapid decline in the flow velocity seems not to be due to viscous effects. Rather, the velocity decay appears to be dependent upon the nature of the wall condition and is also probably influenced by the redistribution of fluid mass as a result of the interaction between the temperature and velocity fields. With the progression of time the imposed constant temperature wall condition provides a declining rate of heat transfer from the wall. This will be shown to be true and discussed in more detail. The redistribution of fluid mass which is brought about by the change in the density field also helps to diminish the effective heat transfer from the wall. The declining rate of heat transfer to the fluid is probably the major factor in the decay of the flow velocity.

The progress of the temperature field as a function of time can be observed in Figure 4.6. The non-dimensional temperature profiles within the fluid are shown at real times of 0.5, 1.0, and 1.5 seconds. These temperature profiles display the behavior of a conduction-dominated process where a relatively small region experiences significant rise in temperature while a large region remains

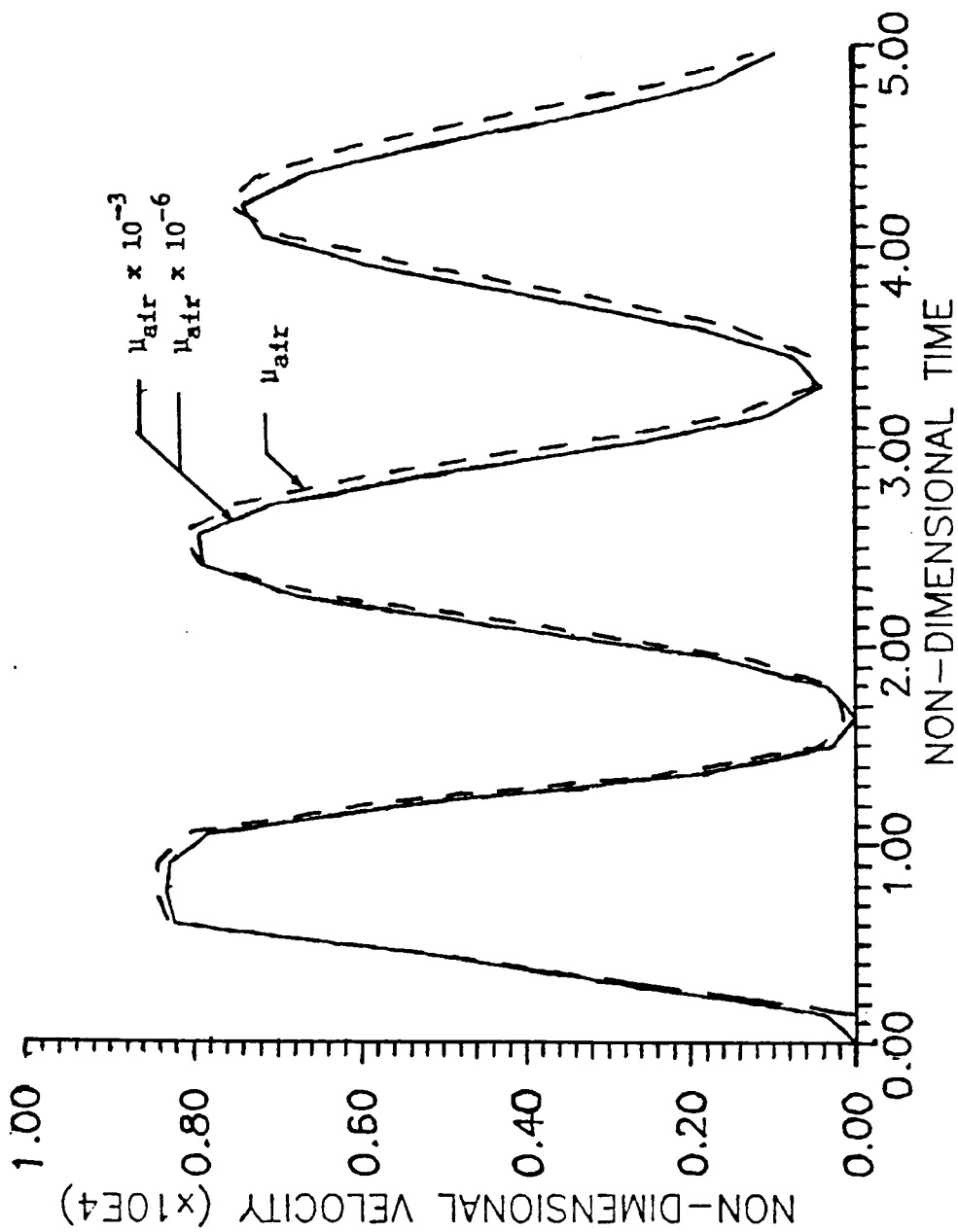


Figure 4.5 The Effect Of Viscosity On Midpoint Velocity

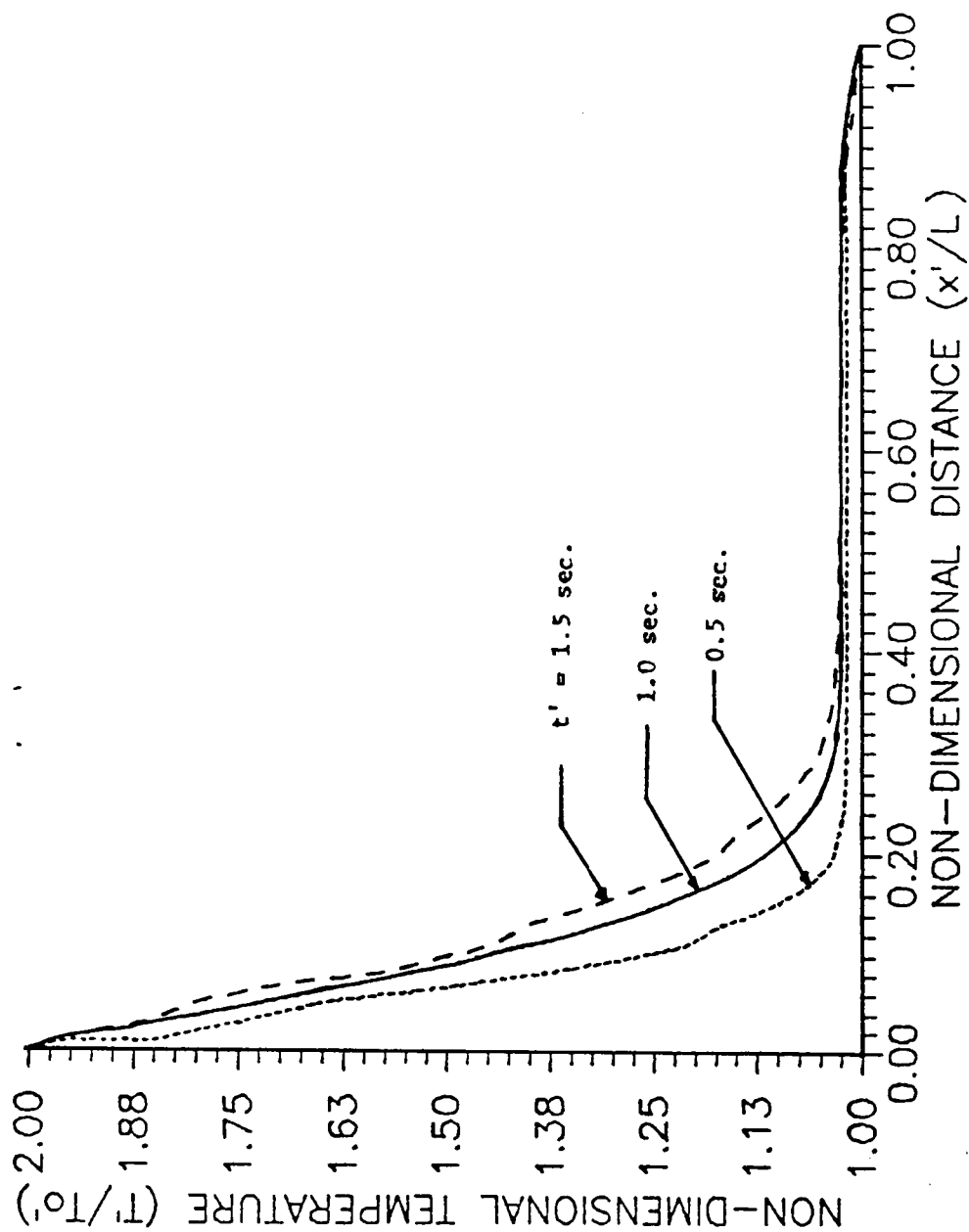


Figure 4.6 Temperature Distribution At Various Times

basically unaffected. For example, after 1.5 seconds, a substantial (65%) region has experienced only a small relative temperature rise ($>5\%$). The region influenced by the heating during the first 0.5 seconds is roughly twice that of the region influenced during the next 1.0 second.

A comparison of the computed temperature profiles in this study with those of Ref. [8] for helium is shown in Figure 4.7. Both the temperature profiles shown are for real times of 0.1 seconds. The steady-state profile is also shown for comparison purposes. The discrepancy between the two results is significant where a substantially lower rate of temperature rise is predicted here as compared to Ref. [8]. While Ref. [8] predicts temperature profiles not far from the steady-state, the temperature profiles predicted by this research resemble more closely those of conduction. Comparison with a rigid conduction model shows that the thermally induced velocity is responsible for an increased rate of heat transfer but not to the level predicted by Ref. [8]. The comparison with a pure conduction model also verifies that the heat transfer process is dominated by the conduction mode. It should be noted that for this computation the time step and the space grid size were changed to 0.04 and $1/44$, respectively, in order to reduce the prohibitive CPU time required for the numerical computation.

The numerically computed temperature profiles are compared with the perturbation scheme of Rhadwan and Kassoy¹² in Figure 4.8. In order to make this comparison, the boundary condition at the cold

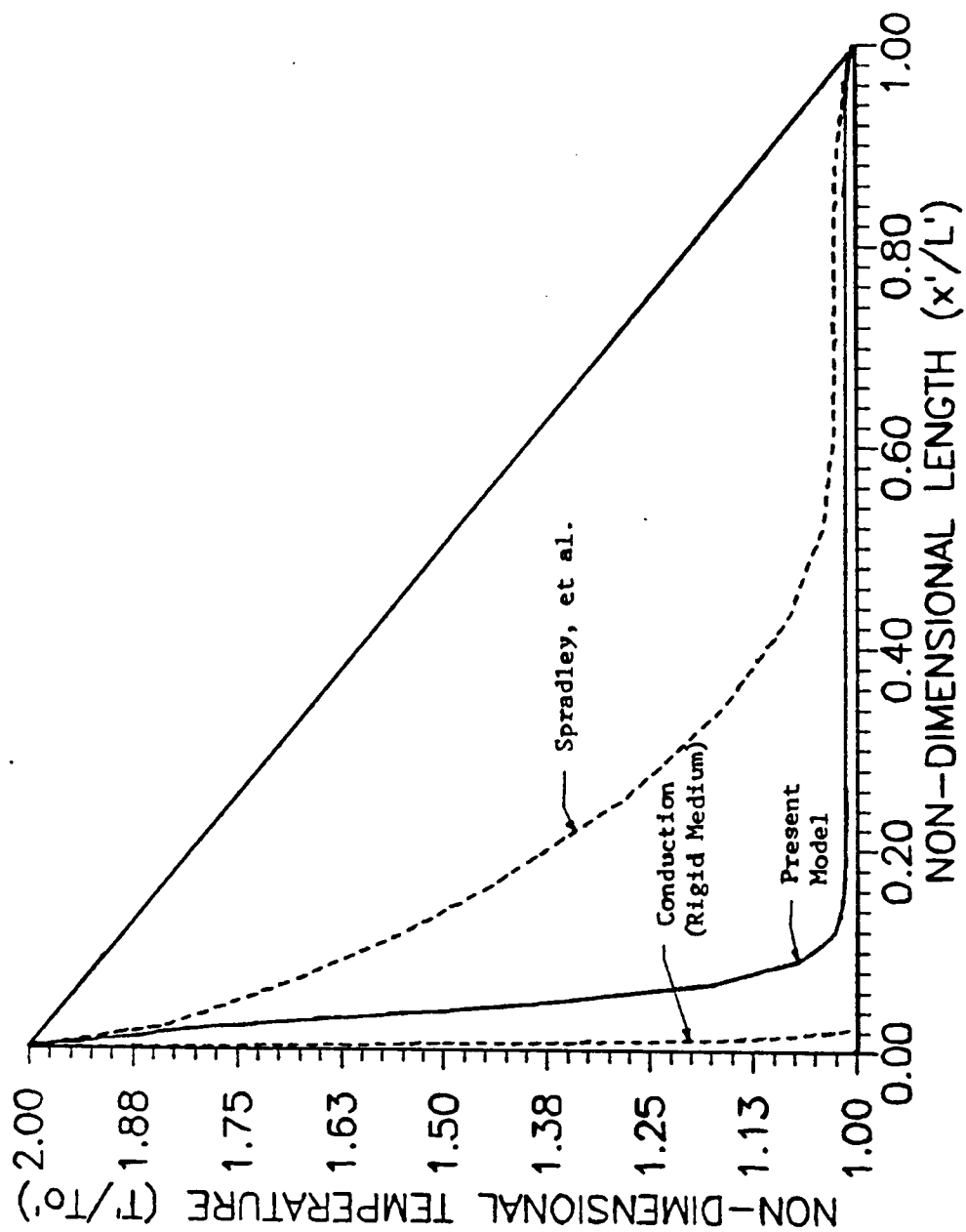


Figure 4.7 Comparison Of Temperature Distributions

wall was changed from an isothermal wall to an adiabatic one while the hot wall remained at a constant temperature as before. The temperature profiles shown in Figure 4.8 are for real times of 9.4 and 44.8 seconds. The two results seem to agree rather well particularly at longer times. The discrepancy of the two schemes at the shorter time is possibly due to the differences in the thermal boundary condition at the heated wall. In the case of Ref. [12] the heated wall is a function of time while here a step change in the temperature of the heated wall was used. A slight increase in the rate of heat transfer is expected in the latter case which maybe the reason for the relatively larger region of affected fluid predicted by this study.

The numerical model with a constant temperature boundary condition was also used to determine the temperature gradient at the heated wall as a function of time. This was accomplished with the use of a 4th-order-accurate polynomial. The results presented in Figure 4.9 show an initial sharp decline in the temperature gradient at the heated wall followed by a gradual leveling trend. It can be shown that the rate of heat transfer is directly proportional to the temperature gradient at the wall. A sharp decline from a value of approximately 300 to 51 occurs in a period from $t=0$ to $t=80$. This decrease levels off and presumably approaches a value of 1.0 at steady-state. The resultant decrease in the heat transferred to the flow is believed to be the reason behind the decay of the acoustic velocity field.

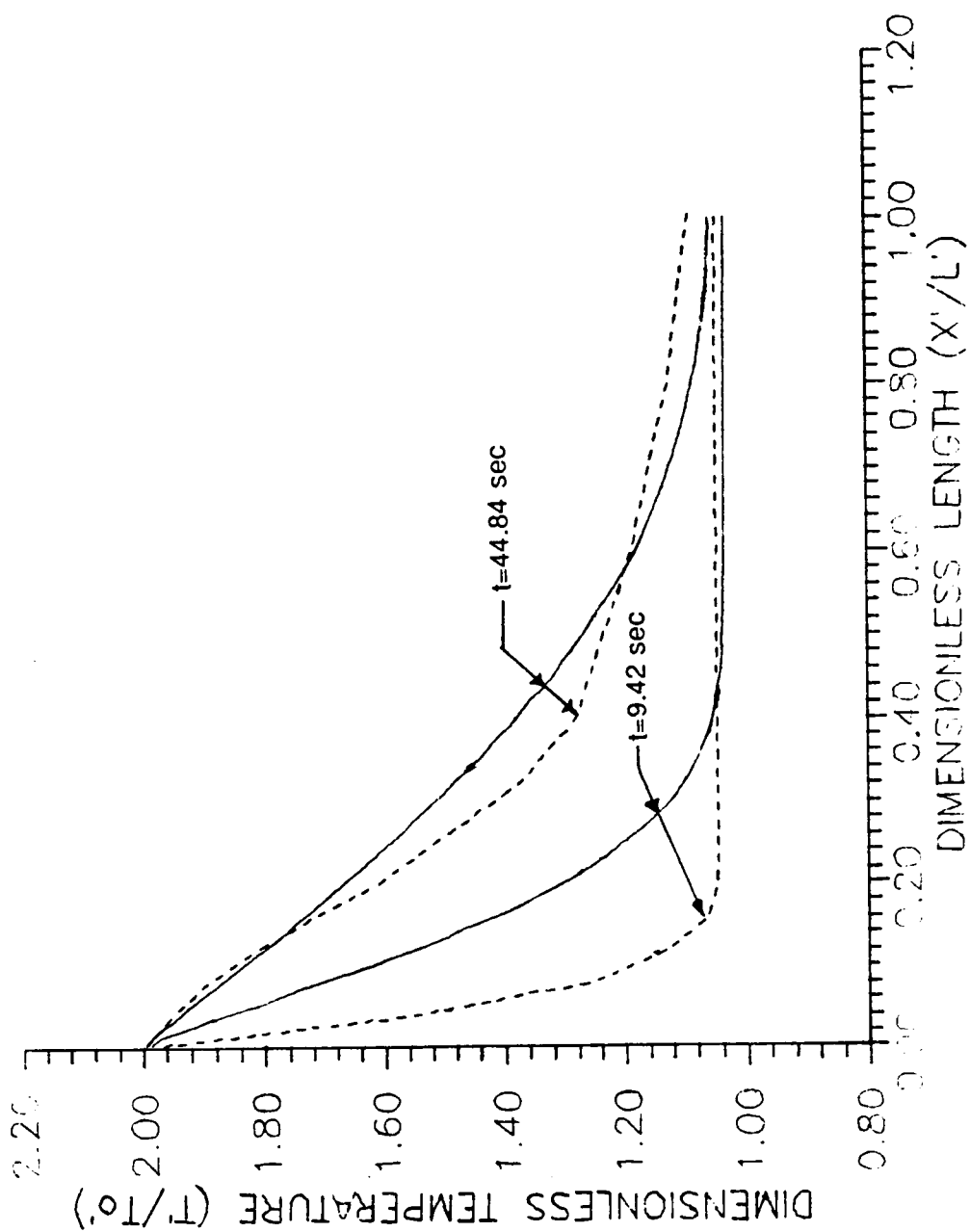


Figure 4.8 Comparison Of Computed Temperature (—) With Results From Ref [12] (----)

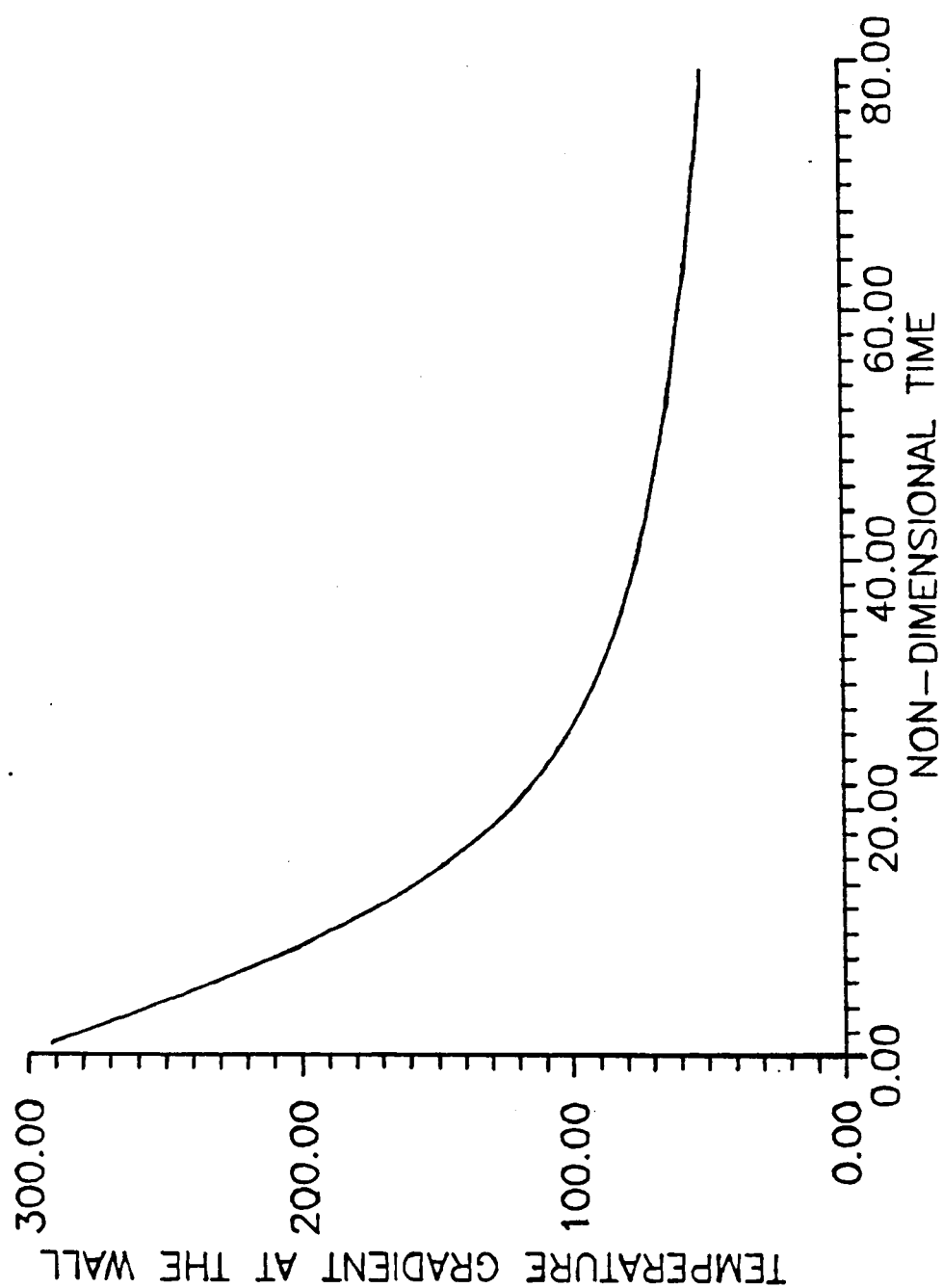


Figure 4.9 Time Variation Of Temperature Gradient At The Wall

The results for the density field are presented in Figure 4.10 for real times of 0.5, 1.0 and 1.5 seconds. The progression of heating results in a slow compression of the fluid. The density appears to include two distinct regions. The first is a region very near the heated surface where the density assumes an almost linear profile. This region includes approximately 20% of the flow field in the first 0.5 seconds, and expands to roughly 25% and 30% as time increases to 1.0 and 1.5 seconds, respectively. The second region is characterized with a relatively constant higher density. This elevated value of density is observed to increase with time as Figure 4.10 suggests.

One of the key areas of interest where TAC may be important is space processing where micro-gravity conditions exist. In such environments, there is an absence of other competing modes of heat transfer such as natural convection. Although these conditions do not exist in ground experiments, the suppression of the natural convection mode can be obtained through heating in a direction parallel to the gravity vector. Thus, it is of interest to compare the importance of TAC in a micro-gravity environment to the heating of a compressible medium from above in a gravity field.

Velocity fields were computed using the model developed for heating from above in varying levels of gravity. The results of this investigation are shown in Figure 4.11. The results indicate only a slight change in velocity amplitude and period of oscillation between gravity and zero-gravity environments. The computation of

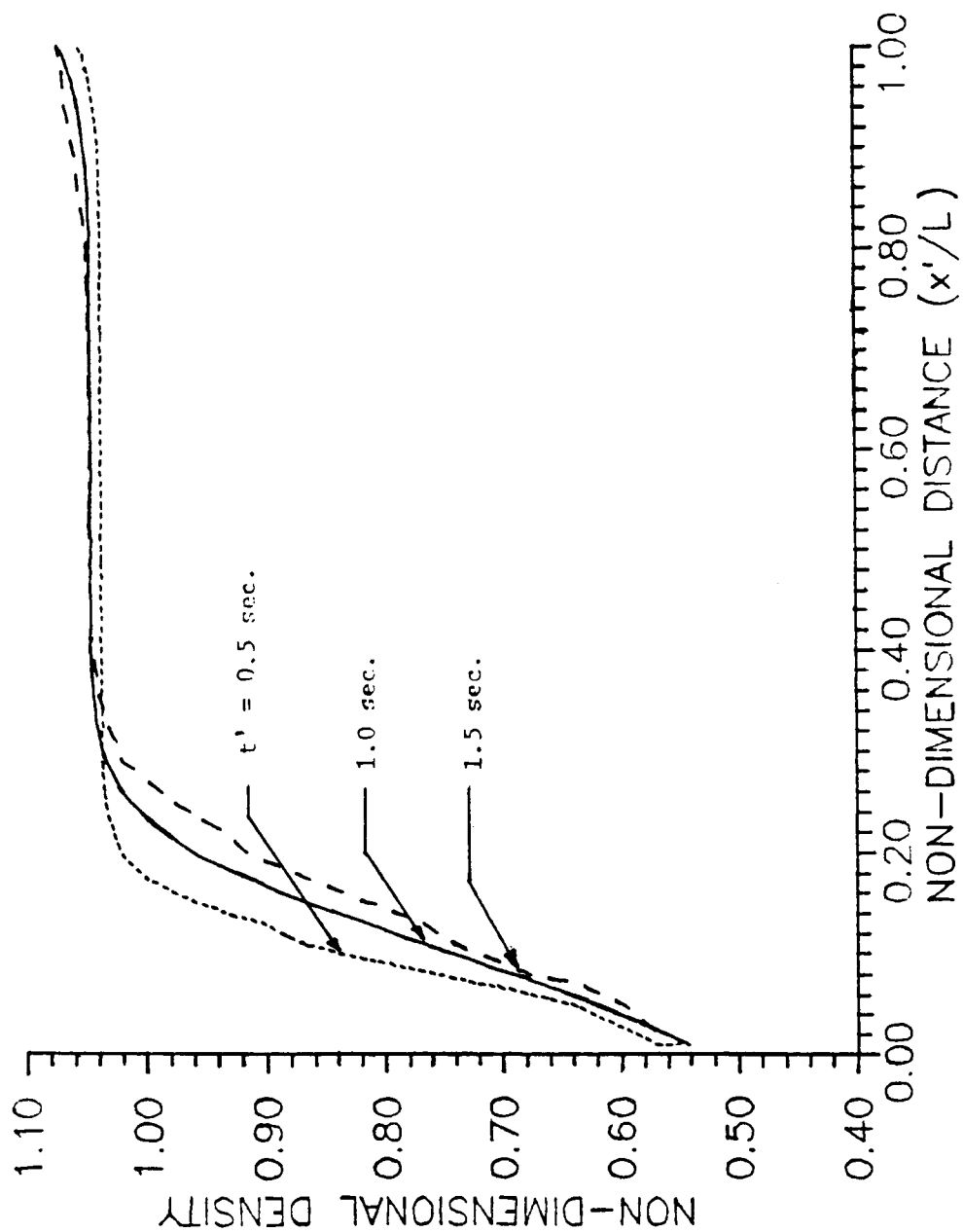


Figure 4.10 Density Profile At Various Times

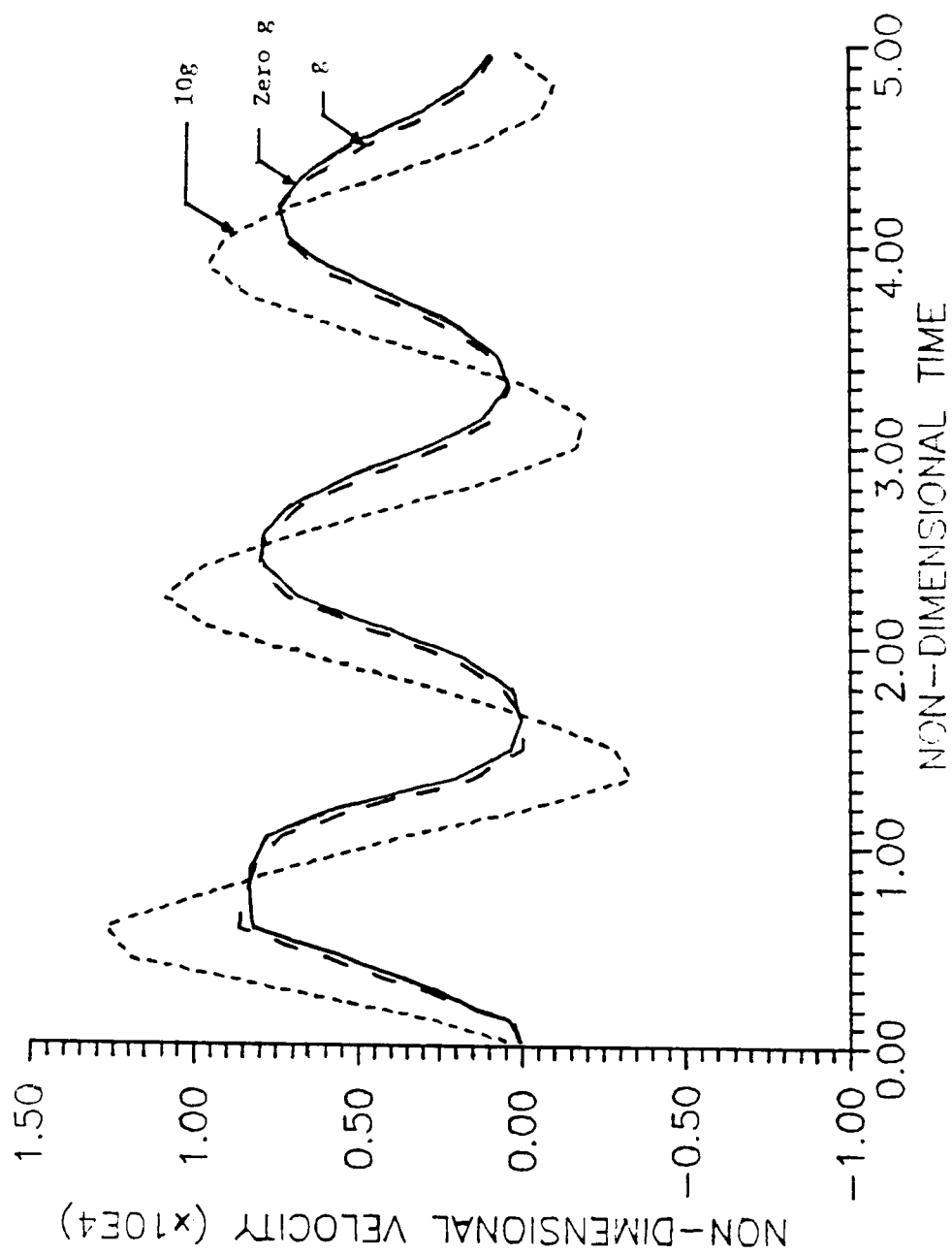


Figure 4.11 The Effect Of Gravity On Midpoint Velocity

the velocity field in a 10g environment, for example, shows modest changes in the velocity amplitude and period of oscillation when compared with the zero-gravity case. The full effects of gravity on the thermal convection process can only be examined by expanding the model to include two or three space dimensions.

