To the Graduate Council:

I am submitting herewith a thesis written by Mark William Yambert entitled "Extension of the Continuity Constraint Algorithm to Variable-Density Flow Simulation." I have examined the final electronic copy of 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 Engineering Science.

A.J. Baker
Major Professor

We have read this thesis
And recommend its acceptance:

Majid Keyhani

Joseph Iannelli

Accepted for the Council:

Anne Mayhew
Vice Provost and Dean of
Graduate Studies

(Original signatures are on file with official student records.)

Extension of the Continuity Constraint Algorithm to Variable-Density Flow Simulation

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

Mark William Yambert
December, 2002

DEDICATION

I wish to dedicate this work to the following:

Those who were: B.J., H.F., M.A., D.W., and P.A.

Those who are: E.G.W., D.S., D.W., C.C., and B.L.

And, those who always will be: J.C.

ACKNOWLEDGMENTS

It would be impossible to individually acknowledge everyone who assisted me during this phase of my education. However, there are a few individuals, without whose help, a successful completion of my graduate studies would not have been possible. Therefore, I would like to single out the following persons for their individual contributions: Joe Orzechowski of the UT CFD Laboratory (for his help with **aPSE** and general computer knowledge), Dr.'s Don Lee and Jim Freels of the Oak Ridge National Laboratory (for a deeper insight into the field of CFD and help in furnishing data used in preparation of this document, respectively), Mr. Matthew Kelleher of ORNL, Mr.'s Jim Burnett and Bob Painter and Dr. Bill Cain of BWXT in Oak Ridge (for helping me troubleshoot computer-related issues related to the preparation of this document and for allowing me to practice my oral presentation on them). Furthermore, I would like to express thanks to my supervisor, David Hetrick of the Oak Ridge National Laboratory for granting me the flexible work schedule needed during the time took to prepare this document. I am also especially indebted to Mr. Mathew Borgers for helping me restore my computer to operational status when I inadvertently attempted to destroy all files on it two weeks before the completion deadline for this thesis.

I also wish to thank the members of my committee, Dr.'s Majid Keyhani and Joseph Iannelli for their comments and understanding during the preparation of this work.

Above all, I wish to thank my advisor, Dr. A. J. Baker for sharing his knowledge of CFD, providing encouragement (which included a few necessary reprimands along the way in order to get me back on track), the fact that he chose not to retire (leaving me academically stranded) during my studies at UT, and above all for his patience during the three years it took to prepare this document.

ABSTRACT

One class of problems commonly encountered in the study of computational fluid dynamics involves the flow of fluids with variable density. Such flows are characterized by density variations too large for the assumption used in most incompressible Navier-Stokes formulations, that small changes in density are linearly proportional to changes in temperature, to be valid. Unlike fully compressible flows, such as the high-speed flow of gases, variable-density flows are often characterized by low Mach numbers. Examples of such flows include 1) combustion problems, where significant density variations may arise due to the large temperature differences present, and 2) flows involving liquids, such as refrigerated hydrogen, whose density varies significantly over small temperature differences.

While fully compressible algorithms can be used to solve problems involving variable-density flows, such calculations are computationally inefficient. As an alternative, a modified version of the Continuity Constraint Algorithm of Williams has been developed for solving problems involving fluids with variable-density. Originally developed as an inexact method for solving incompressible flow problems, the Continuity Constraint Algorithm belongs to a general class of computational algorithms, normally referred to as either "pressure-projection" or "pressure-correction" methods.

This work derives a variable-density form of the computational algorithm and presents the results of its application to a series of two-dimensional benchmark problems. The first of which involves the buoyancy-induced flow of air inside a closed cavity. Results of these initial algorithm tests showed, that while adequate steady-state solutions were obtained for cases corresponding to Rayleigh number values of 10^4 to 10^7, the algorithm experienced some computational difficulties in reaching steady-state conditions.

A second series of calculations were performed to emphasize the effects of the variable-density assumption. For these calculations, solutions for the buoyancy-induced flow of liquid hydrogen inside a closed cavity at a Rayleigh number value of 10^{10} were generated using both the variable-density and incompressible versions of the Continuity Constraint Algorithm. Like the initial algorithm tests, a number of computational difficulties (some of which were significant) were encountered. Examination of the steady-state results, determined via visual inspection of the temporal evolution of velocity and temperature fields, from these analyses showed significant differences between the variable-density and INS solutions.

CONTENTS

LIST OF FIGURES

LIST OF ANIMATIONS

NOMENCLATURE

English Symbols

aPSE	a Problem Solving Environment
atm	atmosphere
c_p	specific heat
CCA	Continuity Constraint Algorithm
CFD	Computational Fluid Dynamics
CNS	Compressible Navier-Stokes
$[\mathbf{D}]$	unit diffusion matrix.
$^\circ$F	degrees Fahrenheit
FE	finite element
f_j	kinematic flux vector
f_j^{v}	viscous flux vector
$\{\mathbf{FQ}\}$	Newton residual vector
ft	feet
g	acceleration of gravity
GWS	Galerkin Weak Statement
INS	Incompressible Navier-Stokes
$^\circ$K	degrees Kelvin
k	thermal conductivity
kg	kilogram
L	length
$\mathbf{L}$	equation
$[\mathbf{JAC}]$	Newton Jacobian matrix
$[\mathbf{M}]$	global square “Mass” matrix
m	meter
N_k	finite element basis function
$\hat{n}$	unit normal vector
P	effective pressure
P'	pressure
Q	temporal trial space function

$\{\mathbf{Q}\}$	vector of discrete, nodal approximations to state variables
q	state or continuum variable
R	ideal gas law constant
$\{\mathbf{R}\}$	global Residual vector
$\mathbf{R}^n$	real number space
S	source term
s	source term, second
s_α	auxiliary source term
T	temperature
t	time
t_o	initial time
u	velocity component
u	velocity vector
$\hat{u}$	unit vector
u_i^*	iterative velocity value
W	temporal test function
w	test function
x	spatial coordinate

Greek Symbols

α	thermal diffusivity = $k/(\rho c_p)$
B	thermal expansion coefficient
Δ	difference (differential)
δ_{ij}	Kroneckar delta
δ	difference between iterative values
μ	dynamic viscosity
ν	kinematic viscosity = μ/ρ
Φ	spatial test function
ϕ	continuity constraint variable (continuity constraint function)
Ψ	spatial trial space function
η_j	transformation coordinate space
$\mathbf{\Omega}$	problem domain
$\partial\mathbf{\Omega}$	problem boundary
ρ	density
σ_{ij}	Reynolds stress tensor
τ	time

Mathematical Symbols

$\approx$	approximately equal
$\frac{\mathbf{d}}{\mathbf{dt}}$	matrix derivative
$\boldsymbol{\theta}$	dimensionless temperature, implicitness factor
$\subset$	element of
$\| \ \|$	energy semi-norm
$\equiv$	exactly equal
∇	gradient operator
$>$	greater than
$d\tau$	integrand
$d\Omega$	integrand
$d\Gamma$	integrand
$<$	less than
[ ]	matrix quantity
∂	partial derivative
$\sum$	summation operator
$\bigcup$	union
{ }	vector quantity

Additional Superscripts

*	dimensionless quantity
h	finite element domain
k	index
M	interpolation index
N	approximate solution, discrete domain index
p	iteration
T	matrix transform

Additional Subscripts

A	auxiliary
e	elemental value
i	index or component
j	index
k	index
n	value at iteration n
R	reference quantity

Dimensionless Groups

Ec	Eckhart number = $u_R^2 /(c_P \Delta T)$
Fr	Froude number = $u_R^2 /(g L_R) \sim \mathrm{B}\Delta T\ \mathrm{Re}^2\mathrm{Pr}/\mathrm{Ra}$
Gr	Grashof number = $g\mathrm{B}\Delta T L_R^3 / \nu^2$
Pe	Peclet number = RePr
Pr	Prandtl number = $\mu c_p / k$
Ra	Rayleigh number = $g\mathrm{B}\Delta T L_R^3 /(\nu\alpha)$ = GrPr
Re	Reynolds number = $\rho u_R L_R / \mu$

1 INTROUCTION

When properly applied, computational fluid dynamics (CFD) is a tool which can provide valuable insight into an almost immeasurable range and complexity of engineering problems. Such problems range from single phase, incompressible flow to that of complicated, multiphase, compressible flows. In practice, the complexity associated with the computational algorithms used to solve these problems often mirrors the complexity of the problem itself. Since, even with the speed and efficiency of modern computers, it is desirable to reduce as much of the computational overhead as possible, new and efficient algorithms which can be applied to specific classes of problems, are continually sought by engineers and scientists.

Computational solutions to flows governed by the fully compressible Navier-Stokes equations (CNS) can be found throughout the literature. Due to the highly nonlinear nature of the governing system of equations, these algorithms are generally quite complex and computationally intensive. Similarly, numerical solutions for problems involving incompressible Navier-Stokes (INS) flows are even more prevalent in the literature. For this specialized class of problems, the governing CNS equations can be simplified based on the fact that variations in the fluid's density remain small over a wide range of conditions. Typical flow problems which fall into this category include those involving many common liquids (such as water) and those involving low speed flow of common gases (such as air). A third class of problems involves fluids which exhibit significant density variations even though the Mach number for the flow may be low. Examples of this include combusting flows and flows of liquid refrigerants.

The primary purpose of this work is to document the development of a computational algorithm which can be applied to problems involving these "weakly-compressible" or, to be more precise, "variable-density" flows, and to demonstrate the practical application of the algorithm itself. The algorithm presented herein is an extension of a modified version [1] of the Continuity Constraint Algorithm (CCA) [2], originally developed by P.T. Williams. The specific goals associated with this research are defined as follows:

1. Derive the governing equations applicable to low-speed, variable-density flows from the fully-compressible set of Navier-Stokes equations.
2. Generate the corresponding Galerkin Weak Statement (GWS) formulation for this set of governing equations for use with discrete, finite element (FE) computational techniques.
3. Create a variable-density computational algorithm using the variable-density GWS formulation.

4. Using this variable-density GWS formulation, modify the computational approach used by the CCA to create a numerical algorithm, applicable to low-speed, variable-density flows.
5. Evaluate algorithm performance by generating solutions to standard benchmark problems (i.e. the classic case of an air-filled thermal cavity).
6. Compare results obtained using the variable-density and INS versions of the CCA via their application to the buoyancy-induced flow of liquid hydrogen, a fluid which experiences significant density variation over a small temperature range near 34˚K.

2 BACKGROUND

2.1 Pressure Projection Algorithms

In recent years, a general class of computational algorithms, commonly referred to as either "pressure-projection" or "pressure-correction" methods, have been developed for use in the field of CFD. Examples of these algorithms include SMAC [3] and variants of SIMPLE [4]. Rather than directly solving the complete set of Navier-Stokes equations, a process which can be quite expensive computationally, these algorithms employ an iterative scheme. First, a functional relation linking a pressure-correction term to velocity is specified. This relation is then substituted into the continuity equation, resulting in an equation linking the pressure-correction term to the error in continuity. The pressure-correction term is also substituted into the momentum equations in order to obtain their simplified forms. The simplified momentum equations, energy equation, and the equation linking the pressure-correction term to continuity error, are then solved. The resulting solution is then used to update the velocity field and pressure-correction term via the latter's relation to the continuity error. This process is then repeated until the continuity error is reduced below an acceptable value.

2.2 Continuity Constraint Algorithm

The CCA [2] is an extension of this class of algorithms. It is a three-dimensional, time-accurate, finite element algorithm developed for modeling flows of incompressible fluids. The basis for the algorithm is the assumption that the divergence error associated with the continuity equation is equal to the gradient of a scalar, potential function, ϕ. William's [2] pressure-correction term is defined as the sum of ϕ values accumulated over a series of iterations.

In its original form [2], the CCA exhibited less than Newton convergence rates. Mitra [1] solved this problem by incorporating the continuity constraint function, ϕ, into the Jacobian for velocity and temperature, yielding a genuine Newton algorithm. It is this latter form of the CCA which was used as the basis for the work presented in this document.

2.3 Extension to Variable-Density Flows

The focus of this work is the extension of the CCA, originally developed for incompressible flows, to problems involving variable-density flows. In contrast, most incompressible flow algorithms are based on the fact that density variations are small. Hence, the

Boussinesq approximation [5], which assumes a linear relationship between temperature and density, is normally used to approximate the effects of buoyancy. Variable-density flows, while not fully compressible, are characterized by density variations that are too large for this approximation to be valid. One example of such flows are those involving combustion. Here, significant density variations can exist. However, the Mach number may remain relatively low due to the increased speed of sound associated with the high temperatures involved. Other applications include flows of liquids whose material properties vary significantly over small temperature ranges. One example is liquid hydrogen. Near 34˚K, its density varies by nearly 50% over a 2˚K temperature range.

3 MATHEMATICAL FORMULATION

3.1 Governing Equations

The laws governing the low speed, laminar flow of a variable-density fluid can be summarized by the following set of coupled, partial differential equations:

$$\frac{\partial \rho}{\partial \tau} + \frac{\partial \rho u_j}{\partial x_j} = 0 \tag{1}$$

$$\frac{\partial \rho u_i}{\partial \tau} + \frac{\partial \rho u_i u_j}{\partial x_j} + \frac{\partial P'}{\partial x_i} - \frac{\partial}{\partial x_j}\mu\left(\frac{\partial u_i}{\partial x_j} - \frac{\partial u_j}{\partial x_i} - \frac{\partial u_k}{\partial x_k}\delta_{ij}\right) + \rho g_i = 0 \tag{2}$$

$$\frac{\rho c_p \partial T}{\partial \tau} + \frac{\rho c_p u_i \partial T}{\partial x_i} - k\frac{\partial}{\partial x_i}\left(\frac{\partial T}{\partial x_i}\right) - \sigma_{ij}\frac{\partial u_i}{\partial x_j} - S = 0 \tag{3}$$

where:

$$\sigma_{ij} - \mu\frac{\partial u_i}{\partial x_j} - \mu\frac{\partial u_j}{\partial x_i} - \mu\frac{\partial u_k}{\partial x_k}\delta_{ij} = 0 \tag{4}$$

and, density is assumed to be known through some auxiliary relation:

$$\rho = \rho\left(P', T\right) \tag{5}$$

Let:

$$P = P' + \rho g_i x_i \tag{6}$$

and define dimensionless variables as:

$$\begin{aligned} u^* &= u/u_R, \\ x^* &= x/L_R, \\ \tau^* &= \tau/(L_R/u_R), \\ \rho^* &= \rho/\rho_R, \\ P^* &= P/(\rho_R u_R^2), \\ \sigma^* &= \sigma/(\mu_R u_R/L_R), \\ T^* &= (T-T_R)/\Delta T, \end{aligned} \tag{7}$$

Then, neglecting the viscous heating term, $\sigma_{ij}\frac{\partial u_i}{\partial x_j}$, in Equation (3) and assuming no internal heat sources [6], Equations (1-4) can be represented in dimensionless form as:

$$\frac{\partial \rho}{\partial \tau} + \frac{\partial \rho u_j}{\partial x_j} = 0 \tag{8}$$

$$\frac{\rho \partial u_i}{\partial \tau} + \frac{\rho u_j \partial u_i}{\partial x_j} + \frac{\partial P}{\partial x_i} - \frac{1}{\mathrm{Re}}\frac{\partial}{\partial x_j}\left(\frac{\partial u_i}{\partial x_j} - \frac{\partial u_j}{\partial x_i} - \frac{\partial u_k}{\partial x_k}\delta_{ij}\right) + \left(\frac{\rho - 1}{Fr}\right)_i = 0 \tag{9}$$

$$\frac{\rho \partial T}{\partial \tau} + \frac{\rho u_i \partial T}{\partial x_i} - \frac{1}{Pe}\frac{\partial}{\partial x_i}\left(\frac{\partial T}{\partial x_i}\right) = 0 \tag{10}$$

$$\rho = \rho(P,T) \tag{11}$$

where the *'s have been dropped for convenience.

3.2 Extension of the Continuity Constraint Algorithm to Variable-density Flow

Rather than solving the continuity equation directly, the CCA uses an iterative strategy to enforce continuity. This is accomplished by replacing the continuity equation with a Poisson equation for the CCA variable, ϕ, substituting a pseudo-pressure term for the actual pressure term in Equation (9), and then solving for the velocity and temperature fields. Once velocity and temperature fields are known, the actual pressure can be solved via a Poisson equation.

3.2.1 Variable-Density Form of Pressure-Poisson Equation

The iterative strategy employed by the CCA to enforce continuity is based on the assumption that the difference between the actual divergence-free velocity, u_i, and an iterative velocity field, u_i^*, can be expressed as the gradient of a scalar potential function:

$$\left(u_i - u_i^*\right) = -\frac{\partial \phi}{\partial x_i} \tag{12}$$

In the case of incompressible flow, Williams [2] showed this potential function can be related to the general velocity field (i.e., on that is not necessarily divergence free) by:

$$\frac{\partial^2 \phi}{\partial x_i^2} - \frac{\partial u_i^*}{\partial x_i} = 0 = \mathbf{L}(\phi) \tag{13}$$

Assuming the contribution from the term involving the time derivative of density from Equation (8) is negligible, it is postulated that for variable-density flow, this relation can be represented by the following form:

$$\frac{\partial^2 \phi}{\partial x_i^2} - \frac{\partial \rho u_i^*}{\partial x_i} = 0 = \mathbf{L}(\phi) \tag{14}$$

3.2.2 Pressure-Poisson Equation

The variable-density form of the pressure-Poisson equation that is used to solve for the actual pressure, P, is obtained by taking the divergence of the momentum equation, Equation (9). The resulting equation has the following form:

$$\frac{\partial}{\partial x_i}\mathbf{L}(u_i) = \frac{\partial^2 P}{\partial x_i^2} + \frac{\partial}{\partial x_i}\left(\frac{\rho \partial u_i}{\partial \tau} + \frac{\rho u_j \partial u_i}{\partial x_j} - \frac{\partial \sigma_{ij}}{\partial x_j} + \left(\frac{\rho - 1}{Fr}\right)_i\right) = 0 = \mathbf{L}(P) \tag{15}$$

where:

$$\sigma_{ij} - \frac{1}{\mathrm{Re}}\frac{\partial}{\partial x_j}\left(\frac{\partial u_i}{\partial x_j} - \frac{\partial u_j}{\partial x_i} - \frac{\partial u_k}{\partial x_k}\delta_{ij}\right) = 0 \tag{16}$$

3.2.3 Summary of Governing Equations for Variable-Density Flow

The variable-density form of the Continuity Constraint Algorithm equations applicable to the low speed flow of a variable-density fluid can be summarized as follows:

$$\frac{\rho \partial u_i}{\partial \tau} + \frac{\rho u_j \partial u_i}{\partial x_j} + \frac{\partial P}{\partial x_i} - \frac{1}{\mathrm{Re}}\frac{\partial}{\partial x_j}\left(\frac{\partial u_i}{\partial x_j} - \frac{\partial u_j}{\partial x_i} - \frac{\partial u_k}{\partial x_k}\delta_{ij}\right) + \left(\frac{\rho - 1}{Fr}\right)_i = 0 = \mathbf{L}(\mathbf{u}) \tag{17}$$

$$\frac{\rho \partial T}{\partial \tau} + \frac{\rho u_i \partial T}{\partial x_i} - \frac{1}{Pe}\frac{\partial}{\partial x_i}\left(\frac{\partial T}{\partial x_i}\right) = 0 = \mathbf{L}(T) \tag{18}$$

$$\frac{\partial^2 \phi}{\partial x_i^2} - \frac{\partial \rho u_i}{\partial x_i} = 0 = \mathbf{L}(\phi) \tag{19}$$

$$\frac{\partial^2 P}{\partial x_i^2} + \frac{\partial}{\partial x_i}\left(\frac{\rho \partial u_i}{\partial \tau} + \frac{\rho u_j \partial u_i}{\partial x_j} - \frac{\partial \sigma_{ij}}{\partial x_j} + \left(\frac{\rho - 1}{Fr} \right)_i \right) = 0 = \mathbf{L}(P) \tag{20}$$

where the Reynolds stresses are given by:

$$\sigma_{ij} - \frac{1}{\mathrm{Re}} \frac{\partial}{\partial x_j} \left(\frac{\partial u_i}{\partial x_j} - \frac{\partial u_j}{\partial x_i} - \frac{\partial u_k}{\partial x_k} \delta_{ij} \right) = 0 \tag{21}$$

and density is assumed to be related to temperature and/or pressure via an additional relation:

$$\rho = \rho(P,T) = \mathbf{L}(\rho) \tag{22}$$

3.3 Weak Statement Form for Developed Equations

Equations (17-22) define a set of coupled, nonlinear, partial differential equations defined on some domain, $\mathbf{\Omega} \subset \mathbf{R}^n$, for some time $t \geq t_o$. Equations (17) and (18) are of the generalized form:

$$\mathbf{L}(q) = \frac{\partial q}{\partial \tau} + \frac{\partial \left(f_j - f_j^{\mathrm{v}} \right)}{\partial x_j} - s = 0 \tag{23}$$

where, q, f_j, f_j^{v}, and s are defined as:

$$q = \begin{Bmatrix} \rho u_i \\ \rho T \end{Bmatrix} \tag{24}$$

$$f_j = \begin{Bmatrix} \rho u_j u_i + P \delta_{ij} \\ \rho u_j T \end{Bmatrix} \tag{25}$$

$$f_j^{\mathrm{v}} = \begin{Bmatrix} \frac{1}{\mathrm{Re}} \frac{\partial}{\partial x_j} \left(\frac{\partial u_i}{\partial x_j} - \frac{\partial u_j}{\partial x_i} - \frac{\partial u_k}{\partial x_k} \delta_{ij} \right) \\ \frac{1}{Pe} \frac{\partial T}{\partial x_j} \end{Bmatrix} \tag{26}$$

$$s = \begin{Bmatrix} -\left(\frac{\rho - 1}{Fr} \right)_i \\ 0 \end{Bmatrix} \tag{27}$$

Equations (19) and (20) are Poisson equations of the general form:

$$\mathbf{L}(q) = \nabla^2 q - s_\alpha(q) = 0 \tag{28}$$

where, q and s_α are given by:

$$q = \begin{Bmatrix} \phi \\ P \end{Bmatrix} \tag{29}$$

$$s_\alpha = \begin{Bmatrix} \nabla \cdot (\rho u_i) \\ \nabla \cdot (\mathbf{L}(\mathbf{u}) - \nabla P) \end{Bmatrix} \tag{30}$$

3.3.1 Galerkin Weak Statement

The discrete form of these equations can be obtained by replacing the state variables by a continuous approximation which assumes time and space are separable:

$$q(x_j, t) \approx q^N(x_j, t) \equiv \sum_{i=1}^{N} \Psi_i(x_j) Q_i(t) \tag{31}$$

where $\Psi_i(x_j)$ and $Q_i(t)$ are sets of trial functions and unknown coefficients, respectively. Through use a continuum form of the *method of weighted residuals* known as the *weak statement*, the error in q^N can be constrained.