4.2 TWO-DIMENSIONAL RESULTS

Validation of the Model

The two-dimensional model was first validated by simulating a one-dimensional TAC heat flow problem in zero-gravity. Simulation of one-dimensional conditions was accomplished by modeling a square cavity in zero-gravity with an insulated top and bottom surfaces. Figure 4.12 shows the midpoint, x-direction, velocity as a function of dimensionless time with air as the fluid. Similar to the one-dimensional model the fluid velocity exhibits an oscillatory behavior indicative of the acoustic nature of the problem. The fact that the numerical results for the first case were computed in the absence of a gravity field results in essentially no velocity in the direction perpendicular to the heat flux vector.

The maximum amplitude of the velocity field shown in Figure 4.12 is approximately 6.0×10^{-5} . This represents a 25% difference in the maximum amplitude in the velocity field predicted by the one-dimensional model. This discrepancy is probably due to the higher resolution used in the two-dimensional model as compared to the

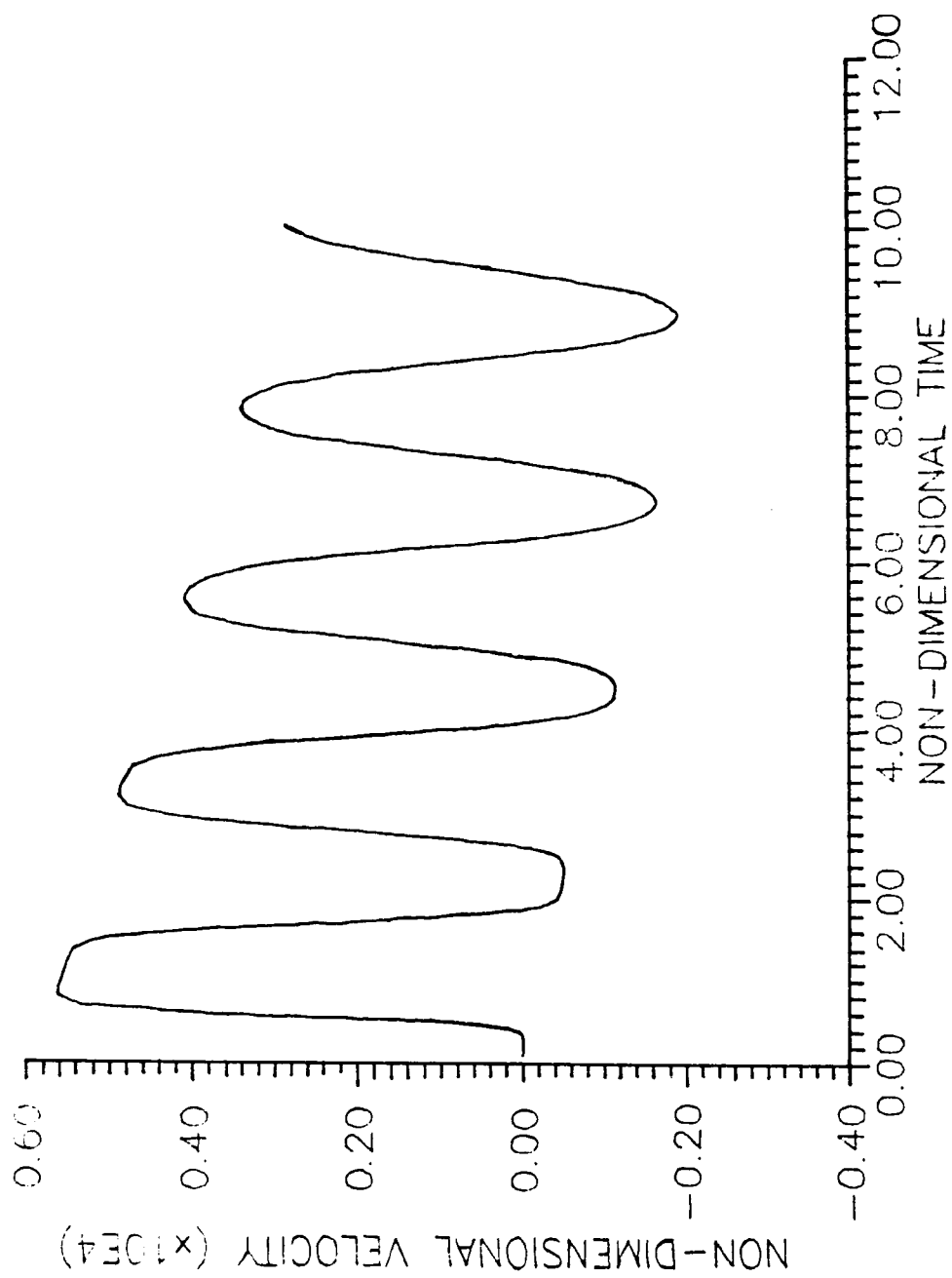


Figure 4.12 Unsteady Center Velocity

one-dimensional model. However, it is observed that differences in the mesh size seems not to produce significant differences in the velocity field except in the initial acoustic cycles.

A comparison of temperature profiles produced by the two-dimensional and one-dimensional models is shown in Figure 4.13. The results of the two models agree rather well with the one-dimensional model predicting a slightly higher rate of heat transfer than the two-dimensional model.

The Effect of Gravity

The second case investigated with the two-dimensional model was the effect of gravity on the flow field and the resulting mixed mode TAC and natural convection heat transfer process. For comparison purposes the model chosen was that of a square cavity 1.0"x1.0" filled with helium. The top and bottom surfaces (perpendicular to the gravity vector) were chosen to be insulated and the left and right surfaces were chosen to be isothermal with the left surface as the source of the thermal disturbance. The thermal disturbance at the left boundary is provided by a step change in wall temperature to twice its initial value.

The resulting velocity field is dominated by TAC oscillations at small times. Figure 4.14 shows the variation of midpoint velocity in the direction parallel to the heat flux vector with time. Similar to other cases the oscillatory nature of the velocity is also clearly evident.

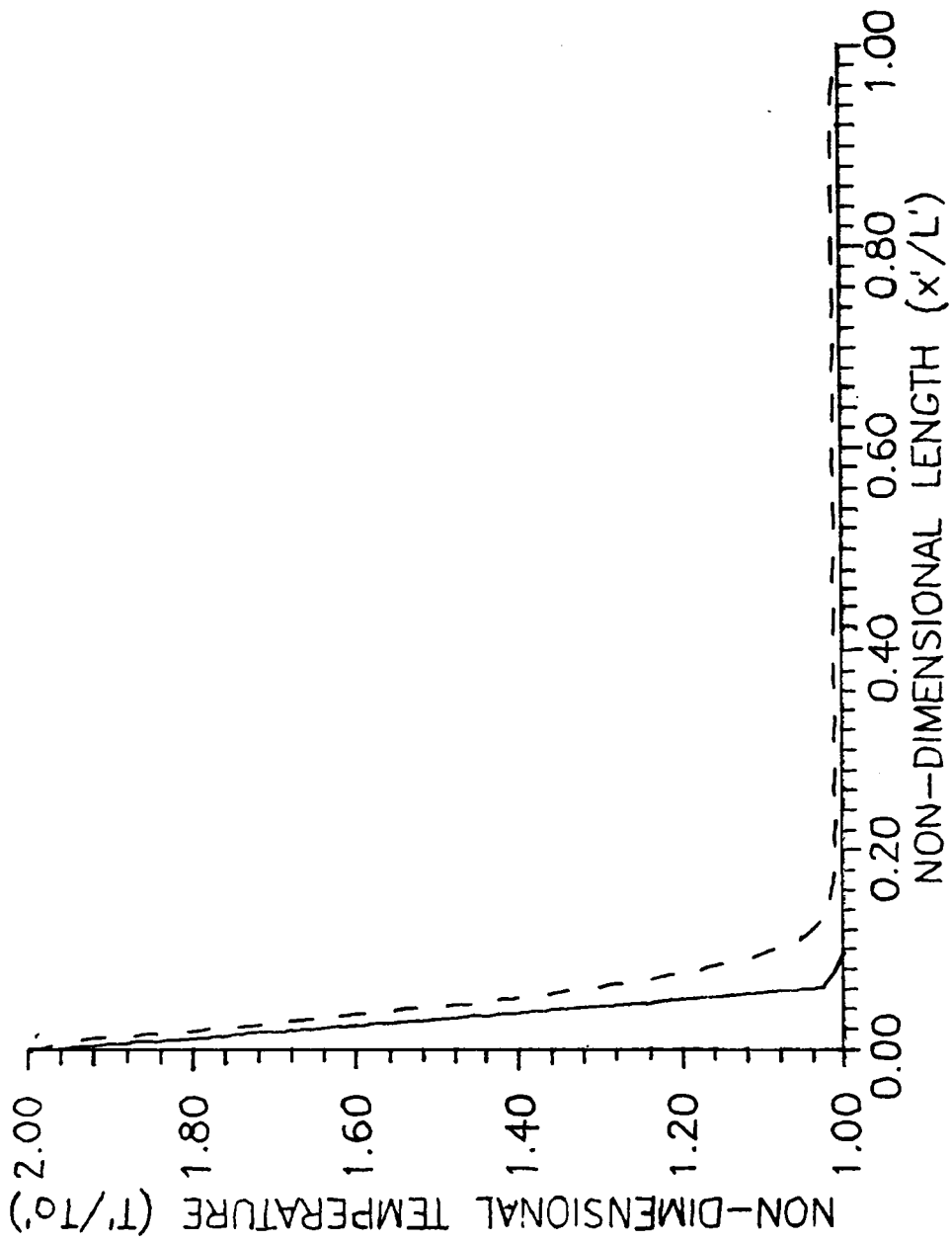


Figure 4.13 Comparison Of Temperature Profiles At 0.1 Seconds

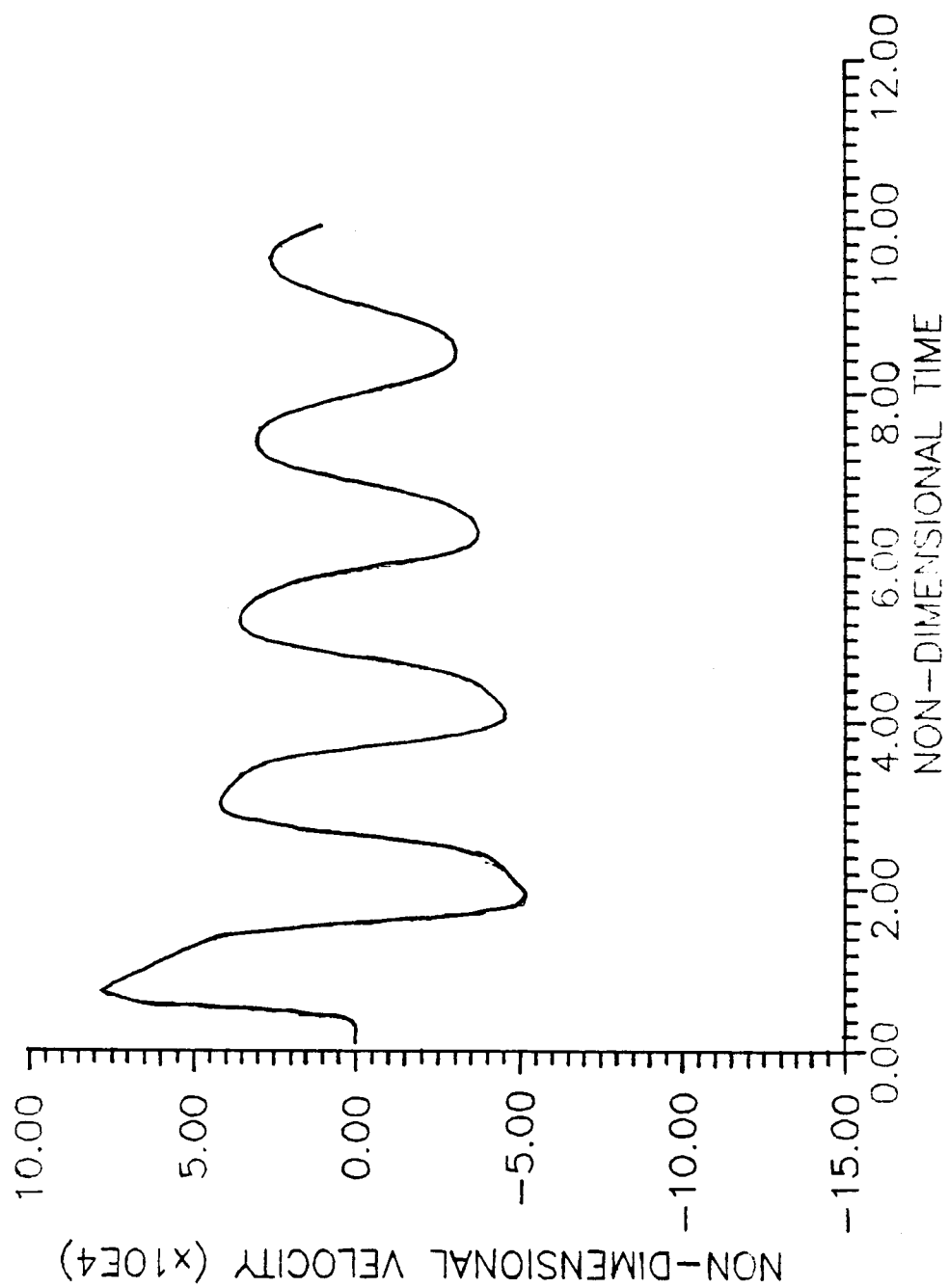


Figure 4.14 Unsteady Center Velocity

The initial amplitude of the velocity wave is approximately 8.0×10^{-4} . This represents a significant increase in the amplitude of the velocity wave as compared with previous cases. The speed of sound in helium is approximately 2.6 times larger than that of air which is presumably one reason for this increase. Also another, and probably the primary cause of this behavior, is the selected small size of the modeled rectangular cavity. The 1.0"x1.0" cavity is roughly 36.0 times smaller than the one modeled by the two-dimensional program in the one-dimensional simulation. Since a smaller volume of fluid is actually being heated in this case. Without changing the boundary conditions, the amount of energy added per unit mass of fluid is significantly higher. Thus this smaller amount of fluid responds with a more rapid thermal expansion and consequently a higher amplitude for the acoustic wave.

Temperature maps showing isotherms at dimensionless times of 10.0, 20.0 and 40.0 are shown in Figures 4.15 through 4.17 respectively. The effects of thermal convection at these very short times is evident in fluid, particularly near the heated wall. However, at such small times (approximately 10^{-4} seconds) no evidence of buoyancy driven convection is evident. This seems reasonable considering that only a small time has elapsed since the initiation of heating. Also the Grashof number for this problem is approximately 2×10^4 representing modest relative magnitudes of buoyancy force in the problem.

NON-DIMENSIONAL TEMPERATURE (T/T_0)

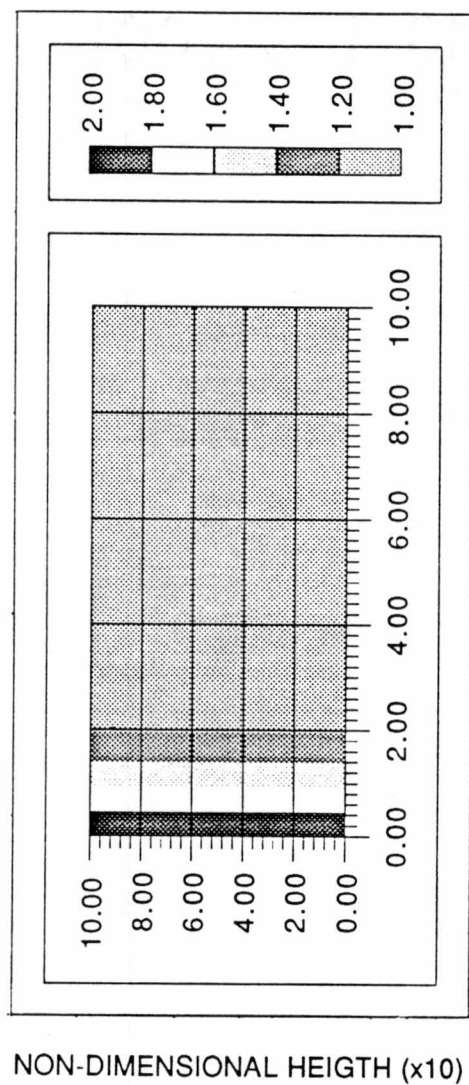


Figure 4.15 Temperature Map At $t=10.0$

NON-DIMENSIONAL TEMPERATURE (T/T_0)

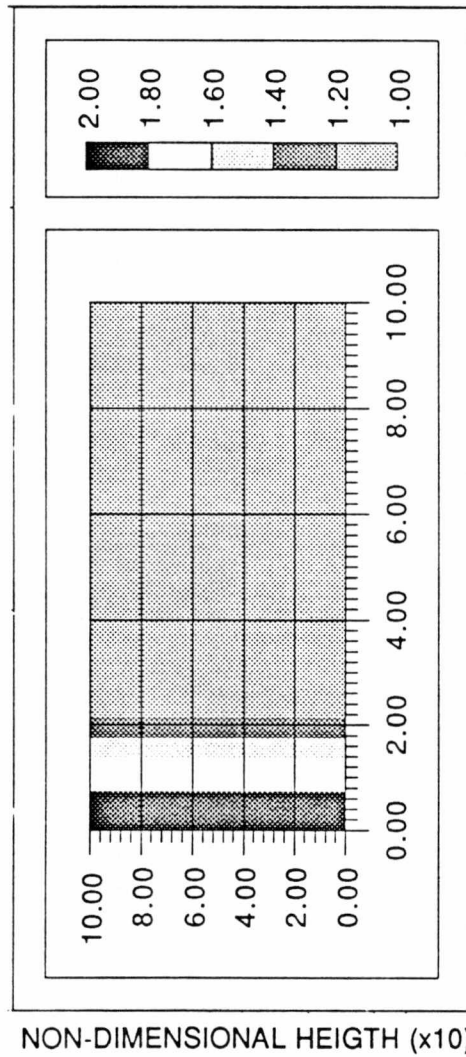


Figure 4.16 Temperature Map At $t=20.0$

NON-DIMENSIONAL TEMPERATURE (T/T_0)

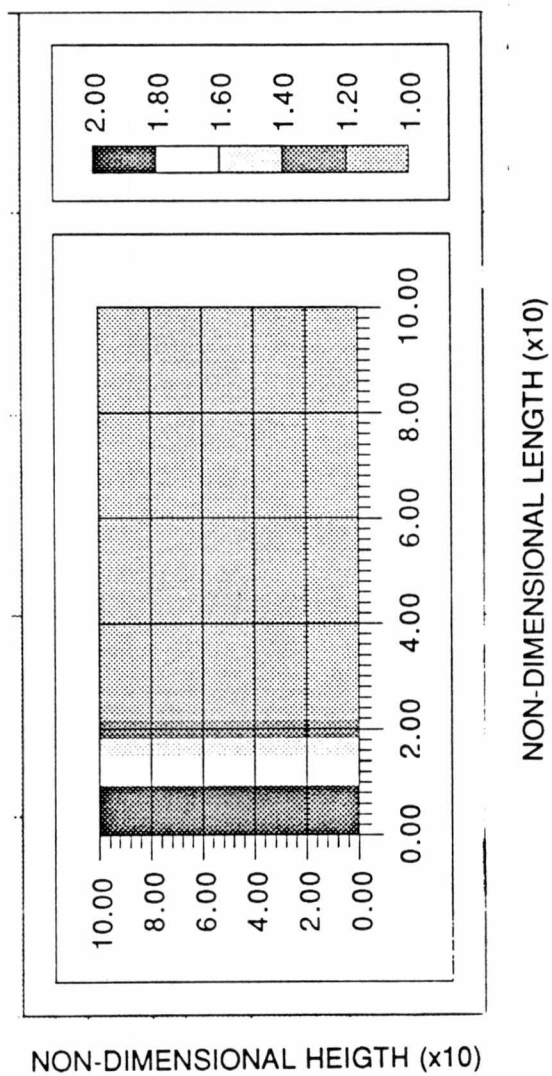


Figure 4.17 Temperature Map At $t=40.0$

Figure 4.18 shows the predicted temperature map from the model developed by Spradley and Churchill⁹ at a dimensionless time of 20.0. Although Figure 4.18 uses a different scaling than used in Figures 4.15 to 4.17 the shape of the temperature map in Figure 4.18 shows a heat transfer process rapidly approaching steady-state profiles. This type of behavior is clearly not shown in the temperature maps (Figures 4.15 to 4.17) developed in this study.

NON-DIMENSIONAL TEMPERATURE (T/T_0)

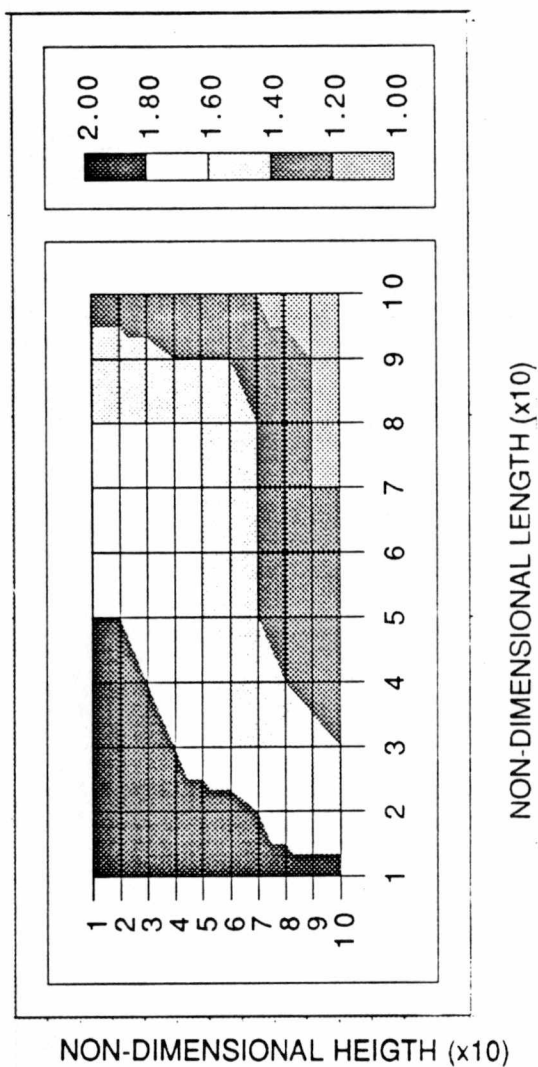


Figure 4.18 Temperature Map From Ref [9] At $t=20.0$

CHAPTER 5

CONCLUSIONS

The major purpose of this study has been the development and evaluation of a numerical model for the velocity and temperature fields in a thermoacoustic convection process. Numerical modeling of the TAC process as developed and discussed in Refs. [8] and [9], has been shown to be inadequate in the computation of reasonably accurate solutions. The numerical scheme discussed in Ref. [8] has exhibited significant dependence on the chosen space mesh size. The numerically erratic velocity field predicted by Ref. [8] improves drastically with the selection of an implicit method of solution. The large amplitude of the predicted velocity field is also observed to decrease with a decrease in mesh size. The rapid transient solution predicted in Ref. [8] slows dramatically as the mesh size decreases and comes to resemble a more conduction-dominated transient process.

A numerical model based on the SIMPLER¹⁵ algorithm has been developed for the governing equations of the TAC process. The modification of this algorithm to include compressibility effects is of fundamental importance in the proper modeling of TAC. This has been accomplished by using the Ideal gas equation of state in conjunction with Newton's Second Law and Conservation of Mass. The model predicts velocity solutions with less erratic behavior and

smaller amplitude, and slower developing temperature profiles than those predicted in Refs. [8]and [9].

The temperature profiles predicted for the one-dimensional case are much slower than those predicted in previous numerical schemes. They do, however, represent a larger rate of heat transfer than predicted by a rigid-conduction model. The temperature solutions from the one-dimensional model also shows some agreement with the solution by perturbation method given in Ref. [12]. A qualitative comparison of numerical results with the experimental data shows some agreement. Full verification of the numerical results with experimental data remains to be accomplished.

Solutions from the two-dimensional model have also shown much slower transient behavior than predicted in Ref. [9]. The simulation of one-dimensional heat flow by the two-dimensional model is in good agreement with the one-dimensional model. The simulation of the mixed mode TAC and natural convection problem described in Ref. [9] indicate that the flow patterns common to a natural convection flow of this type do not develop at such short times. This behavior is evidenced by the fact that no significant velocity perpendicular to the heat flux vector exists, and there is no skewing of the isotherms toward the upper section of the container.

Velocity, temperature, and density solutions indicate that the flow can be divided into two regions. One very near the heated wall where changes occur rapidly and a boundary layer behavior is

exhibited. The second is located far away from the heated surface where changes occur more slowly. In the two-dimensional case a decoupling of the x and y momentum equations also occurs at short times. These distinct regions of the flow become more difficult to differentiate as the solution progresses toward steady-state. In long time cases fluid motion is substantially decreased and conduction and/or conduction and natural convection become the dominate modes of heat transfer.

REFERENCES

- 1) Salah-Eddine, A. "An Experimental and Analytical Investigation of Thermoacoustic Convection Heat Transfer in Gravity and Low-Gravity Environments" Thesis, University of Tennessee, Knoxville, March 1986
- 2) Catt, J. "A Study of The Transient Effects of Thermoacoustic Convection Heat Transfer in Air" Thesis, University of Tennessee, Knoxville, August 1988
- 3) Parang, M. "Modeling of Thermoacoustic Convection Heat Transfer Phenomenon" JPL CR 958343, April 1990
- 4) Sondhauss, C. "Uber die Schallschwingungender Luft in erhitzten Glasrohren ind in gedeckten Pfeifen von ungleicher Weite" Annln. Phys., Vol. 79, 1850
- 5) Rijke, P. "Notice of a New Method of Causing a Vibration of the Air Contained in a Tube Open at Both Ends" Phil. Mag., Ser. 4, Vol. 17, pp. 419-422, 1859
- 6) Rayleigh, Lord "The Explanation of Acoustical Phenomenon" Nature, Vol. 18, p. 319, 1878

- 7) Larkin, B. "Heat Flow to a Confined Fluid in Zero-Gravity" Prog. in Astronautics and Astronautics, Planetary Bodies, Vol. 20, pp. 819-833, 1967
- 8) Spradley, L., Bourgeois, S., Fan, C., Grodzka, P. "A Numerical Solution for Thermoacoustic Convection of Fluids in Low Gravity" NASA CR-2267, May 1973
- 9) Spradley, L., Churchill, S, "Pressure-and Buoyancy-Driven Thermal Convection in a Rectangular Enclosure" Journal of Fluid Mechanics, Vol. 70, Part 4, pp. 705-720, 1975
- 10) Wilkes, J., Churchill, S. A.I.Ch.E. Journal Vol. 12, p. 161, 1966
- 11) Kassoy, D. "The Response of a Confined Gas to a Thermal Disturbance. I : Slow Transients" Journal of Applied Mathematics, Vol. 36, No. 3, pp. 624-634, June 1979
- 12) Radhwan, A., Kassoy, D. "The Response of a Confined Gas to a Thermal Disturbance : Rapid Boundary Heating" Journal of Engineering Mathematics, Vol. 18, p. 136-156, 1984
- 13) Poland, J., Kassoy, D., "The Induction Period of a Thermal Explosion in a Gas Between Infinite Parallel Plates" Combustion and Flame, Vol. 50, pp. 259-274, 1983

- 14) Clark, J., Kassoy, D., Riley, N. "Shocks Generated in a Confined Gas due to Rapid Heat Addition at the Boundary. I : Weak Shocks" Proc. R. Soc. Lond., A 393, pp. 309-329, 1984
- 15) Patankar, S., Spalding, D. "A Calculation Procedure for Heat, Mass, and Momentum Transfer in Three-Dimensional Parabolic Flows" Int. J. of Heat Mass Transfer, Vol. 15, p. 1787, 1972
- 16) Courant, R., Issacson, E., Rees, M. "On the Solution of Non-Linear Hyperbolic Differential Equations by Finite Differences" Comm. Pure Appl. Math, Vol. 5, p. 243, 1952
- 17) Patankar, S. Numerical Heat Transfer and Fluid Flow, McGraw-Hill-Hemisphere, 1980
- 18) Preliminary experimental results yet to be published

APPENDICES

APPENDIX A

**NORMALIZED AND DISCRETIZED
EQUATIONS FOR THE
TWO-DIMENSIONAL TAC PROGRAM**

CONTINUITY EQUATION IN FINITE DIFFERENCE FORM

Continuity Equation:

$$\frac{\partial \rho}{\partial t} + \frac{\partial(\rho u)}{\partial x} + \frac{\partial(\rho v)}{\partial y} = 0$$

Finite Difference Equation:

$$(\rho_p - \rho_p^0) \frac{\Delta x \Delta y}{\Delta t} + [(\rho u)_e - (\rho u)_w] \Delta y + [(\rho v)_n - (\rho v)_s] \Delta x = 0$$

Pressure Equation (from Continuity)

$$a_{PP} P_P = a_{EP} P_E + a_{WP} P_W + a_{NP} P_N + a_{SP} P_S + b_P$$

where:

$$a_{EP} = (\rho_e / a_e) \Delta y : a_{WP} = (\rho_w / a_{e(i-1)}) \Delta y$$

$$a_{NP} = (\rho_n / a_n) \Delta x : a_{SP} = (\rho_s / a_{n(j-1)}) \Delta x$$

$$b_P = -(\rho_p - \rho_p^0) \frac{\Delta x \Delta y}{\Delta t} - [(\rho u)_e - (\rho u)_w] \Delta y - [(\rho v)_n - (\rho v)_s] \Delta x = 0$$

$$a_{PP} = a_{EP} + a_{WP} + a_{NP} + a_{SP}$$

with

a_e and a_n defined in the
x and y-momentum equations respectively.

Pressure Correction Equation (also from Continuity)

$$a_{PP}P'_P = a_{EP}P'_E + a_{WP}P'_W + a_{NP}P'_N + a_{SP}P'_P + b_P$$

where:

$$a_{EP} = [(\rho_e^*/a_e) - u_e^*/(T_E + T_P)]\Delta y \quad : \quad a_{WP} = [(\rho_w^*/a_{e(i-1)}) + u_w^*/(T_W + T_P)]\Delta y$$

$$a_{NP} = [(\rho_n^*/a_n) - v_n^*/(T_N + T_P)]\Delta x \quad : \quad a_{SP} = [(\rho_s^*/a_{n(j-1)}) + v_s^*/(T_S + T_P)]\Delta x$$

$$b_P = -(\rho_P^* - \rho_P^0) \frac{\Delta x \Delta y}{\Delta t} - [(\rho u)^*_e - (\rho u)^*_w]\Delta y - [(\rho v)^*_n - (\rho v)^*_s]\Delta x = 0$$

$$a_{PP} = [(\rho_e^*/a_e) + u_e^*/(T_E + T_P) + (\rho_w^*/a_{e(i-1)}) - u_w^*/(T_W + T_P)]\Delta y + \\ [(\rho_n^*/a_n) + v_n^*/(T_N + T_P) + (\rho_s^*/a_{n(j-1)}) - v_s^*/(T_S + T_P)]\Delta x$$

with

a_e and a_n defined in the
x and y-momentum equations respectively.

NEWTON'S SECOND LAW IN FINITE DIFFERENCE FORM

X-Momentum Equation:

$$\rho \left(\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} \right) = \frac{\rho \sin \theta}{Fr} - \frac{\partial P}{\partial x} + \frac{1}{Re} \left[\frac{4}{3} \frac{\partial^2 u}{\partial x^2} + \frac{1}{3} \frac{\partial^2 v}{\partial x \partial y} + \left(\frac{L}{H} \right)^2 \frac{\partial^2 u}{\partial y^2} \right]$$

Finite Difference Equations:

$$a_e u_e = a_{ne} u_e + a_w u_w + a_{nne} u_{nnenn} + a_{nse} u_{nse} + b_u + (P_P - P_E) \Delta y$$

where :

$$a_{ne} = [-(\rho u)_e, 0] \Delta y_n + (4/3)(1/Re)(\Delta y_n / \Delta x) A(|Pe_e|)$$

$$a_w = [(\rho u)_w, 0] \Delta y_n + (4/3)(1/Re)(\Delta y_n / \Delta x) A(|Pe_w|)$$

$$a_{nne} = [-(\rho v)_n, 0] \Delta x_e + (1/Re)(L/H)^2 (\Delta x_e / \Delta y) A(|Pe_n|)$$

$$a_{nse} = [(\rho v)_s, 0] \Delta x_e + (1/Re)(L/H)^2 (\Delta x_e / \Delta y) A(|Pe_s|)$$

$$a_e^0 = \rho_e^0 \Delta x_e \Delta y / \Delta t$$

with $A(|Pe|)$ Defined as:

$$A(|Pe|) = (1 - 0.1|Pe|)^5$$

$$b_u = \frac{\rho_P \sin \theta}{Fr} \Delta x_e \Delta y + \frac{1}{Re} \left(\frac{1}{3} \right) [v_{nne} - v_{nse} - v_{nn} + v_s] + a_e^0 u_e^0$$

$$Pe_{e/w} = ((\rho u)_{e/w} \Delta x_{e/w}) / (4/3)(1/Re)$$

$$Pe_{n/s} = ((\rho v)_{n/s} \Delta y) / (1/Re)(L/H)^2$$

$$a_e = a_{ne} + a_{nw} + a_{nne} + a_{nse} + a^0_e$$

$\hat{u}$ Equation :

$$a_e \hat{u}_e = \sum a_{nb} u_{nb} + b_u$$

u^* Equation :

$$a_e u_e^* = \sum a_{nb} u_{nb}^* + b_u + (P_P^* - P_E^*) \Delta y$$

Velocity Correction Equation :

$$u_e = u_e^* + (P'_P - P'_E) \Delta y / a_e$$

Y-Momentum Equation :

$$\rho \left(\frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} \right) = -\frac{\rho}{Fr} \left(\frac{L}{H} \right) \cos \theta - \left(\frac{L}{H} \right)^2 \frac{\partial P}{\partial y} + \frac{1}{Re} \left[\frac{4}{3} \left(\frac{L}{H} \right)^2 \frac{\partial^2 v}{\partial y^2} + \frac{1}{3} \left(\frac{L}{H} \right)^2 \frac{\partial^2 u}{\partial x \partial y} + \frac{\partial^2 v}{\partial x^2} \right]$$

Finite Difference Equations :

$$a_n v_n = a_{nn} v_{nn} + a_s v_s + a_{nnw} v_{nw} + a_{nne} v_{ne} + b_v + (P_P - P_N) \Delta x$$

where :

$$a_{nn} = [-(\rho v)_n, 0] \Delta x + (4/3)(1/Re)(L/H)^2(\Delta x/\Delta y)A(|P_e|)$$

$$a_s = [(\rho v)_s, 0] \Delta x + (4/3)(1/Re)(L/H)^2(\Delta x/\Delta y)A(|P_e|)$$

$$a_{nne} = [-(\rho u)_e, 0] \Delta y_n + (1/Re)(\Delta y_n/\Delta x_e)A(|P_e|)$$

$$a_{nnw} = [(\rho u)_w, 0] \Delta y_n + (1/Re)(\Delta y_n/\Delta x_e)A(|P_e|)$$

$$a_n^o = \rho_n^o \Delta x \Delta y_n / \Delta t$$

with $A(|Pe|)$ Defined as:

$$A(|Pe|) = (1 - 0.1|Pe|)^5$$

$$b_v = -\frac{\rho_P}{Fr} \left(\frac{L}{H} \right) \cos \theta \Delta x \Delta y_n + \frac{1}{Re} \left(\frac{1}{3} \right) \left(\frac{L}{H} \right)^2 [u_{nne} - u_{nnw} - u_e + u_w] + a_n^o v_n^o$$

$$Pe_{n/s} = ((\rho v)_{n/s} \Delta y) / (4/3)(1/Re)(L/H)^2$$

$$Pe_{e/w} = ((\rho u)_{e/w} \Delta x_{e/w}) / (1/Re)$$

$$a_n = a_{nne} + a_{nnw} + a_{nn} + a_s + a_n^o$$

$\hat{v}$ Equation :

$$a_n \hat{v}_n = \sum a_{nb} v_n + b_v$$

v^* Equation :

$$a_n v_n^* = \sum a_{nb} v_{nb}^* + b_v + (P_P^* - P_N^*) \Delta x$$

Velocity Correction Equation:

$$v_n = v_n^* + (P'_P - P'_N) \Delta x / a_n$$

ENERGY EQUATION IN FINITE DIFFERENCE FORM

Energy Equation :

$$\rho \left[\frac{\partial T}{\partial t} + u \frac{\partial T}{\partial x} + v \frac{\partial T}{\partial y} \right] = -(\gamma - 1)P \left[\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \right] + \frac{\gamma}{\text{RePr}} \left[\frac{\partial^2 T}{\partial x^2} + \left(\frac{L}{H} \right)^2 \frac{\partial^2 T}{\partial y^2} \right]$$

Finite Difference Equations :

$$a_P T_P = a_E T_E + a_W T_W + a_N T_N + a_S T_S + b_T$$

where :

$$a_E = [-(\rho u)_e, 0] \Delta y + (\gamma / \text{RePr}) (\Delta y / \Delta x_e) A(|\text{Pe}_e|)$$

$$a_W = [(\rho u)_w, 0] \Delta y + (\gamma / \text{RePr}) (\Delta y / \Delta x_w) A(|\text{Pe}_w|)$$

$$a_N = [-(\rho v)_n, 0] \Delta x + (\gamma / \text{RePr}) (L/H)^2 (\Delta x / \Delta y_n) A(|\text{Pe}_n|)$$

$$a_S = [(\rho v)_s, 0] \Delta x + (\gamma / \text{RePr}) (L/H)^2 (\Delta x / \Delta y_s) A(|\text{Pe}_s|)$$

$$a_{P^0} = \rho P^0 \Delta x \Delta y / \Delta t$$

with $A(|\text{Pe}|)$ Defined as:

$$A(|\text{Pe}|) = (1 - 0.1|\text{Pe}|)^5$$

$$b = -(\gamma - 1)P_P [(u_e - u_w) \Delta y + (v_n - v_s) \Delta x] + a_{P^0} T_{P^0}$$

$$\text{Pe}_{e/w} = ((\rho u)_{e/w} \Delta x_{e/w}) / (\gamma / \text{Re})$$

$$\text{Pe}_{n/s} = ((\rho v)_{n/s} \Delta y_{n/s}) / (\gamma / \text{Re})$$

$$a_P = a_E + a_W + a_N + a_S + a_{P^0}$$

APPENDIX B

MODIPL.FOR COMPUTER CODE

C THIS IS A PROGRAM TO SOLVE THE THERMOACOUSTIC
 C CONVECTION HEAT TRANSFER PROBLEM BETWEEN TWO
 C PARALLEL PLATES AS PRESENTED BY L. W. SPRADLEY.
 C THIS PROGRAM IS MODIFIED FROM THE ORIGINAL TACH.FOR.
 C THIS PROGRAM UTILIZES AN IMPLICIT SOLUTION RATHER THAN
 C AN EXPLICIT ONE, MARCHING FORWARD IN TIME.
 C THE WORKING FLUID PROPERTIES OF HELIUM REMAIN THE
 C SAME.

PROGRAM NAME: MODIPL.FOR

PROGRAMMER : JOHN RAULSTON Jr.

VARIABLE DEFINITION:

C DT-NON-DIMENSIONAL TIME STEP (0.005)
 C DX-NON-DIMENSIONAL GRID SPACE ($L/n = 1.0/88.0$)
 C L- LENGTH BETWEEN THE TWO PLATES (15.3 CENTIMETERS)
 C TEO- INITIAL HE TEMPERATURE (273 K)
 C DEO- INITIAL HE DENSITY (2.38E-3 GM/CM**3)
 C V-INITIAL DYNAMIC HE VISCOSITY (1.83E-4 GM/CM-SEC)
 C CV- HE SPECIFIC HEAT (0.01085 CAL/GM-K)
 C K-INITIAL HE THERMAL CONDUCTIVITY (1.91E-5
 C CAL/CM-SEC-K)
 C PEO- INITIAL HE PRESSURE (1.01E6 DYNES/CM**2)
 C G- RATIO OF SPECIFIC HEATS ($C_p/C_v = 1.4$)
 C RTO-THE SQR ROOT OF $R*TEO$ (2.799E4 CM/SEC)
 C RE- REYNOLDS NUMBER ($L'*DEO'*(RTO)/V'$)
 C PR- PRANDTL NUMBER ($V'*C_p'/K'$)
 C U-NON-DIMENSIONAL VELOCITY ($U'/(RTO)$)
 C P-NON-DIMENSIONAL PRESSURE (P/PEO)
 C DE-NON-DIMENSIONAL DENSITY (DE/DEO)
 C TE-NON-DIMENSIONAL TEMPERATURE (TE/TEO)
 C TW-HEATER SURFACE TEMPERATURE (546 K)

THE PROGRAM BEGINS

DEFINING VARIABLES

REAL DEO,V,CV,G,PEO,TEO,TW,RTO,K,KL,DT,DX,RE,PR,TM,MAXDF
 INTEGER L,N,I,X,T,PN,Z,M
 DOUBLE PRECISION U(88,3),DE(88,3),TE(88,3),P(88,3)

\$,DXDEUU(88,3),DXP(88,3),DX2U(88,3),DXDEU(88,3)
 \$,DXT(88,3),DX2T(88,3),DEU(88,3),DXU(88,3),EU(88)

INITIALIZING VARIABLES

DEO=1.2028E-3
 V=1.83E-5
 CV=0.01085
 G=1.40
 PEO=1.01E6
 TEO=273

```

TW=546
RTO=2.799E4
KL=15.3
K=3.93E-7
Z=73206
M=0

C
C      INITIALIZING SPACE AND TIME STEPS DX,DT
C

PN=88.0
DX=1.0/PN
DT=0.005

C
C      CALCULATING REYNOLDS AND PRANDTL NUMBERS
C
RE=(KL*DEO*(RTO))/(V)
PR=(V*G*CV)/K
RE=5.66E6
PR=0.7213

C      INITIAL CONDITIONS
C

      DO 10 X=1,PN,1
          U(X,1)=0.0
          TE(X,1)=1.0
          P(X,1)=1.0
          DE(X,1)=1.0
          DEU(X,1)=0
10      CONTINUE

C
C      ASSUMED U,TE,P,DE VALUES FOR TIME=N+1

      DO 15 X=1,PN,1
          U(X,2)=0
          TE(X,2)=1.0
          P(X,2)=1.0
          DE(X,2)=1.0
          DEU(X,2)=0
          U(X,3)=0
          TE(X,3)=1.0
          P(X,3)=1.0
          DE(X,3)=1.0
          DEU(X,3)=0
15      CONTINUE
      OPEN(6,FILE='IMPTEMP882.DAT',STATUS='UNKNOWN')
      OUTER (TIME) LOOP

      DO 20 TM=DT,370,DT
          N=1
          TM=T*DT
          MIDDLE GRID SPACING LOOP
C
C
C

```

30

MAXDF=0.0

DO 40 I=1,PN,1

C
C
C

MOMENTUM EQUATION

```

IF ((U(I,N+1) .LT. 0) .AND. ((I .GT. 1)
$      .AND.
$      (I .LT. PN))) THEN
$      DXDEUU(I,N+2)=((DE(I+1,N+1)*U(I+1,N+1)
$      **2)-
$      (DE(I,N+1)*U(I,N+1)**2))/DX
$      ELSE IF (((U(I,N+1) .GT. 0) .OR.
$      (U(I,N+1).EQ.
$      0)) .AND. ((I .GT. 1) .AND. (I .LT. PN)))
$
$      THEN
$      DXDEUU(I,N+2)=((DE(I,N+1)*U(I,N+1)**2)-
$      (DE(I-1,N+1)*U(I-1,N+1)**2))/DX
$      END IF
$      IF (I .EQ. 1) THEN
$      DXDEUU(I,N+2)=((DE(I,N+1)*U(I,N+1)**2)-
$
$      (DE(I+1,N+1)*U(I+1,N+1)**2))/(2*DX)
$      ELSE IF (I .EQ. PN) THEN
$      DXDEUU(I,N+2)=-((DE(I,N+1)*U(I,N+1)**2)-
$      (DE(I-1,N+1)*U(I-1,N+1)**2))/(2*DX)
$      END IF
C
$      IF ((I .GT. 1) .AND. (I .LT. PN)) THEN
$      DXP(I,N+2)=(P(I+1,N+1)-P(I-1,N+1))/(2*DX)
$      ELSE IF (I .EQ. 1) THEN
$      DXP(I,N+2)=(P(I+1,N+1)-P(I,N+1))/DX
$      ELSE IF (I .EQ. PN) THEN
$      DXP(I,N+2)=(P(I,N+1)-P(I-1,N+1))/DX
$      END IF
C
$      IF ((I .GT. 1) .AND. (I .LT. PN)) THEN
$      DX2U(I,N+2)=(U(I+1,N+1)-2*U(I,N+1)+U(I-
$      1,N+1))/(DX**2)
$      ELSE IF (I .EQ. 1) THEN
$      DX2U(I,N+2)=(U(I+1,N+1)-3*U(I,N+1)
$      )/(DX**2)
$      ELSE IF (I .EQ. PN) THEN
$      DX2U(I,N+2)=(U(I-1,N+1)-3*U(I,N+1))
$      /(DX**2)
$      END IF
C

```

```

$      DEU(I,N+2)=DEU(I,N)-DT*(DXDEUU(I,N+2)+
$      DXP(I,N+2)-(4/3)*(1/RE)

```

```

$
40      *DX2U(I,N+2))
      CONTINUE

C
C
C      CONTINUITY EQUATION

DO 50 I=1,PN,1
  IF((I .GT. 1) .AND. (I .LT. PN)) THEN
    DXDEU(I,N+2)=(DEU(I+1,N+2)-DEU(I-
$      1,N+2))/(2*DX)
    ELSE IF (I .EQ. 1) THEN

$      DXDEU(I,N+2)=(DEU(I,N+2)+DEU(I+1,N+2))/
$      (2*DX)
    ELSE IF (I .EQ. PN) THEN
$      DXDEU(I,N+2)=-(DEU(I,N+2)+DEU(I-1,N+2))/
$      (2*DX)
    END IF

C      DE(I,N+2)=DE(I,N)-DT*(DXDEU(I,N+2))
      U(I,N+2)=DEU(I,N+2)/DE(I,N+2)
      IF (ABS(U(I,N+1)) .NE. 0.00) THEN
$      EU(I)=ABS(((U(I,N+2)-
$      U(I,N+1))))
      END IF
      IF (EU(I) .GT. MAXDF) THEN
        MAXDF=EU(I)
      END IF
50    CONTINUE

C
C      ENERGY EQUATION
DO 60 I=1,PN,1

  IF ((I .GT. 1) .AND. (I .LT. PN)) THEN
    DXU(I,N+2)=(U(I+1,N+2)-U(I-1,N+2))/(2*DX)
    ELSE IF (I .EQ. 1) THEN
      DXU(I,N+2)=(U(I,N+2)+U(I+1,N+2))/(2*DX)
    ELSE IF (I .EQ. PN) THEN
      DXU(I,N+2)=-(U(I,N+2)+U(I-1,N+2))/(2*DX)
    END IF

C      IF (((U(I,N+2) .EQ. 0) .OR.
$      (U(I,N+2) .GT. 0))
$      .AND. ((I .GT. 1) .AND.
$      (I .LT. PN))) THEN
$      DXT(I,N+2)=(TE(I,N+1)-TE(I-1,N+1))/DX
$      ELSE IF ((U(I,N+2) .LT. 0)
$      .AND. ((I .GT. 1)
$      .AND. (I .LT. PN))) THEN

```

```

DXT(I,N+2)=(TE(I+1,N+1)-TE(I,N+1))/DX
ELSE IF ((U(I,N+2) .LT. 0)
$      .AND. (I .EQ. 1))
$      THEN
DXT(I,N+2)=(TE(I+1,N+1)-TE(I,N+1))/DX
$      ELSE IF ((U(I,N+2) .GT. 0)
$      .OR. (U(I,N+2)
$      .EQ. 0)) .AND. (I .EQ. 1)) THEN
DXT(I,N+2)=(TE(I,N+1)-(TW/TEO))/(0.5*DX)
END IF

IF (((U(I,N+2) .GT. 0)
$      .OR. (U(I,N+2) .EQ.
$      0)) .AND. (I .EQ. PN)) THEN
DXT(I,N+2)=(TE(I,N+1)-TE(I-1,N+1))/DX
$      ELSE IF ((U(I,N+2) .LT. 0) .AND. (I .EQ.
$      PN)) THEN
DXT(I,N+2)=((TEO/TEO)-TE(I,N+1))/(0.5*DX)
END IF

C
IF ((I .GT. 1) .AND. (I .LT. PN)) THEN
DX2T(I,N+2)=(TE(I+1,N+1)
$      -2*TE(I,N+1)+TE(I-
$      1,N+1))/(DX**2)
ELSE IF (I .EQ. 1) THEN

DX2T(I,N+2)=(4/3)*(TE(I+1,N+1)
$      -3*TE(I,N+1)+
$      2*(TW/TEO))/(DX**2)
ELSE IF (I .EQ. PN) THEN

DX2T(I,N+2)=(4/3)*(TE(I-1,N+1)
$      -3*TE(I,N+1)
$      +2*(TEO/TEO))/(DX**2)
END IF

C
TE(I,N+2)=TE(I,N)-DT*
$      (U(I,N+2)*DXT(I,N+2)+
$      ((G-1)/DE(I,N+2))*P(I,N+1)
$      *DXU(I,N+2)-
$      (G/DE(I,N+2))
$      *(1/RE*PR)*DX2T(I,N+2))
60      CONTINUE
C
C
C      IDEAL GAS EQUATION SOLVE FOR P(I,N+2)
      THAT IS CALCULATED P AT TIME t+dt
      DO 65 I=1,PN,1
          P(I,N+2)=DE(I,N+2)*TE(I,N+2)
65      CONTINUE
      DO 66 I=1,PN,1
          U(I,N+1)=U(I,N+2)

```

```

        P(I,N+1)=P(I,N+2)
        TE(I,N+1)=TE(I,N+2)
        DE(I,N+1)=DE(I,N+2)
66      CONTINUE
      IF (MAXDF .GT. 0.0001 ) THEN
        GOTO 30
      END IF

      PRINTING THE RESULTS

11      FORMAT(2X,'DEO(I,N)=' ,E10.4,7X,
      +        'U(I,N)='
      +        ,E10.4,6X,'AT I=' ,I3)
13      FORMAT(2X,'TE(I,N)=' ,E10.4,8X,
      +        'P(I,N)=' ,E10.4,6X,
      +        'AT I=' ,I3)
14      FORMAT(2X,'NON-DIMENSIONAL TIME=' ,1X,F8.5)
17      FORMAT(2X,'U(I,N+2)=' ,E10.4,7X,'U(I,N+1)=' ,
      +        E10.4,6X,'AT I=' ,I3)

18      FORMAT(2X,'MAXDF=' ,1X,E12.5)

      WRITE(6,14) TM
      WRITE(*,14) TM
      M=M+1
      IF (M .EQ. Z) THEN
        WRITE(*,18) MAXDF
        DO 70 I=1,PN,1
          WRITE(6,*) U(I,N+2),I, TM
          WRITE(*,*) U(I,N+2),I, TM
          WRITE(6,*) TE(I,N+2),I, TM
          WRITE(*,13) TE(I,N+2),P(I,N+2),I
70      CONTINUE
        M=0
      END IF

      RESETTING CURRENT TIME VALUES
      DO 80 I=1,PN,1
        DEU(I,N)=DEU(I,N+2)
        U(I,N)=U(I,N+2)
        P(I,N)=P(I,N+2)
        TE(I,N)=TE(I,N+2)
        DE(I,N)=DE(I,N+2)
80      CONTINUE
20      CONTINUE
      STOP
      END

```

APPENDIX C

TAC1D.FOR COMPUTER CODE

```

C *****
C *          PROGRAM NAME : 1DTAC.FOR          *
C * THIS IS A PROGRAM TO STUDY THE EFFECTS OF THERMO- *
C * ACOUSTIC CONVECTION HEAT TRANSFER BETWEEN TWO *
C * PARALLEL INFINITE PLATES. THE EQUATIONS DEVELOPED FOR*
C * THIS PROGRAM ARE BASED ON THE WORK OF SUHAS V. *
C * PATANKAR IN HIS BOOK NUMERICAL HEAT TRANSFER AND *
C * FLUID FLOW. THIS PROGRAM UTILIZED A STAGGARED GRID *
C * SYSTEM WHERE VELOCITY IS CALCULATED AT THE BOUNDRIES *
C * OF THE CONTROL VOLUME OF EACH GRID. THE EQUATIONS *
C * DEVELOPED USE A POWER LAW TYPE SCHEME. *
C * ALTHOUGH ANY OF PATANKAR'S SCHEMES CAN BE EASILY *
C * ADOPTED TO THIS PROGRAM BY REDEFINING THE COEFF- *
C * ICIENTS OF THE MOMENTUM, PRESSURE, AND ENERGY EQUATION*
C *
C * VARIABLE DEFINITION : ( FLUID PROPERTIES ARE OF AIR) *
C * DT- NON-DIMENSIONAL TIME STEP ( 0.025) *
C * DX- NON-DIMENSIONAL GRID SPACE (1/44) *
C * TL - LENGTH BETWEEN TWO PLATES (15.3 cm) *
C * TEO-INITIAL TEMPERATURE (273 K) *
C * DEO- INITIAL DENSITY (1.2126E-3 gm/cm^3) *
C * V- DYNAMIC VISCOSITY (1.83E-4 gm/cm sec) *
C * CV-SPECIFIC HEAT (0.1715 cal/ gm K) *
C * K- THERMAL CONDUCTIVITY (6.08E-5 cal/cm sec K) *
C * PEO- INITIAL PRESSURE (1.01E6 dynes/cm**2) *
C * G- RATIO OF SPECIFIC HEATS (1.4) *
C * RTO- SPEED OF SOUND (SQ RT OF R*T) (28001 cm/sec) *
C * NOTE : ALTHOUGH SQ RT GAMMA*R*T IS CORRECT RTO AS *
C * DEFINED IS BASED ON THE ORIGINAL WORK OF L.W.SPRADLEY*
C * TW- TEMPERATURE OF HEATED WALL (546 K) *
C * RE-REYNOLDS NUMBER (L*DEO*RTO/V) *
C * PR-PRANDTL NUMBER (V*G*CV/K) *
C *
C * UNKNOWN VARIABLES TO BE CALCULATED : *
C * U(I,N)-VELOCITIY WHERE *
C * N=1 (K) DENOTES A PAST QUANTITY *
C * N=2(K+1)DENOTES A FUTURE QUANTITY *
C * N=3(K+2)DENOTES A u^ TERM *
C * N=4(K+3)DENOTES A u* TERM (ACCORDING TO PATANKAR'S *
C * NOTATION) *
C *
C * T(I,N)-NON-DIMENSIONAL TEMPERATURE *
C *
C * DE(I,N)-NON-DIM DENSITY *
C *
C * P(I,N)- NON-DIM PRESSURE WHERE *
C * N=1(K)DENOTES A PAST QUANTITY *
C * N=2(K+1)DENOTES A FUTURE QUANTITY *
C * N=3(K+2)DENOTES A P* QUANTITY *
C * N=4(K+3)DENOTES A P' QUANTITY (ACCORDING TO PATANKAR'S*
C * NOTATION)

```

```

C *
C * THIS PROGRAM MAKES USE OF AN IMPLICIT METHOD OF
C * SOLUTION FOR THE VARIOUS EQUATIONS.
C *****
C
C             THE PROGRAM BEGINS
C
C DIMENSION ARRAYS
C
      IMPLICIT REAL*4 (A-Z)
      REAL DX,DT,RE,PR,TL,TEO,DEO,V,CV,KD,G,PEO,TW,TM,RTO,
$ EROR,AMAX1,H,FTIME,TWALL,TW2,DTD,DTDXR
      INTEGER I,J,K,N,Q,M,L,Z,A,BNI,C,X,Z2,M2
      DOUBLE PRECISION U(89,5),T(89,5),
$ DE(89,5),P(89,5),AU(88),AT(88),MAXDF(88)
      DOUBLE PRECISION PEU(88),PWU(88),FEU(88),FWU(88),
$ ANEU(88),ANWU(88),AEO(88)
      DOUBLE PRECISION AEU1(88),B(88),AE(88),AW(88),
$ AP(88),BP(88)
      DOUBLE PRECISION BMS(88),PE(88),PW(88),FE(88),
$ FW(88)
      DOUBLE PRECISION ATE(88),ATW(88),APTO(88),
$ APT1(88),BT(88)
      DIMENSION IARRAY(1000,1000)
C
C ASSIGNING VARIABLES
C
C NOTE AIR PROPERTIES
      N=88
      DT=0.015
      DX=1.0/N
      TL=15.3
      TEO=273
      DEO=1.2126E-3
      V=1.83E-4
      CV=0.1715
      KD=3.44E-4
      PEO=1.01E6
      G=1.44
      RTO=28001
      TW=546
      Q=N-1
      M=0
      M2=0
      Z=24400
      FTIME=367.0
      X=1
      Z2=50000
      TWALL=TW/TEO
      TW2=TEO/TEO
      GRAV=0

```

```

C
C REYNOLDS AND PRANDTL NUMBER
C
      RE=(TL*DEO*RTO)/V
      PR=V*G*CV/KD
      FR=GRAV*TL/(RTO*RTO)
C INITIAL CONDITIONS
C
      DO 6 I=1,N,1
        DO 7 J=3,4,1
          U(I,J)=0
          P(I,J)=0
          DE(I,J)=0
          T(I,J)=0
7      CONTINUE
6      CONTINUE
      DO 10 I=1,N,1
        DO 20 J=1,2,1
          U(I,J)=0.0
          P(I,J)=1.0
          T(I,J)=1.0
          DE(I,J)=1.0

20     CONTINUE
10     CONTINUE

C
C OPENING OUTPUT DATA FILE
C
      OPEN(7,FILE='2DCOMP.DAT',STATUS='UNKNOWN')

C
C TIME LOOP
      M=0
      DO 30 TM=DT,FTIME,DT
        K=1
5      EROR=0

C
C      MOMENTUM EQUATION
C
      DO 40 I=1,Q,1
        IF ((I .GT. 1) .AND. (I .LT. Q)) THEN
          PEU(I)=(((DE(I,K+1)+DE(I+1,K+1))/2)
$           *U(I,K+1)*(DX))/((4/3)*(1/RE)))
          PWU(I)=(((DE(I,K+1)+DE(I-
$           1,K+1))/2)*U(I-1,K+1)*(DX))/((4/3)*(1/RE)))
          FEU(I)=((DE(I,K+1)+DE(I+1,K+1))/2)*U(I,K+1)
          FWU(I)=((DE(I,K+1)+DE(I-1,K+1))/2)*U(I-1,K+1)

C
C
      IF (-FEU(I) .GT. 0) THEN
        AMAX1=-FEU(I)
      ELSE IF ((-FEU(I) .LT. 0)