Formulation of the *weak statement* can be described as follows. One first seeks to define a set of test functions, $w(x_j, t)$, such that:

$$\int_\Omega w(x_j, t) \mathbf{L}(q^N) \, d\tau = 0 \tag{32}$$

For any set of test functions, it is assumed that $w(x_j, t)$ can be represented by:

$$w^M(x_j, t) \equiv \sum_{i=1}^{M} \Phi_i(x_j) W_i(t) \tag{33}$$

By making the integral stationary with respect to the set of functions, $W_i(t)$, the requirement defined by Equation (32) can be enforced for any set of test functions, $w(x_j, t)$. The resulting w*eak statement*, is given by:

$$WS^N = \frac{\partial}{\partial W_i} \int_\Omega w^M(x_j, t) \mathbf{L}(q^N) \, d\tau = \int_\Omega \Phi_i(x_j) \mathbf{L}(q^N) \, d\tau = 0 \text{ for } 1 \le i \le M \tag{34}$$

The optimal choice according to Baker [7], for the test function set, Φ_i can be made by setting it equal to the trial function set, $\Psi_i(x_j)$, defined in Equation (31). The resultant form of Equation (34), termed the *Galerkin weak statement* (GWS), is given by:

$$GWS^N = \int_\Omega \Psi_i(x_j) \mathbf{L}(q^N) \, d\tau = 0 \text{ for } 1 \le i \le N \tag{35}$$

The GWS can be applied to Equation (23) to yield the general form of:

$$GWS^N = \int_\Omega \Psi_i(x_j)\left[\frac{\partial q^N}{\partial \tau} + \frac{\partial\left(f_j - f_j^{\mathbf{v}}\right)^N}{\partial x_j} - s^N\right] d\Omega = 0 \tag{36}$$

Application of the Green-Gauss theorem yields:

$$\begin{aligned} GWS^N &= \int_\Omega \Psi_i\left(\frac{\partial q^N}{\partial \tau} + \frac{\partial f_j}{\partial x_j} - s^N\right) d\Omega \\ &\quad - \int_\Omega \frac{\partial \Psi_i}{\partial x_j}\left(-f_j^{\mathbf{v}}\right)^N d\Omega + \int_{\partial\Omega} \Psi_i\left(-f_j^{\mathbf{v}}\right)\hat{n}_j\, d\Gamma = 0 \end{aligned} \tag{37}$$

3.3.2 GWS Form of Identified System

Application of the GWS to the momentum equation, Equation (17) yields:

$$\begin{aligned} \int_\Omega \Psi\, \mathbf{L}(\mathbf{u})\, d\Omega &= \int_\Omega \Psi\left(\frac{\partial \rho u_i}{\partial \tau} + \frac{\partial\left(\rho u_j u_i + P\delta_{ij}\right)}{\partial x_j} + \left(\frac{\rho - 1}{Fr}\right)_i\right) d\Omega \\ &\quad - \int_\Omega \frac{\partial \Psi}{\partial x_j}\frac{1}{\mathrm{Re}}\left(\frac{\partial u_i}{\partial x_j} - \frac{\partial u_j}{\partial x_i} - \frac{\partial u_k}{\partial x_k}\delta_{ij}\right) d\Omega \\ &\quad + \int_{\partial\Omega} \Psi \frac{1}{\mathrm{Re}}\left(\frac{\partial u_i}{\partial x_j} - \frac{\partial u_j}{\partial x_i} - \frac{\partial u_k}{\partial x_k}\delta_{ij}\right)\hat{n}_j\, d\Gamma = 0 \end{aligned} \tag{38}$$

or:

$$\begin{aligned} \int_\Omega \Psi\, \mathbf{L}(\mathbf{u})\, d\Omega &= \int_\Omega \Psi\left(\rho\frac{\partial u_i}{\partial \tau} + \rho u_j \frac{\partial u_i}{\partial x_j} + \frac{\partial P}{\partial x_i} + \left(\frac{\rho - 1}{Fr}\right)_i\right) d\Omega \\ &\quad - \frac{1}{\mathrm{Re}}\int_\Omega \frac{\partial \Psi}{\partial x_j}\left(\frac{\partial u_i}{\partial x_j} - \frac{\partial u_j}{\partial x_i} - \frac{\partial u_k}{\partial x_k}\delta_{ij}\right) d\Omega \\ &\quad + \frac{1}{\mathrm{Re}}\int_{\partial\Omega} \Psi\left(\frac{\partial u_i}{\partial x_j} - \frac{\partial u_j}{\partial x_i} - \frac{\partial u_k}{\partial x_k}\delta_{ij}\right)\hat{n}_j\, d\Gamma = 0 \end{aligned} \tag{39}$$

The GWS representation of the energy equation, Equation (18) is given by:

$$\int_\Omega \Psi\, \mathbf{L}(T)\, d\Omega = \int_\Omega \Psi\left(\rho\frac{\partial T}{\partial \tau} + \rho u_j \frac{\partial T}{\partial x_j}\right) d\Omega - \int_\Omega \frac{\partial \Psi}{\partial x_j}\frac{1}{Pe}\left(\frac{\partial T}{\partial x_j}\right) d\Omega \tag{40}$$

$$+\int_{\partial\Omega}\Psi\frac{1}{Pe}\left(\frac{\partial T}{\partial x_j}\right)\hat{n}_j\,d\Gamma \;=\; 0$$

Using Equation (37), the GWS form of Equation (19), for the continuity constraint variable, ϕ, is given by:

$$\begin{aligned}\int_\Omega \Psi\,\mathbf{L}(\phi)\,d\Omega \;&=\; \int_\Omega \Psi\left(\frac{\partial^2\phi}{\partial x_i^2} \;-\; \frac{\partial \rho u_i}{\partial x_i}\right)d\Omega \;=\; 0 \qquad (41)\\ &= \int_\Omega \frac{\partial \Psi}{\partial x_i}\frac{\partial \phi}{\partial x_i} \;+\; \Psi\rho\frac{\partial u_i}{\partial x_i} \;+\; \Psi u_i\frac{\partial \rho}{\partial x_i}\,d\Omega\\ &-\int_{\partial\Omega}\Psi\frac{\partial\phi}{\partial x_i}n_i\,d\Gamma \;=\; 0\end{aligned}$$

Similarly, the GWS form of the pressure-Poisson equation, Equation (20) is given by:

$$\begin{aligned}\int_\Omega \Psi\,\mathbf{L}(P)\,d\Omega \;&=\; \int_\Omega \Psi\frac{\partial}{\partial x_i}\left(\frac{\partial P}{\partial x_i}+\frac{\rho\partial u_i}{\partial\tau}+\frac{\rho u_j\partial u_i}{\partial x_j}-\frac{\partial\sigma_{ij}}{\partial x_j}+\left(\frac{\rho-1}{Fr}\right)_i\right)d\Omega = 0 \qquad (42)\\ &= \int_\Omega \frac{\partial\Psi}{\partial x_i}\left(\frac{\partial P}{\partial x_i} \;+\; \frac{\rho u_j\partial u_i}{\partial x_j} \;-\; \frac{\partial\sigma_{ij}}{\partial x_j} \;+\; \left(\frac{\rho-1}{Fr}\right)_i\right)d\Omega\\ &-\int_\Omega\Psi\frac{\partial}{\partial x_i}\left(\frac{\rho\partial u_i}{\partial\tau}\right)d\Omega\\ &-\int_{\partial\Omega}\Psi\left(\frac{\partial P}{\partial x_i} \;+\; \frac{\rho u_j\partial u_i}{\partial x_j} \;-\; \frac{\partial\sigma_{ij}}{\partial x_j} \;+\; \left(\frac{\rho-1}{Fr}\right)_i\right)n_i\,d\Gamma = 0\end{aligned}$$

or:

$$\begin{aligned}\int_\Omega \Psi\,\mathbf{L}(P)\,d\Omega \;&=\; \int_\Omega \frac{\partial\Psi}{\partial x_i}\left(\frac{\partial P}{\partial x_i}+\rho u_j\frac{\partial u_i}{\partial x_j} \;-\; \frac{\partial\sigma_{ij}}{\partial x_j}+\left(\frac{\rho-1}{Fr}\right)_i\right)d\Omega \qquad (43)\\ &-\int_\Omega\Psi\left(\rho\frac{\partial}{\partial x_i}\left(\frac{\partial u_i}{\partial\tau}\right)+\frac{\partial u_i}{\partial\tau}\frac{\partial\rho}{\partial x_i}\right)d\Omega\\ &-\int_{\partial\Omega}\Psi\left(\rho\frac{\partial u_i}{\partial\tau}\right)n_i\,d\Gamma = 0\end{aligned}$$

Where, the GWS form of the auxiliary equation used to calculate Reynolds stresses, Equation (21) is:

$$\int_\Omega \Psi\,\mathbf{L}(\sigma_{ij})\,d\Omega \;=\int_\Omega\Psi\left(\sigma_{ij}-\frac{1}{\mathrm{Re}}\frac{\partial u_i}{\partial x_j}-\frac{1}{\mathrm{Re}}\frac{\partial u_j}{\partial x_i}+\frac{2}{3}\frac{1}{\mathrm{Re}}\frac{\partial u_k}{\partial x_k}\delta_{ij}\right)d\Omega = 0 \qquad (44)$$

3.3.3 Summary of Galerkin Weak Statement Equations

The GWS form of the equations governing low speed, variable-density flow can be summarized as follows:

$$GWS(\mathbf{u}) = \int_\Omega \Psi \mathbf{L}(\mathbf{u})\, d\Omega \qquad = \int_\Omega \Psi \left(\rho \frac{\partial u_i}{\partial \tau} + \rho u_j \frac{\partial u_i}{\partial x_j} + \frac{\partial P}{\partial x_i} + \left(\frac{\rho - 1}{Fr} \right)_i \right) d\Omega \quad (45)$$

$$- \frac{1}{\mathrm{Re}} \int_\Omega \frac{\partial \Psi}{\partial x_j} \left(\frac{\partial u_i}{\partial x_j} - \frac{\partial u_j}{\partial x_i} - \frac{\partial u_k}{\partial x_k} \delta_{ij} \right) d\Omega$$

$$+ \frac{1}{\mathrm{Re}} \int_{\partial\Omega} \Psi \left(\frac{\partial u_i}{\partial x_j} - \frac{\partial u_j}{\partial x_i} - \frac{\partial u_k}{\partial x_k} \delta_{ij} \right) \hat{n}_j \, d\Gamma = 0$$

$$GWS(T) = \int_\Omega \Psi \mathbf{L}(T)\, d\Omega \qquad = \int_\Omega \Psi \left(\rho \frac{\partial T}{\partial \tau} + \rho u_j \frac{\partial T}{\partial x_j} \right) d\Omega \quad (46)$$

$$- \int_\Omega \frac{\partial \Psi}{\partial x_j} \frac{1}{Pe} \left(\frac{\partial T}{\partial x_j} \right) d\Omega$$

$$+ \int_{\partial\Omega} \Psi \frac{1}{Pe} \left(\frac{\partial T}{\partial x_j} \right) \hat{n}_j \, d\Gamma = 0$$

$$GWS(\phi) = \int_\Omega \Psi \mathbf{L}(\phi)\, d\Omega \qquad = \int_\Omega \left(\frac{\partial \Psi}{\partial x_i} \frac{\partial \phi}{\partial x_i} + \Psi \rho \frac{\partial u_i}{\partial x_i} + \Psi u_i \frac{\partial \rho}{\partial x_i} \right) d\Omega \quad (47)$$

$$- \int_{\partial\Omega} \Psi \frac{\partial \phi}{\partial x_i} n_i \, d\Gamma = 0$$

$$GWS(P) = \int_\Omega \Psi \mathbf{L}(P)\, d\Omega \qquad = \int_\Omega \frac{\partial \Psi}{\partial x_i} \left(\frac{\partial P}{\partial x_i} + \rho u_j \frac{\partial u_i}{\partial x_j} - \frac{\partial \sigma_{ij}}{\partial x_j} + \left(\frac{\rho - 1}{Fr} \right)_i \right) d\Omega \quad (48)$$

$$- \int_\Omega \Psi \left(\rho \frac{\partial}{\partial x_i} \left(\frac{\partial u_i}{\partial \tau} \right) + \frac{\partial u_i}{\partial \tau} \frac{\partial \rho}{\partial x_i} \right) d\Omega$$

$$- \int_{\partial\Omega} \Psi \left(\rho \frac{\partial u_i}{\partial \tau} \right) n_i \, d\Gamma = 0$$

$$GWS(\sigma_{ij}) = \int_\Omega \Psi \mathbf{L}(\sigma_{ij})\, d\Omega \qquad = \int_\Omega \Psi \left(\sigma_{ij} - \frac{1}{\mathrm{Re}} \frac{\partial u_i}{\partial x_j} \right) d\Omega \quad (49)$$

$$+ \int_\Omega \Psi \left(- \frac{1}{\mathrm{Re}} \frac{\partial u_j}{\partial x_i} + \frac{2}{3} \frac{1}{\mathrm{Re}} \frac{\partial u_k}{\partial x_k} \delta_{ij} \right) d\Omega = 0$$

3.3.4 Finite Element Representation

The finite element method, relies on a spatial discretization, $\Omega^{\mathbf{h}}$, of the of the continuum domain, Ω, which consists of the union of a set of non-overlapping sub-domains, $\Omega_{\mathbf{e}}$ (or elements), such that:

$$\Omega \approx \Omega^n \equiv \bigcup_e \Omega_{\mathbf{e}} \tag{50}$$

Then, q^N, the discrete approximation of the continuous variable, q, is formed by the union of the finite element approximations, $q_{\mathbf{e}}$, on $\Omega_{\mathbf{e}}$ such that:

$$q(x_j,t) \approx q^N(x_j,t) \equiv q^h(x_j,t) = \bigcup_e q_e(x_j,t) \tag{51}$$

The finite element approximation, $q_{\mathbf{e}}$, is given by:

$$q_e(x_j,t) \equiv \bigcup_e \left\{N_k(\eta_j)\right\}^T \left\{Q(t)\right\}_e \tag{52}$$

where the row vector, $\left\{N_k(\eta_j)\right\}^T$, called the finite element *basis set*, is composed of k_{th} degree polynomial functions. There are as many polynomials as there are nodal degrees of freedom in $\Omega_{\mathbf{e}}$.

3.4 Linear Algebra

Evaluating Equations (45) and (46) on an elemental basis results in a set of matrix expressions. Assembling these into a global matrix statement yields an algebraic equation set of the form:

$$GWS^h = [\mathbf{M}]\frac{\mathbf{d}\{\mathbf{Q}\}}{\mathbf{dt}} + \{\mathbf{R}(\mathbf{Q})\} = \{\mathbf{0}\} \tag{53}$$

where $[\mathbf{M}]$ is the global square "Mass" matrix, the "Residual", $\{\mathbf{R}\}$ is a global column vector, and $\{\mathbf{Q}\} \equiv \{\mathbf{Q}(\mathbf{t})\}$ is a vector containing the nodal approximations of the state variables. By approximating the time derivative in Equation (53) using an implicit Euler formulation, the finite element representation of the state variables at time, n+1, are given by:

$$\{\mathbf{FQ}\}_{n+1} = [\mathbf{M}]\{\mathbf{Q}_{n+1} - \mathbf{Q}_n\} + \Delta\mathbf{t}\left(\boldsymbol{\theta}\{\mathbf{R}_{n+1}\} + (\mathbf{1}-\boldsymbol{\theta})\{\mathbf{R}_n\}\right) = \{\mathbf{0}\} \tag{54}$$

The solution of the coupled, nonlinear system of algebraic system represented by Equations (45) – (49) at each time station is accomplished via an iterative procedure. In the case of a true Newton method, assuming $\{\mathbf{Q}\}_n$ and $\{\mathbf{FQ}\}_n$ are known, the corresponding matrix equation which must be solved over a series of iterations is given by:

$$\left[\mathbf{JAC_e}\right]^{p}_{n+1}\left\{\delta\mathbf{Q}\right\}^{p+1}_{n+1} = -\left\{\mathbf{FQ}\right\}^{p}_{n+1} \tag{55}$$

where, the elemental Jacobian, $\left[\mathbf{JAC_e}\right]$, is given by:

$$\left[\mathbf{JAC_e}\right] = \left[\mathbf{M}\right] + \boldsymbol{\theta}\Delta\mathbf{t}\,\frac{\partial\left\{\mathbf{R}\right\}}{\partial\left\{\mathbf{Q}\right\}} \tag{56}$$

and,

$$\left\{\mathbf{Q}\right\}^{p+1}_{n+1} = \left\{\mathbf{Q}\right\}^{p}_{n+1} + \left\{\delta\mathbf{Q}\right\}^{p+1}_{n+1} \tag{57}$$

Elemental values for pressure, the continuity constraint variable, ϕ, and the Reynolds stress term, σ_{ij}, defined in Equations (47-49) are evaluated directly using an algebraic equation of the form:

$$\left\{\mathbf{FQ}_A\right\} = \left[\mathbf{D}\right]\left\{\mathbf{Q}_A\right\} \tag{58}$$

where $\left[\mathbf{D}\right]$ is a unit diffusion matrix.

3.5 Continuity Constraint Algorithm Treatment of Pressure Term

As demonstrated by [2], the actual pressure at a given time, P_{n+1}, can be represented by the sum of the known pressure at time n, P_n, and the accumulation of iterative values of the continuity constraint variable, ϕ according to the following relationship:

$$P^{p}_{n+1} = P_n + \frac{1}{\boldsymbol{\theta}\Delta\mathbf{t}}\sum_{k=1}^{p}\phi^k \tag{59}$$

Substituting this relation for the pressure term in Equation (45) results in 1) an additional term in the mass matrix [M] (as the $\boldsymbol{\theta}\Delta\mathbf{t}$ terms in the numerator and denominator cancel) and 2) an additional term in Equation (47), $\frac{\partial P_n}{\partial x_i}$, which involves the pressure calculated at the previous time station.

3.6 Software and Computational Platform

The finite element formulation and computational strategy described previously sections were implemented using **aPSE**. **aPSE** (which stands for "a Problem Solving Environment") is a finite element software package developed at the UT CFD Laboratory. Unlike traditional finite element codes, which are normally designed for a specific class of problems (i.e., fluid flow, solid mechanics, etc.), **aPSE** is a generic system applicable to any class of problem for which a finite element representation can be formulated. This is accomplished via the use of a "template" file which uses a "shorthand" notation to describe the finite element system governing the problem.

Problem "data" (i.e., material properties, mesh parameters, etc) is entered into a separate, "model" file. **aPSE** also provides the user with the additional ability to make problem-specific modifications such as changes to the numerical solution process or specification of auxiliary relations (such as equations of state) via use of hooks" files. All calculations were performed on a SUN Sparc Ultra-80 UNIX workstation running SunOS Version 5.8.

3.7 Density Calculation

Unlike the work of [1] and [2] where density is constant, this work focuses on low-speed flows which can have significant changes in density both spatially and temporally. Two different fluids were analyzed, air and liquid hydrogen. For low speed flows involving air density was calculated using the ideal gas law, the dimensional form of which is given by:

$$\rho = RPT^{-1} \tag{60}$$

From this relation, the partial derivative of density with respect to temperature for an ideal gas can be shown to equal:

$$\frac{\partial \rho}{\partial T} = -RPT^{-2} = -\rho T^{-1} \tag{61}$$

Computationally, the calculation of density and its partial derivative with respect to temperature was implemented by modifying the user programming interface or "hooks" section of **aPSE**.

For liquid hydrogen calculations, an equation of state relating density to temperature was not available. Instead, values for density were linearly interpolated from detailed empirical data [8] found in Table 1 of APPENDIX B. This data was also initially used to estimate the partial derivative of density with respect to temperature via a first-order accurate Taylor series. Computationally, these calculations were again implemented by modifications to the "hooks" section of **aPSE**.

3.8 Variable-Density Algorithm Computational Strategy

The computational strategy used by the variable-density form of CCA closely followed that of the incompressible forms outlined in both [1] and [2]. The chief difference introduced in this work was the inclusion of an additional step in which terms involving density (and derivatives of density) were updated. The iterative strategy used by the variable-density algorithm at time step n+1 is outlined as follows:

1. Initialize state variables based on values at the previous time step (time n).
2. Calculate density based on nodal values for the state variables T and P as well as terms involving derivatives of density with respect to time and pressure (either exactly or via approximate methods).
3. Solve momentum, energy, and ϕ equation set for the p^{th} iteration.
4. Update the sum of CCA variable, ϕ.
5. Update the nodal density values based on the current nodal values of the state variables T and the value of P from the last completed time step.
6. Repeat steps 2-5 until a converged solution is obtained.

7. Solve the auxiliary Reynolds stress equation and the pressure-Poisson equation for the current time step (time n+1).
8. Advance to next time step and repeat steps 2-7 until a steady state solution is reached. The steady-state stopping criteria is defined in terms of the energy semi-norm which, for any given variable, q, the elemental representation is given by:

$$\| q^p \| \equiv \left. \frac{\frac{1}{2}\int_\Omega \frac{\partial q}{\partial x_j}\frac{\partial q}{\partial x_j} d\Omega}{\int_\Omega d\Omega} \right|_{\mathbf{n+1}}^{p} \tag{62}$$

4 ALGORITHM TESTS

Natural convection or buoyancy-induced flows have long been of interest from both a practical as well as computational standpoint. Practically, these problems are of interest in areas such as building ventilation flows. Computational solutions to flows which arise due to temperature variations in the fluid were used by both [1] and [2] to benchmark the incompressible forms of the CCA algorithm. Specifically, these authors investigated buoyancy-driven flows involving air inside a closed cavity. Having formulated the governing equations, their corresponding GWS formulation, and the associated computational strategy, such flows also serve as an initial means of evaluating the performance of the variable-density form of the CCA.

4.1 Background: Buoyancy-Induced Flow Inside a Closed Cavity

The problem considered here involves a closed, rectangular, fluid-filled cavity. The top and bottom of the cavity are treated as adiabatic, while the two vertical walls on opposite sides of the cavity are held at constant temperatures. Historically, the value of a dimensionless quantity known as the Rayleigh number (Ra) has been used to characterize internal, natural convection flows. At lower values ($Ra \leq 10^3$), The heat transfer in the cavity is dominated by conduction [9]. The corresponding flow follows a steady, unicellular rotation pattern centered in the middle of the cavity [9]. At higher values ($10^3 < Ra < 10^5$) convective boundary layers develop near the walls as the flow transitions from one characterized by pure conduction. As the value for Ra increases, convection becomes the dominant force as most of the flow is confined to a small region near the cavity walls. The central region is almost stagnant. At even higher Ra values ($>10^6$, secondary recirculation patterns appear until the flow becomes turbulent somewhere around $Ra = 10^9$.

4.2 Initial Algorithm Tests: Buoyancy-Induced Flow of Air

Both [1] and [2] applied their algorithms to problems involving buoyancy induced flow within a closed cavity. Using this as a guideline, the buoyancy-induced flow of air inside a closed, thermal cavity was used as a test problem for initial algorithm benchmarks.

4.2.1 Problem Description and Goals

The two primary goals of the first phase of the analysis were to 1) verify algorithm performance for lower Ra values (i.e., $Ra < 10^4$), and 2) having thus established a baseline of the

algorithm's performance, explore its behavior at higher Ra values. Initial algorithm tests were conducted using a square cavity with sides of unit length as the domain. The cavity was assumed to be filled with an ideal gas, air, initially at 70˚F and 1 atm. Temperatures on both vertical walls were fixed. The left vertical wall was held at a constant 70˚F, while the temperature on the right wall was varied in order to achieve the desired Ra value. The upper and lower walls of the cavity were assumed to be adiabatic. In addition, no-slip conditions were assumed to exist on all four walls.

4.2.2 Reference Values and Additional Model Input

As indicated in Section 3.1, reference values were used in obtaining the dimensionless forms of the governing equations. For initial algorithm tests, U.S. System units were used for all input parameters and model variables. Reference values for density, viscosity, thermal conductivity and specific heat were based on thermal and mechanical properties of air at 70˚F and 1 atm. A value of 0.06 ft/s, corresponding to a Re value of approximately 383, was selected as the reference velocity for the analysis. At this Re value, the wall temperature difference corresponding to a Ra value of 10^4 is approximately 0.006 ˚F. The model time scale was set equal to the (unit) reference length divided by this reference velocity.

4.2.3 Finite Element Mesh

Proper choice of the finite element mesh (in terms of both size and spacing) is critical if one hopes to obtain a converged and meaningful solution on the computational domain. This is especially true in the case of the closed thermal cavity. At low Ra values (i.e. 10^3), adequate solutions can be obtained using relatively coarse meshes. However, as the Ra increases causing boundary–layer regions to develop near the cavity walls, increasingly complex and refined meshes are required in order to support a solution. Because of the exploratory nature of the initial algorithm tests, it was necessary to select a mesh size which could potentially support solutions over a range of Ra values. Mitra [1] was able to obtain solutions up to Ra = 10^7 using a non-uniform, 32x32 node mesh. Optimal spacing for this mesh size can be achieved using a geometric mesh progression ratio (i.e. the ratio of the lengths of any element and an adjacent element on the side closest to the cavity wall) of 1.1. Hence, initial variable-density algorithm evaluations were conducted using a non-uniform, 32x32 node mesh with a mesh progression ratio of 1.1.

4.2.4 Summary of Computational Strategy

The computational strategy used in the analysis is outlined as follows:

1. For nodes located along vertical walls, assign fixed temperatures corresponding to a Ra value of 10^4, which, for air at Re ~ 383, corresponds to a wall temperature differential of approximately 0.006 ˚F.
2. Interior nodal temperatures are initialized by linear interpolation of fixed wall values.
3. Assign a uniform, initial pressure distribution.
4. Initialize density via the ideal gas law using the temperature and pressure distributions specified in steps 1-3.
5. Arbitrarily initialize remaining algorithm variables to zero.
6. Generate steady-state, finite element solution using the variable-density form of the CCA.
7. Examine the quality of the solution produced, checking for extremization of energy semi-norms, Equation (62), and monotonicity of solution, etc. If necessary, adjust input

parameters (i.e., time step size, convergence criteria, etc) until a satisfactory solution is obtained.

8. Increment the Ra value by an order of magnitude (i.e., set Ra = 10^5) by reassigning temperature values at nodes located along the right, vertical wall. Initialize remaining model variables using the non-dimensional, steady-state results obtained for the previous (Ra = 10^4) solution.
9. Repeat Steps 6 and 7 until a converged solution corresponding to a Ra value of 10^5 is obtained.
10. Increase the Ra value by an order of magnitude and repeat Steps 8 and 9 until a converged solution can no longer be obtained. Note, this process of using results obtained for a given Ra value as initial conditions for additional computations at a larger Ra values is often referred to as "Rayleigh continuation".
11. Evaluate algorithm performance at each Ra level. Determine areas which need improvement/modification.

4.3 Practical Algorithm Application: Buoyancy-Induced Flow of Liquid Hydrogen

The theoretical advantage of the variable-density formulation of the CCA over its INS form is its applicability to flows where 1) density variations are no longer small or 2) the linear relation between density and temperature (assumption in the Boussinesq approximation) is no longer valid. Examples of flows which exhibit these properties include combusting flows where, although the Mach number can remain relatively low due to extreme temperatures (hence the flow is not fully-compressible), density variations can be too large for the flow to be modeled as incompressible. Additional flows which deviate from the standard INS assumption are those involving refrigerants such as liquid hydrogen. Liquid hydrogen is used as a coolant in nuclear reactor applications. At temperatures near absolute zero, its mechanical and thermodynamic properties vary significantly with temperature. For example, at 36˚K, the density of liquid hydrogen is almost 50% less than its value at 34˚K. Furthermore, unlike an ideal gas where density is inversely proportional to temperature, the relation between density and temperature for liquid hydrogen in this temperature range is nonlinear. Having established a baseline of its performance using air, the capabilities of the variable-density form of the CCA were further tested via its application to flows involving low-temperature, liquid hydrogen.

4.3.1 Problem Description and Goals

For this analysis, the variable-density form of the CCA was used to obtain steady-state solutions for the buoyancy induced flow of liquid hydrogen inside a closed thermal cavity. Results from this analysis were compared to those generated from an analogous set of calculations made using the INS form of the CCA. As with initial algorithm tests, a square, closed, thermal cavity was selected as the computational domain for this phase of algorithm development. In order to emphasize the differences between results obtained using the variable-density and incompressible algorithm forms, a 2˚K temperature range (between 34˚K and 36˚K) where density variations are the greatest, was selected as the target for computations.

Substitution of liquid hydrogen for air in the thermal cavity introduces a significant computational difficulty due to differences in its thermodynamic and material properties. For the same cavity size, the Ra number for a liquid hydrogen-filled cavity is on the order of 10^{16}, almost ten orders of magnitude higher than the largest Ra value for which a converged solution using air

as the fluid was obtainable! Even if this Ra value was reduced by ten orders of magnitude, it was unlikely, given the computational difficulties encountered during the initial algorithm tests with air, that a single calculation would produce converged and meaningful results. Therefore, as with the initial algorithm tests, a series of calculations were made in which both the variable-density and INS forms of the CCA were first used to calculate solutions for the buoyancy-induced flow liquid hydrogen inside a closed thermal cavity at a modest Ra value. These solutions were progressively used as initial conditions for calculations at the next, higher Ra value until the desired, two degree temperature difference was obtained.