```

```

$      .OR. (-FEU(I) .EQ. 0)) THEN
          AMAX1=0
        END IF
        AU(I)=0
        H=(1-0.1*(ABS(PEU(I))))**5
        IF (H .GT. AU(I)) THEN
            AU(I)=H
        END IF
        ANEU(I)=((4/(3*RE*DX))*AU(I)+AMAX1)
C
C
IF (FWU(I) .GT. 0) THEN
    AMAX1=FWU(I)
ELSE IF ((FWU(I) .LT. 0)
$      .OR. (FWU(I) .EQ. 0)) THEN
    AMAX1=0
    END IF
    AU(I)=0
    H=(1-0.1*(ABS(PWU(I))))**5
    IF (H .GT. AU(I)) THEN
        AU(I)=H
    END IF
    ANWU(I)=((4/(3*RE*DX))*AU(I)+AMAX1)
C
C
    AEO(I)=(DE(I,1)*(DX/DT))
    AEU1(I)=(ANEU(I)+ANWU(I)+AEO(I))
    B(I)=AEO(I)*U(I,K)+FR*DE(I,K+1)*DX
C
    U(I,K+2)=(ANEU(I)*U(I+1,K+1)
$              +ANWU(I)*U(I-1,K+1)+B(I)
$              )/AEU1(I)
    END IF
    IF (I .EQ. 1) THEN
        PEU(I)=(((DE(I,K+1)+DE(I+1,K+1))/2)
$          *U(I,K+1)*(DX))/((4/3)*(1/RE)))
        PWU(I)=0.000
        FEU(I)=((DE(I,K+1)+DE(I+1,K+1))/2)*U(I,K+1)
        FWU(I)=0.0
C
    IF (-FEU(I) .GT. 0) THEN
        AMAX1=-FEU(I)
    ELSE IF ((-FEU(I) .LT. 0)
$      .OR. (-FEU(I) .EQ. 0)) THEN
        AMAX1=0
    END IF
    AU(I)=0
    H=(1-0.1*(ABS(PEU(I))))**5
    IF (H .GT. AU(I)) THEN
        AU(I)=H

```

```

END IF
ANEU(I)=(AMAX1+AU(I)*(4/(3*RE*DX)))
AMAX1=0
ANWU(I)=((4/(3*RE*DX))+AMAX1)
AEO(I)=(DE(I,K)*(DX/DT))
AEU1(I)=(ANEU(I)+ANWU(I)+AEO(I))
B(I)=AEO(I)*U(I,K)+FR*DE(I,K+1)*DX

C
$
U(I,K+2)=(ANEU(I)*U(I+1,K+1)+ANWU(I)*0+B(I))
/ AEU1(I)
END IF
IF (I .EQ. Q) THEN
$
PWU(I)=(((DE(I,K+1)+DE(I-1,K+1))/2)
*U(I-1,K+1)*(DX))/((4/3)*(1/RE)))
PEU(I)=0
FWU(I)=((DE(I,K+1)+DE(I-1,K+1))/2)*U(I-1,K+1)
FEU(I)=0

C
AMAX1=0
ANEU(I)=((4/(3*RE*DX)))

C
IF (FWU(I) .GT. 0) THEN
AMAX1=FWU(I)
ELSE IF ((FWU(I) .LT. 0)
$
.OR. (FWU(I) .EQ. 0)) THEN
AMAX1=0
END IF
AU(I)=0
H=(1-0.1*(ABS(PWU(I))))**5
IF (H .GT. AU(I)) THEN
AU(I)=H
END IF
ANWU(I)=(AMAX1+AU(I)*(4/(3*RE*DX)))

C
C
AEO(I)=(DE(I,K)*(DX/DT))
AEU1(I)=(ANEU(I)+ANWU(I)+AEO(I))
B(I)=AEO(I)*U(I,K)+FR*DE(I,K+1)*DX

C
$
U(I,K+2)=(ANWU(I)*U(I-1,K+1)+ANEU(I)*0+B(I))
/ AEU1(I)
END IF
40 CONTINUE
C PRESSURE EQUATION
C
DO 50 I=1,N,1
IF ((I .GT. 1) .AND. (I .LT. N)) THEN
AE(I)=(((DE(I,K+1)+DE(I+1,K+1))/2)*(1/AEU1(I)))
AW(I)=(((DE(I-1,K+1)+DE(I,K+1))/2)

```

```

$          *(1/AEU1(I-1)))
AP(I)=(AE(I)+AW(I))
BP(I)=(DE(I,K)-DE(I,K+1))*(DX/DT)+((DE(I-1,K+1)+
$          DE(I,K+1))/2)*U(I-1,K+2)-((DE(I+1,K+1)+
$          DE(I,K+1))/2)*U(I,K+2)
$          P(I,K+1)=(AE(I)*P(I+1,K+1)
$          +AW(I)*P(I-1,K+1)+BP(I))
$          /AP(I)
END IF
IF (I .EQ. 1) THEN
    AE(I)=((DE(I,K+1)+DE(I+1,K+1))/2)*(1/AEU1(I)))
    AW(I)=0.0
    AP(I)=(AE(I)+AW(I))
    BP(I)=(DE(I,K)-DE(I,K+1))*(DX/DT)+(0)
$          -((DE(I+1,K+1)+
$          DE(I,K+1))/2)*U(I,K+2)
C
$          P(I,K+1)=(AE(I)*P(I+1,K+1)
$          +AW(I)*P(I+1,K+1)+BP(I))
$          /AP(I)
END IF
IF (I .EQ. N) THEN
    AE(I)=0.0
    AW(I)=((DE(I-1,K+1)+DE(I,K+1))/2)
$          *(1/AEU1(I-1)))
    AP(I)=(AE(I)+AW(I))
    BP(I)=(DE(I,K)-DE(I,K+1))*(DX/DT)+((DE(I-1,K+1)+
$          DE(I,K+1))/2)*U(I-1,K+2)
$          P(I,K+1)=(AE(I)*P(I+1,K+1)
$          +AW(I)*P(I-1,K+1)+BP(I))
$          /AP(I)
END IF
C
50      CONTINUE
C
C      USING P AS P* SOLVING MOMENTUM EQN
C
DO 60 I=1,N,1
    P(I,K+2)=P(I,K+1)
60      CONTINUE
    I=1
    U(1,K+3)=(ANEU(I)*U(I+1,K+3)+ANWU(I)*0.0+B(I)+
$          (P(I,K+2)-P(I+1,K+2)))/AEU1(I)

DO 70 I=2,Q,1
    U(I,K+3)=(ANEU(I)*U(I+1,K+3)
$          +ANWU(I)*U(I-1,K+3)+B(I)+
$          (P(I,K+2)-P(I+1,K+2)))/AEU1(I)

```

```

70      CONTINUE
        I=Q
        U(Q,K+3)=(ANEU(I)*0.0+ANWU(I)*U(I-1,K+3)+B(I)+
$         (P(I,K+2)-P(I+1,K+2)))/AEU1(I)
        U(N,K+3)=0.0
C
C      TREATING DE AS DE* SOLVE THE MASS SOURCE TERM
C
      DO 80 I=1,N,1
        DE(I,K+2)=DE(I,K+1)
80      CONTINUE
C
C      MASS SOURCE CALCULATION
C
      DO 90 I=1,N,1
        IF ((I .GT. 1) .AND. (I .LT. N)) THEN
          BMS(I)=(DE(I,K)-DE(I,K+2))*(DX/DT)+
$          ((DE(I,K+2)+DE(I-1,K+2))/2)*U(I-1,K+3)-
$          ((DE(I,K+2)+DE(I+1,K+2))/2)*U(I,K+3)
          END IF
          IF (I .EQ. 1) THEN
            BMS(I)=(DE(I,K)-DE(I,K+2))*(DX/DT)-
$            ((DE(I,K+2)+DE(I+1,K+2))/2)*U(I,K+3)
          END IF
          IF (I .EQ. N) THEN
            BMS(I)=(DE(I,K)-DE(I,K+2))*(DX/DT)+
$            ((DE(I,K+2)+DE(I-1,K+2))/2)*U(I-1,K+3)-0
          END IF
90      CONTINUE
C
C      SOLVE THE P' EQUATION
C
      DO 100 I=1,N,1
        IF ((I .GT. 1) .AND. (I .LT. N)) THEN
          AE(I)=(((DE(I,K+2)+DE(I+1,K+2))/2)*(1/AEU1(I))-
$          U(I,K+3)/(T(I,K+1)+T(I+1,K+1)))
          AW(I)=(((DE(I,K+2)+
$          DE(I-1,K+2))/2)*(1/AEU1(I-1))+
$          U(I-1,K+3)/(T(I-1,K+1)+T(I,K+1)))
          AP(I)=(AE(I)+AW(I)+
$          (DX/DT)*(1/T(I,K+1)))
C
          P(I,K+3)=(AE(I)*P(I+1,K+3)+AW(I)*P(I-
$          1,K+3)+BMS(I))/(AP(I))
          END IF
C
          IF (I .EQ. 1) THEN
            AE(I)=(((DE(I,K+2)+DE(I+1,K+2))/2)*(1/AEU1(I))-
$            U(I,K+3)/(T(I,K+1)+T(I+1,K+1)))
            AW(I)=0

```

```

C      AP(I)=(AE(I)+AW(I)+
$      (DX/DT)*(1/T(I,K+1)))
C
$      P(I,K+3)=(AE(I)*P(I+1,K+3)
$      +BMS(I))/(AP(I))
C      END IF
C
IF (I .EQ. N) THEN
    AE(I)=0
    AW(I)=(((DE(I,K+2)+DE(I-1,K+2))/2)*(1/AEU1(I-1))+
$      U(I-1,K+3)/(T(I-1,K+1)+T(I,K+1)))
$      AP(I)=(AE(I)+AW(I)+
$      (DX/DT)*(1/T(I,K+1)))
C
$      P(I,K+3)=(AW(I)*P(I-
$      1,K+3)+BMS(I))/(AP(I))
C      END IF
C
100    CONTINUE
C
C      CORRECT THE VELOCITY AND DENSITY FIELD'S
C
DO 110 I=1,Q,1
    U(I,K+1)=U(I,K+3)+(1/AEU1(I))
$      *(P(I,K+3)-P(I+1,K+3))
    DE(I,K+1)=(P(I,K+1)+P(I,K+3))/(T(I,K+1))
110    CONTINUE
C
C      SOLVE THE ENERGY EQUATION
C
DO 120 I=1,N,1
    IF ((I .GT. 1) .AND. (I .LT. N)) THEN
        PE(I)=(((DE(I,K+1)
$      +DE(I+1,K+1))/2)*U(I+1,K+1)*(DX))
$      /(G/(RE*PR)))
        PW(I)=(((DE(I,K+1)
$      +DE(I-1,K+1))/2)*U(I,K+1)*(DX))
$      /(G/(RE*PR)))
        FE(I)=((DE(I,K+1)+DE(I+1,K+1))/2)*U(I,K+1)
        FW(I)=((DE(I,K+1)+DE(I-1,K+1))/2)*U(I-1,K+1)
C
        IF (-FE(I) .GT. 0) THEN
            AMAX1=-FE(I)
        ELSE IF ((-FE(I) .LT. 0)
$      .OR. (-FE(I) .EQ. 0)) THEN
            AMAX1=0
        END IF
        AT(I)=0
        H=(1-0.1*(ABS(PE(I))))**5
        IF (H .GT. AT(I)) THEN

```

```

      AT(I)=H
      END IF
      ATE(I)=((G/(RE*PR*DX))*AT(I)+AMAX1)

C
C
      IF (FW(I) .GT. 0) THEN
        AMAX1=FW(I)
      ELSE IF ((FW(I) .LT. 0)
$      .OR. (FW(I) .EQ. 0)) THEN
        AMAX1=0
      END IF
      AT(I)=0
      H=(1-0.1*(ABS(PW(I))))**5
      IF (H .GT. AT(I)) THEN
        AT(I)=H
      END IF
      ATW(I)=((G/(RE*PR*DX))*AT(I)+AMAX1)

C
      APTO(I)=(DE(I,K)*(DX/DT))
      BT(I)=APTO(I)*T(I,K)
      APT1(I)=(ATE(I)+ATW(I)+APTO(I)+(G-1)*DE(I,K+1)*
$      (U(I,K+1)-U(I-1,K+1)))
$      T(I,K+2)=(ATE(I)*T(I+1,K+1)
      +ATW(I)*T(I-1,K+2)+BT(I))/APT1(I)
      END IF
      IF (I .EQ. 1) THEN
        PE(I)=(((DE(I,K+1)
$      +DE(I+1,K+1))/2)*U(I,K+1)*(DX))
$      /(G/(RE*PR)))
        PW(I)=0
        FE(I)=((DE(I,K+1)+DE(I+1,K+1))/2)*U(I,K+1)
        FW(I)=0

C
      IF (-FE(I) .GT. 0) THEN
        AMAX1=-FE(I)
      ELSE IF ((-FE(I) .LT. 0)
$      .OR. (-FE(I) .EQ. 0)) THEN
        AMAX1=0
      END IF
      AT(I)=0
      H=(1-0.1*(ABS(PE(I))))**5
      IF (H .GT. AT(I)) THEN
        AT(I)=H
      END IF
      ATE(I)=(((G)/(RE*PR*DX))*AT(I)+AMAX1)

C
      AMAX1=0
      ATW(I)=(((2*G)/(RE*PR*DX))*(1)+AMAX1)
      APTO(I)=(DE(I,K)*(DX/DT))
      BT(I)=APTO(I)*T(I,K)
      APT1(I)=(ATE(I)+ATW(I)+APTO(I)+(G-1)*DE(I,K+1)*

```

```

$          (U(I,K+1)-0))
$          T(I,K+2)=(ATE(I)*T(I+1,K+1)+ATW(I)*(TWALL)+BT(I)
$              )/APT1(I)
END IF
IF (I .EQ. N) THEN
    PE(I)=0
    PW(I)=(((DE(I,K+1)+DE(I-1,K+1))/2)*U(I-1,K+1)*
$          DX)/(G/(RE*PR)))
    FE(I)=0
    FW(I)=((DE(I,K+1)+DE(I-1,K+1))/2)*U(I-1,K+1)
C
    AMAX1=0
    ATE(I)=(((2*G)/(RE*PR*DX))*(1)+AMAX1)
C
    IF (FW(I) .GT. 0) THEN
        AMAX1=FW(I)
    ELSE IF ((FW(I) .LT. 0)
$        .OR. (FW(I) .EQ. 0)) THEN
        AMAX1=0
    END IF
    AT(I)=0
    H=(1-0.1*(ABS(PW(I))))**5
    IF (H .GT. AT(I)) THEN
        AT(I)=H
    END IF
    ATW(I)=((G)/(RE*PR*DX))*AT(I)+AMAX1
    APTO(I)=(DE(I,K)*(DX/DT))
    BT(I)=APTO(I)*T(I,K)
    APT1(I)=(ATE(I)+ATW(I)+APTO(I)+(G-1)*DE(I,K+1)*
$        (0-U(I-1,K+1)))
$        T(I,K+2)=(ATE(I)*TW2+ATW(I)*T(I-1,K+2)+BT(I))/
$            (APT1(I))
    END IF
120    CONTINUE
    DO 126 I=1,N,1
        T(I,K+1)=T(I,K+2)
126    CONTINUE
C
C
C    MASS SOURCE CALCULATION
DO 129 I=1,N,1
    IF ((I .GT. 1) .AND. (I .LT. N)) THEN
        BMS(I)=(DE(I,K)-DE(I,K+1))*(DX/DT)+
$        ((DE(I,K+1)+DE(I-1,K+1))/2)*U(I-1,K+1)-
$        ((DE(I,K+1)+DE(I+1,K+1))/2)*U(I,K+1)
    END IF
    IF (I .EQ. 1) THEN
        BMS(I)=(DE(I,K)-DE(I,K+1))*(DX/DT)-
$        ((DE(I,K+1)+DE(I+1,K+1))/2)*U(I,K+1)
    END IF
    IF (I .EQ. N) THEN

```

```

      BMS(I)=(DE(I,K)-DE(I,K+1))*(DX/DT)+
$      ((DE(I,K+1)+DE(I-1,K+1))/2)*U(I-1,K+1)-0
      END IF
      IF (BMS(I) .GT. EROR) THEN
        EROR=BMS(I)
      END IF
C
129      CONTINUE

C      CONVERGENCE CRITERION
C      WRITE(*,*)EROR
      IF (EROR .GT. 1E-6) THEN
        GOTO 5
      END IF
      WRITE(*,14)TM
      DTDX=(T(3,2)-(9/2)
$      *T(2,2)+9*T(1,2)-(11/2)*TWALL)
$      /(3*DX)
      DTDXR=(-18.125*TWALL+18*T(1,2)
$      +1.125*T(2,2)-T(3,2))/(5.25*DX)

C
C      PRINTING THE RESULTS
C
11      FORMAT(2X,'DE(I,N)=' ,E10.4,7X,
+      'U(I,N)=' ,E10.4,6X,'AT I=' ,I3)
13      FORMAT(2X,'T(I,N)=' ,E10.4,8X,'P(I,N)=' ,E10.4,6X,
+      'AT I=' ,I3)
14      FORMAT(2X,'NON-DIMENSIONAL TIME=' ,1X,E12.5)
15      FORMAT(2X,'EROR=' ,1X,E12.5)
17      FORMAT(2X,E10.4,2X,E10.4,2X
+      ,E10.4,2X,E10.4,1X,I3,1X,E10.4)
32      FORMAT(2X,E10.4,2X,E10.4,2X,E10.4,2X,E10.4,1X,
+      I3,1X,F6.3)
31      FORMAT(2X,'DE(I,K+1)=' ,E10.4,7X,'AT I=' ,I3)
33      FORMAT(2X,'RE=' ,E10.4)
34      FORMAT(2X,'PR=' ,E10.4)
35      FORMAT(2X,'ANEU(I)=' ,E10.4,2X,'AT I=' ,I3)
36      FORMAT(2X,'ANWU(I)=' ,E10.4,2X,'AT I=' ,I3)
41      FORMAT(2X,'AEO(I)=' ,E10.4,2X,'AT I=' ,I3)
42      FORMAT(2X,'AEU1(I)=' ,E10.4,2X,'AT I=' ,I3)
43      FORMAT(2X,'B(I)=' ,E10.4,2X,'AT I=' ,I3)
47      FORMAT(2X,'PEU(I)=' ,E10.4,2X,'AT I=' ,I3)
48      FORMAT(2X,'PWU(I)=' ,E10.4,2X,'AT I=' ,I3)
51      FORMAT(2X,'FEU(I)=' ,E10.4,2X,'AT I=' ,I3)
53      FORMAT(2X,'FWU(I)=' ,E10.4,2X,'AT I=' ,I3)
61      FORMAT(2X,'AE(I)=' ,E10.4,2X,'AT I=' ,I3)
62      FORMAT(2X,'AW(I)=' ,E10.4,2X,'AT I=' ,I3)
63      FORMAT(2X,'AP(I)=' ,E10.4,2X,'AT I=' ,I3)
64      FORMAT(2X,'BP(I)=' ,E10.4,2X,'AT I=' ,I3)
66      FORMAT(2X,'PE(I)=' ,E10.4,2X,'AT I=' ,I3)
67      FORMAT(2X,'PW(I)=' ,E10.4,2X,'AT I=' ,I3)

```

```

68     FORMAT(2X,'FE(I)=' ,E10.4,2X,'AT I=' ,I3)
69     FORMAT(2X,'FW(I)=' ,E10.4,2X,'AT I=' ,I3)
71     FORMAT(2X,'APT1(I)=' ,E10.4,2X,'AT I=' ,I3)
72     FORMAT(2X,'APTO(I)=' ,E10.4,2X,'AT I=' ,I3)
73     FORMAT(2X,'ATW(I)=' ,E10.4,2X,'AT I=' ,I3)
74     FORMAT(2X,'ATE(I)=' ,E10.4,2X,'AT I=' ,I3)
76     FORMAT(2X,'BT(I)=' ,E10.4,2X,'AT I=' ,I3)
77     FORMAT(2X,'BMS(I)=' ,E10.4,2X,'AT I=' ,I3)
84     FORMAT(2X,'U(I,K+3)=' ,E10.4,2X,'AT I=' ,I3)
85     FORMAT(2X,'U(I-1,K+3)=' ,E10.4,2X,'AT I=' ,I3)
86     FORMAT(2X,'P(I,K+3)=' ,E10.4,2X,'AT I=' ,I3)
87     FORMAT(2X,'P(I,K+1)=' ,E10.4,2X,'AT I=' ,I3)
92     FORMAT(2X,E12.4,2X,I3,2X,E12.4)
93     FORMAT(2X,'T(I,K+2)=' ,E10.4,1X,'
+     AT I=' ,I3,2X,'AT TM=' ,E10.4)
94     FORMAT(2X,E12.5,2X,I3,2X,E12.4)

C
C PRINTING THE RESULTS
      M2=M2+1
      M=M+1
      IF (M2 .EQ. Z2) THEN
        WRITE(7,*)T(7,K+2),TM
        M2=0
      END IF
      IF (M .EQ. Z) THEN
        WRITE(*,14)TM
        WRITE(*,15)EROR
        WRITE(*,*)DTDx,TM,DTDxR
        WRITE(7,*)DTDx,TM,DTDxR

C
      DO 130 I=1,N,X
        WRITE(*,*)T(I,K+2),DE(I,K+2),P(I,K+2),I,TM
        WRITE(7,*)T(I,K+2),DE(I,K+1),P(I,K+1),I,TM
        WRITE(7,31)DE(I,K+1),I
        WRITE(*,93)T(I,K+2),I,TM
        WRITE(*,94)(U(I,K+1)*1E4),I,TM
        WRITE(7,*)(U(I,K+1)*1E4),TM
130    CONTINUE
      I=N
      WRITE(7,92)T(I,K+2),I,TM
      WRITE(*,31)DE(I,K+1),I
      WRITE(*,93)T(I,K+2),I
      M=0
      END IF

C
C SETTING PRESENT VARIABLES EQ. TO FUTURE VARIABLES
89    DO 140 I=1,N,1
      T(I,K)=T(I,K+2)
      T(I,K+1)=T(I,K+2)
      DE(I,K)=DE(I,K+1)
      P(I,K)=P(I,K+1)

```

```
140    CONTINUE
      DO 150 I=1,Q,1
        U(I,K)=U(I,K+1)
150    CONTINUE
30     CONTINUE
C
      STOP
      END
```

APPENDIX D

TAC2D.FOR COMPUTER CODE

```

C*****
C*                                PROGRAM NAME : 2DTAC.FOR          *
C*                                PROGRAMMER   : JOHN RAULSTON JR.   *
C*                                                                 *
C* THIS IS A FINITE DIFFERENCE PROGRAM TO SOLVE                   *
C* THE PROBLEM OF THERMOACOUSTIC CONVECTION (TAC) HEAT           *
C* TRANSFER, AND MIXED MODE TAC AND FREE CONVECTION WITHIN      *
C* AN ENCLOSED RECTANGULAR CAVITY WITH SPECIFIED                 *
C* TEMPERATURE BOUNDARY CONDITIONS. THE ALGORITHM USED IN      *
C* THIS PROGRAM IS BASED ON THE SIMPLER (SEMI IMPLICIT          *
C* METHOD FOR PRESSURE LINKED EQUATIONS) ALGORITHM               *
C* DEVELOPED BY S.V. PATANKAR AND D.B. SPALDING AND              *
C* DESCRIBED IN 'NUMERICAL HEAT TRANSFER AND FLUID FLOW' BY     *
C* S.V. PATANKAR; HEMISPHERE PUBLISHING MCGRAW HILL BOOK        *
C* COMPANY 1980.                                                 *
C* THE DISCRETIZED EQUATIONS USED IN THIS PROGRAM WERE          *
C* DEVELOPED BY INTEGRATION OF THE GOVERNING EQUATIONS          *
C* ACROSS A FINITE CONTROL VOLUME AND THROUGH A FINITE TIME*   *
C* STEP.                                                         *
C*                                                                 *
C* VARIABLE DEFINITION:                                          *
C*                                                                 *
C* DT-NON-DIMENSIONAL TIME STEP                                  *
C* DX-NON-DIMENSIONAL SPACE STEP IN THE X-DIRECTION             *
C* DXE-NON-DIMENSIONAL DISTANCE TO THE NEIGHBOR NODE TO THE*   *
C* EAST                                                           *
C* DXW-NON-DIMENSIONAL DISTANCE TO THE NEIGHBOR NODE TO THE*   *
C* WEST                                                           *
C* DY-NON-DIMENSIONAL SPACE STEP IN THE Y-DIRECTION             *
C* DYN-NON-DIMENSIONAL DISTANCE TO THE NEIGHBOR NODE TO THE*   *
C* NORTH                                                          *
C* DYS-NON-DIMENSIONAL DISTANCE TO THE NEIGHBOR NODE TO THE*   *
C* SOUTH                                                          *
C* FTIME-NON-DIMENSIONAL TIME TO WHICH THE PROGRAM IS           *
C* ALLOWED TO RUN                                                 *
C* TIME-NON-DIMENSIONAL TIME (VARIABLE)                          *
C*                                                                 *
C* FLUID PROPERTIES:                                             *
C*                                                                 *
C* MU- DYNAMIC VISCOSITY (GM/CM-SEC)                             *
C* CV-CONSTANT VOLUME SPECIFIC HEAT (CAL/GM K)                  *
C* COND-THERMAL CONDUCTIVITY (CAL/CM K SEC)                      *
C* GAMMA-RATIO OF SPECIFIC HEATS                                 *
C* PEO-INITIAL PRESSURE (DYNES/CM**2)                            *
C* TEO-INITIAL TEMPERATURE (K)                                    *
C* DEO-INITIAL DENSITY (GM/CM**3)                                 *
C* SQRTO-THE SQUARE ROOT OF IDEAL GAS CONST. MULTIPLIED BY    *
C* THE INITIAL TEMPERATURE (R*TEO)**0.5 USED AS A              *
C* REFERENCE                                                      *
C* VELOCITY IN THE SCALING OF THE GOVERNING EQN'S.             *
C* GRAVITY-GRAVITATIONAL CONSTANT (CM/SEC**2)                  *

```

```

C*                                     *
C* GEOMETRIC VARIABLES                                     *
C*                                     *
C* L-SPACE OF THE RECTANGULAR CAVITY IN THE X-DIR (CM)   *
C* H-SPACE OF THE RECTANGULAR CAVITY IN THE Y-DIR (CM)   *
C* THETA-ANGULAR DISPALCEMENT OF THE RECTRANGULAR CAVITY *
C*      WITH RESPECT TO THE X-AXIS                         *
C*                                     *
C* BOUNDARY VARIABLES :                                     *
C*                                     *
C* TEMP1-NON-DIMENSIONAL BOUNDARY TEMPERATURE OF THE LEFT *
C*      WALL CORRESPONDING TO X=0.                         *
C* TEMP2-NON-DIMENSIONAL BOUNDARY TEMPERATURE OF THE TOP OF *
C*      THE ENCLOSURE CORRESPONDING TO Y=HGTH.             *
C* TEMP3-NON-DIMENSIONAL BOUNDARY TEMPERATURE OF THE RIGHT *
C*      WALL CORRESPONDING TO X=LNGLTH.                   *
C* TEMP4-NON-DIMENSIONAL BOUNDARY TEMPERATURE OF THE FLOOR *
C*      OF THE ENCLOSURE CORRESPONDING TO Y=0.             *
C*                                     *
C* NON-DIMENSIONAL GROUPS                                 *
C*                                     *
C* RE-REYNOLDS NUMBER (LNGLTH*DEO*SQRTO/MU)               *
C* PR-PRANDTL NUMBER  (MU*(GAMMA*CV)/COND)                 *
C* INVFR-INVERSE OF THE FROUDE NUMBER (GRAVITY*L)/         *
C*      (SQRTO**2)                                          *
C*                                     *
C* FLOW VARIABLES :                                       *
C*                                     *
C* U(I,J,K)-NON-DIMENSIONAL VELOCITY IN THE X-DIRECTION OF *
C*      NODE I,J AT THE PAST TIME                          *
C* U(I,J,K+1)-NON-DIMENSIONAL VELOCITY IN THE X-DIRECTION *
C*      OF NODE I,J AT A FUTURE TIME                      *
C* U(I,J,K+2)-NON-DIMENSIONAL VELOCITY U^ VELOCITY IN THE *
C*      X-DIR. AT NODE I,J                                *
C* U(I,J,K+3)-NON-DIMENSIONAL U* VELOCITY IN THE X-DIR AT *
C*      NODE I,J                                           *
C*                                     *
C* V(I,J,K)-NON-DIMENSIONAL VELOCITY IN THE Y-DIR.        *
C*      AT NODE I,J AT A PAST TIME                        *
C* V(I,J,K+1)-NON-DIMENSIONAL VELOCITY IN THE Y-DIR. AT *
C*      NODE I,J AT A FUTURE TIME                        *
C* V(I,J,K+2)-NON-DIMENSIONAL V^ VELOCITY IN THE Y-DIR. AT *
C*      NODE I,J                                           *
C* V(I,J,K+3)-NON-DIMENSIONAL V* VELOCITY IN THE Y-DIR. AT *
C*      NODE I,J                                           *
C*                                     *
C* DE(I,J,K)-NON-DIMENSIONAL DENSITY AT NODE I,J, AT A PAST *
C*      TIME                                                *
C* DE(I,J,K+1)-NON-DIMENSIONAL DENSITY AT NODE I,J AT A *
C*      FUTURE TIME                                         *
C* DE(I,J,K+2)-NON-DIMENSIONAL DE* DENSITY AT NODE I,J    *

```

```

C*
C* P(I,J,K)-NON-DIMENSIONAL PRESSURE OF NODE I,J AT A PAST *
C*      TIME *
C* P(I,J,K+1)-NON-DIMENSIONAL PRESSURE OF NODE I,J AT A *
C*      FUTURE TIME *
C* P(I,J,K+2)-NON-DIMENSIONAL P* PRESSURE AT NODE I,J *
C* P(I,J,K+3)-NON-DIMENSIONAL P' PRESSURE AT NODE I,J *
C*
C* T(I,J,K)-NON-DIMENSIONAL TEMPERATURE AT NODE I,J AT A *
C*      PAST TIME *
C* T(I,J,K+1)-NON-DIMENSIONAL TEMPERATURE OF NODE I,J AT A *
C*      FUTURE TIME. *
C*
C* VELOCITY EQN'S VARIABLES *
C*
C* X-MOMENTUM EQN'S *
C*
C* FEX-MASS FLUX LEAVING THE NODAL CONTROL VOLUME TO *
C*      THE EAST *
C* FWX-MASS FLUX ENTERING THE NODAL CONTROL VOLUME FROM THE *
C*      WEST *
C* FNX-MASS FLUX LEAVING THE NODAL CONTROL VOLUME TO THE *
C*      NORTH *
C* FSX-MASS FLUX ENTERING THE NODAL CONTROL VOLUME FROM THE *
C*      SOUTH *
C*
C* PEXE-NODAL PECLET NUMBER FOR THE NODAL NEIGHBOR TO THE *
C*      EAST *
C* PEXW-NODAL PECLET NUMBER FOR THE NODAL NEIGHBOR TO THE *
C*      WEST *
C* PEXN-NODAL PECLET NUMBER FOR THE NODAL NEIGHBOR TO THE *
C*      NORTH *
C* PEXS-NODAL PECLET NUMBER FOR THE NODAL NEIGHBOR TO THE *
C*      SOUTH *
C*
C* AXNE-COEFFICIENT OF NEIGHBOR EAST VELOCITY *
C* AXW-COEFFICIENT OF NEIGHBOR WEST VELOCITY *
C* AXNN-COEFFICIENT OF NEIGHBOR NORTH VELOCITY *
C* AXNS-COEFFICIENT OF NEIGHBOR SOUTH VELOCITY *
C* AXE-COEFFICIENT OF VELOCITY OF INTEREST *
C* AXEO-COEFFICIENT OF VELOCITY OF INTEREST AT A PAST TIME *
C* BX-SOURCE TERM IN THE VELOCITY EQN *
C*
C* Y-MOMENTUM EQN'S *
C*
C* FEY-MASS FLUX LEAVING THE NODAL CONTROL VOLUME TO THE *
C*      EAST *
C* FWY-MASS FLUX ENTERING THE NODAL CONTROL VOLUME FROM THE *
C*      WEST *
C* FNY-MASS FLUX LEAVING THE NODAL CONTROL VOLUME TO THE *
C*      NORTH *

```

```

C* FSY-MASS FLUX ENTERING THE NODAL CONTROL VOLUME FROM THE*
C*   SOUTH                                                    *
C*                                                            *
C* PEYE-NODAL PECLET NUMBER FOR THE NODAL NEIGHBOR TO THE  *
C*   EAST                                                    *
C* PEYW-NODAL PECLET NUMBER FOR THE NODAL NEIGHBOR TO THE  *
C*   WEST                                                    *
C* PEYN-NODAL PECLET NUMBER FOR THE NODAL NEIGHBOR TO THE  *
C*   NORTH                                                   *
C* PEYS-NODAL PECLET NUMBER FOR THE NODAL NEIGHBOR TO THE  *
C*   SOUTH                                                   *
C*                                                            *
C* AYE-COEFFICIENT OF NEIGHBOR EAST VELOCITY               *
C* AYNW-COEFFICIENT OF NEIGHBOR WEST VELOCITY              *
C* AYNN-COEFFICIENT OF NEIGHBOR NORTH VELOCITY             *
C* AYS-COEFFICIENT OF NEIGHBOR SOUTH VELOCITY              *
C* AYN-COEFFICIENT OF VELOCITY OF INTEREST                 *
C* AYN-COEFFICIENT OF VELOCITY OF INTEREST AT A PAST TIME *
C* BY-SOURCE TERM IN THE VELOCITY EQN                     *
C*                                                            *
C* PRESSURE EQUATION VARTIABLES                             *
C*                                                            *
C* de-DY/AXE(I,J)                                           *
C* dw-DY/AXE(I-1,J)                                         *
C* dn-DX/AYN(I,J)                                           *
C* ds-DX/AYN(I,J-1)                                         *
C*                                                            *
C* APE-COEFFICIENT OF NEIGHBOR PRESSURE TO THE EAST        *
C* APW-COEFFICIENT OF NEIGHBOR PRESSURE TO THE WEST        *
C* APN-COEFFICIENT OF NEIGHBOR PRESSURE TO THE NORTH       *
C* APS-COEFFICIENT OF NEIGHBOR PRESSURE TO THE SOUTH       *
C* APP-COEFFICIENT OF PRESSURE OF INTEREST                  *
C* BP-SOURCE TERM FOR THE PRESSURE EQN                     *
C*                                                            *
C* TEMPERATURE EQUATION VARIABLES                           *
C*                                                            *
C* FE-MASS FLOW OUT OF NODAL CONTROL VOLUME TO THE EAST    *
C* FW-MASS FLOW INTO NODAL CONTROL VOLUME FROM THE WEST    *
C* FN-MASS FLOW OUT OF NODAL CONTROL VOLUME TO THE NORTH    *
C* FS- MASS FLOW INTO NODAL CONTROL VOLUME FROM THE SOUTH   *
C*                                                            *
C* PEE-NODAL PECLET NUMBER OF NEIGHBOR EAST                *
C* PEW-NODAL PECLET NUMBER OF NEIGHBOR WEST                *
C* PEN-NODAL PECLET NUMBER OF NEIGHBOR NORTH               *
C* PES-NODAL PECLET NUMBER OF NEIGHBOR SOUTH               *
C*                                                            *
C* AE-COEFFICIENT OF NEIGHBOR TEMPERATURE TO THE EAST      *

```

```

C* AW-COEFFICIENT OF NEIGHBOR TEMPERATURE TO THE WEST      *
C* AN-COEFFICIENT OF NEIGHBOR TEMPERATURE TO THE NORTH     *
C* AS-COEFFICIENT OF NEIGHBOR TEMPERATURE TO THE SOUTH     *
C* AP-COEFFICIENT OF THE TEMPERATURE OF INTEREST           *
C* APO-COEFFICIENT OF THE TEMPERATURE OF INTEREST AT A PAST*
C*     TIME                                                  *
C* B-SOURCE TERM IN THE TEMPERATURE EQUATION               *
C*                                                         *
C*                                                         *
C* MISCELLANEOUS VARIABLES                                  *
C*                                                         *
C* XPOS-X-POSITION USED IN PRINTING OF RESULTS              *
C* YPOS-Y-POSITION USED IN PRINTING OF RESULTS              *
C* PTIME-PRINT TIME; COUNTER USED IN DETERMINING TIMES TO   *
C*     PRINT                                                  *
C* COUNTER'S- MISCELLANEOUS COUNTERS                        *
C* STEP'S-MISCELLANEOUS STEPS                               *
C* EROR- THE TOLERANCE LIMIT FOR CONVERGENCE                *
C* MAXDF-AN INTERIUM VARIABLE FOR THE DETERMINATION OF      *
C*     CONVERGENCE                                           *
C*                                                         *
C*                                                         *
C*                                                         *
C* SUBROUTINE AMAX1(X,Y)                                     *
C* THIS SUBROUTINE IS DESIGNED TO FIND THE LARGER OF TWO    *
C* NUMBERS X OR Y.                                           *
C*                                                         *
C* SUBROUTINE A(X,Y)                                         *
C* THIS SUBROUTINE IS DESIGNED TO PROVIDE A POWER LAW CURVE*
C* FIT FOR A GIVEN PECLET NUMBER X AND TO RETURN THE        *
C* FUNCTION VALUE Y.                                         *
C*****
C
C
C
C* THE PROGRAM BEGINS
C*     IMPLICIT NONE
C
C
C* VARIABLE DEFINITION
C
C     REAL DT,DX,DY,TEO,MU,CV,COND,GAMMA,
C     $     SQRTO,PEO,DEO,RE,PR,INVFR,L,H
C     REAL THETA,FTIME,TIME,EROR,MAXDF,
C     $     GRAVITY,TEMP1,TEMP2,TEMP3,TEMP4
C     REAL DXE,DXW,DYN,DYS,XPOS,YPOS,BRCKT,APEE,PTIME
C
C
C     INTEGER I,J,K,SL,M,N,COUNTER1,COUNTER2,STEP1,STEP2
C
C* DFN PECLET NUMBERS AND MASS FLOWS

```

```

C
    REAL PEXE, PEXW, PEXN, PEXS, FEX, FWX, FNX, FSX
    REAL PEYE, PEYW, PEYS, PEYN
    REAL FEY, FWY, FNY, FSY, PEE, PEW, PEN, PES, FE, FW, FN, FS
C
C* DFN de CONSTANTS USED IN THE PRESSURE EQN'S
C
    REAL DDE, DW, DN, DS
C
C* DFN FLOW VARIABLES
C
    DOUBLE PRECISION U(200,200,4), V(200,200,4),
    $                  DE(200,200,4), P(200,200,4),
    $                  T(200,200,2)
C
C* DFN COEFFICIENTS OF X-MOMENTUM
C
    DOUBLE PRECISION AXEO(200,200), AXE(200,200),
    $                  AXNE(200,200), AXW(200,200),
    $                  AXNN(200,200), AXNS(200,200),
    $                  BX(200,200)
C
C* DFN COEFFICIENTS OF Y-MOMENTUM
C
    DOUBLE PRECISION AYNE(200,200), AYNW(200,200),
    $                  AYNN(200,200), AYS(200,200),
    $                  AYN(200,200), AYNO(200,200),
    $                  BY(200,200)
C
C* DFN COEFFICIENTS OF P EQN'S
C
    DOUBLE PRECISION APE(200,200), APW(200,200),
    $                  APN(200,200), APS(200,200),
    $                  APP(200,200), BP(200,200)
C
C* DFN COEFFICIENTS OF ENERGY EQN
C
    DOUBLE PRECISION AE(200,200), AW(200,200),
    $                  AN(200,200), AS(200,200), AP(100,100),
    $                  APO(200,200), B(200,200)
C
C
C* INITIALIZE VARIABLES
C
    L=15.3
    H=15.3
    THETA=0
C* FLOW PROPERTIES
    TEO=293.0
    DEO=1.2126E-3
    PEO=1.01E6

```

```

MU=1.83E-4
CV=0.1715
COND=6.08E-5
GAMMA=1.4
SQRTO=28001.0
GRAVITY=0.0
C* TIME AND SPACE DIMESIONS AND STEPS
N=100
M=100
K=1
FTIME=10.0
PTIME=10.0
DT=0.01
DX=1.0/N
DY=1.0/M
EROR=1.0E-6
COUNTER1=0
STEP1=51
STEP2=51
C* BOUNDARY CONDITIONS
TEMP1=586.0/TEO
TEMP2=TEO/TEO
TEMP3=TEO/TEO
TEMP4=TEO/TEO
C* NON-DIMENSIONAL GROUPS
RE=(L*DEO*SQRTO)/MU
PR=(MU*GAMMA*CV)/COND
INVFR=(GRAVITY*L)/(SQRTO*SQRTO)
C* INITIAL CONDITIONS
DO 10 SL=1,2,1
DO 20 J=1,M,1
DO 30 I=1,N,1
U(I,J,SL)=0.0
V(I,J,SL)=0.0
P(I,J,SL)=1.0
DE(I,J,SL)=1.0
T(I,J,SL)=1.0
30 CONTINUE
20 CONTINUE
10 CONTINUE
C* INITIALIZING ^,*, ' VALUES
DO 40 SL=3,4,1
DO 50 J=1,M,1
DO 60 I=1,N,1
U(I,J,SL)=0.0
V(I,J,SL)=0.0
P(I,J,SL)=0.0
DE(I,J,SL)=1.0
60 CONTINUE
50 CONTINUE
40 CONTINUE

```

```

C
C* OPEN DATA FILE
C
      OPEN(9, FILE='2DC01.DAT', STATUS='UNKNOWN')
C
C* TIME LOOP
C
      DO 70 TIME = DT, FTIME, DT
5      MAXDF=0.0
C
C*      U^ EQN
C
      DO 80 J=1, M, 1
      DO 90 I=1, (N-1), 1
      IF (J .EQ. 1) THEN
      IF (I .EQ. 1) THEN
      DXE=DX
      DXW=DX/2.0
      DYN=DY
      DYS=DY/2.0
C
      FEX=((DE(I,J,K+1)+DE(I+1,J,K+1))/2.0)
      $      *U(I,J,K+1)
      FWX=0
      FNX=((DE(I,J+1,K+1)+DE(I,J,K+1))/2.0)
      $      *V(I,J,K+1)
      FSX=0
C
      PEXE=FEX*DX/((4.0/3.0)*(1.0/RE))
      PEXW=FWX*DX/((4.0/3.0)*(1.0/RE))
      PEXN=FNX*DYN/((L/H)*(L/H)*(1.0/RE))
      PEXS=FSX*DYS/((L/H)*(L/H)*(1.0/RE))
C
      CALL AMAX1(-FEX, BRCKT)
      CALL A(PEXE, APEE)
      AXNE(I,J)=BRCKT*DY+(4.0/3.0)*(1.0/RE)
      $      *APEE*(DY/DX)
C
      CALL AMAX1(FWX, BRCKT)
      CALL A(PEXW, APEE)
      AXW(I,J)=BRCKT*DY+(4.0/3.0)
      $      *(1.0/RE)*APEE*(DY/DX)
C
      CALL AMAX1(FSX, BRCKT)
      CALL A(PEXS, APEE)
      AXNS(I,J)=BRCKT*DXE+(1.0/RE)
      $      *(L/H)*(L/H)
      $      *APEE*(DXE/DYS)
C
      CALL AMAX1(-FNX, BRCKT)

```

```

CALL A(PEXN,APEE)
AXNN(I,J)=BRCKT*DXE+(1.0/RE)*(L/H)
C      $      *(L/H)*APEE*(DXE/DYN)

AXEO(I,J)=( (DE(I,J,K)+
C      $      DE(I+1,J,K))/2.0)
C      $      *DXE*DY*(1.0/DT)

BX(I,J)=DE(I,J,K+1)*SIN(THETA)
C      $      *DXE*DY*(INVFR)+(1.0/RE)
C      $      *(1.0/3.0)*(V(I+1,J,K+1)-0
C      $      -V(I,J,K+1)+0)+
C      $      AXEO(I,J)*U(I,J,K)

AXE(I,J)=AXNE(I,J)+AXW(I,J)
C      $      +AXNN(I,J)+AXNS(I,J)+
C      $      AXEO(I,J)

U(I,J,K+2)=(AXNE(I,J)*U(I+1,J,K+1)
C      $      +AXW(I,J)*0+
C      $      AXNN(I,J)*U(I,J+1,K+1)
C      $      +AXNS(I,J)*0
C      $      +BX(I,J))/AXE(I,J)

END IF
IF ((I .GT. 1) .AND. (I .LT. (N-1))) THEN
    DXE=DX
    DXW=DX
    DYN=DY
    DYS=DY/2.0
C
    FEX=((DE(I,J,K+1)+DE(I+1,J,K+1))/2.0)
C      $      *U(I,J,K+1)
    FWX=((DE(I-1,J,K+1)+DE(I,J,K+1))/2.0)
C      $      *U(I-1,J,K+1)
    FNX=((DE(I,J+1,K+1)+DE(I,J,K+1))/2.0)
C      $      *V(I,J,K+1)
    FSX=0
C
    PEXE=FEX*DX/((4.0/3.0)*(1.0/RE))
    PEXW=FWX*DX/((4.0/3.0)*(1.0/RE))
    PEXN=FNX*DYN/((L/H)*(L/H)*(1.0/RE))
    PEXS=FSX*DYS/((L/H)*(L/H)*(1.0/RE))
C
    CALL AMAX1(-FEX,BRCKT)
    CALL A(PEXE,APEE)
    AXNE(I,J)=BRCKT*DY+(4.0/3.0)
C      $      *(1.0/RE)*APEE*(DY/DX)
C
    CALL AMAX1(FWX,BRCKT)
    CALL A(PEXW,APEE)
    AXW(I,J)=BRCKT*DY+(4.0/3.0)

```

```

C      $      * (1.0/RE) *APEE* (DY/DX)

      CALL AMAX1 (FSX, BRCKT)
      CALL A (PEXS, APEE)
      AXNS (I, J) = BRCKT * DXE + (1.0/RE)
        $      * (L/H) * (L/H)
        $      *APEE* (DXE/DYS)
C
      CALL AMAX1 (-FNX, BRCKT)
      CALL A (PEXN, APEE)
      AXNN (I, J) = BRCKT * DXE + (1.0/RE) * (L/H)
        $      * (L/H) *APEE* (DXE/DYN)
C
      AXEO (I, J) = ( (DE (I, J, K) +
        $      DE (I+1, J, K) ) / 2.0)
        $      *DXE*DY* (1.0/DT)
C
      BX (I, J) = DE (I, J, K+1) * SIN (THETA) * DXE * DY
        $      * (INVFR) + (1/RE)
        $      * (1.0/3.0) * (V (I+1, J, K+1) - 0
        $      - V (I, J, K+1) + 0) +
        $      AXEO (I, J) * U (I, J, K)
C
      AXE (I, J) = AXNE (I, J) + AXW (I, J)
        $      + AXNN (I, J) + AXNS (I, J) +
        $      AXEO (I, J)
C
      U (I, J, K+2) = (AXNE (I, J) * U (I+1, J, K+1)
        $      + AXW (I, J) * U (I-1, J, K+1) +
        $      AXNN (I, J) * U (I, J+1, K+1)
        $      + AXNS (I, J) * 0 + BX (I, J) )
        $      / AXE (I, J)
      END IF
      IF ( I. EQ. (N-1) ) THEN
        DXE = DX / 2.0
        DXW = DX
        DYN = DY
        DYS = DY / 2.0
C
        FEX = 0
        FWX = ( (DE (I-1, J, K+1) + DE (I, J, K+1) ) / 2.0)
          $      * U (I-1, J, K+1)
        FNX = ( (DE (I, J+1, K+1) + DE (I, J, K+1) ) / 2.0)
          $      * V (I, J, K+1)
        FSX = 0
C
        PEXE = FEX * DX / ( (4.0/3.0) * (1.0/RE) )
        PEXW = FWX * DX / ( (4.0/3.0) * (1.0/RE) )
        PEXN = FNX * DYN / ( (L/H) * (L/H) * (1.0/RE) )
        PEXS = FSX * DYS / ( (L/H) * (L/H) * (1.0/RE) )
C

```


		FWX=0
	\$	FNX=((DE(I,J+1,K+1)+DE(I,J,K+1))/2.0)
		*V(I,J,K+1)
	\$	FSX=((DE(I,J,K+1)+DE(I,J-1,K+1))/2.0)
		*V(I,J-1,K+1)
C		
		PEXE=FEX*DX/((4.0/3.0)*(1.0/RE))
		PEXW=FWX*DX/((4.0/3.0)*(1.0/RE))
		PEXN=FNX*DYN/((L/H)*(L/H)*(1.0/RE))
		PEXS=FSX*DYS/((L/H)*(L/H)*(1.0/RE))
C		
		CALL AMAX1(-FEX,BRCKT)
		CALL A(PEXE,APEE)
	\$	AXNE(I,J)=BRCKT*DY+(4.0/3.0)
		* (1.0/RE) *APEE*(DY/DX)
C		
		CALL AMAX1(FWX,BRCKT)
		CALL A(PEXW,APEE)
	\$	AXW(I,J)=BRCKT*DY+(4.0/3.0)
		* (1.0/RE) *APEE*(DY/DX)
C		
		CALL AMAX1(FSX,BRCKT)
		CALL A(PEXS,APEE)
	\$	AXNS(I,J)=BRCKT*DXE+(1.0/RE)*(L/H)
		* (L/H) *APEE*(DXE/DYS)
C		
		CALL AMAX1(-FNX,BRCKT)
		CALL A(PEXN,APEE)
	\$	AXNN(I,J)=BRCKT*DXE+(1.0/RE)*(L/H)
		* (L/H) *APEE*(DXE/DYN)
C		
		AXEO(I,J)=((DE(I,J,K)
	\$	+DE(I+1,J,K))/2.0)
	\$	*DXE*DY*(1.0/DT)
C		
		BX(I,J)=DE(I,J,K+1)*SIN(THETA)
	\$	*DXE*DY*(INVFR)+(1.0/RE)
	\$	* (1.0/3.0) * (V(I+1,J,K+1)
	\$	-V(I+1,J-1,K+1)
	\$	-V(I,J,K+1)+V(I,J-1,K+1))+
	\$	AXEO(I,J)*U(I,J,K)
C		
		AXE(I,J)=AXNE(I,J)+AXW(I,J)
	\$	+AXNN(I,J)+AXNS(I,J)+
	\$	AXEO(I,J)
C		
		U(I,J,K+2)=(AXNE(I,J)
	\$	*U(I+1,J,K+1)+AXW(I,J)*0+
	\$	AXNN(I,J)*U(I,J+1,K+1)
	\$	+AXNS(I,J)*U(I,J-1,K+1)
	\$	+BX(I,J))/AXE(I,J)

```

END IF
IF (( I .GT. 1) .AND. (I .LT. (N-1))) THEN
  DXE=DX
  DXW=DX
  DYN=DY
  DYS=DY

C
$
  FEX=( (DE(I,J,K+1)+DE(I+1,J,K+1))/2.0)
    *U(I,J,K+1)
$
  FWX=( (DE(I-1,J,K+1)+DE(I,J,K+1))/2.0)
    *U(I-1,J,K+1)
$
  FNX=( (DE(I,J+1,K+1)+DE(I,J,K+1))/2.0)
    *V(I,J,K+1)
$
  FSX=( (DE(I,J-1,K+1)+DE(I,J,K+1))/2.0)
    *V(I,J-1,K+1)

C
  PEXE=FEX*DX/((4.0/3.0)*(1.0/RE))
  PEXW=FWX*DX/((4.0/3.0)*(1.0/RE))
  PEXN=FNX*DYN/((L/H)*(L/H)*(1.0/RE))
  PEXS=FSX*DYS/((L/H)*(L/H)*(1.0/RE))

C
  CALL AMAX1(-FEX,BRCKT)
  CALL A(PEXE,APEE)
$
  AXNE(I,J)=BRCKT*DY+(4.0/3.0)
    *(1.0/RE)*APEE*(DY/DX)

C
  CALL AMAX1(FWX,BRCKT)
  CALL A(PEXW,APEE)
$
  AXW(I,J)=BRCKT*DY+(4.0/3.0)
    *(1.0/RE)*APEE*(DY/DX)

C
  CALL AMAX1(FSX,BRCKT)
  CALL A(PEXS,APEE)
$
  AXNS(I,J)=BRCKT*DXE+(1.0/RE)*(L/H)
    *(L/H)*APEE*(DXE/DYS)

C
  CALL AMAX1(-FNX,BRCKT)
  CALL A(PEXN,APEE)
$
  AXNN(I,J)=BRCKT*DXE+(1.0/RE)*(L/H)
    *(L/H)*APEE*(DXE/DYN)

C
  AXEO(I,J)=( (DE(I,J,K)
    +DE(I+1,J,K))/2.0)
    *DXE*DY*(1.0/DT)

C
  BX(I,J)=DE(I,J,K+1)*SIN(THETA)*DXE*DY
    *(INFR)+(1.0/RE)
    *(1.0/3.0)*(V(I+1,J,K+1)
    -V(I+1,J-1,K+1)
    -V(I,J,K+1)+V(I,J-1,K+1))+
    AXEO(I,J)*U(I,J,K)
$
$
$
$
$

```

```

C
$
$
C
$
$
$
$
END IF
IF (I .EQ. (N-1)) THEN
    DXE=DX/2.0
    DXW=DX
    DYN=DY
    DYS=DY
C
FEX=0
FWX=((DE(I-1,J,K+1)+DE(I,J,K+1))/2.0)
$
    *U(I-1,J,K+1)
FNX=((DE(I,J+1,K+1)+DE(I,J,K+1))/2.0)
$
    *V(I,J,K+1)
FSX=((DE(I,J-1,K+1)+DE(I,J,K+1))/2.0)
$
    *V(I,J-1,K+1)
C
PEXE=FEX*DX/((4.0/3.0)*(1.0/RE))
PEXW=FWX*DX/((4.0/3.0)*(1.0/RE))
PEXN=FNX*DYN/((L/H)*(L/H)*(1.0/RE))
PEXS=FSX*DYS/((L/H)*(L/H)*(1.0/RE))
C
CALL AMAX1(-FEX,BRCKT)
CALL A(PEXE,APEE)
AXNE(I,J)=BRCKT*DY+(4.0/3.0)
$
    *(1.0/RE)*APEE*(DY/DX)
C
CALL AMAX1(FWX,BRCKT)
CALL A(PEXW,APEE)
AXW(I,J)=BRCKT*DY+(4.0/3.0)
$
    *(1.0/RE)*APEE*(DY/DX)
C
CALL AMAX1(FSX,BRCKT)
CALL A(PEXS,APEE)
AXNS(I,J)=BRCKT*DXE+(1.0/RE)*(L/H)
$
    *(L/H)*APEE*(DXE/DYS)
C
CALL AMAX1(-FNX,BRCKT)
CALL A(PEXN,APEE)
AXNN(I,J)=BRCKT*DXE+(1.0/RE)*(L/H)
$
    *(L/H)*APEE*(DXE/DYN)
C
AXEO(I,J)=((DE(I,J,K)

```

```

C      $      +DE(I+1,J,K))/2.0)
      $      *DXE*DY*(1.0/DT)

      BX(I,J)=DE(I,J,K+1)*SIN(THETA)
      $      *DXE*DY*(INVFR)+(1.0/RE)
      $      *(1.0/3.0)*(V(I+1,J,K+1)
      $      -V(I+1,J-1,K+1)
      $      -V(I,J,K+1)+V(I,J-1,K+1))+
      $      AXEO(I,J)*U(I,J,K)

C      AXE(I,J)=AXNE(I,J)+AXW(I,J)
      $      +AXNN(I,J)+AXNS(I,J)+
      $      AXEO(I,J)

C      U(I,J,K+2)=(AXNE(I,J)*0
      $      +AXW(I,J)*U(I-1,J,K+1)+
      $      AXNN(I,J)*U(I,J+1,K+1)
      $      +AXNS(I,J)*U(I,J-1,K+1)
      $      +BX(I,J))/AXE(I,J)

      END IF
      END IF
      IF ( J .EQ. M) THEN
        IF ( I .EQ. 1) THEN
          DXE=DX
          DXW=DX/2.0
          DYN=DY/2.0
          DYS=DY

C      FEX=((DE(I,J,K+1)+DE(I+1,J,K+1))/2.0)
      $      *U(I,J,K+1)
      FWX=0
      FNX=0
      FSX=((DE(I,J-1,K+1)+DE(I,J,K+1))/2.0)
      $      *V(I,J-1,K+1)

C      PEXE=FEX*DX/((4.0/3.0)*(1.0/RE))
      PEXW=FWX*DX/((4.0/3.0)*(1.0/RE))
      PEXN=FNX*DYN/((L/H)*(L/H)*(1.0/RE))
      PEXS=FSX*DYS/((L/H)*(L/H)*(1.0/RE))

C      CALL AMAX1(-FEX,BRCKT)
      CALL A(PEXE,APEE)
      AXNE(I,J)=BRCKT*DY+(4.0/3.0)
      $      *(1.0/RE)*APEE*(DY/DX)

C      CALL AMAX1(FWX,BRCKT)
      CALL A(PEXW,APEE)
      AXW(I,J)=BRCKT*DY+(4.0/3.0)
      $      *(1.0/RE)*APEE*(DY/DX)

C      CALL AMAX1(FSX,BRCKT)

```

```

C      $      CALL A(PEXS,APEE)
      $      AXNS(I,J)=BRCKT*DXE+(1.0/RE)*(L/H)
      $      * (L/H)*APEE*(DXE/DYS)

C      $      CALL AMAX1(-FNX,BRCKT)
      $      CALL A(PEXN,APEE)
      $      AXNN(I,J)=BRCKT*DXE+(1.0/RE)*(L/H)
      $      * (L/H)*APEE*(DXE/DYN)

C      $      AXEO(I,J)=(DE(I,J,K)
      $      +DE(I+1,J,K))/2.0
      $      *DXE*DY*(1.0/DT)

C      $      BX(I,J)=DE(I,J,K+1)*SIN(THETA)
      $      *DXE*DY*(INVFR)+(1.0/RE)
      $      *(1.0/3.0)*(0-V(I+1,J-1,K+1)-0
      $      +V(I,J-1,K+1)
      $      )+AXEO(I,J)*U(I,J,K)

C      $      AXE(I,J)=AXNE(I,J)+AXW(I,J)
      $      +AXNN(I,J)+AXNS(I,J)+
      $      AXEO(I,J)

C      $      U(I,J,K+2)=(AXNE(I,J)*U(I+1,J,K+1)
      $      +AXW(I,J)*0+
      $      AXNN(I,J)*0
      $      +AXNS(I,J)*U(I,J-1,K+1)
      $      +BX(I,J))/AXE(I,J)

      END IF
      IF ((I .GT. 1) .AND. (I .LT. (N-1))) THEN
        DXE=DX
        DXW=DX
        DYN=DY/2.0
        DYS=DY

C      $      FEX=((DE(I,J,K+1)+DE(I+1,J,K+1))/2.0)
      $      *U(I,J,K+1)
      $      FWX=((DE(I-1,J,K+1)+DE(I,J,K+1))/2.0)
      $      *U(I-1,J,K+1)
      $      FSX=((DE(I,J-1,K+1)+DE(I,J,K+1))/2.0)
      $      *V(I,J-1,K+1)
      $      FNX=0

C      $      PEXE=FEX*DX/((4.0/3.0)*(1.0/RE))
      $      PEXW=FWX*DX/((4.0/3.0)*(1.0/RE))
      $      PEXN=FNX*DYN/((L/H)*(L/H)*(1.0/RE))
      $      PEXS=FSX*DYS/((L/H)*(L/H)*(1.0/RE))

C      $      CALL AMAX1(-FEX,BRCKT)
      $      CALL A(PEXE,APEE)
      $      AXNE(I,J)=BRCKT*DY+(4.0/3.0)