Unlike the initial algorithm evaluations, the goal of these intermediate analyses was simply to provide converged solutions as input conditions for the case involving the next higher Ra value. Obtaining steady-state solutions was not practical due to the excessive solution times involved. Consequently, depending on the Ra value, the intermediate solutions were terminated after 100 to 500 time steps.

4.3.2 Reference Values Used in Non-Dimensionalization of Model Variables

For this phase of algorithm tests, SI units, consistent with the tabular data from [8] (See APPENDIX B) were used for all input parameters and model variables. As previously indicated, the thermal properties of liquid hydrogen near 34˚K are such that, for a closed thermal cavity and a reference length of 1.0 ft (0.305 m), a 2˚K wall temperature difference corresponds to a Ra value of approximately 10^{16}. As this value is well beyond even the largest Ra number for which Mitra [1] was able to obtain a solution with the INS form of the CCA, it was necessary to modify the reference length used in this phase of the analysis to one which would correspond to a realistic Ra value at the required, 2˚K wall temperature differential. Consequently, a reference length scale of 0.01 m was chosen for use in the analysis. At the required 2˚K temperature difference, this results in a Ra value of 10^{10}, which still exceeds the range of values for which Mitra [1] was able to obtain solutions. Reference values for density, viscosity, thermal conductivity and specific heat used in both the variable-density as well as INS calculations were based on the tabulated thermal and mechanical properties of liquid hydrogen at 34˚K [8]. For INS calculations the value used for the thermal expansion coefficient, B, was based on the average temperature of the two vertical walls.

4.3.3 Finite Element Mesh

Without a doubt, the most important requirement for any FE computation's success (or failure) is the ability of the computational mesh to support a converged solution. As seen with the initial algorithm tests involving air, a Ra value of 10^7 appeared to be the limit for which solutions were obtainable using a 32x32 node mesh. Therefore, in order to obtain solutions for liquid hydrogen filled cavity at the desired Ra value of 10^{10}, it was understood that a more sophisticated mesh would be required. The most common method of accomplishing this is to simply increase the mesh size. Mitra reported using mesh sizes ranging from 64x64 to 128x128 in order to obtain solutions for Ra value ~ 10^8. Unfortunately, simply increasing the mesh size, introduces additional complications, namely an exponential increase in computational time. An alternative, which is commonly used in place of or in addition to increasing the mesh size, is to modify the nodal spacing. Mitra [1] demonstrated how even subtle changes in the mesh progression ratio could lead to significant changes in computational efficiency and accuracy.

Hence, in generating solutions for the variable-density flow of liquid hydrogen within a closed thermal cavity, numerous combinations of mesh size and spacing were evaluated. Use of both uniform and non-uniform 64x64 and 128x128 node meshes produced solutions which either

failed to converge at all, diverged after an initial period of convergence, or exceeded the practical, computational time limits required to obtain steady-state solutions (See Section 5.2.1).

Experimentation with multiple mesh configurations resulted in a modified 64x64 node mesh composed of an inner and outer region. The inner region extends from the center of the cavity outward to within 0.001 m of the cavity walls. While area-wise, it encompasses the majority of the computational domain, this region is coarsely gridded, containing only 15% of the mesh's nodes. The remaining portion of the computational mesh is concentrated in a finely gridded region that extends outward from the center region to the cavity's walls. This mesh is pictured in Figures 22 and 23 (Note: see APPENDIX C for all figures).

4.3.4 Additional Considerations

Initially, attempts were made to calculate the Jacobian terms involving the partial derivatives of density with respect to temperature using a first-order accurate Taylor series representation for the variable-density problem formulation. Over the course of model runs it became apparent that the terms introduced by this process were interfering with solution convergence. The offending term was replaced with the corresponding INS formulation which uses the ratio of Grashof number (Gr) divided by Re^2. Since this term only appears in the Jacobian section and not the residual, it only affects the problem convergence rate, not the actual solution.

4.3.5 Summary of Computational Strategy

The computational strategy used in calculating the variable-density flow of liquid hydrogen is nearly identical to that used in the initial algorithm tests involving air and summarized below:

1. For the variable-density CCA formulation, assign fixed temperatures along vertical wall nodes corresponding to a Ra value of 10^5.
2. Initialize interior nodal temperatures by linear interpolation of fixed wall values.
3. Initialize density field according to initial nodal temperatures via interpolation of tabular data.
4. Assign uniform, initial distributions to remaining algorithm variables: velocity, pressure, ϕ, etc.
5. Generate a finite element solution using the variable-density form of the CCA corresponding to a specified number of time steps.
6. Follow analogous procedure described in Steps 1-5 to generate a finite element solution using the INS form of the CCA.
7. Examine the quality of the solution produced, checking for extremization of energy norms, Equation (62), and monotonicity of solution, etc. If necessary, adjust input parameters (i.e., reference parameters, time step size, convergence criteria, etc) until a satisfactory solution is obtained for a moderate (i.e. $< 10^3$) Re value.
8. Upon verification of solution quality, reassign fixed temperatures along vertical walls corresponding to the next, order of magnitude higher Ra value.
9. Initialize remaining velocity, temperature, pressure and CCA variable, ϕ, using the non-dimensional results from the steady-state solution obtained for the previous Ra value.
10. For the variable-density algorithm case, initialize density according to initial temperature distribution.

11. Repeat Steps 5-10 generating finite element solutions using both the variable-density and INS forms of the CCA until 2˚K wall temperature difference is achieved, i.,e. apply "Rayleigh continuation" process.
12. For the final case corresponding to a 2˚K temperature difference, continue the solution process until a steady-state solution is reached for both the variable-density and INS problem formulations.
13. Compare solutions obtained from the variable-density and INS forms of the CCA noting significant differences, similarities, etc.

5 RESULTS

Quantitative results obtained from the analyses described in Sections 4.2 and 4.3 are presented in the following sections. Results obtained from the initial application of the variable-density form of the CCA algorithm to the buoyancy-induced flow of air inside a closed cavity are presented in Section 5.1. Those obtained from the application of the variable-density and INS forms of the CCA to flows involving liquid hydrogen are given in Section 5.2.

5.1 Initial Variable-Density Algorithm Evaluations

The results of the initial algorithm tests in which the variable-density form of the CCA was used to analyze the buoyancy-induced flow of air inside a closed cavity are presented in the following sections.

5.1.1 Initial Evaluations: Ra = 10^4

Initial algorithm tests at a modest Ra value of 10^4 using the variable-density form of the CCA proved successful. Generally, the Newton algorithm was able to converge within three iterations at each time station. Examination of the energy semi-norms (Figure 1) reveals that the algorithm converges to a true, steady-state solution within 120 time steps. Energy norms for velocity and temperature show a monotonic and rapid convergence. Initially (i.e., for time step values < 60) the norm for pressure is somewhat less well behaved, exhibiting oscillations before it converges. In contrast to the well behaved norms for velocity and temperature, the energy norm for the CCA variable, ϕ, is extremely oscillatory in nature during the period in which the flow is developing. These oscillations gradually dampen out as the solution approaches steady-state. This behavior of the CCA variable is consistent with that reported for the INS version of the algorithm [1].

Recalling Section 3.8, the variable-density form of the CCA calculates nodal density values using the pressure field from the previous time station. This introduces a slight temporal inconsistency as the temperature values used in the ideal gas equation are taken from the current time station. Initially, an attempt was made to overcome this discrepancy and make the algorithm truly time-accurate by calculating densities based on the sum of the previous pressure field and the CCA variable, ϕ. Unfortunately the oscillatory nature of the CCA ϕ function (as seen by the evolution of its energy norm) translated to a poorly behaved density field which made convergence to steady-state impossible. Hence this initial approach was dropped in favor of one using a "retarded" pressure. Plots of the temperature, velocity, CCA variable, and pressure fields

are shown in Figures 2-5, respectively. These results are consistent with those obtained by [1] and [10] (see Figure 21).

5.1.2 Initial Evaluations: Ra = 10^5

As indicated in Section 4.2.4, the results obtained at Ra = 10^4 for the velocity and temperature fields were used as initial conditions for an analysis at Ra = 10^5. Again, the algorithm appeared to exhibit Newton or near Newton convergence rates during the initial time steps. However, as the solution progressed in time, it became apparent that a steady-state value could not be reached.

To investigate the cause of this, equivalent INS forms of the model and template files were created based on the input parameters used in the variable-density formulation. A simulation using this INS formulation was then used to confirm that a steady-state solution could be obtained. Having confirmed the performance CCA using the INS formulation, a series of simulations were performed in which the INS form of algorithm was gradually converted to the variable-density formulation by reintroducing terms from the latter back into the template file. Upon examination of the solutions obtained from this exercise, the primary source of the convergence problems was traced to be the terms in the pressure equation used to approximate the partial derivatives of velocity with respect to time. This was confirmed by running the variable-density formulation with these terms dropped. The resulting solution converged to steady state after approximately 300 iterations. It should be noted that, while removal of the time derivative terms allowed variable-density formulation to reach a steady-state solution, it also meant that the algorithm was no longer technically time-accurate. Hence, from this point on, it is more accurate to describe solutions at a given time step as solutions corresponding to an outer iteration in the process of reaching steady-state conditions.

Plots of the energy semi-norms for Ra = 10^5 are given in Figure 6 These plots show the modified variable-density algorithm converges to steady-state in approximately 300 time steps. With the offending time derivative terms removed, these plots show even better convergence characteristics than for the Ra = 10^4 solutions, particularly in the case of ϕ. Corresponding plots of the temperature, velocity, ϕ, and pressure fields are shown in Figures 7-10, respectively. Again, these results are consistent with those obtained by [1] and [10] (see Figure 21).

5.1.3 Initial Evaluations: Ra = 10^6 and Ra = 10^7

Again, the results obtained for the previous (Ra of 10^5) solution were used as initial conditions for a solution at a Ra of 10^6. Even after removal of the terms used to approximate the partial derivatives of velocity with respect to time, the algorithm appeared to have difficulty in reaching a steady-state solution, as can be seen (Figure 11) by the evolution over time of the energy semi-norms as defined in Equation (62). A closer evaluation of the energy norms for velocity and temperature reveals that after about 250 time steps, the variable-density solution appears to be oscillating about an asymptotic value. Hence, although the energy norms have not converged, the solution appears to have essentially reached a steady-state condition. The temperature, velocity, ϕ, and pressure fields corresponding to this solution after 450 time steps are shown in Figures 12-15, respectively. The Ra = 10^7 test case produced even greater oscillations in energy norms after 500 time steps (See Figure 16). Plots of temperature, velocity, ϕ, and pressure fields can be seen in Figures 17-20, respectively. Using these results as initial conditions, a solution for a final test case at Ra = 10^8 was attempted. In this case instance, no solution past the first few time steps could be obtained. This is consistent with Mitra's [1] results

which showed that the ability of a 32x32 node mesh to support a solution at this Ra value was marginal at best.

5.2 Practical Application Results: Buoyancy-Induced Flow of Liquid Hydrogen

Results obtained from simulations involving the buoyancy-induced flow of liquid hydrogen inside a closed cavity are presented in the following sections. For comparison purposes, solutions calculated using both the variable-density and INS forms of the CCA are presented.

5.2.1 Results: Variable-Density Formulation

The results from the initial algorithms tests served as a clear indicator that extensive refinement to the computational mesh would be needed to obtain solutions corresponding to a 2˚K temperature difference (Ra ~ 10^{10}). Initially a non-uniform, 64x64 node mesh with progression ratios ranging from 1.1-1.2 was used to generate solutions for the liquid hydrogen-filled cavity. This mesh produced converged solutions for lower Ra values (i.e. temperature less than 0.002˚K). However, at Ra values greater than 10^8 (a temperature difference of 0.02˚K), the finite element solution began to diverge after only 100-200 time steps. Results obtained from a similar computation using a uniform, 128x128 node mesh were only marginally better. Furthermore, use of the 128x128 node mesh translated to solution times which averaged ten days per simulation. This was clearly unacceptable.

Additional experimentation with mesh sizes and spacing led to the application of the mesh described in Section 4.3.3 which consisted of a finely-gridded outer region and a coarsely-gridded inner region (Figures 22 and 23). After a few initial attempts in which the mesh spacing near the cavity walls was adjusted, a 500 time step simulation was performed at a Ra value of 10^8. Examination of the results of this simulation showed that, while the system had not yet reached a steady-state condition, there were no signs of numerical problems which had plagued the solutions obtained using previous meshes. The temperature difference was increased to 0.2˚K (Ra = 10^9) and an additional, 500 step simulation was performed. Again, examination of the results showed no signs of numerical problems after 500 time steps. Finally, at the required 2˚K temperature difference, the solution was allowed to progress for 1000 time steps. While the resulting solution showed no obvious signs of numerical difficulties, examination of the energy semi-norms revealed that it had not yet reached a steady-state condition. The simulation was restarted at this point and allowed to continue for an additional 1000 time steps. Again, while a converged solution was obtained, the code showed no signs of detecting steady-state conditions and stopping. After a series of simulations/restarts were performed, it became obvious that the computer code's criteria for detecting steady-state conditions, defined in terms of the energy semi-norms of selected state variables, was not being satisfied. Realizing this inability to converge indicated a weakness in the algorithm's Jacobian, the algorithm's computational strategy was modified so that the nodal density update was performed after converged solutions for temperature, velocity, pressure, and ϕ were obtained at each time station. Using this modified algorithm, the previous solution strategy was repeated. Even with these modifications, the algorithm still failed to converge to steady-state conditions.

As an alternative stopping criteria, temperature fields corresponding to each time step were plotted and combined to create an animation of the solution's progress (See APPENDIX D). Visual inspection of these animations was then used to determine when the solution had reached a

steady-state condition. Plots of temperature, velocity, density, and pressure fields corresponding to this "observed" steady-state condition can be seen in Figures 24 - 32.

5.2.2 Results: INS Formulation

An analogous procedure to that outlined in Section 5.2.1 was used to generate steady-state results using the INS formulation of the CCA for the liquid hydrogen-filled cavity. As with the case involving variable-density flow, a series of simulations at lower Ra numbers were performed until a solution corresponding to the required, 2˚K wall temperature differential was obtained. Again, determination of a steady condition was made manually via inspection of an animation temperature field as it progressed over time. Plots of temperature, velocity, and pressure fields corresponding to this "observed" steady-state condition can be seen in Figures 33 - 38.

6 DISCUSSION OF RESULTS

A qualitative discussion of the results presented in Section 5 is presented in the following section.

6.1 Initial Test Cases

Initial attempts to calculate a time-accurate representation of the buoyancy-induced flow of air inside a closed cavity using variable-density form of the CCA proved successful. At a Ra value of 10^4, both Newton convergence rates as well as a true, steady-state solution (measured in terms of the energy semi-norms of the state variables) were observed. However, as indicated in Section 5.1.2, simply increasing the Ra value by an order of magnitude to 10^5 resulted in failure of the algorithm to achieve a steady-state solution. While removal of offending terms used to approximate the derivatives of velocity with respect to time momentarily corrected this problem, similar difficulties in the algorithm's ability to achieve steady-state conditions began to appear again as Ra was increased to 10^6 and 10^7. Plots of the time-evolution of energy-semi-norms for these cases (Figures 11 and 16, respectively) show an initial period in which the norms seem to converge followed by a period in which they appear to oscillate about an asymptotic value. Translated, the algorithm was having difficulty resolving the physics and the numerics of the problem.

This behavior is indicative of a weakness in the Jacobian and is not without precedence. The energy semi-norms for Ra values between 10^5 and 10^7 are similar in character to those calculated by Mitra [1] using the INS formulation of the CCA with a segregated Jacobian. Furthermore, even with the aid of a coupled Jacobian, Mitra [1] experienced similar difficulties in achieving steady-state solutions at higher Ra values ($\sim 10^8$) as the energy semi-norms he calculated also show an initial period of convergence followed by one characterized by oscillations about an asymptotic value. Hence, the variable-density formulation appears to be magnifying weaknesses in the Jacobian which may already be present in the INS form. It appears that, while the variable-density algorithm formulation appears to resolve the physics of the problem, additional refinements are needed in order to address the convergence issues associated with the numerics of the problem.

One final comment should be made regarding the quality of solutions obtained during these initial algorithm tests. With the possible exception of the case corresponding to Ra = 10^4, the quality of the solutions for all other Ra values would benefit greatly from the use of a more refined mesh. While none of the solutions corresponding to Ra values of $10^5 - 10^7$ show signs of significant dispersion error (typically indicated by the presence of "2-Δx" waves), all would

benefit from the use of a more refined mesh. This is especially true for Ra values of 10^6 and 10^7 (Figures 13 and 18) where the 32x32 node mesh fails to resolve the secondary flow patterns near the top-right and bottom-left corners of the cavity. It should be emphasized that the actual simulation times involved in generating these solutions were often quite lengthy. Solution times of a day or days were not uncommon, particularly for the high Ra (i.e. 10^6 and 10^7) cases. Therefore, since the primary purpose of these initial simulations was simply to demonstrate a successful application of the variable-density formulation, no additional attempts to resolve the flow using more refined meshes were performed.

6.2 Comparison Between Variable-Density and INS Results

The primary difficulty encountered during the initial applications of the variable-density form of the CCA was its ability in converging to a steady-state solution. During the second phase of algorithm tests, in which both the INS and variable-density versions of the CCA were used to analyze the buoyancy-driven flow of liquid hydrogen in a closed cavity, this, as well as a number of even more significant computational difficulties were experienced.

As stated in the previous section, the application of the variable-density form of the CCA to natural convection problems involving air inside a closed cavity was not significantly impacted by the presence of numerical dispersion error. While not optimal, the 32x32 node finite element mesh was capable of producing adequate solutions for Ra values up to10^7. The same cannot be said for cases involving the flow of liquid hydrogen. While the 64x64 node computational mesh initially used in these analyses was capable of generating interim solutions for a small, fixed number of time steps at Ra values less than 10^8, the dispersion error, directly attributed to the inability of the computational mesh to support a solution, made it impossible to generate interim solutions for interim Ra values of 10^8 and 10^9, let alone a steady-state solution at the target Ra value of 10^{10}. In order to obtain solutions at these higher Ra values, extensive modifications to the computational mesh were required (see Figures 22 and 23).

Based on results from the initial algorithm evaluations, it was anticipated that similar computational difficulties would be experienced when trying to achieve a steady-state solution for liquid hydrogen flow at Ra value of 10^{10}. Indeed, having finally developed a suitable computational mesh, attempts to generate a steady-state solution (based on values of the energy semi-norms) were unsuccessful. Again, it should be emphasized that Mitra [1] experienced similar difficulties when calculating solutions at a Ra value of 10^8. The Ra value of 10^{10} for which a steady-state solution is sought here is two orders of magnitude greater than this. Visual inspection of the solution animations described in Section 5.2.1 and 5.2.2, revealed that both the variable-density and INS solutions had effectively reached steady-state conditions (See APPENDIX D).

Both the variable-density and INS solutions for the high, Ra = 10^{10} case show two regions with distinctly different flow patterns. The first is a large, inner region where there is little flow. Here, heat transfer is dominated by conduction. The second is a smaller, outer, boundary layer where heat and mass transfer are dominated by convection. An initial comparison of temperature fields generated by the variable-density (Figure 24) and INS (Figure 33) formulations seems to indicate that there is little difference in the two solutions. This is misleading. While this is certainly true for the inner region, a closer comparison of the boundary layer region near the lower, left (Figures 25 and 34) and upper right (Figures 26 and 35) walls of the cavity reveals significant differences in the variable-density and INS temperature profiles. Also of significance are the differences in symmetry between the solutions obtained using the variable-density and INS algorithms. The temperature contours calculated by the INS algorithm in Figures 34 and 35 line up almost exactly when the plots are rotated 90 degrees (with respect to the outward pointing

normal) and overlayed. In contrast, those calculated using the variable-density formulation are clearly not symmetric. While the temperature field near the bottom left corner resembles the INS solution, the temperature field near the upper right corner of the cavity is significantly different in character. For the INS case, the buoyant force which drives the problem is linearly proportional to the wall temperature differential. For the variable-density case this statement is no longer accurate. Instead, the driving buoyant force is more correctly characterized as being proportional to differences in fluid density. Hence the character of the results obtained for the variable-density case is consistent with what is expected as the data in APPENDIX B clearly shows that density variations with temperature are highly nonlinear in nature.

As seen from this analysis, the differences between solutions calculated using the variable-density and INS forms of the CCA are subtle. Temperature and velocities calculated using the variable-density formulation are very similar to those reported by Mitra [1] which were calculated using the INS formulation. However, as seen from the liquid hydrogen study, these differences become more apparent at high Ra values. Unfortunately, many of the computational difficulties experienced during this study are also a direct result of the fact such high Ra values were required to illustrate algorithm differences. For example, the algorithm's difficulties in reaching steady-state conditions may not be due solely to numerical instabilities. Other authors [11] have reported that buoyancy-driven flows begin to transition to turbulence at similar Ra values. Hence, it is entirely possible that a true steady-state solution may not even exist for the problems considered. While careful selection of model input, such as the choice of fluid (in this case, liquid hydrogen, whose density varies significantly with small changes in temperature) or length scale (here a physically unrealistic length scale of 0.01 m was used) can be used to keep Ra values reasonable, specifying a test problem that will highlight differences between the variable-density and INS forms of the CCA almost guarantees that computational difficulties will arise.

7 CONCLUSIONS

This work has presented a new computer algorithm applicable to CFD problems where the fluid's density is assumed to be variable. The new algorithm is a modified form of Continuity Constraint Algorithm [1, 2], a Galerkin, finite element algorithm originally designed for application to problems involving the flow of incompressible fluids. The new algorithm is designed for use in situations where the Boussinesq assumption (i.e., that density and temperature are linearly related) used by the INS formulation is no longer valid. Both the details of the algorithm's formulation, as well as results of its application to a series of benchmark tests have been presented. The results of these applications were somewhat mixed and may be categorized as both positive and negative.

The positive results obtained using the variable-density form of the CCA can be summarized as follows. First, the functional performance of the algorithm was demonstrated via its application to a series of benchmark problems involving the buoyancy-induced flow of air inside a closed cavity. Results of these analyses showed that the algorithm was able to successfully calculate solutions for a range of moderate $10^4 - 10^7$ Ra numbers. A second series of calculations were made to emphasize the effects of the variable-density assumption. For these calculations, solutions for the buoyancy-induced flow of liquid hydrogen inside a closed cavity were generated using both the variable-density and INS version of the CCA. A close examination of the results of these analyses showed significant differences between the variable-density and INS solutions.

Along with these successes, a number of negative results or deficiencies were identified during the variable-density algorithm's tests. Chief among these was the algorithm's degraded performance at higher Ra values. Benchmark tests involving air showed that, while the algorithm was capable of producing a time-accurate solution at a Ra value of 10^4, numerical difficulties associated with attempts to generate a solution at a Ra value of 10^5 resulted in the complete failure of the algorithm. Subsequent tests traced the source of these numerical difficulties to terms used to approximate the partial derivatives of velocity components with respect to time. Upon removal of these terms, the algorithm produced a converged solution. However, the resultant solution was technically only accurate at steady-state conditions.

A second problem encountered during algorithm applications was its difficulty in converging to a steady-state condition (as defined in terms of changes in the energy semi-norms of the algorithms primitive variables). At moderate Ra values, i.e. 10^6 - 10^7, this translated into poor convergence rates and, hence, unnecessarily lengthy solution times. In addition, the oscillatory nature of energy semi-norms (Fig 6, 11, and 16) was considerably less desirable than that typically obtained from a "well-behaved" solution. Again it should be stressed that the

primary purpose of the initial benchmark calculations, to which these solutions correspond, was to successfully demonstrate the algorithm's applicability over a typical range of Ra values. Had optimal solutions been desired, additional experimentation with other controlling factors (i.e., time step, mesh size, artificial diffusion, etc.) would have been performed. Furthermore, it should be noted that even the INS form of the CCA exhibited similar difficulties at extreme Ra values [1]. While resulting in less than optimal solutions when applied to these initial benchmark cases, these convergence difficulties made it impossible to obtain a true, steady-state solution for the extreme Ra values (i.e. 10^{10}) encountered during the analyses involving liquid hydrogen. While possibly attributable to the actual physics of the problem, this inability to achieve a steady-state solution suggests a weakness in the formulation of the algorithm's Jacobian. While modifications were made to the algorithm's computational strategy in an attempt to correct this deficiency (See Section 4.3.4 and 5.2.1), this problem has not yet been resolved.

In conclusion, while initially successful, the variable-density form of the CCA algorithm appears to be constrained by a number of limitations. The algorithm is functional, but its performance is not optimal. To some extent this may be an artifact of the unrealistic, physical parameters chosen for the analyses to emphasize differences between variable-density and INS algorithm formulations. In addition, the algorithm also suffers from a number of computational limitations. In addition to making the algorithm more efficient computationally, improvements are needed to once again make the algorithm time-accurate at higher Ra values. Finally, the algorithm would benefit from additional improvements which are beyond the scope of this work. These include the robust derivation of the variable-density form of boundary conditions for the CCA variable, ϕ. This would allow application of the algorithm a wider range of problem classes such as that of room air flow caused by differential pressure gradients.

LIST OF REFERENCES

REFERENCES

[1] Mitra, A. J., "On Performance of a Pressure-Projection Algorithm for Incompressible Navier-Stokes", Ms Thesis, University of Tennessee, Knoxville, (2000).

[2] Williams, P. T., "A Three Dimensional Time Accurate Incompressible Navier-Stokes Finite Element CFD Algorithm", PhD Dissertation, University of Tennessee, Knoxville, (1993).

[3] Amsden, A. A. and Harlow, F. H., "The SMAC Method: A Numerical Technique for Calculating Incompressible Fluid Flows, Los Alamos Scientific Laboratory, Los Alamos, New Mexico, University of California, LA-4370, (1970).

[4] Patankar, S. V., *Numerical Heat Transfer and Fluid Flow*, Hemisphere, Washington, D. C., (1980).

[5] Boussinesq, J., *Theorie Analytique de la Chaleur*, Vol. 2. Gauthier-Villars, Paris, (1903).

[6] White, F. M., *Viscous Fluid Flow*, Vol. 2. McGraw-Hill, New York, (1991).

[7] Baker, A. J., *Finite Element Computational Fluid Mechanics*, Hemisphere, New York, (1983).

[8] Lemmon, E. W., *Thermodynamic and Transport Properties of Pure Fluids Database Version 5.0*, National Institute of Standards and Technology (NIST), (2000).