```

```

C      $                                *(1.0/RE)*APEE*(DY/DX)

      CALL AMAX1(FWX,BRCKT)
      CALL A(PEXW,APEE)
      AXW(I,J)=BRCKT*DY+(4.0/3.0)
C      $                                *(1.0/RE)*APEE*(DY/DX)

      CALL AMAX1(FSX,BRCKT)
      CALL A(PEXS,APEE)
      AXNS(I,J)=BRCKT*DXE
C      $                                +(1.0/RE)*(L/H)*(L/H)*
C      $                                APEE*(DXE/DYS)

      CALL AMAX1(-FNX,BRCKT)
      CALL A(PEXN,APEE)
      AXNN(I,J)=BRCKT*DXE+(1.0/RE)*(L/H)
C      $                                *(L/H)*APEE*(DXE/DYN)

      AXEO(I,J)=( (DE(I,J,K)
C      $                                +DE(I+1,J,K))/2.0)
C      $                                *DXE*DY*(1.0/DT)

      BX(I,J)=DE(I,J,K+1)*SIN(THETA)*DXE*DY
C      $                                *(INVFR)+(1.0/RE)
C      $                                *(1.0/3.0)*(0-V(I+1,J-1,K+1)-0
C      $                                +V(I,J-1,K+1))+
C      $                                AXEO(I,J)*U(I,J,K)

      AXE(I,J)=AXNE(I,J)+AXW(I,J)
C      $                                +AXNN(I,J)+AXNS(I,J)+
C      $                                AXEO(I,J)

      U(I,J,K+2)=(AXNE(I,J)*U(I+1,J,K+1)
C      $                                +AXW(I,J)*U(I-1,J,K+1)+
C      $                                AXNS(I,J)*U(I,J-1,K+1)
C      $                                +AXNN(I,J)*0
C      $                                +BX(I,J))/AXE(I,J)

      END IF
      IF (I.EQ.(N-1)) THEN
C      $                                DXE=DX/2.0
C      $                                DXW=DX
C      $                                DYN=DY/2.0
C      $                                DYS=DY

      FEX=0
      FWX=( (DE(I-1,J,K+1)+DE(I,J,K+1))/2.0)
C      $                                *U(I-1,J,K+1)
      FSX=( (DE(I,J-1,K+1)+DE(I,J,K+1))/2.0)
C      $                                *V(I,J-1,K+1)
      FNX=0
C

```

```

PEXE=FEX*DX/((4.0/3.0)*(1.0/RE))
PEXW=FWX*DX/((4.0/3.0)*(1.0/RE))
PEXN=FNX*DYN/((L/H)*(L/H)*(1.0/RE))
PEXS=FSX*DYS/((L/H)*(L/H)*(1.0/RE))

C
CALL AMAX1(-FEX,BRCKT)
CALL A(PEXE,APEE)
AXNE(I,J)=BRCKT*DY+(4.0/3.0)
$          *(1.0/RE)*APEE*(DY/DX)
C
CALL AMAX1(FWX,BRCKT)
CALL A(PEXW,APEE)
AXW(I,J)=BRCKT*DY+(4.0/3.0)
$          *(1.0/RE)*APEE*(DY/DX)
C
CALL AMAX1(FSX,BRCKT)
CALL A(PEXS,APEE)
AXNS(I,J)=BRCKT*DYE+(1.0/RE)*(L/H)
$          *(L/H)*APEE*(DYE/DYS)
C
CALL AMAX1(-FNX,BRCKT)
CALL A(PEXN,APEE)
AXNN(I,J)=BRCKT*DYE+(1.0/RE)*(L/H)
$          *(L/H)*APEE*(DYE/DYN)
C
AXEO(I,J)=(DE(I,J,K)
$          +DE(I+1,J,K))/2.0
$          *DYE*DY*(1.0/DT)
C
BX(I,J)=DE(I,J,K+1)*SIN(THETA)*DYE*DY
$          *(INVFR)+(1.0/RE)
$          *(1.0/3.0)*(0-V(I+1,J-1,K+1)
$          -0+V(I,J-1,K+1))+
$          AXEO(I,J)*U(I,J,K)
C
AXE(I,J)=AXNE(I,J)+AXW(I,J)
$          +AXNN(I,J)+AXNS(I,J)+
$          AXEO(I,J)
C
U(I,J,K+2)=(AXNE(I,J)*0
$          +AXW(I,J)*U(I-1,J,K+1)+
$          AXNS(I,J)*U(I,J-1,K+1)
$          +AXNN(I,J)*0
$          +BX(I,J))/AXE(I,J)
END IF
END IF
90      CONTINUE
80      CONTINUE
C
C
C*      V^ EQN

```

```

C      DO 100 J=1,(M-1),1
      DO 110 I=1,N,1
      IF (J .EQ. 1) THEN
      IF (I .EQ. 1) THEN
      DXE=DX
      DXW=DX/2.0
      DYN=DY
      DYS=DY/2.0
C
      FEY=((DE(I+1,J,K+1)+DE(I,J,K+1))/2.0)
      $      *U(I,J,K+1)
      FWY=0
      FNY=((DE(I,J+1,K+1)+DE(I,J,K+1))/2.0)
      $      *V(I,J,K+1)
      FSY=0
C
      PEYE=FEY*DXE/(1.0/RE)
      PEYW=FWY*DXW/(1.0/RE)
      PEYN=FNY*DY/((4.0/3.0)*(L/H)
      $      *(L/H)*(1.0/RE))
      PEYS=FSY*DY/((4.0/3.0)*(L/H)
      $      *(L/H)*(1.0/RE))
C
      CALL AMAX1(-FEY,BRCKT)
      CALL A(PEYE,APEE)
      AYNE(I,J)=BRCKT*DYN+(1.0/RE)
      $      *APEE*(DYN/DXE)
C
      CALL AMAX1(FWY,BRCKT)
      CALL A(PEYW,APEE)
      AYNW(I,J)=BRCKT*DYN+(1.0/RE)
      $      *APEE*(DYN/DXW)
C
      CALL AMAX1(-FNY,BRCKT)
      CALL A(PEYN,APEE)
      AYNN(I,J)=BRCKT*DX+(1.0/RE)
      $      *(4.0/3.0)*(L/H)
      $      *(L/H)*APEE*(DX/DY)
C
      CALL AMAX1(FSY,BRCKT)
      CALL A(PEYS,APEE)
      AYS(I,J)=BRCKT*DX+(4.0/3.0)
      $      *(1.0/RE)*(L/H)
      $      *(L/H)*APEE*(DX/DY)
C
      AYNO(I,J)=((DE(I,J,K)+DE(I,J+1,K))/2.0)
      $      *DYN*DX*(1.0/DT)
C
      BY(I,J)=-DE(I,J,K+1)*COS(THETA)*DYN*DX
      $      *(L/H)*(INFR)+

```

```

$          (1.0/RE) * (1.0/3.0) * (L/H) * (L/H)
$          * (U(I,J+1,K+1) - 0
$          - U(I,J,K+1) + 0)
$          + AYN0(I,J) * V(I,J,K)
C
$          AYN(I,J) = AYN0(I,J) + AYS(I,J) + AYNE(I,J)
$          + AYNW(I,J) + AYN0(I,J)
C
$          V(I,J,K+2) = (AYN0(I,J) * V(I,J+1,K+1)
$          + AYS(I,J) * 0
$          + AYNE(I,J) * V(I+1,J,K+1)
$          + AYNW(I,J) * 0
$          + BY(I,J)) / AYN(I,J)
END IF
IF ((I .GT. 1) .AND. (I .LT. N)) THEN
DXE=DX
DXW=DX
DYN=DY
DYS=DY/2.0
C
$          FEY = ((DE(I+1,J,K+1) + DE(I,J,K+1)) / 2.0)
$          * U(I,J,K+1)
$          FWY = ((DE(I-1,J,K+1) + DE(I,J,K+1)) / 2.0)
$          * U(I-1,J,K+1)
$          FNY = ((DE(I,J+1,K+1) + DE(I,J,K+1)) / 2.0)
$          * V(I,J,K+1)
$          FSY = 0
C
$          PEYE = FEY * DXE / (1.0/RE)
$          PEYW = FWY * DXW / (1.0/RE)
$          PEYN = FNY * DY / ((4.0/3.0) * (L/H)
$          * (L/H) * (1.0/RE))
$          PEYS = FSY * DY / ((4.0/3.0) * (L/H)
$          * (L/H) * (1.0/RE))
C
$          CALL AMAX1(-FEY, BRCKT)
$          CALL A(PEYE, APEE)
$          AYNE(I,J) = BRCKT * DYN + (1.0/RE)
$          * APEE * (DYN/DXE)
C
$          CALL AMAX1(FWY, BRCKT)
$          CALL A(PEYW, APEE)
$          AYNW(I,J) = BRCKT * DYN + (1.0/RE)
$          * APEE * (DYN/DXW)
C
$          CALL AMAX1(-FNY, BRCKT)
$          CALL A(PEYN, APEE)
$          AYN0(I,J) = BRCKT * DX + (1.0/RE)
$          * (4.0/3.0) * (L/H)
$          * (L/H) * APEE * (DX/DY)
C

```

```

CALL AMAX1(FSY, BRCKT)
CALL A(PEYS, APEE)
AYS(I, J) = BRCKT * DX + (4.0/3.0)
$          * (1.0/RE) * (L/H)
$          * (L/H) * APEE * (DX/DY)
C
AYNO(I, J) = ((DE(I, J, K) + DE(I, J+1, K)) / 2.0)
$          * DYN * DX * (1.0/DT)
C
BY(I, J) = -DE(I, J, K+1) * COS(THETA) * DYN * DX
$          * (L/H) * (INVFR) +
$          (1.0/RE) * (1.0/3.0) * (L/H) * (L/H)
$          * (U(I, J+1, K+1) - U(I-1, J+1, K+1)
$          - U(I, J, K+1) + U(I-1, J, K+1))
$          + AYNO(I, J) * V(I, J, K)
C
AYN(I, J) = AYNN(I, J) + AYS(I, J) + AYNE(I, J)
$          + AYNW(I, J) + AYNO(I, J)
C
V(I, J, K+2) = (AYNN(I, J)
$          * V(I, J+1, K+1) + AYS(I, J) * 0
$          + AYNE(I, J) * V(I+1, J, K+1)
$          + AYNW(I, J) * V(I-1, J, K+1)
$          + BY(I, J)) / AYN(I, J)
END IF
IF (I .EQ. N) THEN
DXE = DX / 2.0
DXW = DX
DYN = DY
DYS = DY / 2.0
C
FEY = 0
FWY = ((DE(I-1, J, K+1) + DE(I, J, K+1)) / 2.0)
$      * U(I-1, J, K+1)
FNY = ((DE(I, J+1, K+1) + DE(I, J, K+1)) / 2.0)
$      * V(I, J, K+1)
FSY = 0
C
PEYE = FEY * DXE / (1.0/RE)
PEYW = FWY * DXW / (1.0/RE)
PEYN = FNY * DY / ((4.0/3.0)
$      * (L/H) * (L/H) * (1.0/RE))
PEYS = FSY * DY / ((4.0/3.0) * (L/H)
$      * (L/H) * (1.0/RE))
C
CALL AMAX1(-FEY, BRCKT)
CALL A(PEYE, APEE)
AYNE(I, J) = BRCKT * DYN + (1.0/RE)
$          * APEE * (DYN/DXE)
C
CALL AMAX1(FWY, BRCKT)

```

```

CALL A(PEYW,APEE)
AYNW(I,J)=BRCKT*DYN+(1.0/RE)
      *APEE*(DYN/DXW)
C
$

CALL AMAX1(-FNY,BRCKT)
CALL A(PEYN,APEE)
AYNN(I,J)=BRCKT*DX+(1.0/RE)
      *(4.0/3.0)*(L/H)
      *(L/H)*APEE*(DX/DY)
C
$
$

CALL AMAX1(FSY,BRCKT)
CALL A(PEYS,APEE)
AYS(I,J)=BRCKT*DX+(4.0/3.0)
      *(1.0/RE)*(L/H)
      *(L/H)*APEE*(DX/DY)
C
$
$

AYNO(I,J)=((DE(I,J,K)+DE(I,J+1,K))/2.0)
      *DYN*DX*(1.0/DT)
C
$

BY(I,J)=-DE(I,J,K+1)*COS(THETA)*DYN*DX
      *(L/H)*(INVFR)+
      (1.0/RE)*(1.0/3.0)*(L/H)*(L/H)
      *(0-U(I-1,J+1,K+1)
      -0+U(I-1,J,K+1))
      +AYNO(I,J)*V(I,J,K)
C
$

AYN(I,J)=AYNN(I,J)+AYS(I,J)+AYNE(I,J)
      +AYNW(I,J)+AYNO(I,J)
C
$

V(I,J,K+2)=(AYNN(I,J)
      *V(I,J+1,K+1)+AYS(I,J)*0
      +AYNW(I,J)*V(I-1,J,K+1)
      +AYNE(I,J)*0
      +BY(I,J))/AYN(I,J)
      END IF
END IF
IF ((J .GT. 1) .AND. (J .LT. (M-1))) THEN
  IF (I .EQ. 1) THEN
    DXE=DX
    DXW=DX/2.0
    DYN=DY
    DYS=DY
C
$

FEY=((DE(I+1,J,K+1)+DE(I,J,K+1))/2.0)
      *U(I,J,K+1)
FWY=0
FNY=((DE(I,J+1,K+1)+DE(I,J,K+1))/2.0)
      *V(I,J,K+1)
FSY=((DE(I,J-1,K+1)+DE(I,J,K+1))/2.0)
      *V(I,J-1,K+1)
C
$

```

```

PEYE=FEY*DXE/(1.0/RE)
PEYW=FWY*DXW/(1.0/RE)
PEYN=FNY*DY/((4.0/3.0)*(L/H)
$      *(L/H)*(1.0/RE))
$
C      PEYS=FSY*DY/((4.0/3.0)*(L/H)
      *(L/H)*(1.0/RE))

      CALL AMAX1(-FEY,BRCKT)
      CALL A(PEYE,APEE)
$      AYNE(I,J)=BRCKT*DYN+(1.0/RE)
      *APEE*(DYN/DXE)
C

      CALL AMAX1(FWY,BRCKT)
      CALL A(PEYW,APEE)
$      AYNW(I,J)=BRCKT*DYN+(1.0/RE)
      *APEE*(DYN/DXW)
C

      CALL AMAX1(-FNY,BRCKT)
      CALL A(PEYN,APEE)
$      AYNN(I,J)=BRCKT*DX+(1.0/RE)
$      *(4.0/3.0)*(L/H)
      *(L/H)*APEE*(DX/DY)
C

      CALL AMAX1(FSY,BRCKT)
      CALL A(PEYS,APEE)
$      AYS(I,J)=BRCKT*DX+(4.0/3.0)*(1.0/RE)
      *(L/H)*(L/H)*APEE*(DX/DY)
C

      AYNO(I,J)=((DE(I,J,K)+DE(I,J+1,K))/2.0)
$      *DYN*DX*(1.0/DT)
C

      BY(I,J)=-DE(I,J,K+1)*COS(THETA)*DYN*DX
$      *(L/H)*(INVFR)+
$      (1.0/RE)*(1.0/3.0)*(L/H)*(L/H)
$      *(U(I,J+1,K+1)-0
$      -U(I,J,K+1)+0)
$      +AYNO(I,J)*V(I,J,K)
C

      AYN(I,J)=AYNN(I,J)+AYS(I,J)+AYNE(I,J)
$      +AYNW(I,J)+AYNO(I,J)
C

      V(I,J,K+2)=(AYNN(I,J)*V(I,J+1,K+1)
$      +AYS(I,J)*V(I,J-1,K+1)
$      +AYNE(I,J)*V(I+1,J,K+1)
$      +AYNW(I,J)*0
$      +BY(I,J))/AYN(I,J)

END IF
IF ((I .GT. 1) .AND. (I .LT. N)) THEN
DXE=DX
DXW=DX
DYN=DY

```

```

C      DYS=DY

      FEY=( (DE(I+1,J,K+1)+DE(I,J,K+1))/2.0)
$      *U(I,J,K+1)
      FWY=( (DE(I-1,J,K+1)+DE(I,J,K+1))/2.0)
$      *U(I-1,J,K+1)
      FNY=( (DE(I,J+1,K+1)+DE(I,J,K+1))/2.0)
$      *V(I,J,K+1)
      FSY=( (DE(I,J-1,K+1)+DE(I,J,K+1))/2.0)
$      *V(I,J-1,K+1)

C      PEYE=FEY*DXE/(1.0/RE)
      PEYW=FWY*DXW/(1.0/RE)
      PEYN=FNY*DY/( (4.0/3.0)
$      *(L/H)*(L/H)*(1.0/RE))
      PEYS=FSY*DY/( (4.0/3.0)
$      *(L/H)*(L/H)*(1.0/RE))

C      CALL AMAX1(-FEY,BRCKT)
      CALL A(PEYE,APEE)
      AYNE(I,J)=BRCKT*DYN+(1.0/RE
$      )*APEE*(DYN/DXE)

C      CALL AMAX1(FWY,BRCKT)
      CALL A(PEYW,APEE)
      AYNW(I,J)=BRCKT*DYN+(1.0/RE
$      )*APEE*(DYN/DXW)

C      CALL AMAX1(-FNY,BRCKT)
      CALL A(PEYN,APEE)
      AYNN(I,J)=BRCKT*DX+(1.0/RE)
$      *(4.0/3.0)*(L/H)
$      *(L/H)*APEE*(DX/DY)

C      CALL AMAX1(FSY,BRCKT)
      CALL A(PEYS,APEE)
      AYS(I,J)=BRCKT*DX+(4.0/3.0)
$      *(1.0/RE)*(L/H)
$      *(L/H)*APEE*(DX/DY)

C      AYNO(I,J)=( (DE(I,J,K)+DE(I,J+1,K))/2.0)
$      *DYN*DX*(1.0/DT)

C      BY(I,J)=-DE(I,J,K+1)*COS(THETA)*DYN
$      *DX*(L/H)*(INVFR)+
$      (1.0/RE)*(1.0/3.0)*(L/H)*(L/H)
$      *(U(I,J+1,K+1)-U(I-1,J+1,K+1)
$      -U(I,J,K+1)+U(I-1,J,K+1))
$      +AYNO(I,J)*V(I,J,K)

C      AYN(I,J)=AYNN(I,J)+AYS(I,J)+AYNE(I,J)

```

```

C      $      +AYNW(I,J)+AYNO(I,J)

      $      V(I,J,K+2)=(AYNN(I,J)*V(I,J+1,K+1)
      $      +AYS(I,J)*V(I,J-1,K+1)
      $      +AYNE(I,J)*V(I+1,J,K+1)
      $      +AYNW(I,J)*V(I-1,J,K+1)
      $      +BY(I,J))/AYN(I,J)

      END IF
      IF (I.EQ. N) THEN
        DXE=DX/2.0
        DXW=DX
        DYN=DY
        DYS=DY

C      FEY=0
      FWY=((DE(I-1,J,K+1)+DE(I,J,K+1))/2.0)
      $      *U(I-1,J,K+1)
      FNY=((DE(I,J+1,K+1)+DE(I,J,K+1))/2.0)
      $      *V(I,J,K+1)
      FSY=((DE(I,J-1,K+1)+DE(I,J,K+1))/2.0)
      $      *V(I,J-1,K+1)

C      PEYE=FEY*DXE/(1.0/RE)
      PEYW=FWY*DXW/(1.0/RE)
      PEYN=FNY*DY/((4.0/3.0)*(L/H)
      $      *(L/H)*(1.0/RE))
      PEYS=FSY*DY/((4.0/3.0)*(L/H)
      $      *(L/H)*(1.0/RE))

C      CALL AMAX1(-FEY,BRCKT)
      CALL A(PEYE,APEE)
      AYNE(I,J)=BRCKT*DYN+(1.0/RE)
      $      *APEE*(DYN/DXE)

C      CALL AMAX1(FWY,BRCKT)
      CALL A(PEYW,APEE)
      AYNW(I,J)=BRCKT*DYN+(1.0/RE)
      $      *APEE*(DYN/DXW)

C      CALL AMAX1(-FNY,BRCKT)
      CALL A(PEYN,APEE)
      AYNN(I,J)=BRCKT*DX
      $      +(1.0/RE)*(4.0/3.0)*(L/H)
      $      *(L/H)*APEE*(DX/DY)

C      CALL AMAX1(FSY,BRCKT)
      CALL A(PEYS,APEE)
      AYS(I,J)=BRCKT*DX+(4.0/3.0)
      $      *(1.0/RE)*(L/H)
      $      *(L/H)*APEE*(DX/DY)

C

```

```

C      $      AYN0(I,J)=( (DE(I,J,K)+DE(I,J+1,K))/2.0)
                        *DYN*DX*(1.0/DT)

C      $      BY(I,J)=-DE(I,J,K+1)*COS(THETA)*DYN*DX
$      $      *(L/H)*(INVFR)+
$      $      (1.0/RE)*(1.0/3.0)*(L/H)*(L/H)
$      $      *(0-U(I-1,J+1,K+1)
$      $      -0+U(I-1,J,K+1))
$      $      +AYNO(I,J)*V(I,J,K)

C      $      AYN(I,J)=AYNN(I,J)+AYS(I,J)+AYNE(I,J)
C      $      +AYNW(I,J)+AYNO(I,J)

C      $      V(I,J,K+2)=(AYNN(I,J)*V(I,J+1,K+1)
$      $      +AYS(I,J)*V(I,J-1,K+1)
$      $      +AYNE(I,J)*0
$      $      +AYNW(I,J)*V(I-1,J,K+1)
$      $      +BY(I,J))/AYN(I,J)

      END IF
      END IF
      IF (J.EQ.(M-1)) THEN
        IF (I.EQ.1) THEN
          DXE=DX
          DXW=DX/2.0
          DYN=DY/2.0
          DYS=DY

C      $      FEY=((DE(I+1,J,K+1)+DE(I,J,K+1))/2.0)
$      $      *U(I,J,K+1)
          FWY=0
          FNY=0
          FSY=((DE(I,J-1,K+1)+DE(I,J,K+1))/2.0)
C      $      *V(I,J-1,K+1)

          PEYE=FEY*DXE/(1.0/RE)
          PEYW=FWY*DXW/(1.0/RE)
          PEYN=FNY*DY/((4.0/3.0)*(L/H)
$      $      *(L/H)*(1.0/RE))
          PEYS=FSY*DY/((4.0/3.0)*(L/H)
C      $      *(L/H)*(1.0/RE))

          CALL AMAX1(-FEY,BRCKT)
          CALL A(PEYE,APEE)
C      $      AYN0(I,J)=BRCKT*DYN+(1.0/RE)
$      $      *APEE*(DYN/DXE)

          CALL AMAX1(FWY,BRCKT)
          CALL A(PEYW,APEE)
C      $      AYNW(I,J)=BRCKT*DYN+(1.0/RE)
$      $      *APEE*(DYN/DXW)

```

```

CALL AMAX1(-FNY, BRCKT)
CALL A(PEYN, APEE)
AYNN(I, J) = BRCKT * DX + (1.0/RE)
$          * (4.0/3.0) * (L/H)
$          * (L/H) * APEE * (DX/DY)
C

CALL AMAX1(FSY, BRCKT)
CALL A(PEYS, APEE)
AYS(I, J) = BRCKT * DX + (4.0/3.0)
$          * (1.0/RE) * (L/H)
$          * (L/H) * APEE * (DX/DY)
C

AYNO(I, J) = ((DE(I, J, K) + DE(I, J+1, K)) / 2.0)
$          * DYN * DX * (1.0/DT)
C

BY(I, J) = -DE(I, J, K+1) * COS(THETA) * DYN * DX
$          * (L/H) * (INVFR) +
$          (1.0/RE) * (1.0/3.0) * (L/H) * (L/H)
$          * (U(I, J+1, K+1) - 0
$          - U(I, J, K+1) + 0)
$          + AYNO(I, J) * V(I, J, K)
C

AYN(I, J) = AYNN(I, J) + AYS(I, J) + AYNE(I, J)
$          + AYNW(I, J) + AYNO(I, J)
C

V(I, J, K+2) = (AYS(I, J)
$          * V(I, J-1, K+1) + AYNN(I, J) * 0
$          + AYNE(I, J) * 0 + AYNW(I, J)
$          * V(I-1, J, K+1)
$          + BY(I, J)) / AYN(I, J)
END IF
IF ((I .GT. 1) .AND. (I .LT. N)) THEN
DXE=DX
DXW=DX
DYN=DY/2.0
DYS=DY
C

FEY=((DE(I+1, J, K+1)+DE(I, J, K+1))/2.0)
$      *U(I, J, K+1)
FWY=((DE(I-1, J, K+1)+DE(I, J, K+1))/2.0)
$      *U(I-1, J, K+1)
FSY=((DE(I, J-1, K+1)+DE(I, J, K+1))/2.0)
$      *V(I, J-1, K+1)
FNY=0
C

PEYE=FEY*DXE/(1.0/RE)
PEYW=FWY*DXW/(1.0/RE)
PEYN=FNY*DY/((4.0/3.0)*(L/H)
$      *(L/H)*(1.0/RE))
PEYS=FSY*DY/((4.0/3.0)*(L/H)
$      *(L/H)*(1.0/RE))

```


		FSY=((DE(I,J-1,K+1)+DE(I,J,K+1))/2.0)
	\$	*V(I,J-1,K+1)
		FNY=0
C		
		PEYE=FEY*DXE/(1.0/RE)
		PEYW=FWY*DXW/(1.0/RE)
	\$	PEYN=FNY*DY/((4.0/3.0)*(L/H)
		(L/H)(1.0/RE))
	\$	PEYS=FSY*DY/((4.0/3.0)*(L/H)
		(L/H)(1.0/RE))
C		
		CALL AMAX1(-FEY,BRCKT)
		CALL A(PEYE,APEE)
	\$	AYNE(I,J)=BRCKT*DYN+(1.0/RE)
		APEE(DYN/DXE)
C		
		CALL AMAX1(FWY,BRCKT)
		CALL A(PEYW,APEE)
	\$	AYNW(I,J)=BRCKT*DYN+(1.0/RE)
		APEE(DYN/DXW)
C		
		CALL AMAX1(-FNY,BRCKT)
		CALL A(PEYN,APEE)
	\$	AYNN(I,J)=BRCKT*DX+(1.0/RE)
		(4.0/3.0)(L/H)
	\$	*(L/H)*APEE*(DX/DY)
C		
		CALL AMAX1(FSY,BRCKT)
		CALL A(PEYS,APEE)
	\$	AYS(I,J)=BRCKT*DX
		+(4.0/3.0)*(1.0/RE)*(L/H)
	\$	*(L/H)*APEE*(DX/DY)
C		
	\$	AYNO(I,J)=((DE(I,J,K)+DE(I,J+1,K))/2.0)
		*DYN*DX*(1.0/DT)
C		
	\$	BY(I,J)=-DE(I,J,K+1)*COS(THETA)*DYN*DX
		(L/H)(INVFR)+
	\$	(1.0/RE)*(1.0/3.0)*(L/H)*(L/H)
	\$	*(0-U(I-1,J+1,K+1)
	\$	-0+U(I-1,J,K+1))
	\$	+AYNO(I,J)*V(I,J,K)
C		
	\$	AYN(I,J)=AYNN(I,J)+AYS(I,J)+AYNE(I,J)
		+AYNW(I,J)+AYNO(I,J)
C		
	\$	V(I,J,K+2)=(AYS(I,J)*V(I,J-1,K+1)
		+AYNN(I,J)*0
	\$	+AYNE(I,J)*0+
	\$	AYNW(I,J)*V(I-1,J,K+1)
	\$	+BY(I,J))/AYN(I,J)


```

$          DE(I,J,K+1))/2.0)
$          *DW*DY
C
$          APN(I,J)=((DE(I,J+1,K+1)
$                   +DE(I,J,K+1))/2.0)*DN*DX
C
$          APS(I,J)=0
C
$          APP(I,J)=APE(I,J)+APW(I,J)
$                   +APN(I,J)+APS(I,J)
C
$          BP(I,J)=(DE(I,J,K)-DE(I,J,K+1))
$                   *DX*DY*(1.0/DT)+
$                   ((DE(I-1,J,K+1)
$                    +DE(I,J,K+1))/2.0)
$                   *U(I-1,J,K+2)
$                   -((DE(I+1,J,K+1)
$                    +DE(I,J,K+1))/2.0)
$                   *U(I,J,K+2))*DY
$                   +(0-((DE(I,J+1,K+1)+
$                    DE(I,J,K+1))/2.0)
$                   *V(I,J,K+2))*DX
C
$          P(I,J,K+1)=(APE(I,J)*P(I+1,J,K+1)
$                    +APW(I,J)*P(I-1,J,K+1)
$                    +APN(I,J)*P(I,J+1,K+1)
$                    +BP(I,J))/APP(I,J)
$
$          END IF
$          IF (I .EQ. N) THEN
C              DW=DY/AXE(I-1,J)
C              DN=DX/AYN(I,J)
C
C              APE(I,J)=0
C
C              APW(I,J)=((DE(I-1,J,K+1)
$                       +DE(I,J,K+1))/2.0)
$                       *DW*DY
C
C              APN(I,J)=((DE(I,J+1,K+1)+
$                       DE(I,J,K+1))/2.0)
$                       *DN*DX
C
C              APS(I,J)=0
C
C              APP(I,J)=APE(I,J)+APW(I,J)
$                       +APN(I,J)+APS(I,J)
C
$              BP(I,J)=(DE(I,J,K)-
$                       DE(I,J,K+1))*DX*DY*(1.0/DT)+
$                       ((DE(I-1,J,K+1)+
$                       DE(I,J,K+1))/2.0)