[9] Gebhart, B., Jaluria, Y., Mahajan, R. L., and Sammakia, B., *Buoyancy-Induced Flows and Transport*, Hemisphere, Cambridge, (1988).

[10] de Vahl Davis, G. and Jones, I. P., "Natural Convection in a Square Cavity: A Comparison Exercise", *International Journal of Numerical Methods in Fluids*, Vol 3., 227-248, (1983).

[11] Bejan, A., *Convection Heat Transfer*, John Wiley & Sons, Inc., New York, (1995).

APPENDICIES

APPENDIX A: SAMPLE INPUT FILES

Typical “Model” Input File:

```
TITLE       **** THERMAL CAVITY  MOD.CAV **** (1/99)
     PHI THERMAL CAVITY
INTEGRATION FACTORS
   0.        $   INITIAL_TIME
   1.E6      $   FINAL_TIME
   1.0E-4    $   PROBLEM_CONVERGENCE_CRITERIA
   0.50      $   MAXIMUM_CHANGE_IN_Q_(DQ)
   -0.0500   $   INITIAL_TIME_STEP
   1.25      $   TIME_STEP_MULTIPLIER
  20.00      $   MAXIMUM_TIME_STEP
   0         $   CRITERIA_TO_RAISE_MAX_TIME_STEP
  500        $   MAXIMUM_NUMBER_OF_STEPS
  10         $   MAXIMUM_NUMBER_OF_ITERATIONS_PER_STEP
   1.0E-4    $   ITERATION_CONVERGENCE_CRITERIA
   1.0       $   THETA_IMPLICITNESS_FACTOR
   1.        $   CONVERGENCE_VARIABLE
GLOBAL SCALARS
  [ 0. 1. -1. 0. 0.  $  HT ONE MONE REI RE2I
    0. 0. .5 0. 0.0  $  GRSH PEI HALF SPHISW BETT
    0.0 0.0 0.0      $  EU EC FR2I
    0.0 2.0 0.666667 1.33333 ] $  BETA TWO TTHRD FTHRD
MATERIALS
    2      1
  [ 0.  ] $  DUM1
  [ 0.  ] T  DUM1
OTHERS
 0. 0 26           T RARRAY(26)
  1.0E-1           T PHI_DQ_LEVEL  (EPSPHI)
  0. 0 71          T RARRAY(71)
   1.0             T PHIFCT
  0 28 T    ISCALAR(28)
  1 T       NBITER
  7.              T VECTOR 7
$     COMMON /CTHERM/
$    1    ALC,       GAMMAF,        AEXT0,        AEXT1,        AEXT2,
       1.          1.4           0.0           0.0           0.       # BGU
$    2   UREF,         AREF,       GASCON,         PREF,        PZERO,
     6.0897E-2    0.          4.972E4     2.1168E3     2.1168E3    # BGU
```

```
$    3 CONDCT,      GRAVCN,      SPHEAT,       VISCK,          XMW,
       4.175889E-6  32.17398   .237    1.591337E-4     28.95748   # BGU
$    4   TREF,        TMAX,        TMIN,       TZERO,      DELTEM,
       70.0        6.10126E-03   0.0        459.6        0.          # BGU
$    5    BTU,      XMAREF,       XMAIN,      XMAOUT,      TINTFT,
       778.0       0.          0.          0.          12.         # BGU
$    6     CF,        HREF,       HZERO,       CEXT1,       CEXT2,
       0.          0.          0.          0.          0.          # BGU
$    7   SREF,        SMAX,        SMIN,       SZERO,      DELSAL,
       0.          0.          0.          0.          0.          # BGU
$    8  DEXT1,       DEXT2,       DEXT3,       DEXT4,       DEXT5,
       0.          0.          0.          0.          0.        T
$    9 ZXT(60)
#     You can add (or replace) variables that you have incorporated
#        into your 'hooks' files.
#     Numeric entries are free format.
#     Entries, such as AEXT1 etc. are open for your use.
#  N.B.  ALC is the ONLY variable used in the 'main' subroutines.
#          It is set to 1.0, if you do not set it here.
#          It MUST be in position 1 in this COMMON block.
#   END OF CTHERM
#     COMMON /WALLFN/ TINTPR(100),THKNES(100),SKNFRN(100),TMHICK(100)
  0  32            T
  0               T  ISMPHI
   0 10           T  ISCALR(10)
   1              T  IETKJA
PRINT
    33    33     1   # GRID SIZE    TOTAL OF   1089 NODES
                10   # PRINT INTERVAL (STEPS)
           -1   33   # I WINDOW
            1   33   # J WINDOW
            1    1   # K WINDOW
NONE # VARIABLES IN COUT
NONE # VARIABLES FIRST STEP ONLY
((1P,6(1X,5E14.6/),1X,3E14.6)) # FORMAT FOR 33x33
GRAPHICS
 X1 X2 TEMP U1 U2 PHI PRES  T
THREE_INTEGER    T  INTEGER PRINT FIELD
 TEMP U1 U2 PHI SPHI PRES S11 S22 T
DEBUG
 NONE     $ STEP NUMBER TO PRINT ALL ITERATIONS
 NONE     $ TYPE OF DEBUG PRINT
 1 $ ITERATION NUMBER
 1087,1,1 $ I,J,K  LOCATION, (K=0, ALL PLANES)
 ALL   $ VARIABLE OR ALL
 ALL $ TERM NUMBER OR (- FOR SET NO.), OR ALL
BOUNDARY CONDITIONS
 WALL_NS  1 [ 33I1 1 31I33 34 31I33 66 33I-1 1089 ]
 DIRI_T   1 [ 33I33 1 33I33 33 ]
INITIAL CONDITIONS
  [1089*0.] $ U1
  [1089*0.] $ U2
  [33(0.00 16R1.1 3.05063E-03 16R0.909 6.10126E-03)] $ TEMP # INTER.,
FOR RAYLEIGH NO.
  [1089*0.] $ PHI
  [1089*0.] $ S11
  [1089*0.] $ S12
  [1089*0.] $ S22
```

```
[1089*2116.8] $ PRES
[1089*0.] $ OMGA
[1089*0.] $ PSI
[1089*0.] $ DUNC
[1089*0.] $ TRBC
[1089*0.] $ UM11
[1089*0.] $ UM12
[1089*0.] $ UM22
[1089*0.] $ UMAG
[1089*0.] $ NUSL
[1089*0.] $ SPHI
[1089*0.] $ SPHL
[1089*0.] $ TAU2
[1089*0.] $ SRCT
[1089*1.] $ RHO
[1089*1.] $ RHOM1
[1089*1.] $ DRDT
[1089*1.] $ OLDRHO
[1089*1.] $ OL2RHO
[1089*2116.8] $ OLDPRES
[1089*0.] $ U1OLD
[1089*0.] $ U2OLD
[1089*0.] $ U1PR
[1089*0.] $ U2PR
[1089*0.] $ RHOPR
[1089*1.] $ EPMN
[1089*0.] $ U1E
[1089*0.] $ U2E
[1089*0.] $ TMPE
[1089*0.] $ PHIE
[1089*0.] $ S11E
[1089*0.] $ S12E
[1089*0.] $ S22E
[1089*0.] $ PRSE
[1089*0.] $ OMGE
[1089*0.] $ PSIE
[1089*0.] $ SCAL
[1089*0.] $ OTLN
[1089*0.] $ MTRL
[1089*0.] $ DETJ
[1089*0.] $ DETC
[1089*0.] $ ETKJ
[1089*0.] $ EJK2
[1089*0.] $ EJK3
[1089*0.] $ EJK4
[1089*0.] $ EJK5
[1089*0.] $ EJK6
[1089*0.] $ EJK7
[1089*0.] $ EJK8
[1089*0.] $ EJK9
[1089*0.] $ U1L
[1089*0.] $ U2L
[1089*0.] $ TEML
[1089*0.] $ PHIL
[1089*0.] $ S11L
[1089*0.] $ S12L
[1089*0.] $ S22L
[1089*0.] $ PREL
```

```
  [1089*0.] $ OMGL
  [1089*0.] $ PSIL
  [1089*0.] $ DUNL
  [1089*0.] $ F1
  [1089*0.] $ F2
  [1089*0.] $ F3
  [1089*0.] $ F4
  [1089*0.] $ F5
  [1089*0.] $ F6
  [1089*0.] $ F7
  [1089*0.] $ F8
  [1089*0.] $ F9
  [1089*0.] $ F10
  [1089*0.] $ F11
  [1089*0.] $ G1
  [1089*0.] $ G2
  [1089*0.] $ G3
  [1089*0.] $ G4
  [1089*0.] $ G5
  [1089*0.] $ G6
  [1089*0.] $ G7
  [1089*0.] $ G8
  [1089*0.] $ G9
  [1089*0.] $ G10
  [1089*0.] $ G11
  [1089*0.] $ H1
  [1089*0.] $ H2
  [1089*0.] $ H3
  [1089*0.] $ H4
  [1089*0.] $ H5
  [1089*0.] $ H6
  [1089*0.] $ H7
  [1089*0.] $ H8
  [1089*0.] $ H9
  [1089*0.] $ H10
  [1089*0.] $ H11
  [ 33(0. 16R1.10 0.5 16R0.909 1.0) ] $ X1
  [ -33(0. 16R1.10 0.5 16R0.909 1.0) ] $ X2
  [1089*0.] T ONES
END
```

Typical “Template” Input File:

```
##### **** temp.pns **** {U1 U2 TEMP PHI} {PRES} {OMGA} {PSI} {NORMS}
INTEGRATION FACTORS
   INITIAL_TIME
   FINAL_TIME
   PROBLEM_CONVERGENCE_CRITERIA
   MAXIMUM_CHANGE_IN_Q_(DQ)
   INITIAL_TIME_STEP
   TIME_STEP_MULTIPLIER
   MAXIMUM_TIME_STEP
   CRITERIA_TO_RAISE_MAX_TIME_STEP
   MAXIMUM_NUMBER_OF_STEPS
   MAXIMUM_NUMBER_OF_ITERATIONS_PER_STEP
   ITERATION_CONVERGENCE_CRITERIA
   THETA_IMPLICITNESS_FACTOR
   CONVERGENCE_VARIABLE

TRANSFORMATION ARRAYS
      ETKJ      1.
      DETJ      1

BOUNDARY CONDITIONS
          U1 U2 TEMP PHI S11 S12 S22 PRES OMGA PSI    # ORDER
  FLUX     3  0  0   0  0   0   0   0   0   0    # CONVECTION HEAT FLUX
  HFLUX    4  0  0   0  0   0   0   0   0   0    # HEAT FLUX
  RFLUX    5  0  0   0  0   0   0   0   0   0    # RADIATION HEAT FLUX
  HRFLX    6  0  0   0  0   0   0   0   0   0    # RADIATION HEAT FLUX
  INLT_P   3  0  0   0  0   0   0   0   0   0    # INLET
  WALL_SL  4  0  0   0  0   0   0   0   0   0    # SLIP WALL
  DIRI_U   D  0  0   0  0   0   0   0   0   0    # DIRICHLET U
  DIRI_V   0  D  0   0  0   0   0   0   0   0    # DIRICHLET V
  WALL_NS  D  D  0   0  0   0   0   0   0   D    # NO SLIP WALL
  DIRI_T   0  0  D   0  0   0   0   0   0   0    # WALL TEMPERATURE
  DIRI_PHI 0  0  0   D  0   0   0   0   0   0    # THROUGHFLOW PHI
  DIRI_S11 0  0  0   0  D   0   0   0   0   0    # THROUGHFLOW PRESSURE
  DIRI_S12 0  0  0   0  0   D   0   0   0   0    # THROUGHFLOW PRESSURE
  DIRI_S22 0  0  0   0  0   0   D   0   0   0    # THROUGHFLOW PRESSURE
  DIRI_P   0  0  0   0  0   0   0   D   0   0    # THROUGHFLOW PRESSURE
  DIRI_OMG 0  0  0   0  0   0   0   0   D   0    # DIRICHLET OMGA
  DIRI_PSI 0  0  0   0  0   0   0   0   0   D    # DIRICHLET PSI
  BLANK    D  D  D   D  D   D   D   D   D   D    # BLANK REGION

TITLE
     PHI CONSTRAINT INS GWS ALGORITHM, 2D, QUASI-COUPLED NEWTON JACOBIAN
(12/21/99)

RESIDUALS
  U1   1    #  VARBL, SET NO.,  --- TEMPORAL SET (U1)
 ()()(RHO)(0;1)(B3000)(-U1)
+()()()(1;0)(B201)(PHI)
+()()()(3;0)(B202)(PHI)
+()()()(1;0)(B201)(SPHI)
+()()()(3;0)(B202)(SPHI)

  U1   2    #  VARBL, SET NO.,  --- SPATIAL  SET (U1)
 ()(RHO)(U1+U2)(1020;0)(B3001)(U1)
```

```
+()(RHO)(U1+U2)(3040;0)(B3002)(U1)
+()()()(1;0)(B201)(PRES)
+()()()(3;0)(B202)(PRES)

# S11
+(REI)()()(1122;-1)(B211)(U1)
+(REI)()()(3344;-1)(B222)(U1)
+(REI)()()(1234;-1)(B221)(U1)
+(REI)()()(1234;-1)(B212)(U1)

# S12
+(REI)()()(11;-1)(B211)(U1)
+(REI)()()(13;-1)(B212)(U1)
+(REI)()()(13;-1)(B221)(U1)
+(REI)()()(33;-1)(B222)(U1)
+(REI)()()(12;-1)(B211)(U2)
+(REI)()()(23;-1)(B212)(U2)
+(REI)()()(14;-1)(B221)(U2)
+(REI)()()(34;-1)(B222)(U2)
# terms from 2nd coef of viscosity
+(-,TTHRD,REI)()()(11;-1)(B211)(U1)
+(-,TTHRD,REI)()()(13;-1)(B212)(U1)
+(-,TTHRD,REI)()()(13;-1)(B221)(U1)
+(-,TTHRD,REI)()()(33;-1)(B222)(U1)

  U2   1    #  VARBL, SET NO.,  --- TEMPORAL SET (U2)
 ()()(RHO)(0;1)(B3000)(-U2)
+()()()(2;0)(B201)(PHI)
+()()()(4;0)(B202)(PHI)
+()()()(2;0)(B201)(SPHI)
+()()()(4;0)(B202)(SPHI)

  U2   2    #  VARBL, SET NO.,  --- SPATIAL  SET (U2)
 ()(RHO)(U1+U2)(1020;0)(B3001)(U2)
+()(RHO)(U1+U2)(3040;0)(B3002)(U2)
+()()()(2;0)(B201)(PRES)
+()()()(4;0)(B202)(PRES)

# S22
+(REI)()()(1122;-1)(B211)(U2)
+(REI)()()(3344;-1)(B222)(U2)
+(REI)()()(1234;-1)(B221)(U2)
+(REI)()()(1234;-1)(B212)(U2)

# S12
+(REI)()()(12;-1)(B211)(U1)
+(REI)()()(14;-1)(B212)(U1)
+(REI)()()(23;-1)(B221)(U1)
+(REI)()()(34;-1)(B222)(U1)
+(REI)()()(22;-1)(B211)(U2)
+(REI)()()(24;-1)(B212)(U2)
+(REI)()()(24;-1)(B221)(U2)
+(REI)()()(44;-1)(B222)(U2)
# terms from 2nd coef of viscosity
+(-,TTHRD,REI)()()(22;-1)(B211)(U2)
+(-,TTHRD,REI)()()(24;-1)(B212)(U2)
+(-,TTHRD,REI)()()(24;-1)(B221)(U2)
+(-,TTHRD,REI)()()(44;-1)(B222)(U2)
```

```
+(FR2I)()()(0;1)(B200)(RHOM1)

  TEMP   1    #  VARBL, SET NO.,  --- TEMPORAL SET (TEMP)
 ()()(RHO)(0;1)(B3000)(-TEMP)
  TEMP   2   #  VARBL, SET NO.,  --- SPATIAL  SET (TEMP)
 ()(RHO)(U1+U2)(1020;0)(B3001)(TEMP)
+()(RHO)(U1+U2)(3040;0)(B3002)(TEMP)
+(PEI)()()(1122;-1)(B211)(TEMP)
+(PEI)()()(3344;-1)(B222)(TEMP)
+(PEI)()()(1324;-1)(B221)(TEMP)
+(PEI)()()(1324;-1)(B212)(TEMP)

  PHI   1     #  VARBL, SET NO.,  --- SPATIAL  SET (PHI)
 ()()(RHO)(1;0)(B3001)(U1)
+()()(RHO)(3;0)(B3002)(U1)
+()()(RHO)(2;0)(B3001)(U2)
+()()(RHO)(4;0)(B3002)(U2)
+()()(U1)(1;0)(B3001)(RHO)
+()()(U1)(3;0)(B3002)(RHO)
+()()(U2)(2;0)(B3001)(RHO)
+()()(U2)(4;0)(B3002)(RHO)
+()()()(1122;-1)(B211)(PHI)
+()()()(3344;-1)(B222)(PHI)
+()()()(1324;-1)(B221)(PHI)
+()()()(1324;-1)(B212)(PHI)

JACOBIANS
  U1  U1  1  1  #  VARBL, VARDIF, SET, DIRECTION 1
 ()()(RHO)(;1)(B3000)()
  U1  U1  2  1  #  VARBL, VARDIF, SET, DIRECTION 1
+()(RHO)(U1+U2)(1020;0)(B3001)()
+()(RHO)(U1+U2)(3040;0)(B3002)()
+()(RHO)(U1)(1;0)(B3100)()
+()(RHO)(U1)(3;0)(B3200)()
+(REI)()()(1122;-1)(B211)()
+(REI)()()(3344;-1)(B222)()
+(REI)()()(1234;-1)(B221)()
+(REI)()()(1234;-1)(B212)()
+(REI)()()(11;-1)(B211)()
+(REI)()()(13;-1)(B212)()
+(REI)()()(13;-1)(B221)()
+(REI)()()(33;-1)(B222)()
# terms from 2nd coef of viscosity
+(-,TTHRD,REI)()()(11;-1)(B211)()
+(-,TTHRD,REI)()()(13;-1)(B212)()
+(-,TTHRD,REI)()()(13;-1)(B221)()
+(-,TTHRD,REI)()()(33;-1)(B222)()

  U1  U2  2  1  #
 ()(RHO)(U1)(2;0)(B3100)()
+()(RHO)(U1)(4;0)(B3200)()
+(REI)()()(12;-1)(B211)()
+(REI)()()(23;-1)(B212)()
+(REI)()()(14;-1)(B221)()
+(REI)()()(34;-1)(B222)()

   U1  PHI 1 1 #
 ()()()(1;0)(B201)()
```

```
+()()()(3;0)(B202)()

  U2  U1  2  1  #
 ()(RHO)(U2)(1;0)(B3100)()
+()(RHO)(U2)(3;0)(B3200)()
+(REI)()()(12;-1)(B211)()
+(REI)()()(14;-1)(B212)()
+(REI)()()(23;-1)(B221)()
+(REI)()()(34;-1)(B222)()

  U2  U2  1  1  #
 ()()(RHO)(;1)(B3000)()

  U2  U2  2  1  #
+()(RHO)(U1+U2)(1020;0)(B3001)()
+()(RHO)(U1+U2)(3040;0)(B3002)()
+()(RHO)(U2)(2;0)(B3100)()
+()(RHO)(U2)(4;0)(B3200)()
# terms from S22
+(REI)()()(1122;-1)(B211)()
+(REI)()()(1324;-1)(B212)()
+(REI)()()(1324;-1)(B221)()
+(REI)()()(3344;-1)(B222)()
# terms from S12
+(REI)()()(22;-1)(B211)()
+(REI)()()(24;-1)(B212)()
+(REI)()()(24;-1)(B221)()
+(REI)()()(44;-1)(B222)()
# terms from 2nd coef of viscosity
+(-,TTHRD,REI)()()(22;-1)(B211)()
+(-,TTHRD,REI)()()(24;-1)(B212)()
+(-,TTHRD,REI)()()(24;-1)(B221)()
+(-,TTHRD,REI)()()(44;-1)(B222)()

  U2 TEMP 2  1 #
 (FR2I)()()(;1)(B200)(DRDT)

   U2  PHI  1 1 #
 ()()()(2;0)(B201)()
+()()()(4;0)(B202)()

 TEMP U1 2 1 #
()(RHO)(TEMP)(1;0)(B3100)()+()(RHO)(TEMP)(3;0)(B3200)()

 TEMP U2 2 1 #
()(RHO)(TEMP)(2;0)(B3100)()+()(RHO)(TEMP)(4;0)(B3200)()

  TEMP TEMP 1  1     #
 ()()(RHO)(0;1)(B3000)()

  TEMP TEMP 2  1    #
+()(RHO)(U1+U2)(1020;0)(B3001)()
+()(RHO)(U1+U2)(3040;0)(B3002)()
+(PEI)()()(1122;-1)(B3011)()
+(PEI)()()(3344;-1)(B3022)()
+(PEI)()()(1324;-1)(B3021)()
+(PEI)()()(1324;-1)(B3012)()
```

```
  PHI U1  1 1 #
()()(RHO)(1;0)(B3001)()
+()()(RHO)(1;0)(B3100)()
+()()(RHO)(3;0)(B3002)()
+()()(RHO)(3;0)(B3200)()

  PHI U2 1 1 #
()()(RHO)(2;0)(B3100)()
+()()(RHO)(2;0)(B3001)()
+()()(RHO)(4;0)(B3200)()
+()()(RHO)(4;0)(B3002)()

  PHI PHI  1  1  #  VARBL, VARDIF, SET, ALL DIRECTIONS
 ()()()(1122;-1)(B211)()
+()()()(3344;-1)(B222)()
+()()()(1324;-1)(B221)()
+()()()(1324;-1)(B212)()

GROUP FREQUENCY
      1
SOLUTION TYPE
  DELTA_Q
  ILU_GMRES
  IMPLICIT_EULER
END


# ---- GROUP 2 REYNOLDS STRESSES -------

TITLE
  CNS2D PHI    REYNOLDS STRESSES
#              (and Artificial Diffusion Terms)
#---------- S11 ---------------------

RESIDUALS
  S11 2
 (-,TWO)()()(1;0)(B201)(U1) + (-,TWO)()()(3;0)(B202)(U1)
+(TTHRD)()()(1;0)(B201)(U1) + (TTHRD)()()(3;0)(B202)(U1)
+(TTHRD)()()(2;0)(B201)(U2) + (TTHRD)()()(4;0)(B202)(U2)

#---------- S12 = S21 -------------
  S12 2
 (-)()()(2;0)(B201)(U1) + (-)()()(4;0)(B202)(U1)
+(-)()()(1;0)(B201)(U2) + (-)()()(3;0)(B202)(U2)

#---------- S22 ---------------------
  S22 2
 (-,TWO)()()(2;0)(B201)(U2) + (-,TWO)()()(4;0)(B202)(U2)
+(TTHRD)()()(1;0)(B201)(U1) + (TTHRD)()()(3;0)(B202)(U1)
+(TTHRD)()()(2;0)(B201)(U2) + (TTHRD)()()(4;0)(B202)(U2)

# ---- RE STRESSES TERMS -------
JACOBIANS
  S11  S11   2 1  #  VARBL, VARDIF, SET, # OF TERMS, ALL DIRECTIONS
 ()()()(;1)(B200)()

  S12  S12   2 1  #  VARBL, VARDIF, SET, # OF TERMS, ALL DIRECTIONS
 ()()()(;1)(B200)()
```

```
  S22  S22   2 1  #  VARBL, VARDIF, SET, # OF TERMS, ALL DIRECTIONS
+()()()(;1)(B200)()


GROUP FREQUENCY
  -1
SOLUTION TYPE
  Q
  ILU_GMRES
  IMPLICIT_EULER
END



TITLE
   PHI ALGORITHM,  DELSQ PRESSURE SOLVE

RESIDUALS
  PRES   2   #  VARBL, SET NO.,  --- SPATIAL  SET (PRES)
 ()()(RHO)(1;0)(B3010)(U1PR)+()()(RHO)(3;0)(B3020)(U1PR)
+()()(RHO)(2;0)(B3010)(U2PR)+()()(RHO)(4;0)(B3020)(U2PR)
+()(RHO)(U1)(11;-1)(B3011)(U1)+()(RHO)(U1)(13;-1)(B3012)(U1)
+()(RHO)(U1)(13;-1)(B3021)(U1)+()(RHO)(U1)(33;-1)(B3022)(U1)
+()(RHO)(U2)(12;-1)(B3011)(U1)+()(RHO)(U2)(14;-1)(B3012)(U1)
+()(RHO)(U2)(23;-1)(B3021)(U1)+()(RHO)(U2)(34;-1)(B3022)(U1)
+()(RHO)(U1)(21;-1)(B3011)(U2)+()(RHO)(U1)(23;-1)(B3012)(U2)
+()(RHO)(U1)(14;-1)(B3021)(U2)+()(RHO)(U1)(34;-1)(B3022)(U2)
+()(RHO)(U2)(22;-1)(B3011)(U2)+()(RHO)(U2)(24;-1)(B3012)(U2)
+()(RHO)(U2)(24;-1)(B3021)(U2)+()(RHO)(U2)(44;-1)(B3022)(U2)
+(-,REI)()()(11;-1)(B211)(S11) + (-,REI)()()(13;-1)(B212)(S11)
+(-,REI)()()(13;-1)(B221)(S11) + (-,REI)()()(33;-1)(B222)(S11)
+(-,REI)()()(1122;-1)(B211)(S12)+(-,REI)()()(1234;-1)(B212)(S12)
+(-,REI)()()(1234;-1)(B221)(S12)+(-,REI)()()(3344;-1)(B222)(S12)
+(-,REI)()()(22;-1)(B211)(S22) + (-,REI)()()(24;-1)(B212)(S22)
+(-,REI)()()(24;-1)(B221)(S22) + (-,REI)()()(44;-1)(B222)(S22)
+(FR2I)()()(2;0)(B210)(RHOM1) + (FR2I)()()(4;0)(B220)(RHOM1)

JACOBIANS
  PRES PRES  2  1  #  VARBL, VARDIF, SET, ALL DIRECTIONS
 ()()()(1122;-1)(B211)()
+()()()(3344;-1)(B222)()
+()()()(1324;-1)(B221)()
+()()()(1324;-1)(B212)()

GROUP FREQUENCY
    -1
SOLUTION TYPE
  Q
  ILU_GMRES
  IMPLICIT_EULER
END

TITLE            **** TEMPLATE TEMP.PHI ****
   OMEGA - FROM DELXU  (1/99)

RESIDUALS
```

```
  OMGA   2    #  VARBL, SET NO.,  --- SPATIAL  SET (OMGA)
 ()()()(2;0)(B201)(U1)
+()()()(4;0)(B202)(U1)
+(-)()()(1;0)(B201)(U2)
+(-)()()(3;0)(B202)(U2)

JACOBIANS
  OMGA OMGA  2  1  #  VARBL, VARDIF, SET, ALL DIRECTIONS
 ()()()(0;1)(B200)()

GROUP FREQUENCY
      -10
SOLUTION TYPE
  Q
  ILU_GMRES
  IMPLICIT_EULER
END

TITLE            **** TEMPLATE TEMP.PHI ****
   DELSQ PSI FROM OMEGA  (1/99)

RESIDUALS
  PSI   2  #  VARBL, SET NO.,  --- SPATIAL  SET (PSI)
 (-)()()(0;1)(B200)(OMGA)

JACOBIANS
  PSI PSI  2  1 # VARBL, VARDIF, SET, ALL DIRECTIONS
 ()()()(1122;-1)(B211)()
+()()()(3344;-1)(B222)()
+()()()(1324;-1)(B221)()
+()()()(1324;-1)(B212)()

GROUP FREQUENCY
      -10
SOLUTION TYPE
  Q
  ILU_GMRES
  IMPLICIT_EULER
END

TITLE            **** TEMPLATE TEMP.PHI ****
  ENERGY NORM COMPUTATIONS, ALL VARIABLES/PARAMETERS  (1/99)

RESIDUALS
  DUNC   1   #                    TEMPORAL TERM IN {F(Q)}
 ()()()(0;1)(B200)(-DUNC)

JACOBIANS
  DUNC DUNC   1 1   #             TIME TERM IN {F(Q)}
 ()()()(0;1)(A200)()
  DUNC DUNC   1 2   #             TIME TERM IN {F(Q)}
 ()()()(0;1)(A200)()

NORMS
  OMGE  1  T                           ENERGY NORM FOR OMGA
 (HALF,REI)()(OMGA)(1122;-1)(B211)(OMGA)
 (HALF,REI)()(OMGA)(1324;-1)(B212)(OMGA)
 (HALF,REI)()(OMGA)(1324;-1)(B221)(OMGA)
```

```
 (HALF,REI)()(OMGA)(3344;-1)(B222)(OMGA)

  PSIE  1  T                              ENERGY NORM FOR PSI
 (HALF)()(PSI)(1122;-2)(B211)(PSI)
 (HALF)()(PSI)(1324;-2)(B212)(PSI)
 (HALF)()(PSI)(1324;-2)(B221)(PSI)
 (HALF)()(PSI)(3344;-2)(B222)(PSI)

  U1E  1  T                              ENERGY NORM FOR U1
 (HALF)()(U1)(1122;-1)(B211)(U1)
 (HALF)()(U1)(1324;-1)(B212)(U1)
 (HALF)()(U1)(1324;-1)(B221)(U1)
 (HALF)()(U1)(3344;-1)(B222)(U1)

  U2E  1  T                              ENERGY NORM FOR U2
 (HALF)()(U2)(1122;-1)(B211)(U2)
 (HALF)()(U2)(1324;-1)(B212)(U2)
 (HALF)()(U2)(1324;-1)(B221)(U2)
 (HALF)()(U2)(3344;-1)(B222)(U2)

  PRSE  1  T                              ENERGY NORM FOR PRES
 (HALF)()(PRES)(1122;-1)(B211)(PRES)
 (HALF)()(PRES)(1324;-1)(B212)(PRES)
 (HALF)()(PRES)(1324;-1)(B221)(PRES)
 (HALF)()(PRES)(3344;-1)(B222)(PRES)

  TMPE  1  T                              ENERGY NORM FOR TEMP
 (HALF,PEI)()(TEMP)(1122;-1)(B211)(TEMP)
 (HALF,PEI)()(TEMP)(1324;-1)(B212)(TEMP)
 (HALF,PEI)()(TEMP)(1324;-1)(B221)(TEMP)
 (HALF,PEI)()(TEMP)(3344;-1)(B222)(TEMP)

  PHIE  1  T                              ENERGY NORM FOR PHI
 (HALF)()(PHI)(1122;-1)(B211)(PHI)
 (HALF)()(PHI)(1324;-1)(B212)(PHI)
 (HALF)()(PHI)(1324;-1)(B221)(PHI)
 (HALF)()(PHI)(3344;-1)(B222)(PHI)

GROUP FREQUENCY
       1
SOLUTION TYPE
  DELTA_Q
  FACTORED_GAUSS_ELIMINATION
  IMPLICIT_EULER

END
```

APPENDIX B: LIQUID HYDROGEN DATA

Table 1: Density Data for Liquid Hydrogen (Source [8])

Temperature (K) (˚K)	Density (kg/m^3)		Temperature (˚K)	Density (kg/m^3)		Temperature (˚K)	Density (kg/m^3)
34.00	32.84		34.68	22.59		35.36	19.14
34.02	32.33		34.70	22.45		35.38	19.07
34.04	31.82		34.72	22.30		35.40	19.00
34.06	31.33		34.74	22.16		35.42	18.93
34.08	30.85		34.76	22.03		35.44	18.87
34.10	30.38		34.78	21.90		35.46	18.80
34.12	29.93		34.80	21.77		35.48	18.73
34.14	29.49		34.82	21.65		35.50	18.67
34.16	29.08		34.84	21.53		35.52	18.60
34.18	28.67		34.86	21.41		35.54	18.54
34.20	28.29		34.88	21.29		35.56	18.48
34.22	27.92		34.90	21.18		35.58	18.42
34.24	27.57		34.92	21.07		35.60	18.36
34.26	27.23		34.94	20.96		35.62	18.30
34.28	26.91		34.96	20.86		35.64	18.24
34.30	26.60		34.98	20.75		35.66	18.18
34.32	26.31		35.00	20.65		35.68	18.12
34.34	26.03		35.02	20.56		35.70	18.07
34.36	25.76		35.04	20.46		35.72	18.01
34.38	25.50		35.06	20.36		35.74	17.96
34.40	25.25		35.08	20.27		35.76	17.90
34.42	25.01		35.10	20.18		35.78	17.85
34.44	24.78		35.12	20.09		35.80	17.80
34.46	24.56		35.14	20.01		35.82	17.74
34.48	24.35		35.16	19.92		35.84	17.69
34.50	24.14		35.18	19.84		35.86	17.64
34.52	23.95		35.20	19.75		35.88	17.59
34.54	23.76		35.22	19.67		35.90	17.54
34.56	23.57		35.24	19.59		35.92	17.49
34.58	23.40		35.26	19.51		35.94	17.44
34.60	23.23		35.28	19.44		35.96	17.39
34.62	23.06		35.30	19.36		35.98	17.35
34.64	22.90		35.32	19.29		36.00	17.30
34.66	22.74		35.34	19.21			