```

```

C      $      *U(I-1,J,K+2)
      $      -0)*DY
      $      +(0-((DE(I,J+1,K+1)
      $      +DE(I,J,K+1))/2.0)
      $      *V(I,J,K+2))*DX

      P(I,J,K+1)=(APW(I,J)*P(I-1,J,K+1)
      $      +APN(I,J)*P(I,J+1,K+1)
      $      +BP(I,J))/APP(I,J)

      END IF
END IF
IF ((J .GT. 1) .AND. (J .LT. M)) THEN
  IF (I .EQ. 1) THEN
    DDE=DY/AXE(I,J)
    DN=DX/AYN(I,J)
    DS=DX/AYN(I,J-1)

C      APE(I,J)=((DE(I,J,K+1)+
      $      DE(I+1,J,K+1))/2.0)
      $      *DDE*DY

C      APW(I,J)=0

C      APN(I,J)=((DE(I,J+1,K+1)+
      $      DE(I,J,K+1))/2.0)
      $      *DN*DX

C      APS(I,J)=((DE(I,J-2,K+1)+
      $      DE(I,J,K+1))/2.0)
      $      *DS*DX

C      APP(I,J)=APE(I,J)+APW(I,J)+
      $      APN(I,J)+APS(I,J)

C      BP(I,J)=(DE(I,J,K)-DE(I,J,K+1))
      $      *DX*DY*(1.0/DT)+
      $      (0-((DE(I+1,J,K+1)+DE(I,J,K+1))
      $      /2.0)*U(I,J,K+2))*DY
      $      +((DE(I,J-1,K+1)+DE(I,J,K+1))
      $      /2.0)*V(I,J-1,K+2)
      $      -((DE(I,J+1,K+1)+DE(I,J,K+1))
      $      /2.0)*V(I,J,K+2))*DX

C      P(I,J,K+1)=(APE(I,J)*P(I+1,J,K+1)
      $      +APS(I,J)*P(I,J-1,K+1)
      $      +APN(I,J)*P(I,J+1,K+1)
      $      +BP(I,J))/APP(I,J)

      END IF
IF ((I .GT. 1) .AND. (I .LT. N)) THEN
  DDE=DY/AXE(I,J)
  DW=DY/AXE(I-1,J)

```

```

C      DN=DX/AYN(I,J)
      DS=DX/AYN(I,J-1)

C      APE(I,J)=((DE(I,J,K+1)+DE(I+1,J,K+1))
$          /2.0)*DDE*DY

C      APW(I,J)=((DE(I-1,J,K+1)+DE(I,J,K+1))
$          /2.0)*DW*DY

C      APN(I,J)=((DE(I,J+1,K+1)
$          +DE(I,J,K+1))/2.0)*DN*DX

C      APS(I,J)=((DE(I,J-1,K+1)
$          +DE(I,J,K+1))/2.0)*DS*DX

C      APP(I,J)=APE(I,J)+APW(I,J)
$          +APN(I,J)+APS(I,J)

C      BP(I,J)=(DE(I,J,K)-DE(I,J,K+1))
$          *DX*DY*(1.0/DT)+
$          ((DE(I-1,J,K+1)+
$          DE(I,J,K+1))/2.0)
$          *U(I-1,J,K+2)
$          -((DE(I+1,J,K+1)+
$          DE(I,J,K+1))/2.0)
$          *U(I,J,K+2))*DY
$          +(((DE(I,J-1,K+1)
$          +DE(I,J,K+1))/2.0)
$          *V(I,J-1,K+2)
$          -((DE(I,J+1,K+1)
$          +DE(I,J,K+1))/2.0)
$          *V(I,J,K+2))*DX

C      P(I,J,K+1)=(APE(I,J)*P(I+1,J,K+1)
$          +APW(I,J)*P(I-1,J,K+1)
$          +APN(I,J)*P(I,J+1,K+1)
$          +APS(I,J)*P(I,J-1,K+1)
$          +BP(I,J))/APP(I,J)

      END IF
      IF (I .EQ. N) THEN
C      DW=DY/AXE(I-1,J)
C      DN=DX/AYN(I,J)
C      DS=DX/AYN(I,J-1)

C      APE(I,J)=0

C      APW(I,J)=((DE(I-1,J,K+1)
$          +DE(I,J,K+1))/2.0)
$          *DW*DY

C      APN(I,J)=((DE(I,J+1,K+1)

```

```

$                                +DE(I,J,K+1))/2.0)
$                                *DN*DX
C
$                                APS(I,J)=( (DE(I,J-1,K+1)
$                                +DE(I,J,K+1))/2.0)
$                                *DS*DX
C
$                                APP(I,J)=APE(I,J)+APW(I,J)
$                                +APN(I,J)+APS(I,J)
C
$                                BP(I,J)=(DE(I,J,K)-DE(I,J,K+1))
$                                *DX*DY*(1.0/DT)+
$                                ((DE(I-1,J,K+1)+
$                                DE(I,J,K+1))/2.0)
$                                *U(I-1,J,K+2)
$                                -0)*DY
$                                +((DE(I,J-1,K+1)
$                                +DE(I,J,K+1))/2.0)
$                                *V(I,J-1,K+2)
$                                -(DE(I,J+1,K+1)
$                                +DE(I,J,K+1))/2.0)
$                                *V(I,J,K+2))*DX
C
$                                P(I,J,K+1)=(APW(I,J)*P(I-1,J,K+1)
$                                +APN(I,J)*P(I,J+1,K+1)
$                                +APS(I,J)*P(I,J-1,K+1)
$                                +BP(I,J))/APP(I,J)
$                                END IF
$                                END IF
$                                IF (J .EQ. M) THEN
$                                IF (I .EQ. 1) THEN
$                                DDE=DY/AXE(I,J)
$                                DS=DX/AYN(I,J-1)
C
$                                APE(I,J)=( (DE(I,J,K+1)
$                                +DE(I+1,J,K+1))/2.0)
$                                *DDE*DY
C
$                                APW(I,J)=0
C
$                                APN(I,J)=0
C
$                                APS(I,J)=( (DE(I,J-1,K+1)
$                                +DE(I,J,K+1))/2.0)
$                                *DS*DX
C
$                                APP(I,J)=APE(I,J)+APW(I,J)
$                                +APN(I,J)+APS(I,J)
C
$                                BP(I,J)=(DE(I,J,K)-DE(I,J,K+1))
$                                *DX*DY*(1.0/DT)+

```

```

$          (0
$          -( (DE(I+1,J,K+1)+
$          DE(I,J,K+1))/2.0)
$          *U(I,J,K+2)) *DY
$          +(( (DE(I,J-1,K+1)
$          +DE(I,J,K+1))/2.0)
$          *V(I,J-1,K+2)
$          -0) *DX
C
$          P(I,J,K+1)=(APE(I,J)*P(I+1,J,K+1)
$          +APS(I,J)*P(I,J-1,K+1)
$          +BP(I,J))/APP(I,J)
END IF
IF ((I .GT. 1) .AND. (I .LT. N)) THEN
DDE=DY/AXE(I,J)
DW=DY/AXE(I-1,J)
DS=DX/AYN(I,J-1)
C
$          APE(I,J)=( (DE(I,J,K+1)
$          +DE(I+1,J,K+1))/2.0)
$          *DDE*DY
C
$          APW(I,J)=( (DE(I-1,J,K+1)+
$          DE(I,J,K+1))/2.0)
$          *DW*DY
C
$          APN(I,J)=0
C
$          APS(I,J)=( (DE(I,J-1,K+1)
$          +DE(I,J,K+1))/2.0)
$          *DS*DX
C
$          APP(I,J)=APE(I,J)+APW(I,J)
$          +APN(I,J)+APS(I,J)
C
$          BP(I,J)=(DE(I,J,K)-DE(I,J,K+1))
$          *DX*DY*(1.0/DT)+
$          (( (DE(I-1,J,K+1)
$          +DE(I,J,K+1))/2.0)
$          *U(I-1,J,K+2)
$          -( (DE(I+1,J,K+1)+
$          DE(I,J,K+1))/2.0)
$          *U(I,J,K+2)) *DY
$          +(( (DE(I,J-1,K+1)
$          +DE(I,J,K+1))/2.0)
$          *V(I,J-1,K+2)
$          -0) *DX
C
$          P(I,J,K+1)=(APE(I,J)*P(I+1,J,K+1)
$          +APW(I,J)*P(I-1,J,K+1)
$          +APS(I,J)*P(I,J-1,K+1)

```

```

$                                +BP(I,J))/APP(I,J)
END IF
IF (I .EQ. N) THEN
    DW=DY/AXE(I-1,J)
    DS=DX/AYN(I,J-1)
C
C
    APE(I,J)=0
    APW(I,J)=( (DE(I-1,J,K+1)
$              +DE(I,J,K+1))/2.0)
$              *DW*DY
C
    APN(I,J)=0
C
    APS(I,J)=( (DE(I,J-1,K+1)+
$              DE(I,J,K+1))/2.0)
$              *DS*DX
C
    APP(I,J)=APE(I,J)+APW(I,J)
$              +APN(I,J)+APS(I,J)
C
    BP(I,J)=(DE(I,J,K)-DE(I,J,K+1))*
$              DX*DY*(1.0/DT)+
$              ((DE(I-1,J,K+1)+
$              DE(I,J,K+1))/2.0)
$              *U(I-1,J,K+2)
$              -0)*DY
$              +((DE(I,J-1,K+1)+
$              DE(I,J,K+1))/2.0)
$              *V(I,J-1,K+2)
$              -0)*DX
C
    P(I,J,K+1)=(APW(I,J)*P(I-1,J,K+1)
$              +APS(I,J)*P(I,J-1,K+1)
$              +BP(I,J))/APP(I,J)
END IF
END IF
130 CONTINUE
120 CONTINUE
C
C
C    TREATING PRESSURE P(I,J,K+1) AS P* ( P(I,J,K+2))
DO 132 J=1,M,1
    DO 138 I=1,N,1
        P(I,J,K+2)=P(I,J,K+1)
138 CONTINUE
132 CONTINUE
C
C*
C    U* EQN
U(1,1,K+3)=(AXNE(1,1)*U(2,1,K+3)+AXNN(1,1)*U(1,2,K+3))

```

```

$          +AXNS(1,1)*0+AXW(1,1)*0+BX(1,1)
$          +(P(1,1,K+2)-P(2,1,K+2))*DY)/AXE(1,1)
DO 140 I=2,(N-2),1
$          U(I,1,K+3)=(AXNE(I,1)*U(I+1,1,K+3)+AXW(I,1)
$              *U(I-1,1,K+3)
$              +AXNN(I,1)*U(I,2,K+3)+
$              AXNS(I,1)*0+BX(I,1)
$              +(P(I,1,K+2)-P(I+1,1,K+2))*DY)/AXE(I,1)
140 CONTINUE
$          U(N-1,1,K+3)=(AXW(N-1,1)*U(N-2,1,K+3)+AXNE(N-1,1)*0
$              +AXNN(N-1,1)*U(N-1,2,K+3)
$              +AXNS(N-1,1)*0+BX(N-1,1)
$              +(P(N-1,1,K+2)-P(N,1,K+2))
$              *DY)/AXE(N-1,1)
C
DO 150 J=2,(M-1),1
$          U(1,J,K+3)=(AXNE(1,J)*U(2,J,K+3)+AXW(1,J)*0
$              +AXNN(1,J)*U(1,J+1,K+3)+AXNS(1,J)
$              *U(1,J-1,K+3)+BX(1,J)
$              +(P(1,J,K+2)-P(2,J,K+2))*DY)/AXE(1,J)
DO 160 I=1,(N-2),1
$          U(I,J,K+3)=(AXNE(I,J)*U(I+1,J,K+3)
$              +AXW(I,J)*U(I-1,J,K+3)
$              +AXNN(I,J)*U(I,J+1,K+3)+AXNS(I,J)
$              *U(I,J-1,K+3)+BX(I,J)
$              +(P(I,J,K+2)-P(I+1,J,K+2))*
$              DY)/AXE(I,J)
160 CONTINUE
$          U(N-1,J,K+3)=(AXNE(N-1,J)*0+AXW(1,J)*U(N-2,J,K+3)
$              +AXNN(N-1,J)*U(N-1,J+1,K+3)
$              +AXNS(N-1,J)*U(N-1,J-1,K+3)+BX(N-1,J)
$              +(P(N-1,J,K+2)-P(N,J,K+2))*
$              DY)/AXE(N-1,J)
150 CONTINUE
$          U(1,M,K+3)=(AXNE(1,M)*U(2,M,K+3)+AXW(1,J)*0
$              +AXNN(1,M)*0+AXNS(1,M)*
$              U(1,M-1,K+3)+BX(1,M)
$              +(P(1,M,K+2)-P(2,M,K+2))*DY)/AXE(1,M)
DO 170 I=2,(N-2),1
$          U(I,M,K+3)=(AXNE(I,M)
$              *U(I+1,M,K+3)+AXW(I,M)*U(I-1,M,K+3)
$              +AXNN(1,J)*0+AXNS(I,M)
$              *U(I,M-1,K+3)+BX(I,M)
$              +(P(I,M,K+2)-P(I+1,M,K+2))*DY)/AXE(I,M)
170 CONTINUE
$          U(N-1,M,K+3)=(AXNE(N-1,M)*0+AXW(N-1,M)
$              *U(N-2,M,K+3)
$              +AXNN(N-1,M)*0+AXNS(N-1,M)
$              *U(N-1,M-1,K+3)+BX(N-1,M)
$              +(P(N-1,M,K+2)-P(N,M,K+2))*DY)
$              /AXE(N-1,M)

```

C
C*
C

V* EQN

```

V(1,1,K+3)=(AYNE(1,1)*V(2,1,K+3)+AYNW(1,1)*0
$      +AYNN(1,1)*V(1,2,K+3)+AYS(1,1)*0
$      +BY(1,1)+(P(1,1,K+2)-
$      P(1,2,K+2))*DX)/AYN(1,1)
DO 180 I=1,(N-1),1
$      V(I,1,K+3)=(AYNE(I,1)*V(I+1,1,K+3)
$      +AYNW(I,1)*V(I-1,1,K+3)
$      +AYNN(I,1)*V(I,2,K+3)+AYS(I,1)*0
$      +BY(I,1)+(P(I,1,K+2)-P(I,2,K+2))*DX)
$      /AYN(I,1)
180 CONTINUE
$      V(N,1,K+3)=(AYNE(N,1)*0+AYNW(N,1)*V(N-1,1,K+3)
$      +AYNN(N,1)*V(N,2,K+3)+AYS(N,1)*0
$      +BY(N,1)+(P(N,1,K+2)
$      -P(N,2,K+2))*DX)/AYN(N,1)
DO 190 J=2,(M-2),1
$      V(1,J,K+3)=(AYNE(1,J)*V(2,J,K+3)+AYNW(1,J)*0
$      +AYNN(1,J)*V(1,J+1,K+3)
$      +AYS(1,J)*V(1,J-1,K+3)
$      +BY(1,J)+(P(1,J,K+2)-P(1,J+1,K+2))*DX)
$      /AYN(1,J)
DO 200 I=2,(N-1),1
$      V(I,J,K+3)=(AYNE(I,J)*V(I+1,J,K+3)
$      +AYNW(I,J)*V(I-1,J,K+3)
$      +AYNN(I,J)*V(I,J+1,K+3)+AYS(I,J)
$      *V(I,J-1,K+2)
$      +BY(I,J)+(P(I,J,K+2)-P(I,J+1,K+2))
$      *DX)/AYN(I,J)
200 CONTINUE
$      V(N,J,K+3)=(AYNE(N,J)*0+AYNW(N,J)*V(N-1,J,K+3)
$      +AYNN(N,J)*V(N,J+1,K+3)+
$      AYS(N,J)*V(N,J-1,K+3)
$      +BY(N,J)+(P(N,J,K+2)-P(N,J+1,K+2))
$      *DX)/AYN(N,J)
190 CONTINUE
$      V(1,M-1,K+3)=(AYNE(1,M-1)*V(1,M-1,K+3)
$      +AYNW(1,M-1)*0
$      +AYNN(1,M-1)*0
$      +AYS(1,M-1)*V(1,M-2,K+3)
$      +BY(1,M-1)+(P(1,M-1,K+2)-
$      P(1,M,K+2))*DX)
$      /AYN(1,M-1)
DO 210 I=2,(N-1),1
$      V(I,M-1,K+3)=(AYNE(I,M-1)*V(I+1,M-1,K+3)
$      +AYNW(I,M-1)*V(I-1,M-1,K+3)
$      +AYNN(I,M-1)*0+AYS(I,M-1)*V(I,M-2,K+3)
$      +BY(I,M-1)+(P(I,M-1,K+2)
$      -P(I,M,K+2))*DX)/AYN(I,M-1)

```

```

210      CONTINUE
          V(N,M-1,K+3)=(AYNE(N,M-1)*0
$          +AYNW(N,M-1)*V(N-1,M-1,K+3)
$          +AYNN(N,M-1)*0+AYS(N,M-1)*
$          V(N,M-2,K+3)
$          +BY(N,M-1)+(P(N,M-1,K+2)-P(N,M,K+2))
$          *DX)/AYN(N,M-1)
C
C*      DE* EQN
C
          DO 213 J=1,M,1
            DO 216 I=1,N,1
              DE(I,J,K+2)=P(I,J,K+2)/T(I,J,K+1)
216      CONTINUE
213      CONTINUE
C
C*      P' EQN
C
          DDE=DY/AXE(1,1)
          DN=DX/AYN(1,1)
C
          APE(1,1)=(-(1.0/(T(2,1,K+1)+T(1,1,K+1)))*U(1,1,K+3)+
$          ((DE(2,1,K+2)+DE(1,1,K+2))/2.0)*DDE)*DY
C
          APW(1,1)=0
C
          APN(1,1)=(-(1.0/(T(1,2,K+1)+T(1,1,K+1)))*V(1,1,K+3)+
$          ((DE(1,2,K+2)+DE(1,1,K+2))/2.0)*DN)*DX
C
          APS(1,1)=0
C
          APP(1,1)=APE(1,1)+APW(1,1)+APN(1,1)+APS(1,1)+
$          DX*DY*(1.0/DT)
C
          BP(1,1)=(DE(1,1,K)-DE(1,1,K+2))*DX*DY*(1.0/DT)+
$          (0-
$          ((DE(2,1,K+2)+DE(1,1,K+2))
$          /2.0)*U(1,1,K+3))*DY+
$          (0-
$          ((DE(1,2,K+2)+DE(1,1,K+2))/2.0)
$          *V(1,1,K+3))*DX
$
          P(1,1,K+3)=(APE(1,1)*P(2,1,K+3)
$          +APN(1,1)*P(1,2,K+3)+BP(1,1))/APP(1,1)
C
          DO 220 I=2,(N-1),1
            DDE=DY/AXE(I,1)
            DW=DY/AXE(I-1,1)
            DN=DX/AYN(I,1)

```

```

C
$ APE(I,1)=(-(1.0/(T(I+1,1,K+1)
$ +T(I,1,K+1))) *U(I,1,K+3)+
$ ((DE(I+1,1,K+2)+DE(I,1,K+2))/2.0) *DDE) *DY
C
$ APW(I,1)=((1.0/(T(I,1,K+1)+T(I-1,1,K+1)))
$ *U(I-1,1,K+3)+
$ ((DE(I-1,1,K+2)+DE(I,1,K+2))/2.0)
$ *DW) *DY
C
$ APN(I,1)=(-(1.0/(T(I,2,K+1)+T(I,1,K+1))) *V(I,1,K+3)+
$ ((DE(I,2,K+2)+DE(I,1,K+2))/2.0) *DN) *DX
C
APS(I,1)=0
C
$ APP(I,1)=APE(I,1)+APW(I,1)+APN(I,1)+APS(I,1)+
$ DX*DY*(1.0/DT)
C
$ BP(I,1)=(DE(I,1,K)-DE(I,1,K+2)) *DX*DY*(1.0/DT)+
$ ((DE(I-1,1,K+2)+
$ DE(I,1,K+2))/2.0) *U(I-1,1,K+3)-
$ ((DE(I+1,1,K+2)+
$ DE(I,1,K+2))/2.0) *U(I,1,K+3)) *DY+
$ (0-
$ ((DE(I,2,K+2)
$ +DE(I,1,K+2))/2.0) *V(I,1,K+3)) *DX
$
$ P(I,1,K+3)=(APE(I,1) *P(I+1,1,K+3)
$ +APW(I,1) *P(I-1,1,K+3)
$ +APN(I,1) *P(I,2,K+3)+BP(I,1)) /APP(I,1)
220 CONTINUE
$ DW=DY/AXE(N-1,1)
$ DN=DX/AYN(N,1)
C
APE(N,1)=0
C
$ APW(N,1)=((1.0/(T(N,1,K+1)
$ +T(N-1,1,K+1))) *U(N-1,1,K+3)+
$ ((DE(N-1,1,K+2)+DE(N,1,K+2))/2.0) *DW) *DY
C
$ APN(N,1)=(-(1.0/(T(N,2,K+1)+T(N,1,K+1))) *V(N,1,K+3)+
$ ((DE(N,2,K+2)+DE(N,1,K+2))/2.0) *DN) *DX
C
APS(N,1)=0
C
$ APP(N,1)=APE(N,1)+APW(N,1)+APN(N,1)+APS(N,1)+
$ DX*DY*(1.0/DT)
C
$ BP(N,1)=(DE(N,1,K)-DE(N,1,K+2)) *DX*DY*(1.0/DT)+
$ ((DE(N-1,1,K+2)+
$ DE(N,1,K+2))/2.0) *U(N-1,1,K+3)-

```

```

$          0)*DY+
$          (0-
$          ((DE(N,2,K+2)+
$          DE(N,1,K+2))/2.0)*V(N,1,K+3))*DX
C
$          P(N,1,K+3)=(APW(N,1)*P(N-1,1,K+3)
$          +APN(N,1)*P(N,2,K+3)+BP(N,1))/APP(N,1)
C
DO 230 J=2,(M-1),1
    DDE=DY/AXE(1,J)
    DN=DX/AYN(1,J)
    DS=DX/AYN(1,J-1)
C
$          APE(1,J)=(-(1.0/(T(2,J,K+1)
$          +T(1,J,K+1)))*U(1,J,K+3)+
$          ((DE(2,J,K+2)+DE(1,J,K+2))/2.0)*DDE)*DY
C
$          APW(1,J)=0
C
$          APN(1,J)=(-(1.0/(T(1,J+1,K+1)
$          +T(1,J,K+1)))*V(1,J,K+3)+
$          ((DE(1,J+1,K+2)+DE(1,J,K+2))/2.0)*DN)*DX
C
$          APS(1,J)=((1.0/(T(1,J-1,K+1)
$          +T(1,J,K+1)))*V(1,J-1,K+3)+
$          ((DE(1,J-1,K+2)+DE(1,J,K+2))/2.0)*DS)*DX
C
$          APP(1,J)=APE(1,J)+APW(1,J)+APN(1,J)+APS(1,J)+
$          DX*DY*(1.0/DT)
C
$          BP(1,J)=(DE(1,J,K)-DE(1,J,K+2))*DX*DY*(1.0/DT)+
$          (0-
$          ((DE(2,J,K+2)+DE(1,J,K+2))/2.0)
$          *U(1,J,K+3))*DY+
$          ((DE(1,J-1,K+2)+DE(1,J,K+2))/2.0)
$          *V(1,J-1,K+3)-
$          ((DE(1,J+1,K+2)+DE(1,J,K+2))/2.0)
$          *V(1,J,K+3))*DX
$          P(1,J,K+3)=(APE(1,J)*P(2,J,K+3)
$          +APN(1,J)*P(1,J+1,K+3)
$          +APS(1,J)*P(1,J-1,K+3)
$          +BP(1,J))/APP(1,J)
C
DO 240 I=2,(N-1),1
    DDE=DY/AXE(I,J)
    DW=DY/AXE(I-1,J)
    DN=DX/AYN(I,J)
    DS=DX/AYN(I,J-1)
C
$          APE(I,J)=(-(1.0/(T(I+1,J,K+1)

```

```

$          +T(I,J,K+1)))*U(I,J,K+3)+
$          ((DE(I+1,J,K+2)+
$          DE(I,J,K+2))/2.0)*DDE)*DY
C
APW(I,J)=((1.0/(T(I-1,J,K+1)
$          +T(I,J,K+1)))*U(I-1,J,K+3)+
$          ((DE(I-1,J,K+2)+DE(I,J,K+2))
$          /2.0)*DW)*DY
C
APN(I,J)=(-(1.0/(T(I,J+1,K+1)+
$          T(I,J,K+1)))*V(I,J,K+3)+
$          ((DE(I,J+1,K+2)+DE(I,J,K+2))
$          /2.0)*DN)*DX
C
APS(I,J)=((1.0/(T(I,J-1,K+1)
$          +T(I,J,K+1)))*V(I,J-1,K+3)+
$          ((DE(I,J-1,K+2)+DE(I,J,K+2))
$          /2.0)*DS)*DX
C
APP(I,J)=APE(I,J)+APW(I,J)+APN(I,J)+APS(I,J)+
$          DX*DY*(1.0/DT)
C
BP(I,J)=(DE(I,J,K)-
$          DE(I,J,K+2))*DX*DY*(1.0/DT)+
$          (((DE(I-1,J,K+2)+DE(I,J,K+2))/2.0)
$          *U(I-1,J,K+3)-
$          ((DE(I+1,J,K+2)+DE(I,J,K+2))/2.0)
$          *U(I,J,K+3))*DY+
$          (((DE(I,J-1,K+2)+DE(I,J,K+2))/2.0)
$          *V(I,J-1,K+3)-
$          ((DE(I,J+1,K+2)+DE(I,J,K+2))/2.0)
$          *V(I,J,K+3))*DX
$
$          P(I,J,K+3)=(APE(I,J)*
$          P(I+1,J,K+3)+APW(I,J)*P(I-1,J,K+3)
$          +APN(I,J)*P(I,J+1,K+3)
$          +APS(I,J)*P(I,J-1,K+3)+
$          BP(I,J))/APP(I,J)
C
240      CONTINUE
          DW=DY/AXE(N-1,J)
          DN=DX/AYN(N,J)
          DS=DX/AYN(N,J-1)
C
          APE(N,J)=0
C
          APW(N,J)=((1.0/(T(N-1,J,K+1)
$          +T(N,J,K+1)))*U(N-1,J,K+3)+
$          ((DE(N-1,J,K+2)+
$          DE(N,J,K+2))/2.0)*DW)*DY
C

```

```

C      APN(N,J)=(- (1.0/ (T(N,J+1,K+1)
$          +T(N,J,K+1) ) ) *V(N,J,K+3)+
$          ( (DE(N,J+1,K+2)+DE(N,J,K+2) ) /2.0) *DN) *DX
C
C      APS(N,J)= ( (1.0/ (T(N,J-1,K+1)
$          +T(N,J,K+1) ) ) *V(N,J-1,K+3)+
$          ( (DE(N,J-1,K+2)+DE(N,J,K+2) ) /2.0) *DS) *DX
C
C      APP(N,J)=APE(N,J)+APW(N,J)+APN(N,J)+APS(N,J)+
$          DX*DY*(1.0/DT)
C
C      BP(N,J)=(DE(N,J,K)-DE(N,J,K+2) ) *DX*DY*(1.0/DT) +
$          ( ( (DE(N-1,J,K+2)+DE(N,J,K+2) )
$          /2.0) *U(N-1,J,K+3) -
$          0) *DY+
$          ( ( (DE(N,J-1,K+2)+
$          DE(N,J,K+2) ) /2.0) *V(N,J-1,K+3) -
$          ( (DE(N,J+1,K+2)+
$          DE(N,J,K+2) ) /2.0) *V(N,J,K+3) ) *DX
C
C      P(N,J,K+3)=(APW(N,J) *P(N-1,J,K+3)
$          +APN(N,J) *P(N,J+1,K+3)
$          +APS(N,J) *P(N,J-1,K+3)
$          +BP(N,J) ) /APP(N,J)
C
230  C      CONTINUE
C
C      DDE=DY/AXE(1,M)
C      DS=DX/AYN(1,M-1)
C
C      APE(1,M)=(- (1.0/ (T(2,M,K+1)
$          +T(1,M,K+1) ) ) *U(1,M,K+3)+
$          ( (DE(2,M,K+2)+DE(1,M,K+2) ) /2.0) *DDE) *DY
C
C      APW(1,M)=0
C
C      APN(1,M)=0
C
C      APS(1,M)= ( (1.0/ (T(1,M-1,K+1)+
$          T(1,M,K+1) ) ) *V(1,M-1,K+3)+
$          ( (DE(1,M-1,K+2)+DE(1,M,K+2) ) /2.0) *DS) *DX
C
C      APP(1,M)=APE(1,M)+APW(1,M)+APN(1,M)+APS(1,M)+
$          DX*DY*(1.0/DT)
C
C      BP(1,M)=(DE(1,M,K)-DE(1,M,K+2) ) *DX*DY*(1.0/DT) +
$          (0-
$          ( (DE(2,M,K+2)+DE(1,M,K+2) ) /2.0)
$          *U(1,M,K+3) ) *DY+
$          ( ( (DE(1,M-1,K+2)+DE(1,M,K+2) )
$          /2.0) *V(1,M-1,K+3) -

```

```

$          0)*DX
$
$          P(1,M,K+3)=(APE(1,M)*P(2,M,K+3)+
$                      APS(1,M)*P(1,M-1,K+3)
$                      +BP(1,M))/APP(1,M)
C
DO 250 I=2,(N-1),1
  DDE=DY/AXE(I,M)
  DW=DY/AXE(I-1,M)
  DS=DX/AYN(I,M-1)
C
  APE(I,M)=(-(1.0/(T(I+1,M,K+1)
$             +T(I,M,K+1)))*U(I,M,K+3)+
$             ((DE(I+1,M,K+2)+
$             DE(I,M,K+2))/2.0)*DDE)*DY
C
  APW(I,M)=((1.0/(T(I-1,M,K+1)
$             +T(I,M,K+1)))*U(I-1,M,K+3)+
$             ((DE(I-1,M,K+2)+
$             DE(I,M,K+2))/2.0)*DW)*DY
C
  APN(I,M)=0
C
  APS(I,M)=((1.0/(T(I,M-1,K+1)
$             +T(I,M,K+1)))*V(I,M-1,K+3)+
$             ((DE(I,M-1,K+2)
$             +DE(I,M,K+2))/2.0)*DS)*DX
C
  APP(I,M)=APE(I,M)+APW(I,M)+APN(I,M)+APS(I,M)+
$           DX*DY*(1.0/DT)
C
  BP(I,M)=(DE(I,M,K)-DE(I,M,K+2))
$           *DX*DY*(1.0/DT)+
$           (((DE(I-1,M,K+2)+DE(I,M,K+2))/2.0)
$           *U(I-1,M,K+3)-
$           ((DE(I+1,M,K+2)+DE(I,M,K+2))/2.0)
$           *U(I,M,K+3))*DY+
$           (((DE(I,M-1,K+2)+DE(I,M,K+2))/2.0)
$           *V(I,M-1,K+3)-
$           0)*DX
$
$           P(I,M,K+3)=(APE(I,M)*P(I+1,M,K+3)
$                       +APW(I,M)*P(I-1,M,K+3)
$                       +APS(I,M)*P(I,M-1,K+3)+
$                       BP(I,M))/APP(I,M)
C
250 CONTINUE
  DW=DY/AXE(N-1,M)
  DS=DX/AYN(I,J-1)
C
  APE(N,M)=0