APPENDIX C: FIGURES

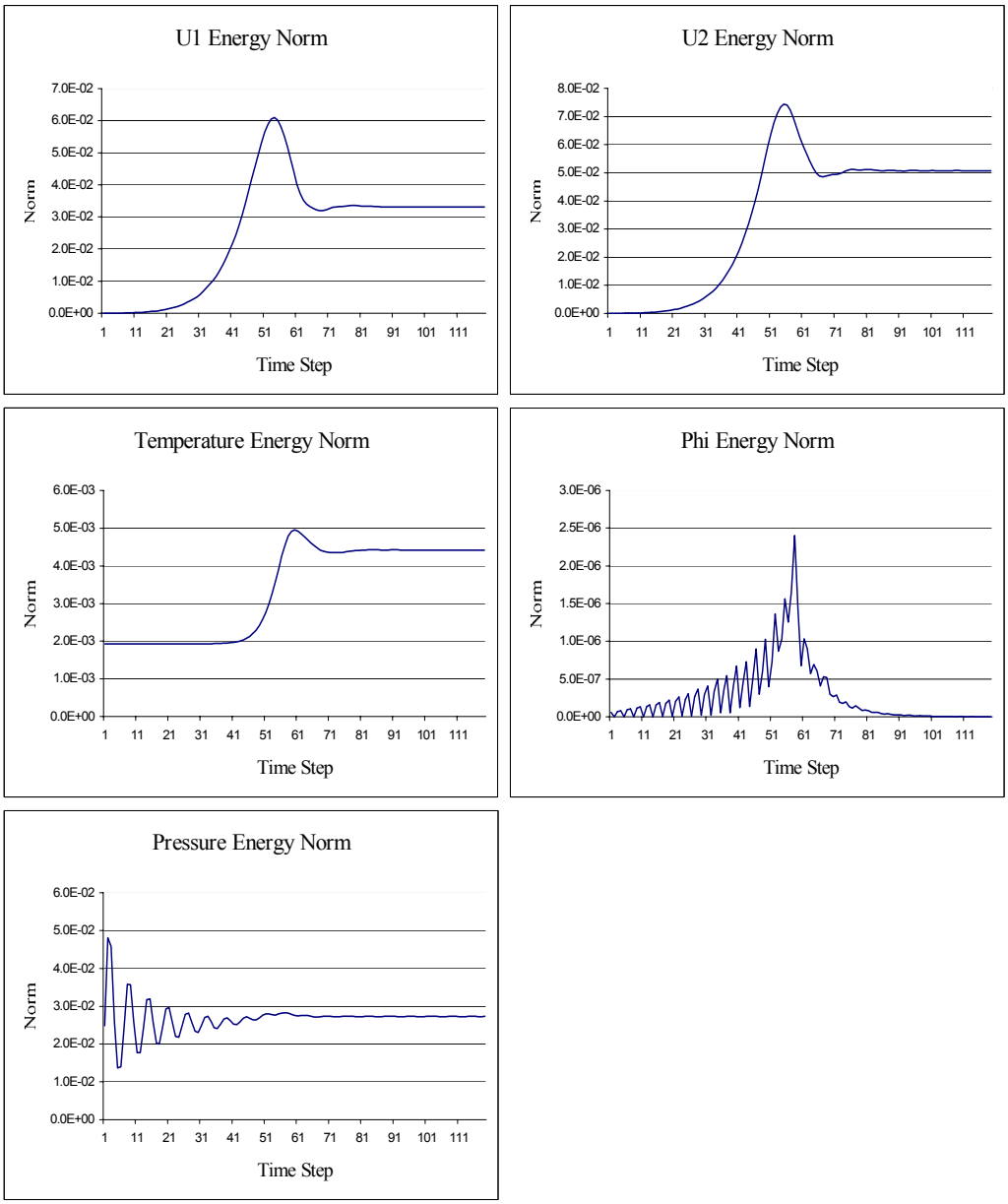

Figure 1: Energy Semi-Norms Over Time for Ra = 10^4

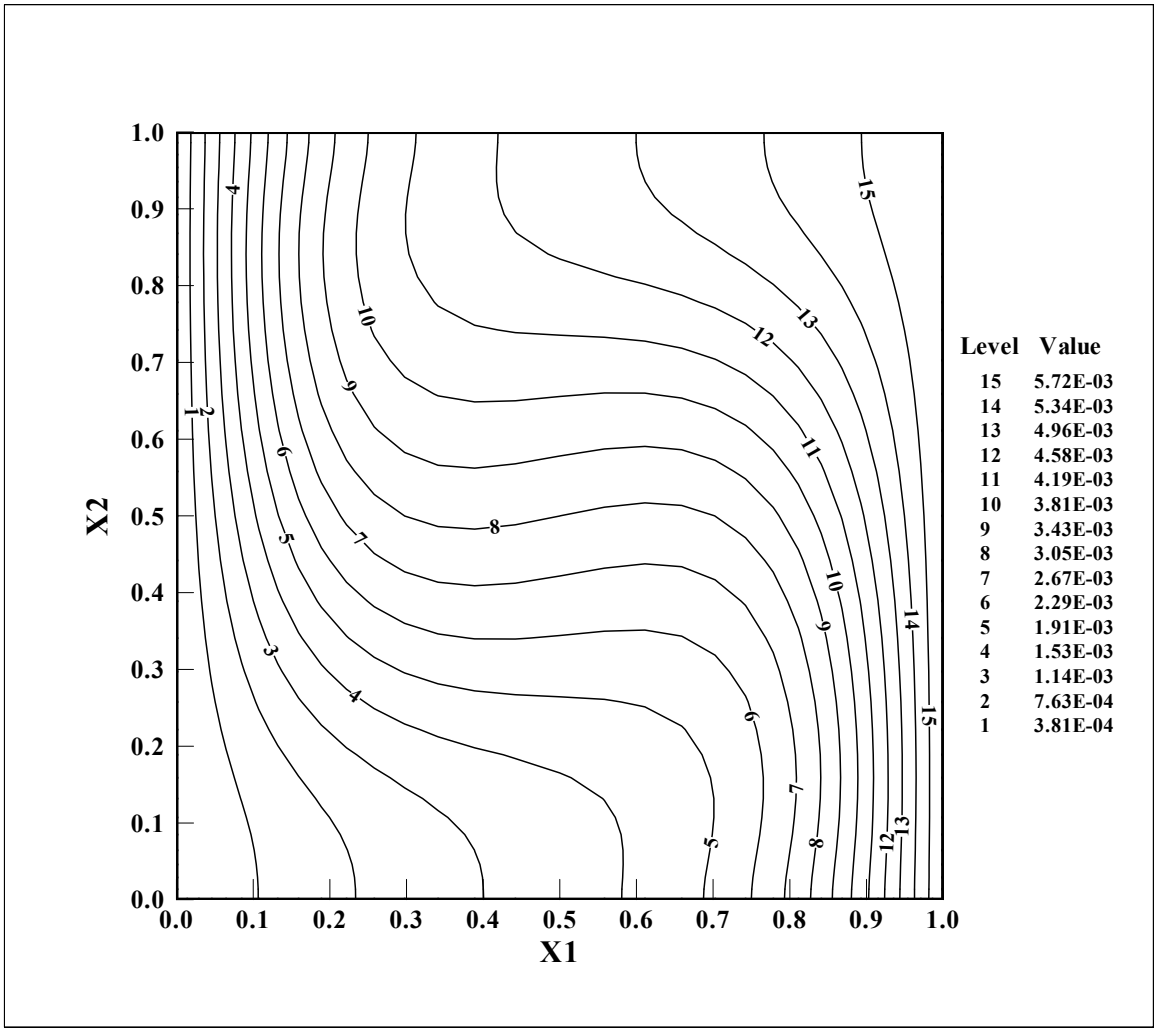

Figure 2: Temperature Contours for Ra = 10^4

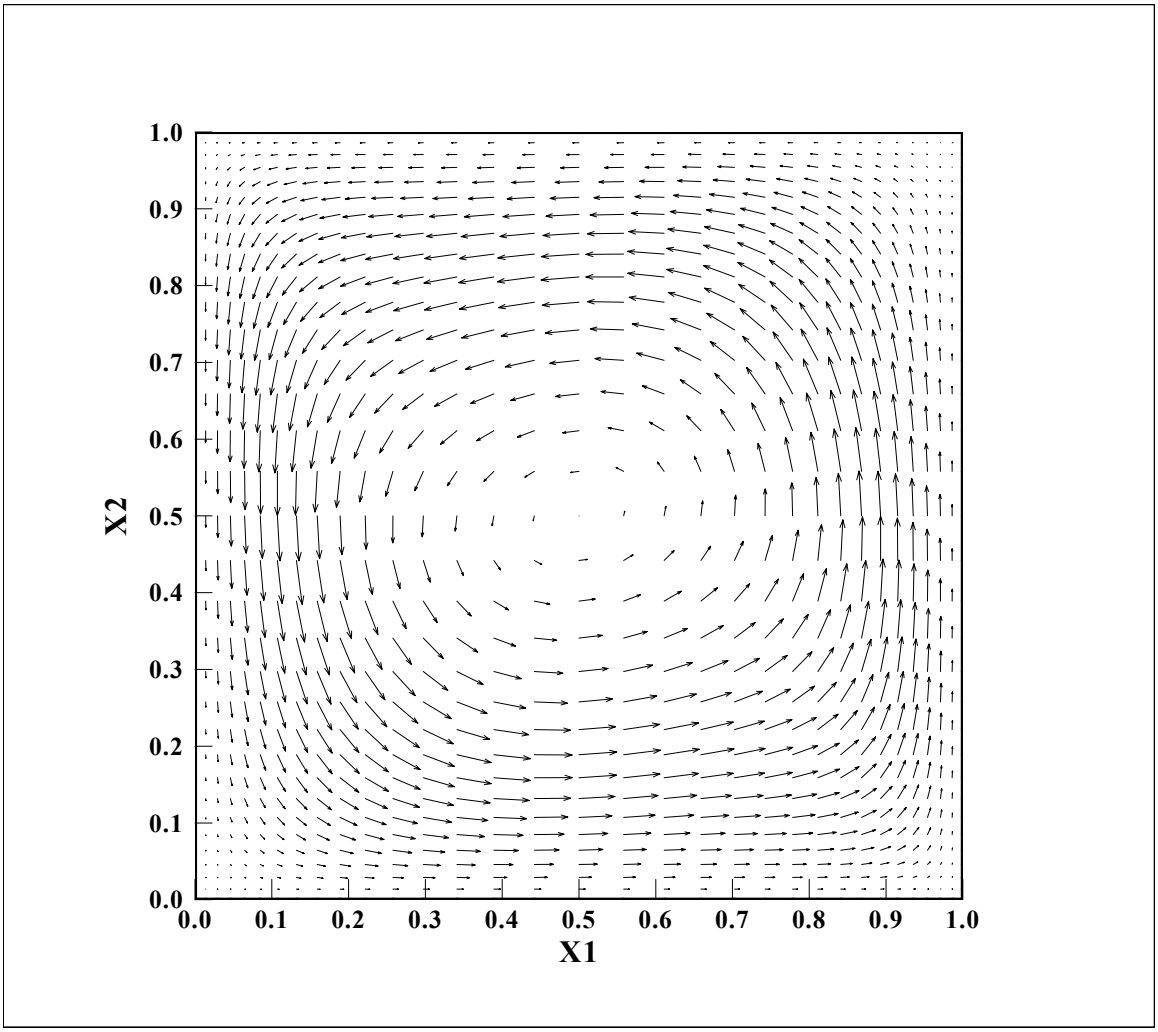

Figure 3: Velocity Vectors for Ra = 10^4

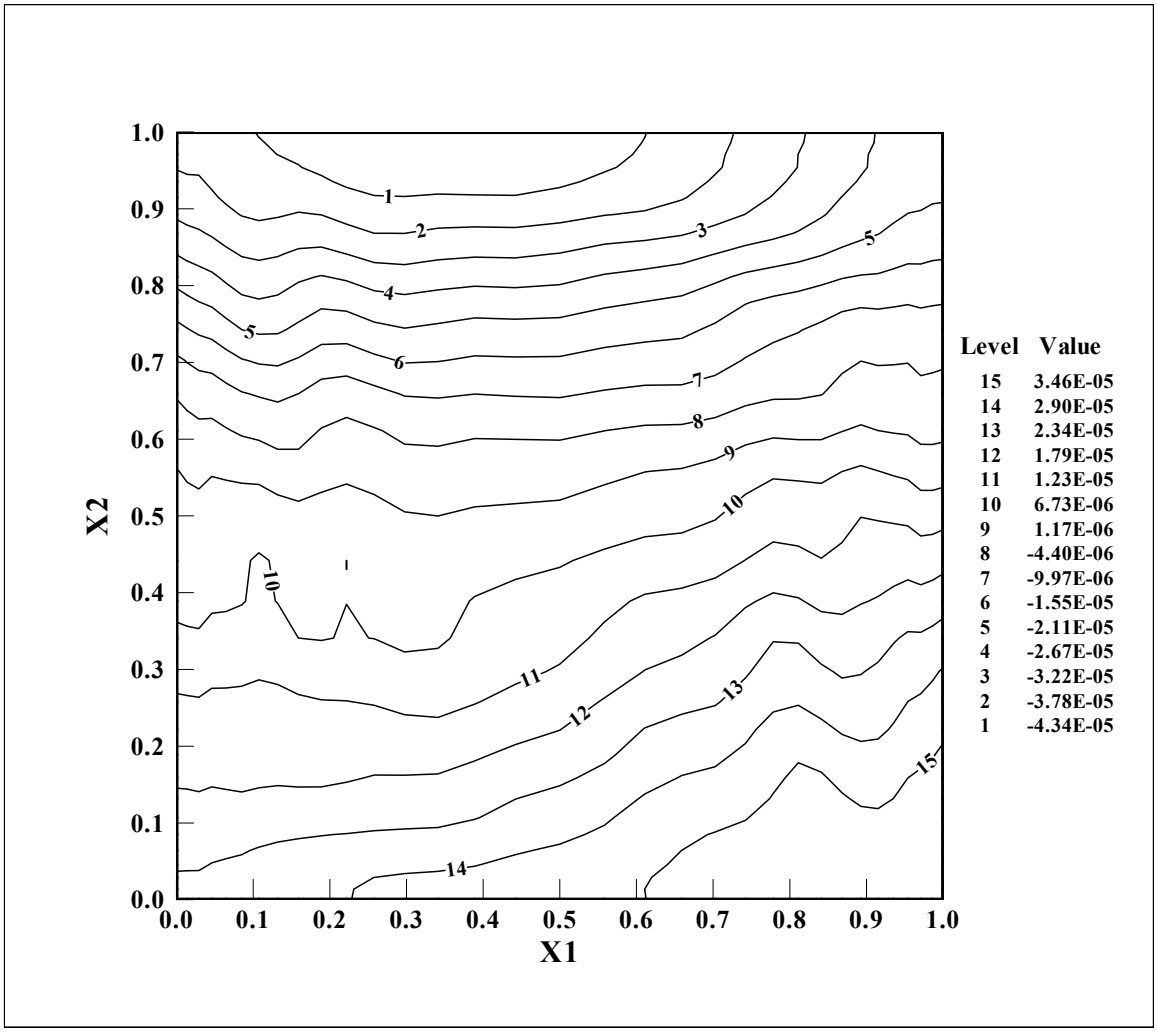

Figure 4: CCA ϕ Variable Contours for Ra = 10^4

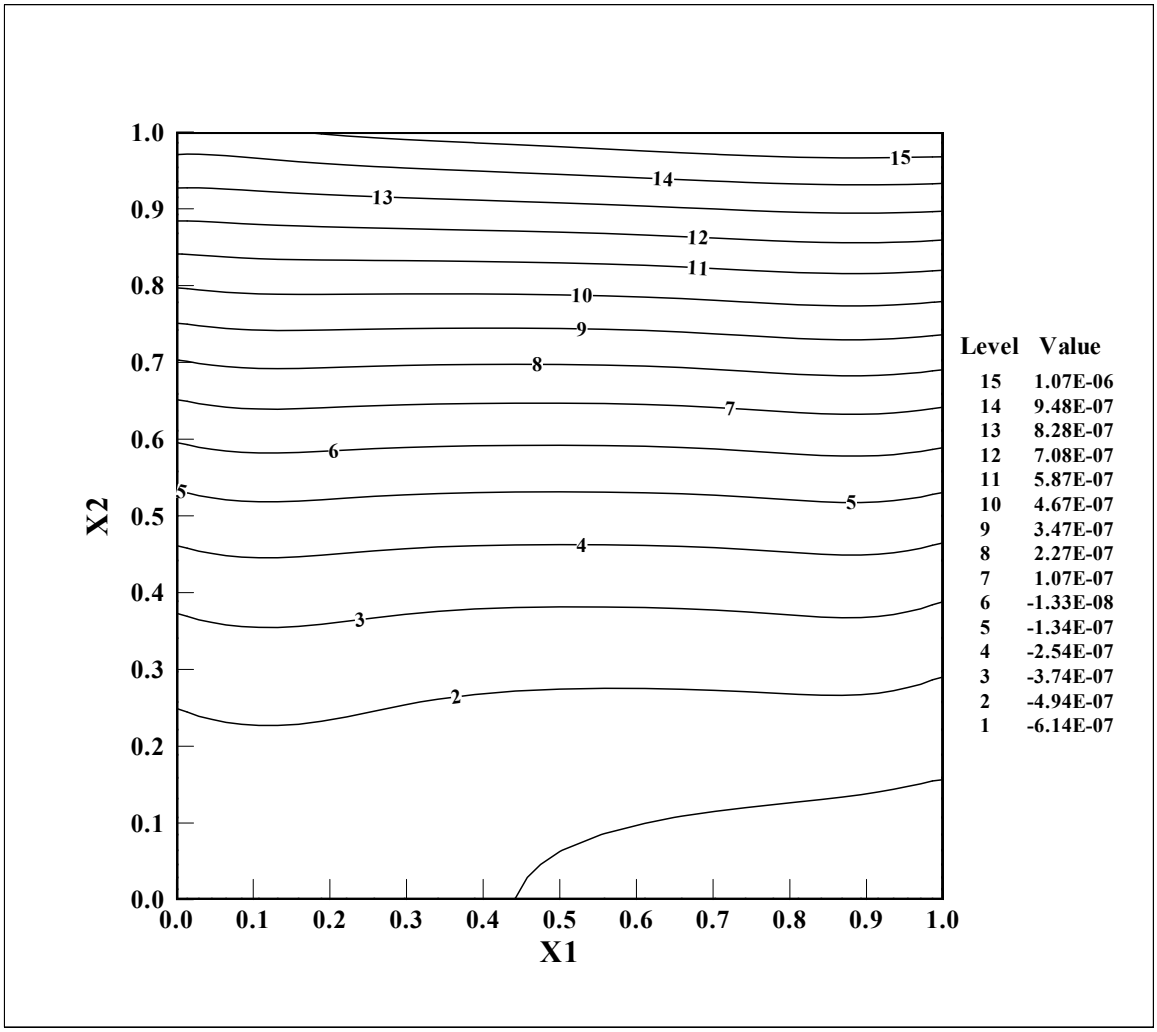

Figure 5: Pressure Contours for Ra = 10^4

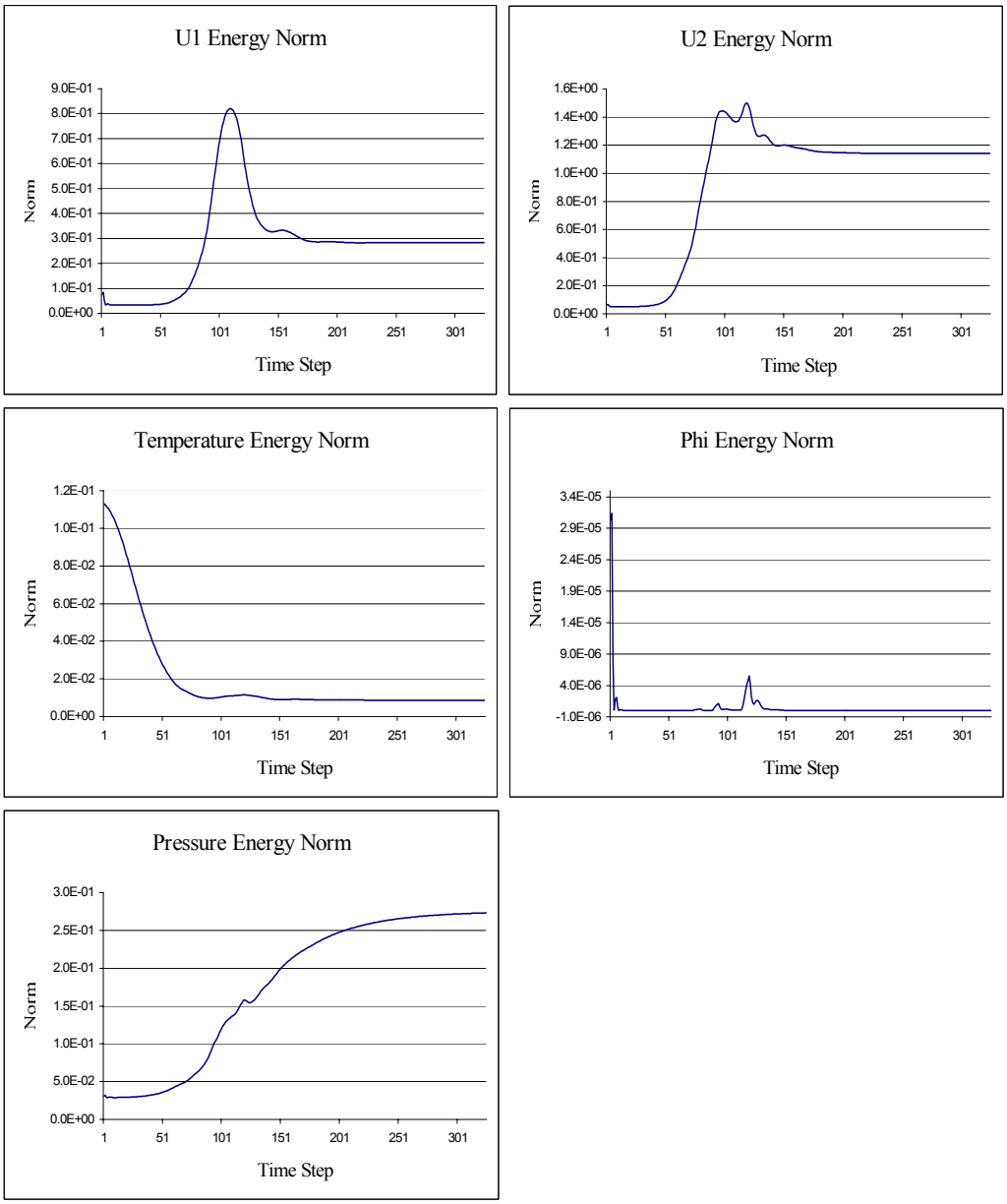

Figure 6: Energy Semi-Norms Over Time for Ra = 10^5

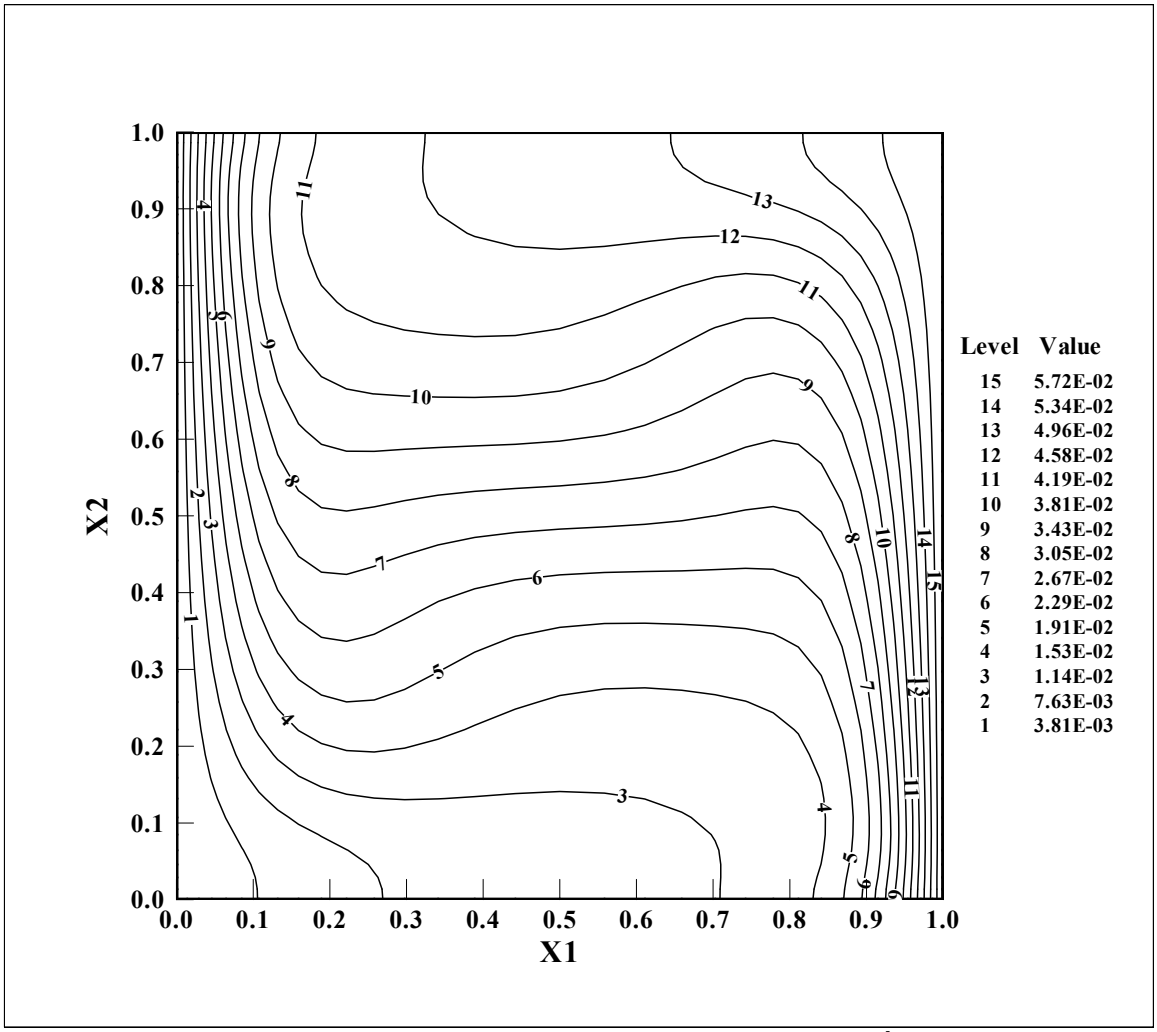

Figure 7: Temperature Contours for Ra = 10^5

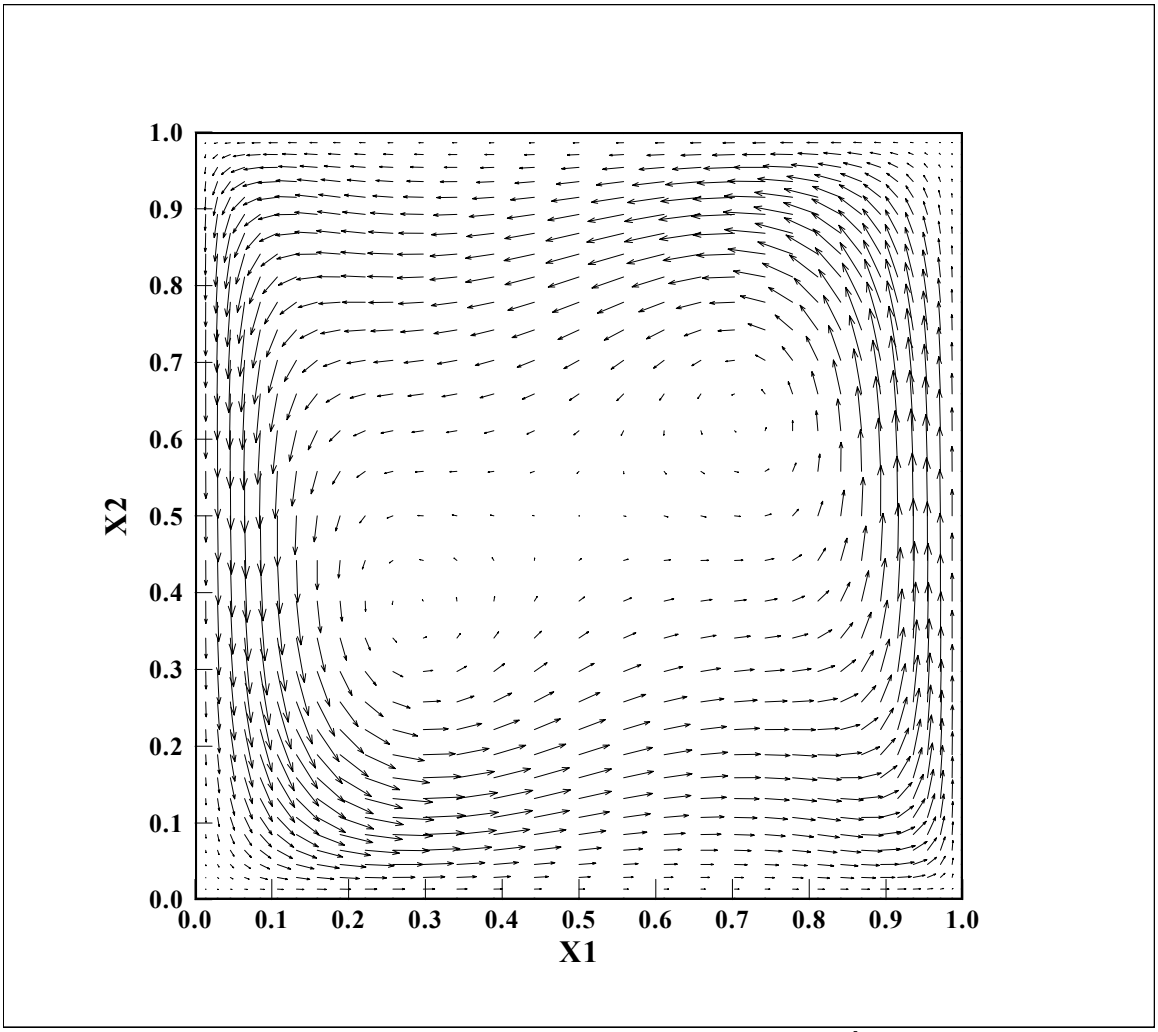

Figure 8: Velocity Vectors for Ra = 10^5

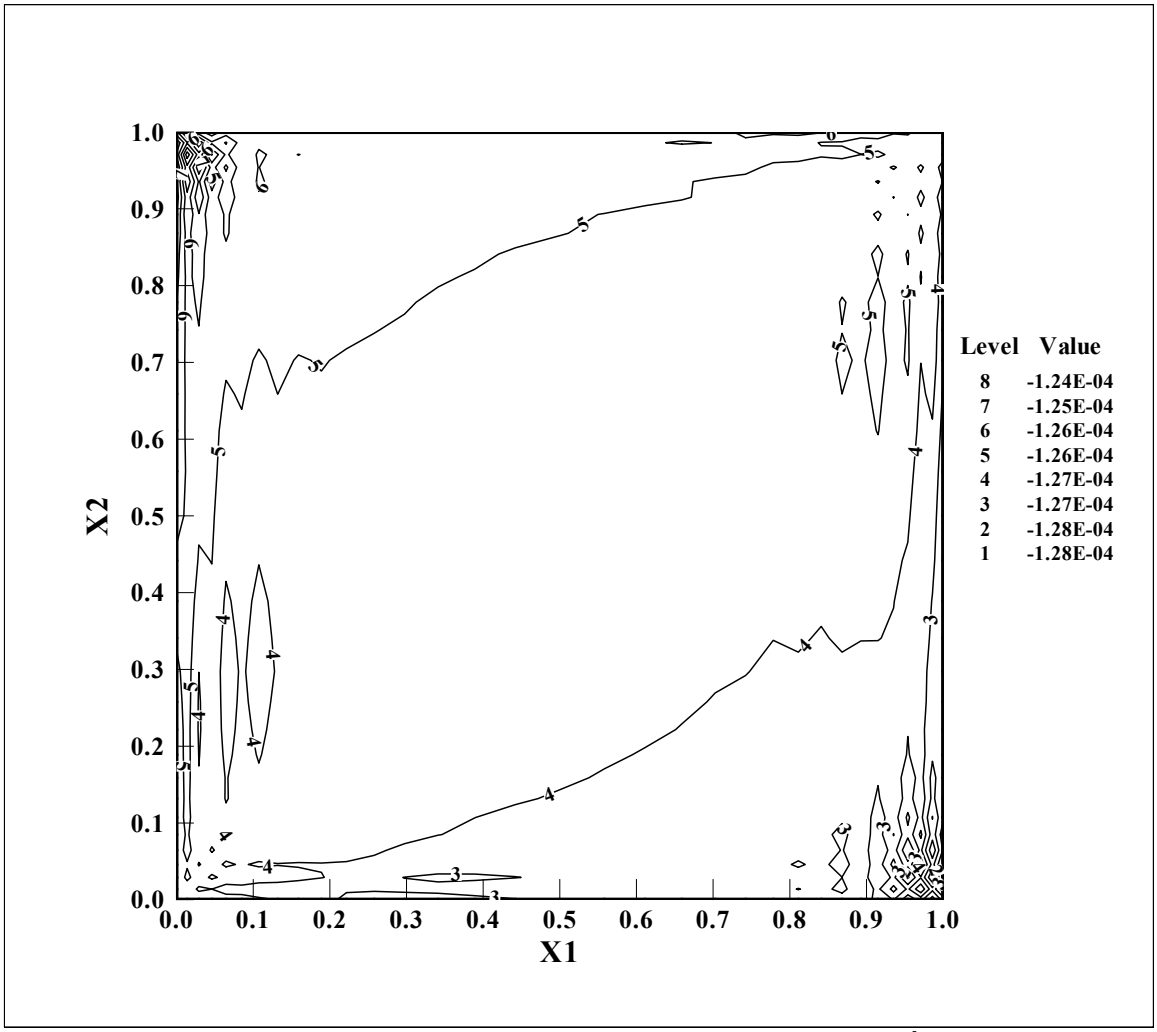

Figure 9: CCA ϕ Variable Contours for Ra = 10^5

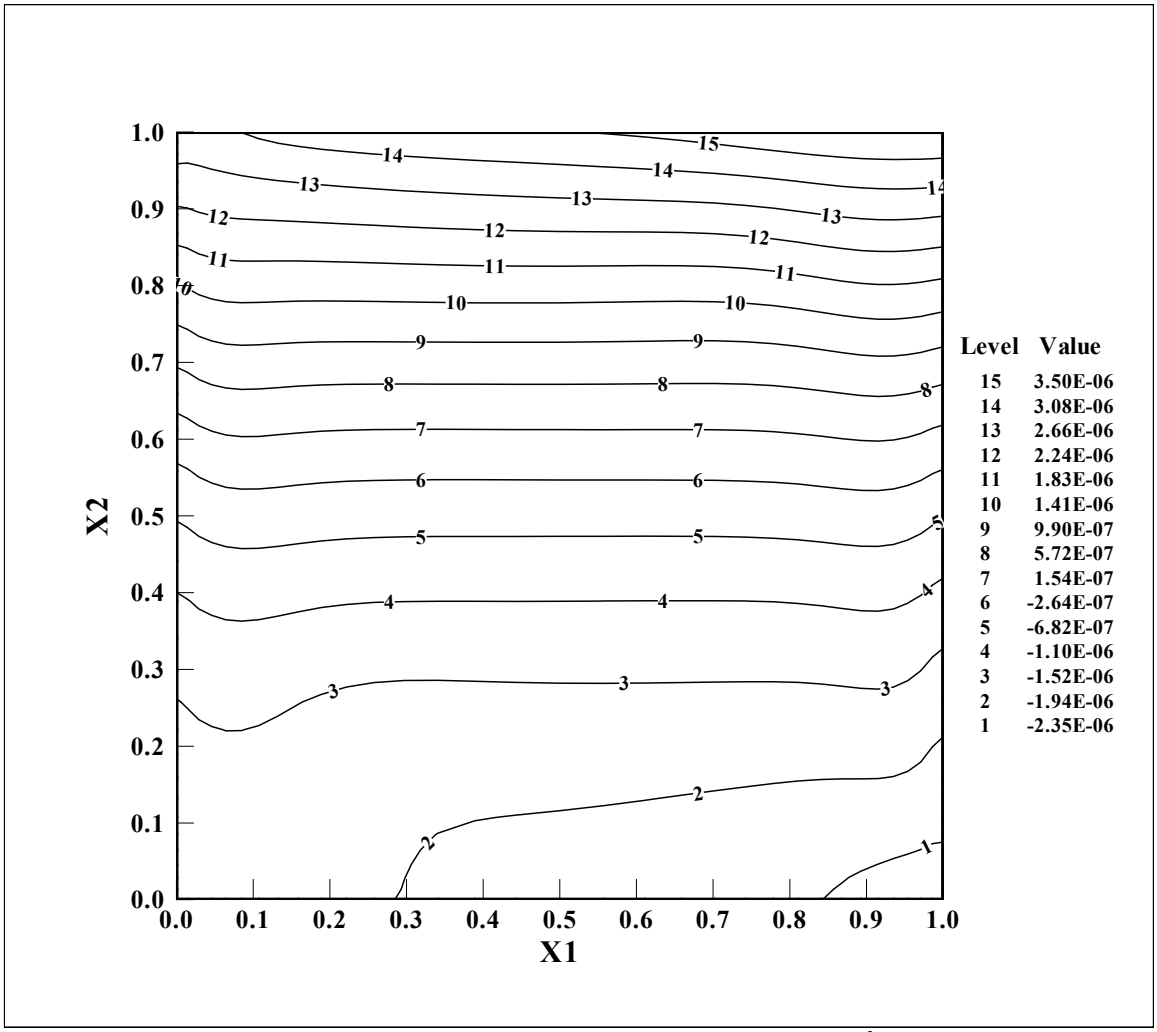

Figure 10: Pressure Contours for Ra = 10^5

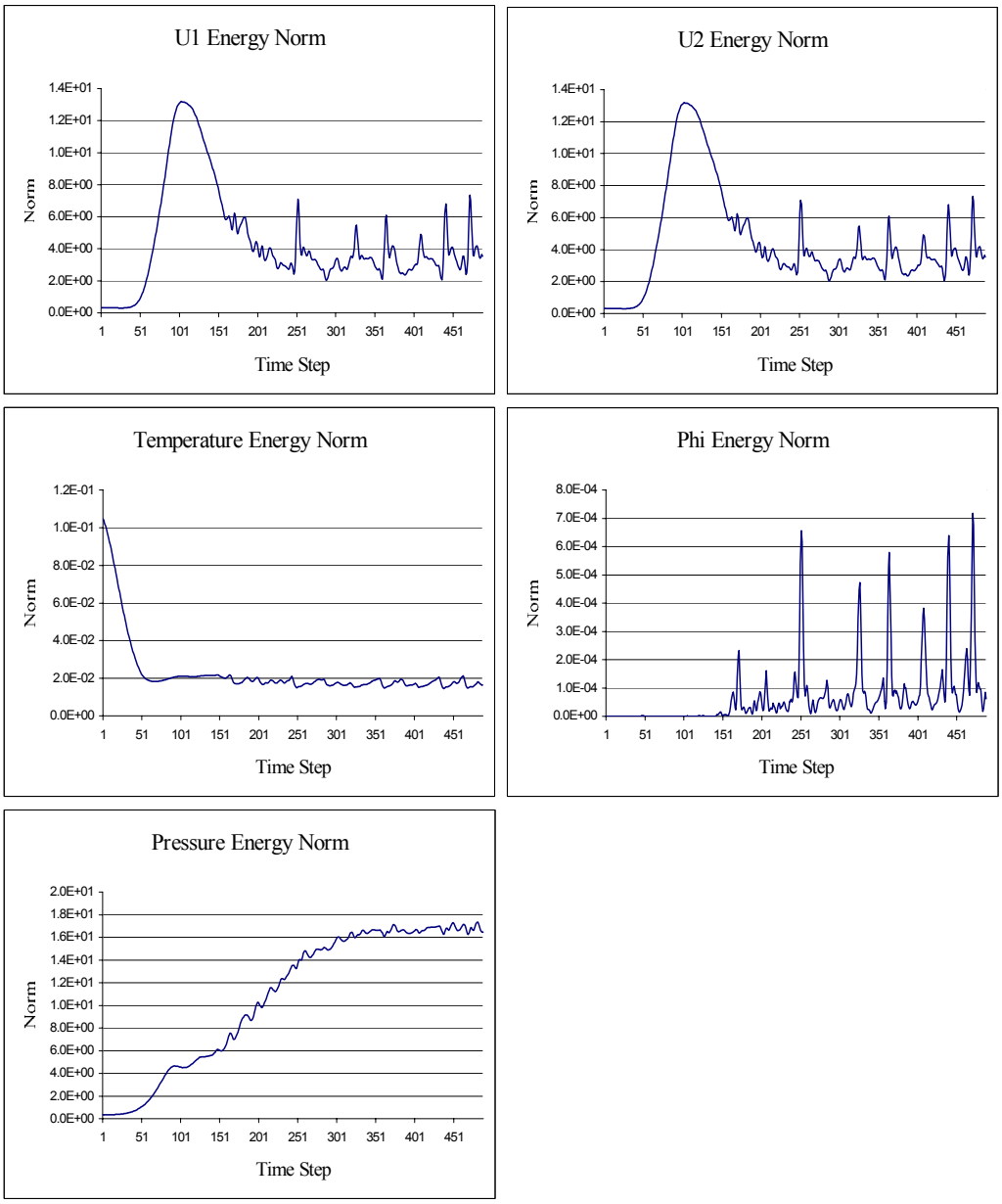

Figure 11: Energy Semi-Norms Over Time for Ra = 10^6

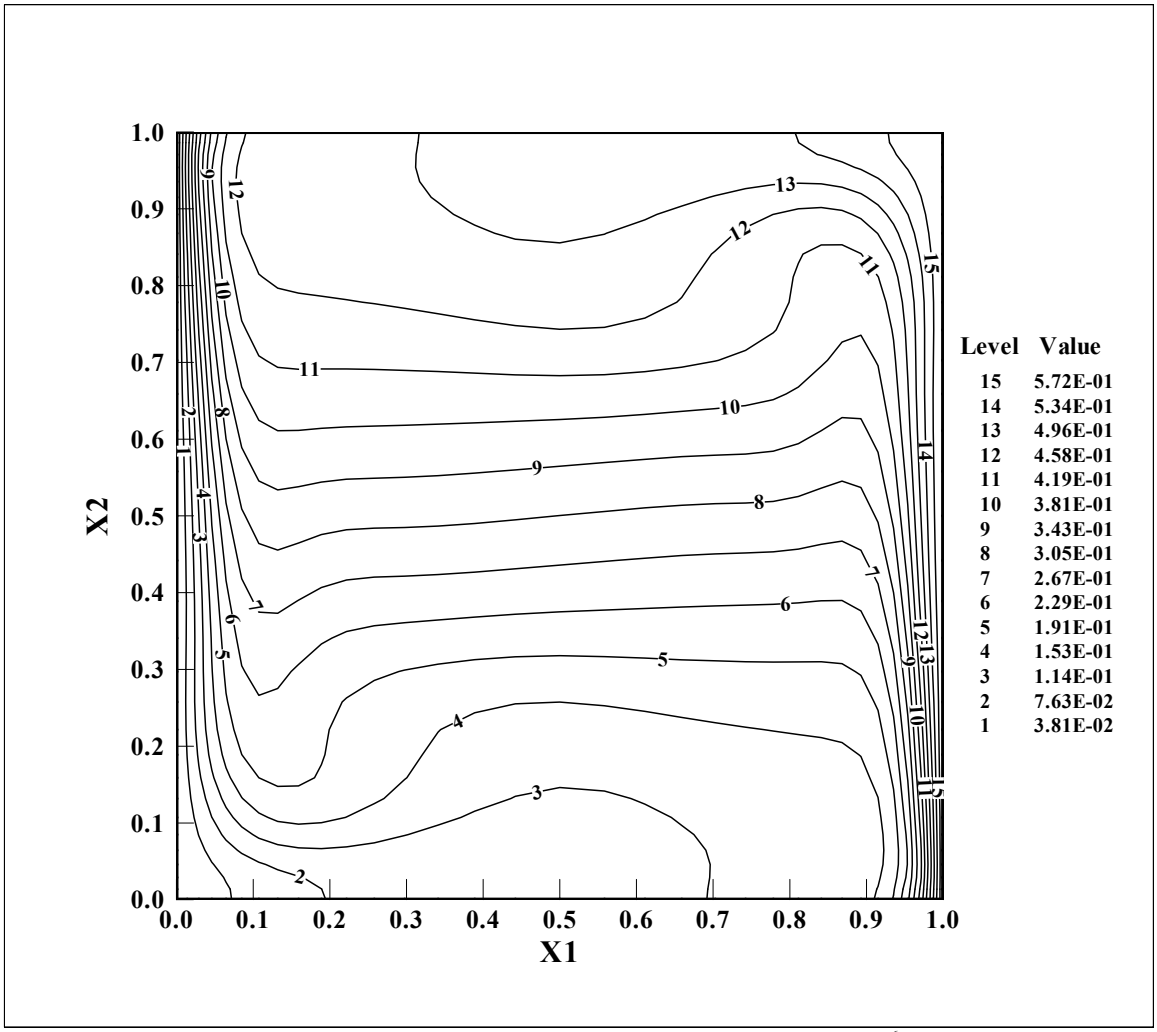

Figure 12: Temperature Contours for Ra = 10^6

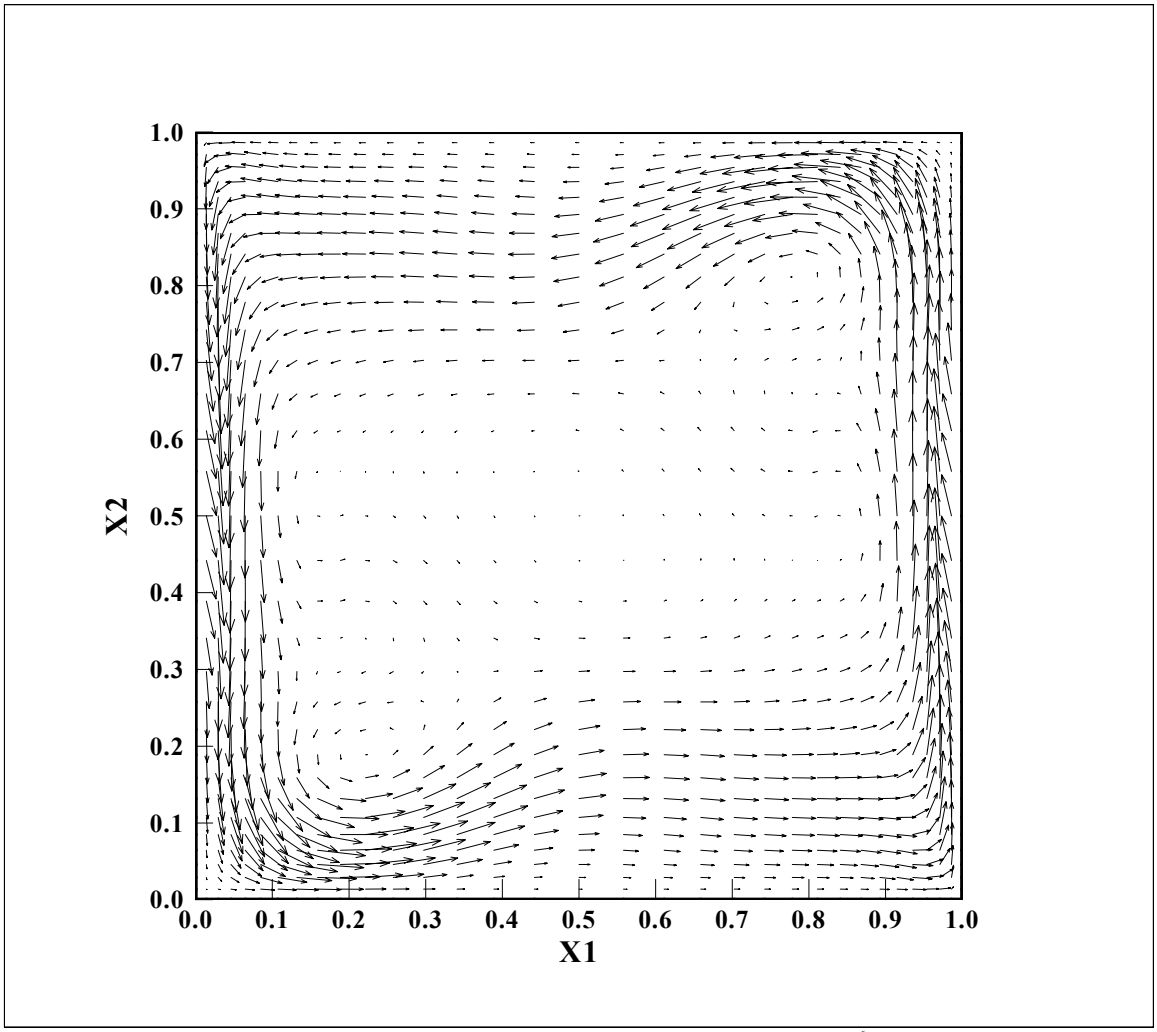

Figure 13: Velocity Vectors for Ra = 10^6

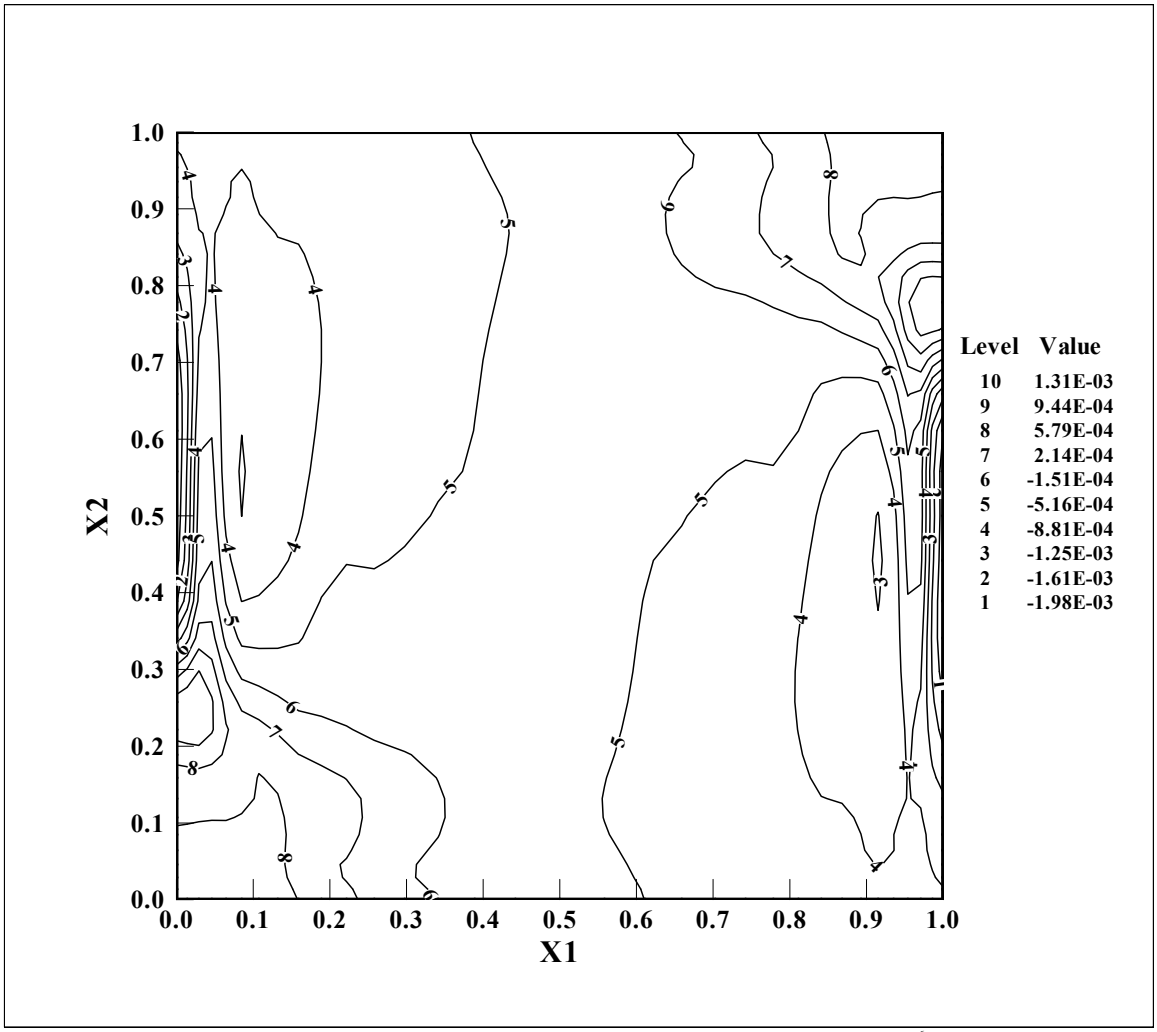

Figure 14: CCA ϕ Variable Contours for Ra = 10^6

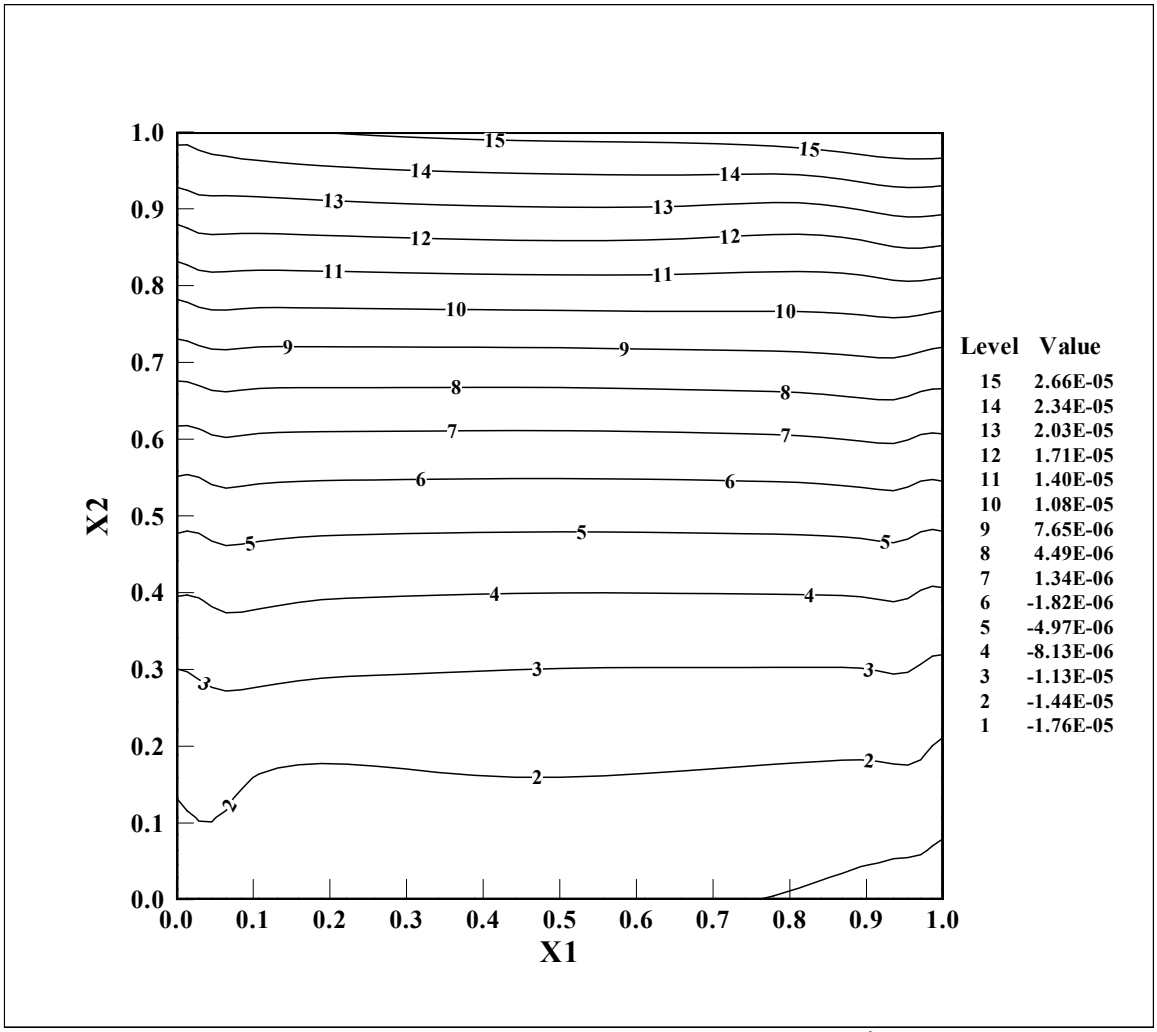

Figure 15: Pressure Contours for Ra = 10^6

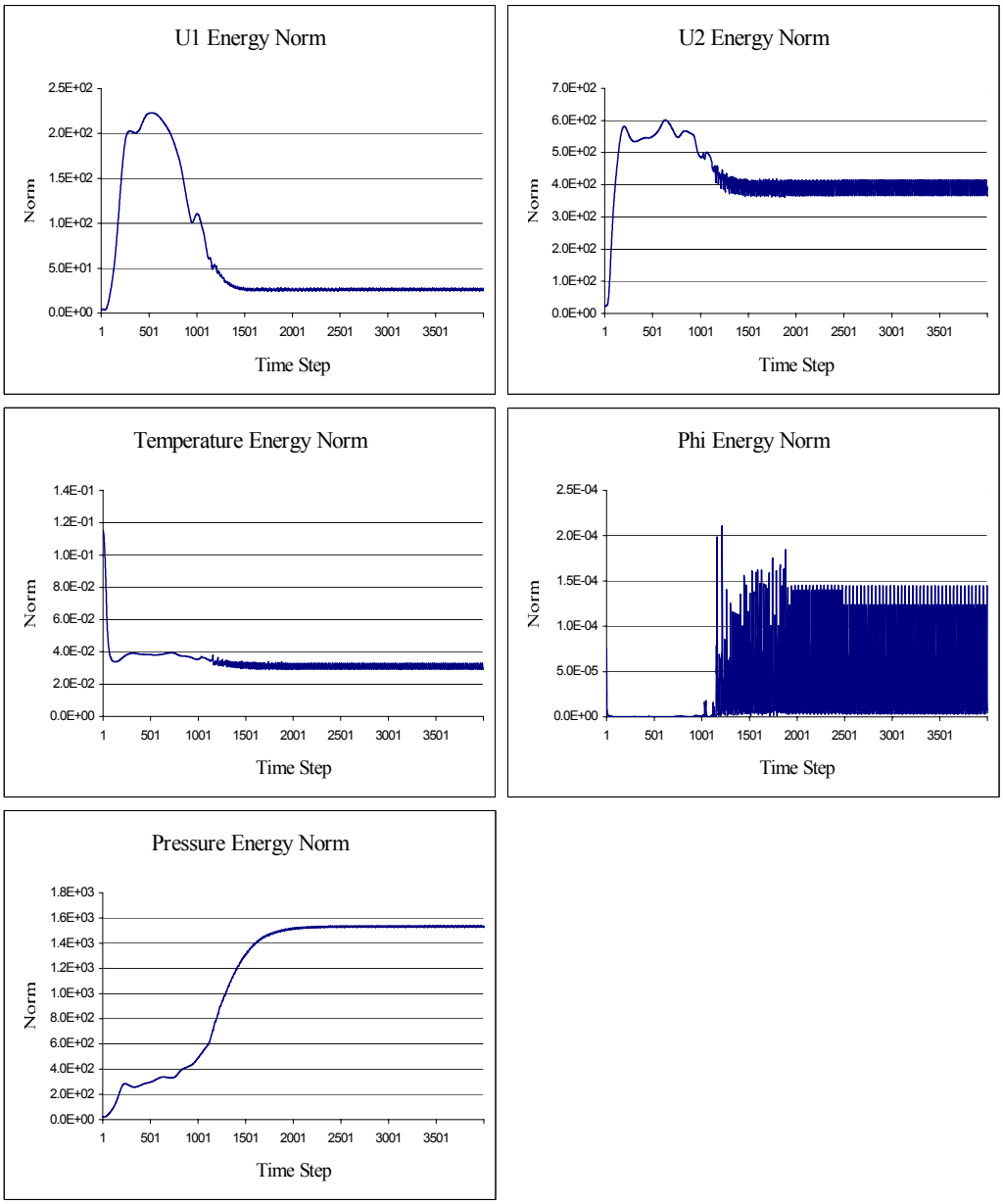

Figure 16: Energy Semi-Norms Over Time for Ra = 10^7

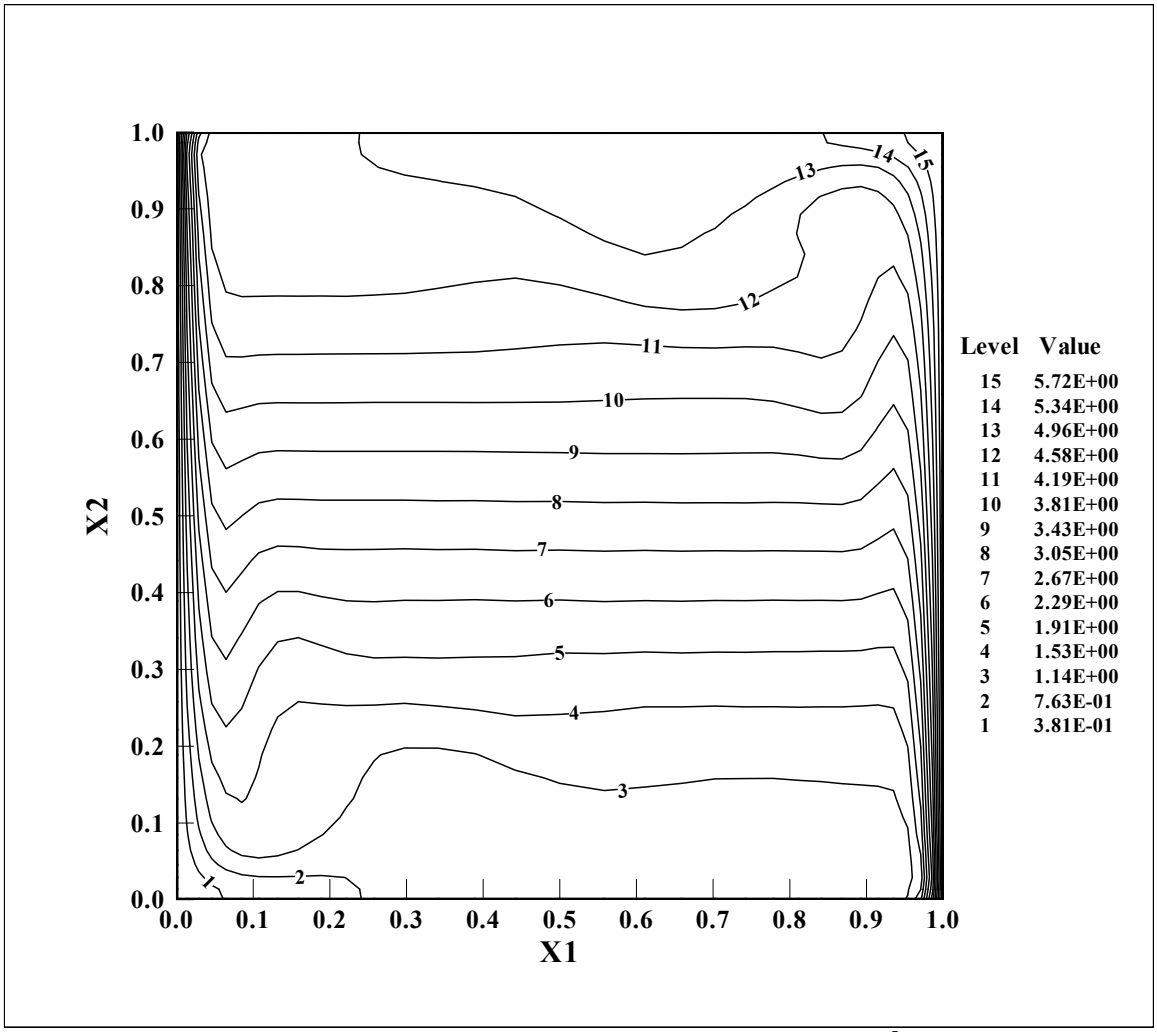

Figure 17: Temperature Contours for Ra = 10^7

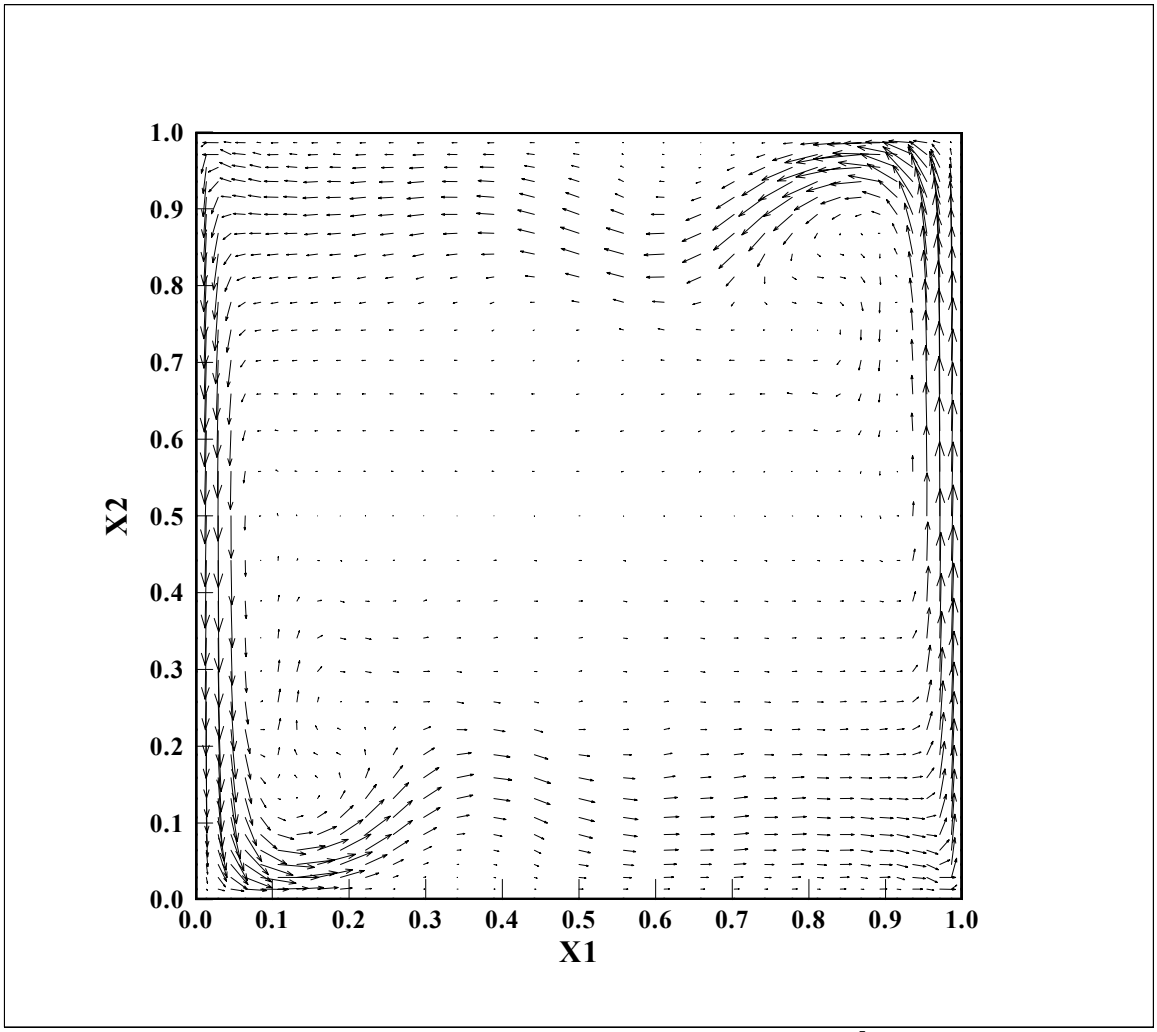

Figure 18: Velocity Vectors for Ra = 10^7