```

```

C
$
$
$
APW(N,M) = ( (1.0 / (T(N-1,M,K+1)
               +T(N,M,K+1))) *U(N-1,M,K+3) +
               ( (DE(N-1,M,K+2)+DE(N,M,K+2))
               /2.0) *DW) *DY
C
APN(N,M) = 0
C
$
$
$
APS(N,M) = ( (1.0 / (T(N,M-1,K+1) +
                    T(N,M,K+1))) *V(N,M-1,K+3) +
               ( (DE(N,M-1,K+2)+DE(N,M,K+2))
               /2.0) *DS) *DX
C
$
APP(N,M) = APE(N,M) + APW(N,M) + APN(N,M) + APS(N,M) +
            DX*DY*(1.0/DT)
C
$
$
$
$
$
$
$
BP(N,M) = (DE(N,M,K) - DE(N,M,K+2))
            *DX*DY*(1.0/DT) +
            ( ( (DE(N-1,M,K+2)+DE(N,M,K+2)) /2.0)
            *U(N-1,M,K+3) -
            0) *DY +
            ( ( (DE(N,M-1,K+2)+DE(N,M,K+2)) /2.0)
            *V(N,M-1,K+3) -
            0) *DX
$
$
P(N,M,K+3) = (APW(N,M) *P(N-1,M,K+3)
               +APS(N,M) *P(N,M-1,K+3) +
               BP(N,M)) /APP(N,M)
C
C*
C
CORRECT U* VELOCITY FIELD
DO 260 J=1,M,1
    DO 270 I=1,(N-1),1
        U(I,J,K+1) = U(I,J,K+3) +
                     (P(I,J,K+3) - P(I+1,J,K+3))
                     *(DY/AXE(I,J))
$
$
270
260
CONTINUE
CONTINUE
C
C*
C
CORRECT THE V* VELOCITY FIELD
DO 280 J=1,(M-1),1
    DO 290 I=1,N,1
        V(I,J,K+1) = V(I,J,K+3) +
                     (P(I,J,K+3) - P(I,J+1,K+3))
                     *(DX/AYN(I,J))
$
$
290
280
CONTINUE
CONTINUE
C
C*
C
CORRECT THE DENSITY FIELD

```

```

DO 300 J=1,M,1
  DO 310 I=1,N,1
    DE(I,J,K+1)=(P(I,J,K+2)
      $
      +P(I,J,K+3))/T(I,J,K+1)
310
300
CONTINUE
CONTINUE

C
C*
C
SOLVE THE ENERGY EQN

DXE=DX
DXW=DX/2.0
DYN=DY
DYS=DY/2.0

C
FE=((DE(1,1,K+1)+DE(2,1,K+1))/2.0)*U(1,1,K+1)
FW=0
FN=((DE(1,2,K+1)+DE(1,1,K+1))/2.0)*V(1,1,K+1)
FS=0

C
PEE=(FE*DXE)/(GAMMA/(RE*PR))
PEW=(FW*DXW)/(GAMMA/(RE*PR))
PEN=(FN*DYN)/(GAMMA/(RE*PR*(L/H)*(L/H)))
PES=(FS*DYS)/(GAMMA/(RE*PR*(L/H)*(L/H)))

C
CALL AMAX1(-FE,BRCKT)
CALL A(PEE,APEE)
AE(1,1)=BRCKT*DY+(GAMMA/(RE*PR*DXE))*APEE*DY

C
CALL AMAX1(FW,BRCKT)
CALL A(PEW,APEE)
AW(1,1)=BRCKT*DY+(GAMMA/(RE*PR*DXW))*APEE*DY

C
CALL AMAX1(-FN,BRCKT)
CALL A(PEN,APEE)
AN(1,1)=BRCKT*DX+(GAMMA/(RE*PR*DYN))
  $
  *(L/H)*(L/H)*APEE*DX

C
CALL AMAX1(FS,BRCKT)
CALL A(PES,APEE)
AS(1,1)=BRCKT*DX+0*
  $
  (GAMMA/(RE*PR*DYS))*(L/H)*(L/H)*APEE*DX

C
APO(1,1)=DE(1,1,K)*DX*DY*(1.0/DT)

C
B(1,1)=- (GAMMA-1)
  $
  *P(1,1,K+1)*((U(1,1,K+1)-0)*DY-
  $
  (V(1,1,K+1)-0)*DX)+APO(1,1)*T(1,1,K)

C
AP(1,1)=AE(1,1)+AW(1,1)
  $
  +AN(1,1)+AS(1,1)+APO(1,1)

C

```

```

C
$
$
T(1,1,K+1)=(AE(1,1)*T(2,1,K+1)+AW(1,1)*TEMP1+
AN(1,1)*T(1,2,K+1)+AS(1,1)*TEMP4
+B(1,1))/AP(1,1)
C
DO 320 I=2,(N-1),1
DXE=DX
DXW=DX
DYN=DY
DYS=DY/2.0
C
FE=((DE(I,1,K+1)+DE(I+1,1,K+1)
)/2.0)*U(I,1,K+1)
FW=((DE(I-1,1,K+1)+DE(I,1,K+1))/2.0)
*U(I-1,1,K+1)
FN=((DE(I,2,K+1)+DE(I,1,K+1))/2.0)
*V(I,1,K+1)
FS=0
C
PEE=(FE*DXE)/(GAMMA/(RE*PR))
PEW=(FW*DXW)/(GAMMA/(RE*PR))
PEN=(FN*DYN)/(GAMMA/(RE*PR*(L/H)*(L/H)))
PES=(FS*DYS)/(GAMMA/(RE*PR*(L/H)*(L/H)))
C
CALL AMAX1(-FE,BRCKT)
CALL A(PEE,APEE)
AE(I,1)=BRCKT*DY+(GAMMA
/(RE*PR*DXE))*APEE*DY
C
CALL AMAX1(FW,BRCKT)
CALL A(PEW,APEE)
AW(I,1)=BRCKT*DY+(GAMMA/
(RE*PR*DXW))*APEE*DY
C
CALL AMAX1(-FN,BRCKT)
CALL A(PEN,APEE)
AN(I,1)=BRCKT*DX+(GAMMA/
(RE*PR*DYN))*(L/H)*(L/H)*APEE*DX
C
CALL AMAX1(FS,BRCKT)
CALL A(PES,APEE)
AS(I,1)=BRCKT*DX+0*
(GAMMA/(RE*PR*DYS))*(L/H)*(L/H)*APEE*DX
C
APO(I,1)=DE(I,1,K)*DX*DY*(1.0/DT)
C
B(I,1)=- (GAMMA-1)*P(I,1,K+1)
*( (U(I,1,K+1)-U(I-1,1,K+1))*DY-
(V(I,1,K+1)-0)*DX
)+APO(I,1)*T(I,1,K)
C
AP(I,1)=AE(I,1)+AW(I,1)+

```

```

C      $      AN(I,1)+AS(I,1)+APO(I,1)

C      $      T(I,1,K+1)=(AE(I,1)*T(I+1,1,K+1)
C      $      +AW(I,1)*T(I-1,1,K+1)
C      $      +AN(I,1)*T(I,2,K+1)
C      $      +AS(I,1)*TEMP4
C      $      +B(I,1))/AP(I,1)

320      CONTINUE
      DXE=DX/2.0
      DXW=DX
      DYN=DY
      DYS=DY/2.0

C      FE=0
      FW=((DE(N-1,1,K+1)+DE(N,1,K+1))
      $      /2.0)*U(N-1,1,K+1)
      $      FN=((DE(N,2,K+1)+DE(N,1,K+1))
      $      /2.0)*V(N,1,K+1)
      FS=0

C      PEE=(FE*DXE)/(GAMMA/(RE*PR))
      PEW=(FW*DXW)/(GAMMA/(RE*PR))
      PEN=(FN*DYN)/(GAMMA/(RE*PR*(L/H)*(L/H)))
      PES=(FS*DYS)/(GAMMA/(RE*PR*(L/H)*(L/H)))

C      CALL AMAX1(-FE,BRCKT)
      CALL A(PEE,APEE)
      $      AE(N,1)=BRCKT*DY+(GAMMA
C      $      /(RE*PR*DXE))*APEE*DY

C      CALL AMAX1(FW,BRCKT)
      CALL A(PEW,APEE)
      $      AW(N,1)=BRCKT*DY+
C      $      (GAMMA/(RE*PR*DXW))*APEE*DY

C      CALL AMAX1(-FN,BRCKT)
      CALL A(PEN,APEE)
      $      AN(N,1)=BRCKT*DX+(GAMMA/
C      $      (RE*PR*DYN))*(L/H)*(L/H)*APEE*DX

C      CALL AMAX1(FS,BRCKT)
      CALL A(PES,APEE)
      $      AS(N,1)=BRCKT*DX+0*(GAMMA
C      $      /(RE*PR*DYS))*(L/H)*(L/H)*APEE*DX

C      APO(N,1)=DE(N,1,K)*DX*DY*(1.0/DT)

C      B(N,1)=-(GAMMA-1)*P(N,1,K+1)
      $      *((0-U(N-1,1,K+1))*DY-
      $      (V(N,1,K+1)-0)*DX)+

```

```

C      $      APO(N,1)*T(N,1,K)

C      $      AP(N,1)=AE(N,1)+AW(N,1)+
C      $      AN(N,1)+AS(N,1)+APO(N,1)

C      $      T(N,1,K+1)=(AE(N,1)*TEMP3+
C      $      AW(N,1)*T(N-1,1,K+1)+
C      $      AN(N,1)*T(N,2,K+1)+AS(N,1)*TEMP4
C      $      +B(N,1))/AP(N,1)

C      DO 330 J=2,(M-1),1
C      DXE=DX
C      DXW=DX/2.0
C      DYN=DY
C      DYS=DY

C      $      FE=((DE(1,J,K+1)+
C      $      DE(2,J,K+1))/2.0)*U(1,J,K+1)
C      $      FW=0
C      $      FN=((DE(1,J+1,K+1)+
C      $      DE(1,J,K+1))/2.0)*V(1,J,K+1)
C      $      FS=((DE(1,J-1,K+1)+DE(1,J,K+1))/2.0)
C      $      *V(1,J-1,K+1)

C      PEE=(FE*DXE)/(GAMMA/(RE*PR))
C      PEW=(FW*DXW)/(GAMMA/(RE*PR))
C      PEN=(FN*DYN)/(GAMMA/(RE*PR*(L/H)*(L/H)))
C      PES=(FS*DYS)/(GAMMA/(RE*PR*(L/H)*(L/H)))

C      CALL AMAX1(-FE,BRCKT)
C      CALL A(PEE,APEE)
C      $      AE(1,J)=BRCKT*DY+(GAMMA/
C      $      (RE*PR*DXE))*APEE*DY

C      CALL AMAX1(FW,BRCKT)
C      CALL A(PEW,APEE)
C      $      AW(1,J)=BRCKT*DY+(GAMMA
C      $      /(RE*PR*DXW))*APEE*DY

C      CALL AMAX1(-FN,BRCKT)
C      CALL A(PEN,APEE)
C      $      AN(1,J)=BRCKT*DX+(GAMMA/(RE*PR*DYN))
C      $      *(L/H)*(L/H)*APEE*DX

C      CALL AMAX1(FS,BRCKT)
C      CALL A(PES,APEE)
C      $      AS(1,J)=BRCKT*DX+(GAMMA/(RE*PR*DYS))
C      $      *(L/H)*(L/H)*APEE*DX

C      APO(1,J)=DE(1,J,K)*DX*DY*(1.0/DT)

```

```

C
$
$
$
B(1,J)=- (GAMMA-1) *P(1,J,K+1) *
          ((U(1,J,K+1)-0) *DY-
          (V(1,J,K+1)-V(1,J-1,K+1)) *DX)
          +APO(1,J) *T(1,J,K)

C
$
AP(1,J)=AE(1,J)+AW(1,J)+AN(1,J)
          +AS(1,J)+APO(1,J)

C
$
$
$
$
T(1,J,K+1)=(AE(1,J) *T(2,J,K+1)
             +AW(1,J) *TEMP1+
             AN(1,J) *T(1,J+1,K+1)
             +AS(1,J) *T(1,J-1,K+1)
             +B(1,J)) /AP(1,J)

C
DO 340 I=2, (N-1), 1
  DXE=DX
  DXW=DX
  DYN=DY
  DYS=DY

C
$
$
$
$
$
FE=((DE(I,J,K+1)+DE(I+1,J,K+1))/2.0)
   *U(I,J,K+1)
FW=((DE(I-1,J,K+1)+DE(I,J,K+1))/2.0)
   *U(I-1,J,K+1)
FN=((DE(I,J+1,K+1)+DE(I,J,K+1))/2.0)
   *V(I,J,K+1)
FS=((DE(I,J-1,K+1)+DE(I,J,K+1))/2.0)
   *V(I,J-1,K+1)

C
$
$
$
$
PEE=(FE*DXE) / (GAMMA / (RE*PR))
PEW=(FW*DXW) / (GAMMA / (RE*PR))
PEN=(FN*DYN) / (GAMMA / (RE*PR * (L/H)
   * (L/H)))
PES=(FS*DYS) / (GAMMA / (RE*PR *
   (L/H) * (L/H)))

C
$
$
$
$
CALL AMAX1 (-FE, BRCKT)
CALL A(PEE, APEE)
AE(I,J)=BRCKT*DY+ (GAMMA/
   (RE*PR*DXE)) *APEE*DY

C
$
$
$
$
CALL AMAX1 (FW, BRCKT)
CALL A(PEW, APEE)
AW(I,J)=BRCKT*DY+ (GAMMA/
   (RE*PR*DXW)) *APEE*DY

C
$
$
$
$
CALL AMAX1 (-FN, BRCKT)
CALL A(PEN, APEE)
AN(I,J)=BRCKT*DX+ (GAMMA / (RE*PR*DYN))
   * (L/H) * (L/H)
   *APEE*DX

```



```

C      $      AW(N,J)=BRCKT*DY+(GAMMA
              /(RE*PR*DXW))*APEE*DY

C      $      CALL AMAX1(-FN,BRCKT)
              CALL A(PEN,APEE)
              AN(N,J)=BRCKT*DX+(GAMMA/
              (RE*PR*DYN))*(L/H)
              *(L/H)*APEE*DX

C      $      CALL AMAX1(FS,BRCKT)
              CALL A(PES,APEE)
              AS(N,J)=BRCKT*DX+(GAMMA/(RE*PR*DYS))
              *(L/H)*(L/H)*APEE*DX

C      $      APO(N,J)=DE(N,J,K)*DX*DY*(1.0/DT)

C      $      B(N,J)=- (GAMMA-1)*P(N,J,K+1)*((0
              -U(N-1,J,K+1))*DY-
              (V(N,J,K+1)-V(N,J-1,K+1))*DX)
              +APO(N,J)*T(N,J,K)

C      $      AP(N,J)=AE(N,J)+AW(N,J)+AN(N,J)+AS(N,J)
              +APO(N,J)

C      $      T(N,J,K+1)=(AE(N,J)*TEMP3
              +AW(N,J)*T(N-1,J,K+1)+
              AN(N,J)*T(N,J+1,K+1)
              +AS(N,J)*T(N,J-1,K+1)
              +B(N,J))/AP(N,J)

C      $      330      CONTINUE
              DXE=DX
              DXW=DX/2.0
              DYN=DY/2.0
              DYS=DY

C      $      FE=((DE(1,M,K+1)+
              DE(2,M,K+1))/2.0)*U(1,M,K+1)
              FW=0
              FN=0
              FS=((DE(1,M-1,K+1)
              +DE(1,M,K+1))/2.0)*V(1,M-1,K+1)

C      $      PEE=(FE*DXE)/(GAMMA/(RE*PR))
              PEW=(FW*DXW)/(GAMMA/(RE*PR))
              PEN=(FN*DYN)/(GAMMA/(RE*PR*(L/H)*(L/H)))
              PES=(FS*DYS)/(GAMMA/(RE*PR*(L/H)*(L/H)))

C      $      CALL AMAX1(-FE,BRCKT)
              CALL A(PEE,APEE)
              AE(1,M)=BRCKT*DY+(GAMMA

```

```

C      $      / (RE*PR*DXE) ) *APEE*DY

CALL AMAX1 (FW, BRCKT)
CALL A (PEW, APEE)
AW(1,M)=BRCKT*DY+( GAMMA
C      $      / (RE*PR*DXW) ) *APEE*DY

CALL AMAX1 (-FN, BRCKT)
CALL A (PEN, APEE)
AN(1,M)=BRCKT*DX+0*( GAMMA/
C      $      (RE*PR*DYN) ) * (L/H) * (L/H) *APEE*DX

CALL AMAX1 (FS, BRCKT)
CALL A (PES, APEE)
AS(1,M)=BRCKT*DX+( GAMMA/(RE*PR*DYS) )
C      $      * (L/H) * (L/H) *APEE*DX

APO(1,M)=DE(1,M,K) *DX*DY*(1.0/DT)
C
C      B(1,M)=- (GAMMA-1) *P(1,M,K+1) * ( (U(1,M,K+1)
C      $      -0) *DY-
C      $      (0-V(1,M-1,K+1) ) *DX)
C      $      +APO(1,M) *T(1,M,K)

AP(1,M)=AE(1,M)+AW(1,M)+AN(1,M)+AS(1,M)
C      $      +APO(1,M)

T(1,M,K+1)=(AE(1,M) *T(2,M,K+1)
C      $      +AW(1,M) *TEMP1+
C      $      AN(1,M) *TEMP2
C      $      +AS(1,M) *T(1,M-1,K+1)
C      $      +B(1,M) )/AP(1,M)

DO 350 I=2, (N-1), 1
      DXE=DX
      DXW=DX
      DYN=DY/2.0
      DYS=DY
C
C      FE=((DE(I,M,K+1)+DE(I+1,M,K+1))/2.0)
C      $      *U(I,M,K+1)
C      $      FW=((DE(I-1,M,K+1)+DE(I,M,K+1))/2.0)
C      $      *U(I-1,M,K+1)
      FN=0
      FS=((DE(I,M-1,K+1)+DE(I,M,K+1))/2.0)
C      $      *V(I,M-1,K+1)

      PEE=(FE*DXE)/( GAMMA/(RE*PR) )
      PEW=(FW*DXW)/( GAMMA/(RE*PR) )
      PEN=(FN*DYN)/( GAMMA
C      $      / (RE*PR*(L/H) * (L/H) ) )

```

		PES=(FS*DYS)/(GAMMA/(RE*PR*
	\$	(L/H)*(L/H)))
C		
		CALL AMAX1(-FE,BRCKT)
		CALL A(PEE,APEE)
	\$	AE(I,M)=BRCKT*DY+(GAMMA/
		(RE*PR*DXE))*APEE*DY
C		
		CALL AMAX1(FW,BRCKT)
		CALL A(PEW,APEE)
	\$	AW(I,M)=BRCKT*DY+(GAMMA
		/(RE*PR*DXW))*APEE*DY
C		
		CALL AMAX1(-FN,BRCKT)
		CALL A(PEN,APEE)
	\$	AN(I,M)=BRCKT*DX+0*(GAMMA
	\$	/(RE*PR*DYN))
		(L/H)(L/H)*APEE*DX
C		
		CALL AMAX1(FS,BRCKT)
		CALL A(PES,APEE)
	\$	AS(I,M)=BRCKT*DX+(GAMMA/(RE*PR*DYS))
		(L/H)(L/H)*APEE*DX
C		
		APO(I,M)=DE(I,M,K)*DX*DY*(1.0/DT)
C		
	\$	B(I,M)=- (GAMMA-1)*
	\$	P(I,M,K+1)*((U(I,M,K+1)
	\$	-U(I-1,M,K+1))*DY-
	\$	(0-V(I,M-1,K+1))*DX)
		+APO(I,M)*T(I,M,K)
C		
	\$	AP(I,M)=AE(I,M)+AW(I,M)
	\$	+AN(I,M)+AS(I,M)
		+APO(I,M)
C		
	\$	T(I,M,K+1)=(AE(I,M)*T(I+1,M,K+1)
	\$	+AW(I,M)*T(I-1,M,K+1)+
	\$	AN(I,M)*TEMP2
	\$	+AS(I,M)*T(I,M-1,K+1)
		+B(I,M))/AP(I,M)
C		
350		CONTINUE
		DXE=DX/2.0
		DXW=DX
		DYN=DY/2.0
		DYS=DY
C		
		FE=0
	\$	FW=((DE(N-1,M,K+1)+DE(N,M,K+1)
		)/2.0)*U(N-1,M,K+1)

```

FN=0
FS=( (DE(N,M-1,K+1)+DE(N,M,K+1))
$      /2.0)*V(N,M-1,K+1)
C

PEE=(FE*DXE)/(GAMMA/(RE*PR))
PEW=(FW*DXW)/(GAMMA/(RE*PR))
PEN=(FN*DYN)/(GAMMA/(RE*PR*(L/H)*(L/H)))
PES=(FS*DYS)/(GAMMA/(RE*PR*(L/H)*(L/H)))
C

CALL AMAX1(-FE,BRCKT)
CALL A(PEE,APEE)
AE(N,M)=BRCKT*DY+(GAMMA
$      /(RE*PR*DXE))*APEE*DY
C

CALL AMAX1(FW,BRCKT)
CALL A(PEW,APE)
AW(N,M)=BRCKT*DY+(GAMMA
$      /(RE*PR*DXW))*APEE*DY
C

CALL AMAX1(-FN,BRCKT)
CALL A(PEN,APEE)
AN(N,M)=BRCKT*DX+0*(GAMMA/(RE*PR*DYN))
$      *(L/H)*(L/H)*APEE*DX
C

CALL AMAX1(FS,BRCKT)
CALL A(PES,APEE)
AS(N,M)=BRCKT*DX+(GAMMA/(RE*PR*DYS))
$      *(L/H)*(L/H)*APEE*DX
C

APO(N,M)=DE(N,M,K)*DX*DY*(1.0/DT)
C

B(N,M)=-(GAMMA-1)*P(N,M,K+1)*((0
$      -U(N-1,M,K+1))*DY-
$      (0-V(N,M-1,K+1))*DX)
$      +APO(N,M)*T(N,M,K)
C

AP(N,M)=AE(N,M)+AW(N,M)+AN(N,M)+AS(N,M)
$      +APO(N,M)
C

T(N,M,K+1)=(AE(N,M)*TEMP3
$      +AW(N,M)*T(N-1,M,K+1)+
$      AN(N,M)*TEMP2
$      +AS(N,M)*T(N,M-1,K+1)
$      +B(N,M))/AP(N,M)
C
C*
C

MASS SOURCE TO CHECK FOR CONVERGENCE

BP(1,1)=(DE(1,1,K)-DE(1,1,K+1))
$      *DX*DY*(1.0/DT)+
$      (0
$      -(DE(2,1,K+1)+DE(1,1,K+1))/2.0)

```

```

$          *U(1,1,K+1)) *DY+
$          (0
$          -((DE(1,2,K+1)+DE(1,1,K+1))/2.0)
$          *V(1,1,K+1)) *DX
DO 360 I=2,(N-1),1
    BP(I,1)=(DE(I,1,K)-DE(I+1,1,K+1))
        *DX*DY*(1.0/DT)+
        (((DE(I-1,1,K+1)+
        DE(I,1,K+1))/2.0)
        *U(I-1,1,K+1)
        -((DE(I+1,1,K+1)+
        DE(I,1,K+1))/2.0)
        *U(I,1,K+1)) *DY+
        (0
        -((DE(I,2,K+1)+
        DE(I,1,K+1))/2.0)
        *V(I,1,K+1)) *DX
360 CONTINUE
    BP(N,1)=(DE(N,1,K)-DE(N,1,K+1))
        *DX*DY*(1.0/DT)+
        (((DE(N-1,1,K+1)+
        DE(N,1,K+1))/2.0)
        *U(N-1,1,K+1)
        -0) *DY+
        (0
        -((DE(N,2,K+1)+DE(N,1,K+1))/2.0)
        *V(N,1,K+1)) *DX
DO 370 J=2,(M-1),1
    BP(1,J)=(DE(1,J,K)-DE(1,J,K+1))
        *DX*DY*(1.0/DT)+
        (0
        -((DE(2,J,K+1)+
        DE(1,J,K+1))/2.0)
        *U(1,J,K+1)) *DY+
        (((DE(1,J-1,K+1)+
        DE(1,J,K+1))/2.0)
        *V(1,J-1,K+1)
        -((DE(1,J+1,K+1)+
        DE(1,J,K+1))/2.0)
        *V(1,J,K+1)) *DX
DO 380 I=1,(N-1),1
    BP(I,J)=(DE(I,J,K)-DE(I,J,K+1))
        *DX*DY*(1.0/DT)+
        (((DE(I-1,J,K+1)+
        DE(I,J,K+1))/2.0)
        *U(I-1,J,K+1)
        -((DE(I+1,J,K+1)+
        DE(I,J,K+1))/2.0)
        *U(I,J,K+1)) *DY+
        (((DE(I,J-1,K+1)+
        DE(I,J,K+1))/2.0)

```



```

C          WRITE(9,*)V(I,J,K+1),XPOS,YPOS,TIME
C
C          WRITE(*,*)'U=',U(I,J,K+1)*10E3
C          $          , 'AT', I, J, TIME
C          WRITE(*,*)'V=',V(I,J,K+1), 'AT', I, J, TIME
C
C          COUNTER1=0
C          470      CONTINUE
C          460      CONTINUE
C          END IF
C
C          RESETTING VARIABLES
C
C          DO 480 J=1,M,1
C              DO 490 I=1,N,1
C                  T(I,J,K)=T(I,J,K+1)
C                  DE(I,J,K)=DE(I,J,K+1)
C                  P(I,J,K)=P(I,J,K+1)
C          490      CONTINUE
C          480      CONTINUE
C
C          DO 500 J=1,M,1
C              DO 510 I=1,N-1,1
C                  U(I,J,K)=U(I,J,K+1)
C          510      CONTINUE
C          500      CONTINUE
C
C          DO 520 J=1,M-1,1
C              DO 530 I=1,N,1
C                  V(I,J,K)=V(I,J,K+1)
C          530      CONTINUE
C          520      CONTINUE
C          70      CONTINUE
C          STOP
C          END
C
C*****
C          SUBROUTINE AMAX1
C          SUBROUTINE AMAX1(X,Y)
C              REAL X,Y
C              IF (X .GT. 0) THEN
C                  Y=X
C              ELSE IF (X .LE. 0) THEN
C                  Y=0
C              END IF
C              RETURN
C              END
C
C*****
C*****
C*****

```

C

```
                SUBROUTINE A(A,Y)
SUBROUTINE A(X,Y)
  REAL X,Y,Z
  Z=(1-0.1*ABS(X))**5
  IF (Z .GT. 0) THEN
    Y=Z
  ELSE IF (Z .LE. 0) THEN
    Y=0
  END IF
  RETURN
END
```

APPENDIX E

**CHART OF VARIOUS SPACE AND TIME INCREMENTS
USED IN THE PROGRAM MODIPL.FOR; TAC1D.FOR;
TAC2D.FOR**

SPACE AND TIME INCREMENTS USED IN MODIPL.FOR

$\Delta x \backslash \Delta t$	0.015	0.005	0.001875
1/22	RUNS	RUNS	RUNS
1/44	RUNS	RUNS	RUNS
1/88	RUNS	RUNS	RUNS
1/110	FAILS TO CONVERGE	RUNS	RUNS
1/156	FAILS TO CONVERGE	FAILS TO CONVERGE	RUNS
1/176	FAILS TO CONVERGE	FAILS TO CONVERGE	FAILS TO CONVERGE

SPACE AND TIME INCREMENTS USED IN TAC1D.FOR

$\Delta x \backslash \Delta t$	0.040	0.035	0.030	0.020	0.015	0.010
1/33	RUNS	RUNS	RUNS	FAILS TO CONVERGE	FAILS TO CONVERGE	FAILS
1/44	RUNS	RUNS	RUNS	RUNS	FAILS TO CONVERGE	FAILS
1/50	RUNS	RUNS	RUNS	RUNS	RUNS	FAILS
1/60	RUNS	RUNS	RUNS	RUNS	RUNS	FAILS
1/70	RUNS	RUNS	RUNS	RUNS	RUNS	FAILS
1/80	RUNS	RUNS	RUNS	RUNS	RUNS	FAILS
1/88	RUNS	RUNS	RUNS	RUNS	RUNS	FAILS
1/92	FAILS TO CONVERGE	FAILS TO CONVERGE	FAILS TO CONVERGE	FAILS TO CONVERGE	FAILS TO CONVERGE	FAILS

SPACE AND TIME INCREMENTS USED IN TAC2D.FOR

Δt values					
mesh sizes X by Y nodes	$\Delta t=0.1$	$\Delta t=0.04$	$\Delta t=0.02$	$\Delta t=0.01$	$\Delta t=.001$
10x10	FAILS TO CONVERGE	RUNS	RUNS	FAILS TO CONV.	FAILS TO CONV.
20x20	FAILS TO CONVERGE	RUNS	RUNS	FAILS TO CONV.	FAILS TO CONV.
40x40	FAILS TO CONVERGE	RUNS	RUNS	RUNS	FAILS TO CONV.
80x80	RUNS	RUNS	RUNS	RUNS	RUNS
100x100	RUNS	RUNS	RUNS	RUNS	RUNS
200x200	FAILS TO CONVERGE	FAILS TO CONV.	FAILS TO CONVERGE	RUNS	FAILS TO CONV.

VITA

John P. Raulston Jr. was born in Chattanooga, Tennessee to John and Jackie Raulston on August 9, 1967. He graduated from Oswego High School, Oswego, New York in 1985. In December, 1989 he received a Bachelor of Science degree in Mechanical Engineering at the University of Tennessee, Knoxville and was commissioned as a Second Lieutenant in the United States Air Force. On May 12, 1990 he married the former Cynthia Bowles of Powell, Tennessee. In December of the same year he completed the requirements for a Master of Science degree in Mechanical Engineering at the University of Tennessee at Knoxville and reported for undergraduate pilot training at Williams AFB, Chandler, Arizona.