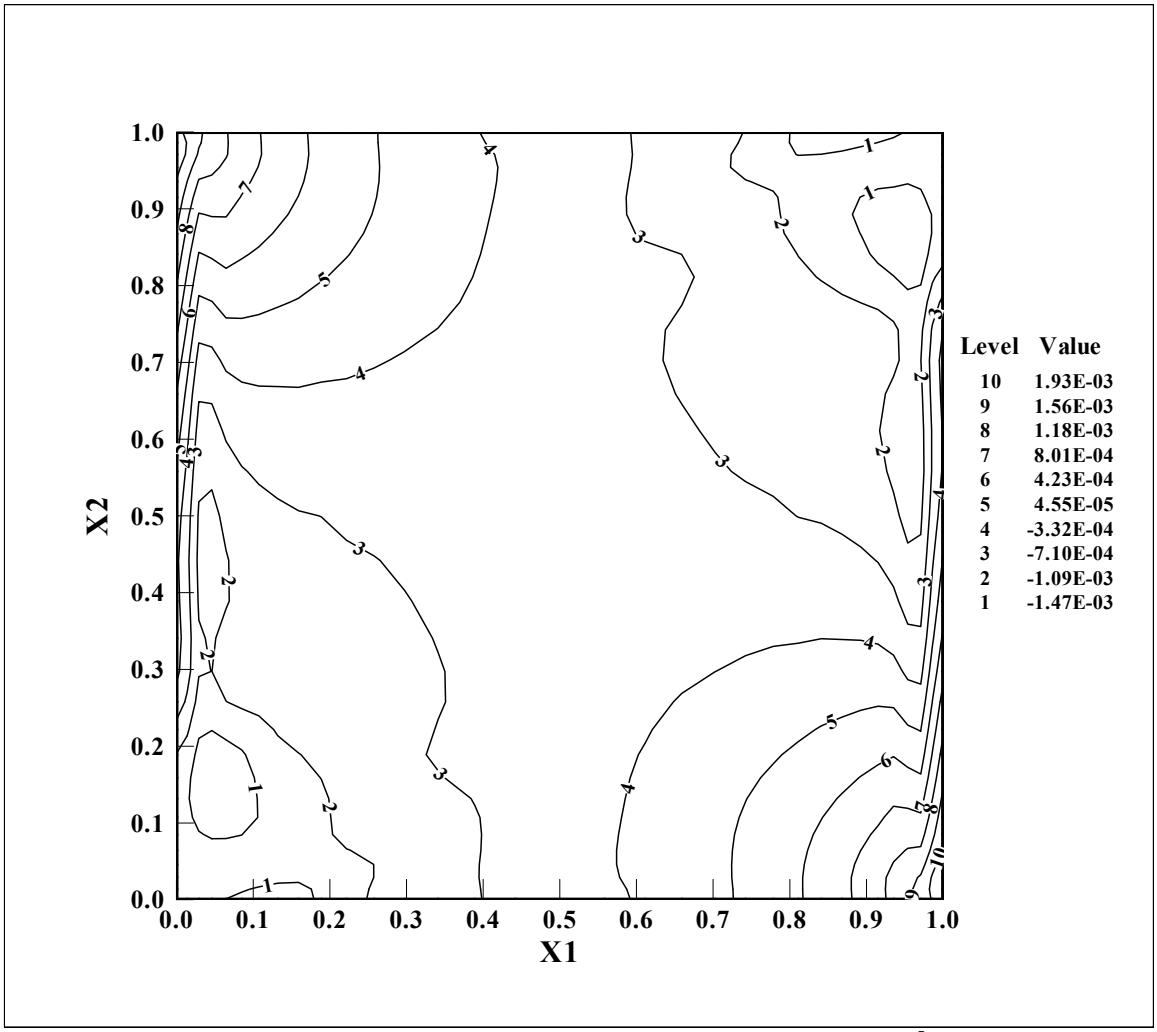

Figure 19: CCA ϕ Variable Contours for Ra = 10^7

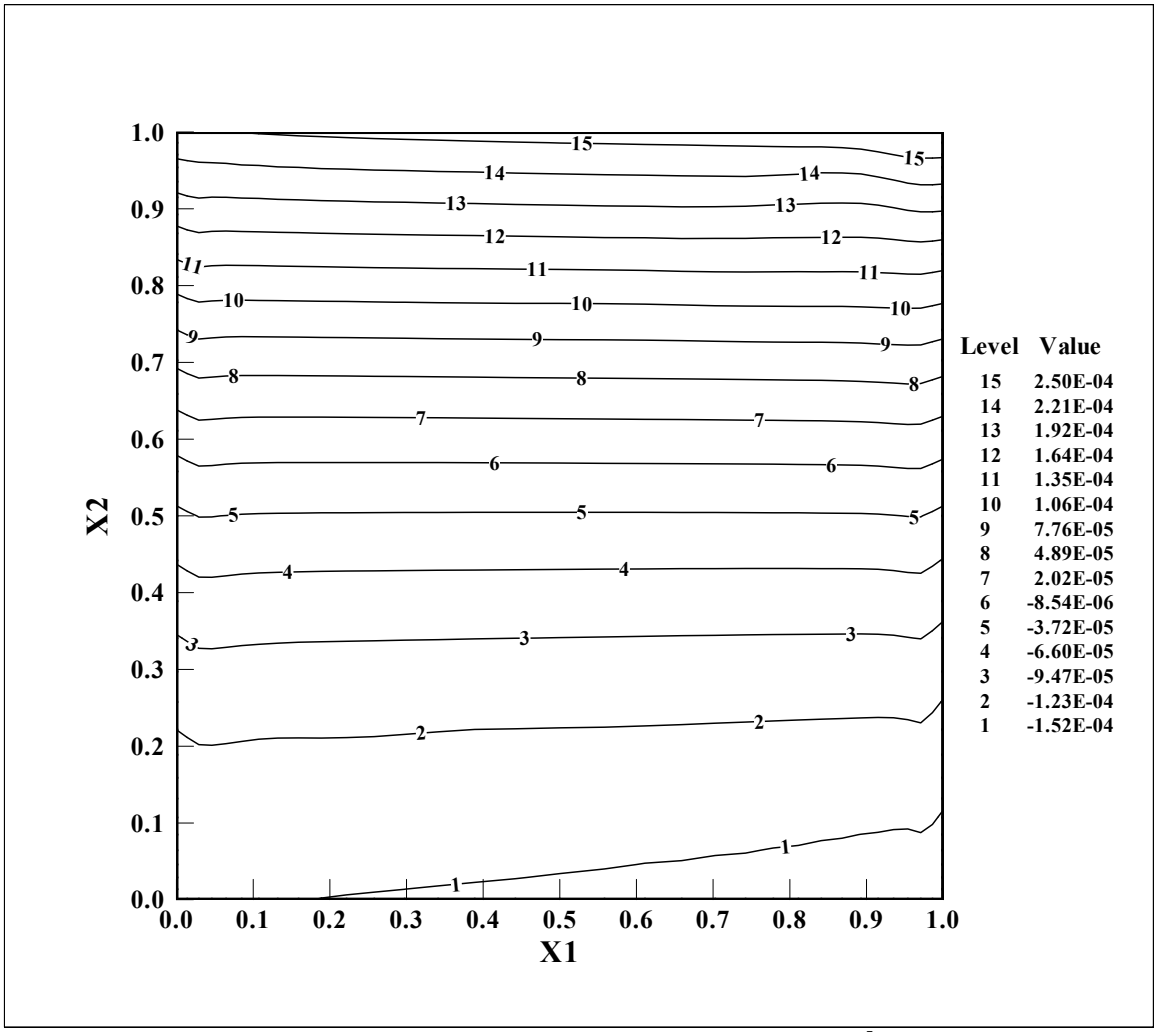

Figure 20: Pressure Contours for Ra = 10^7

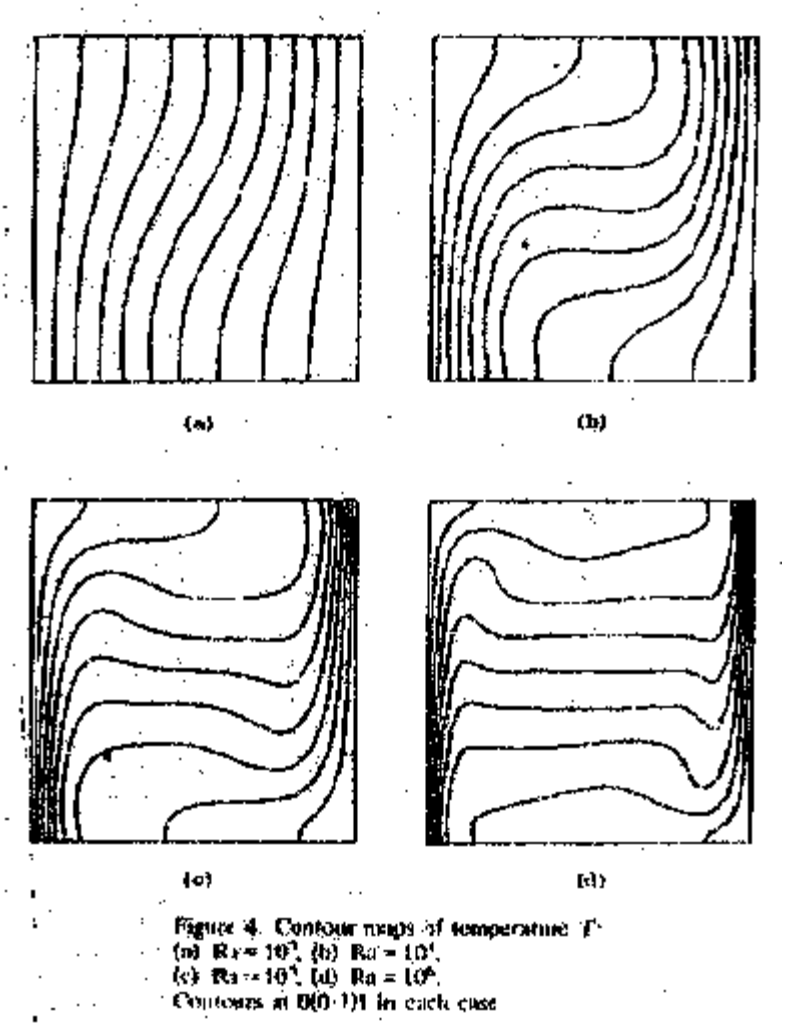

Figure 21: Temperature Profiles for Air at Ra = 10^3, 10^4, 10^5, 10^6 (Source [10])

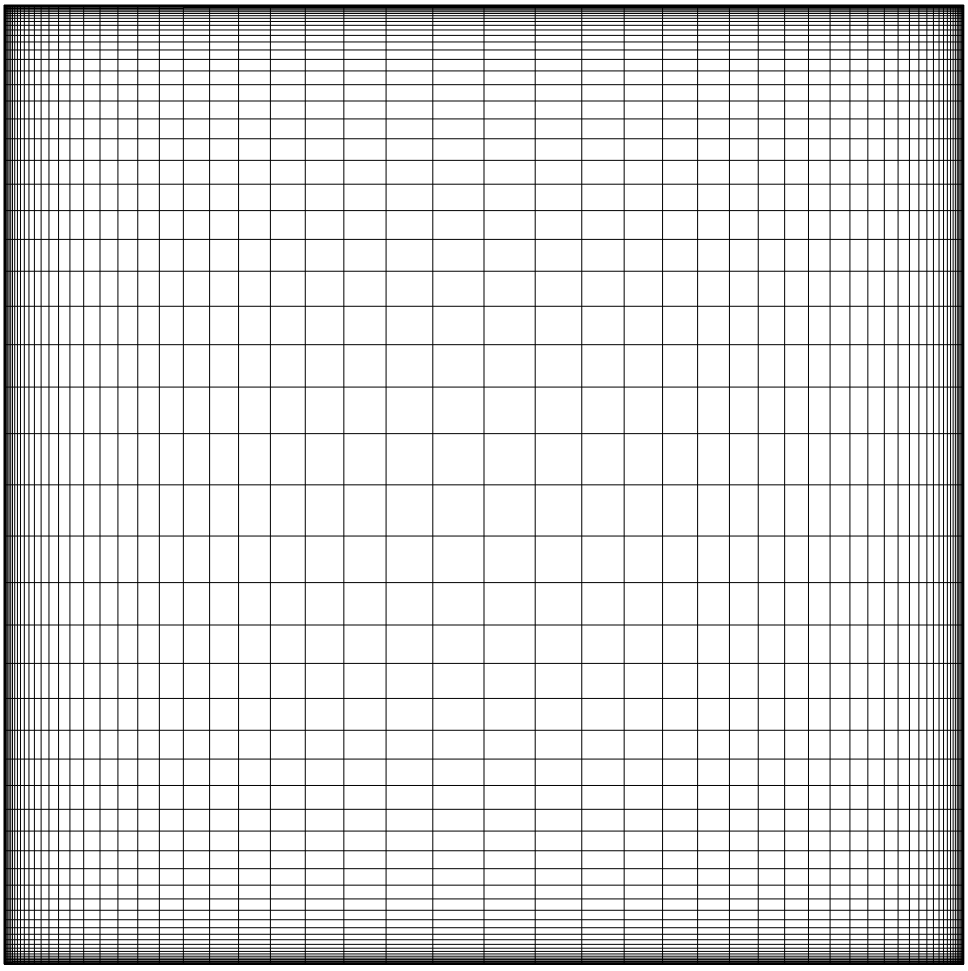

Figure 22: Liquid Hydrogen Analysis 64x64 Node Mesh

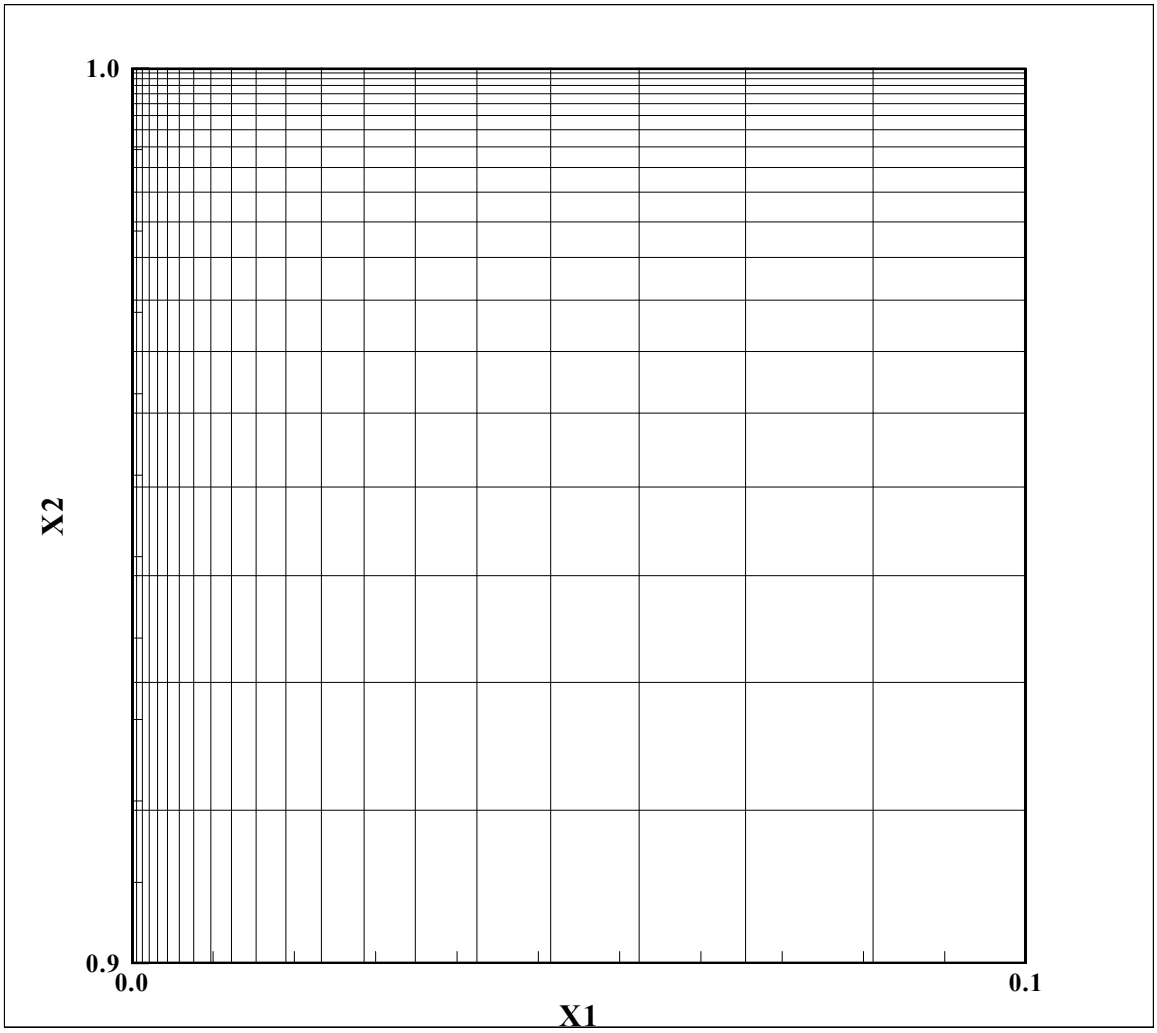

Figure 23: Liquid Hydrogen Analysis 64x64 Node Mesh (Magnified)

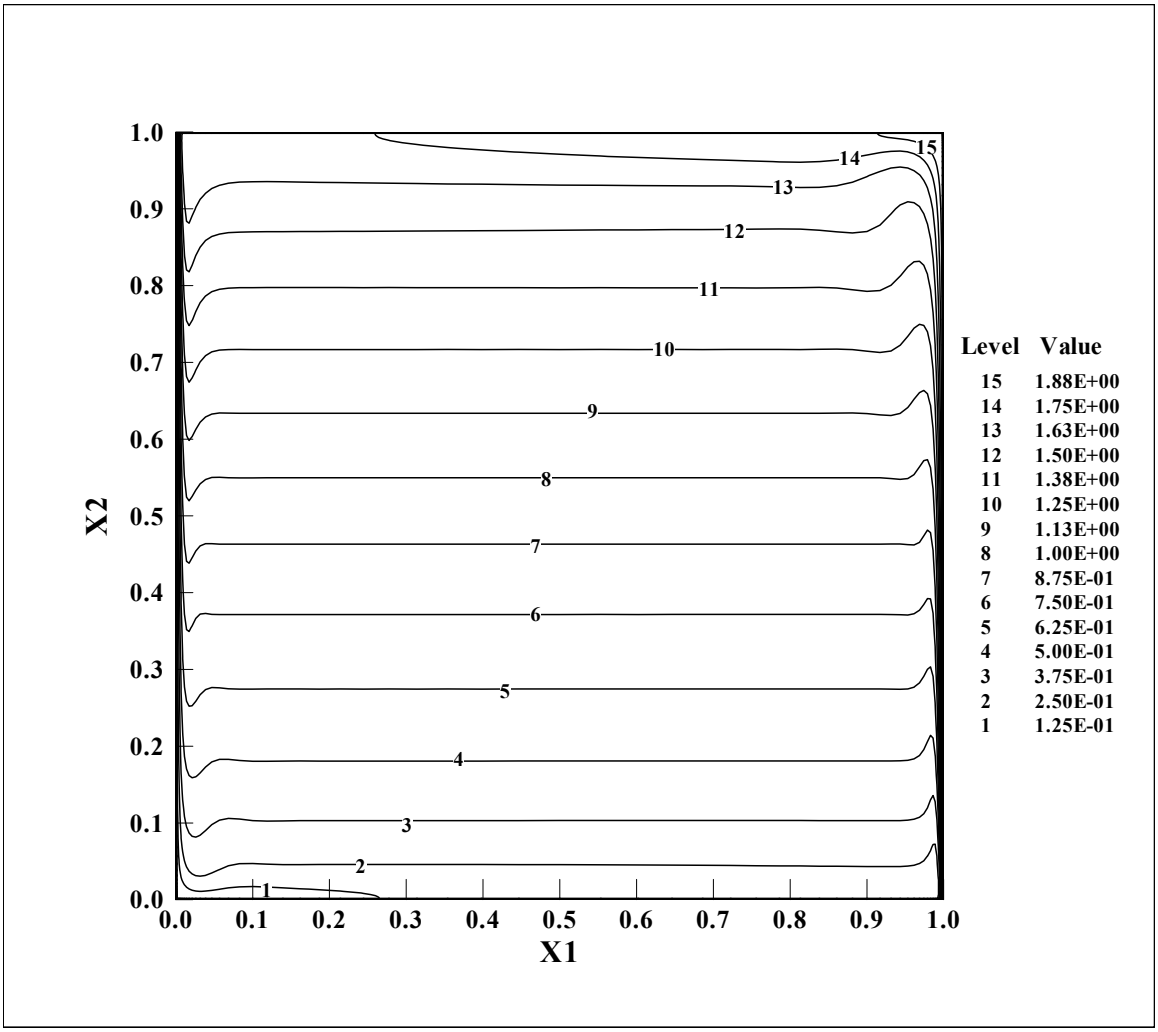

Figure 24: Liquid Hydrogen Analysis Temperature Contours

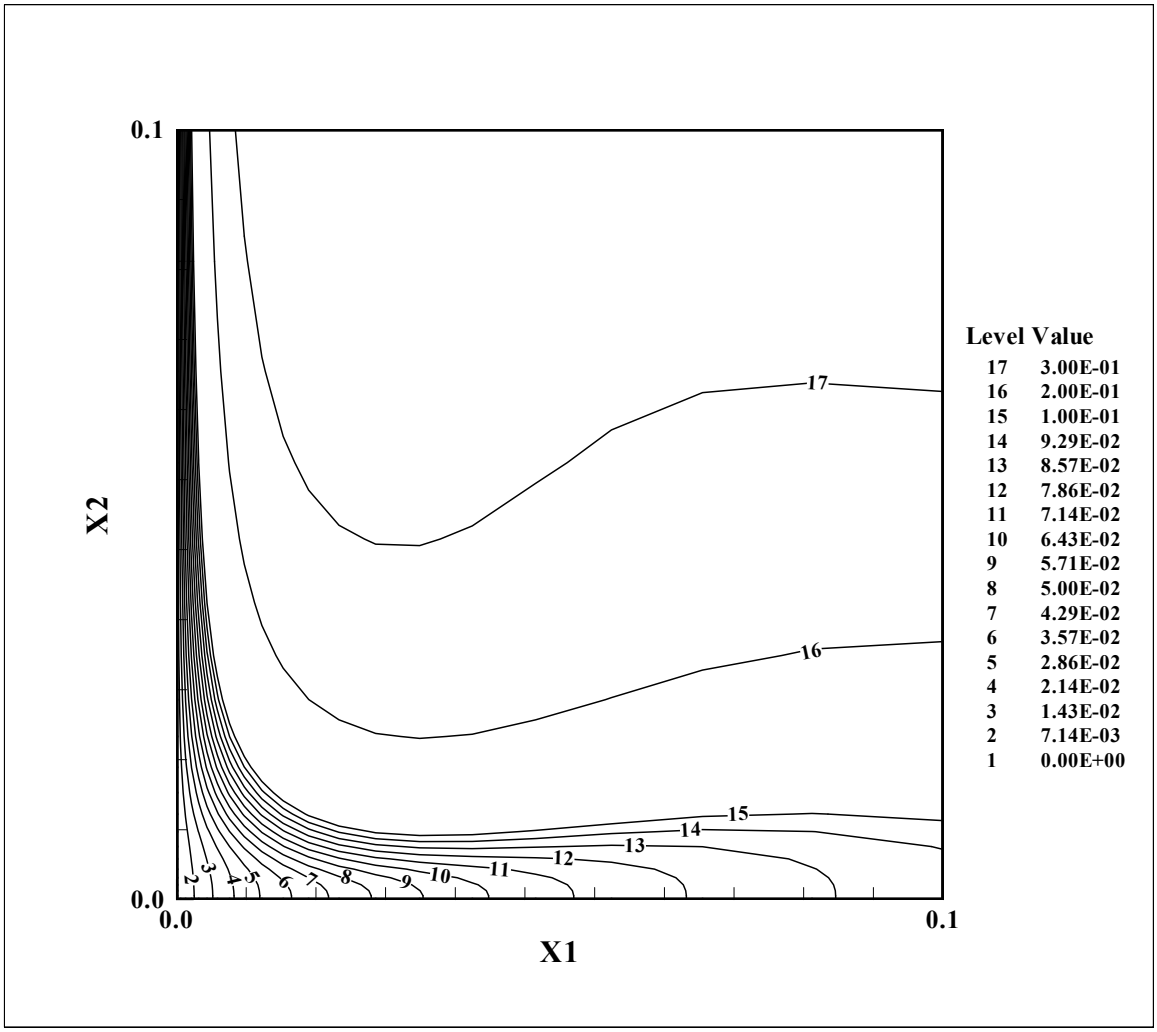

Figure 25: Liquid Hydrogen Analysis Temperature Contours (Magnified 1)

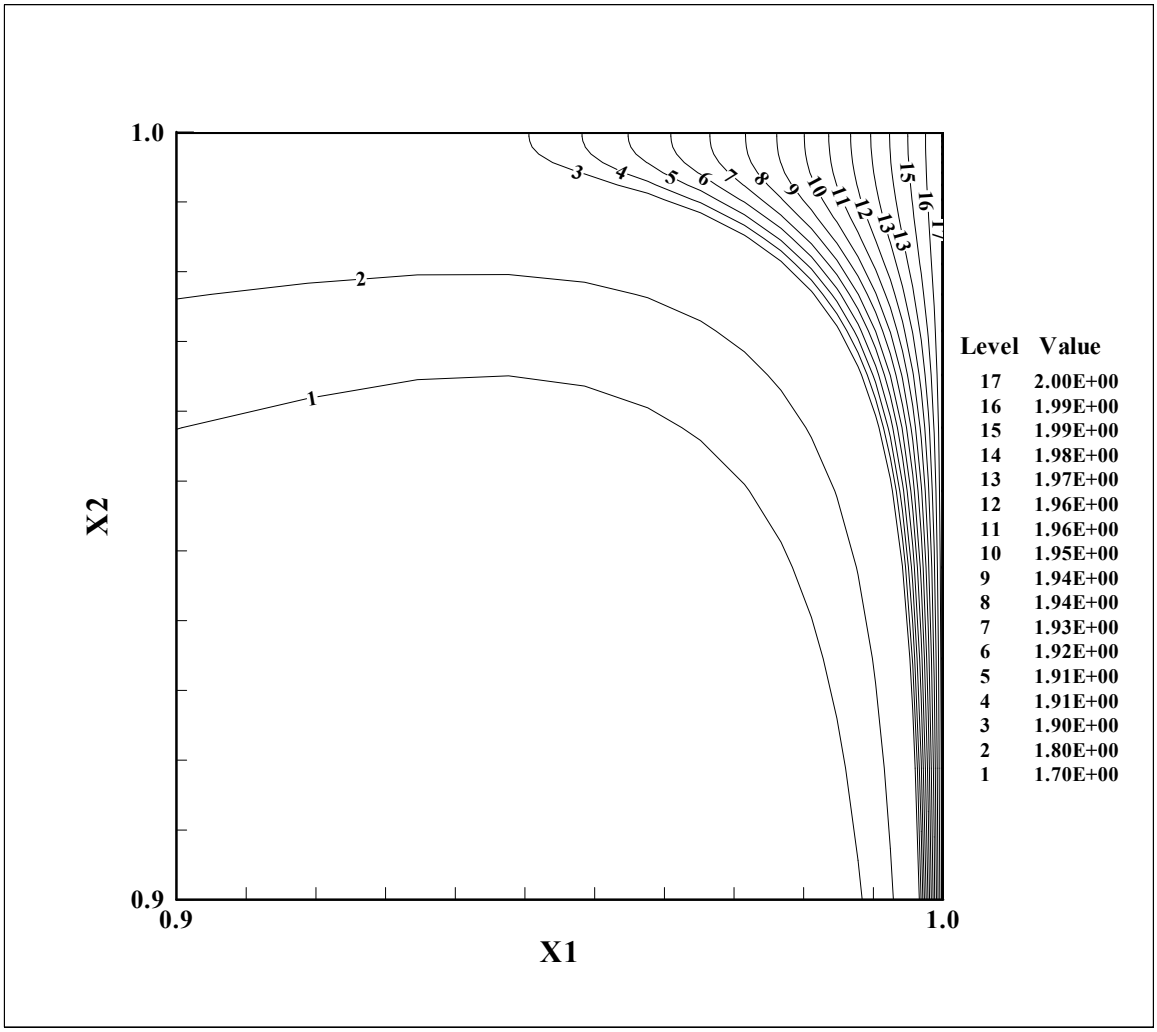

Figure 26: Liquid Hydrogen Analysis Temperature Contours (Magnified 2)

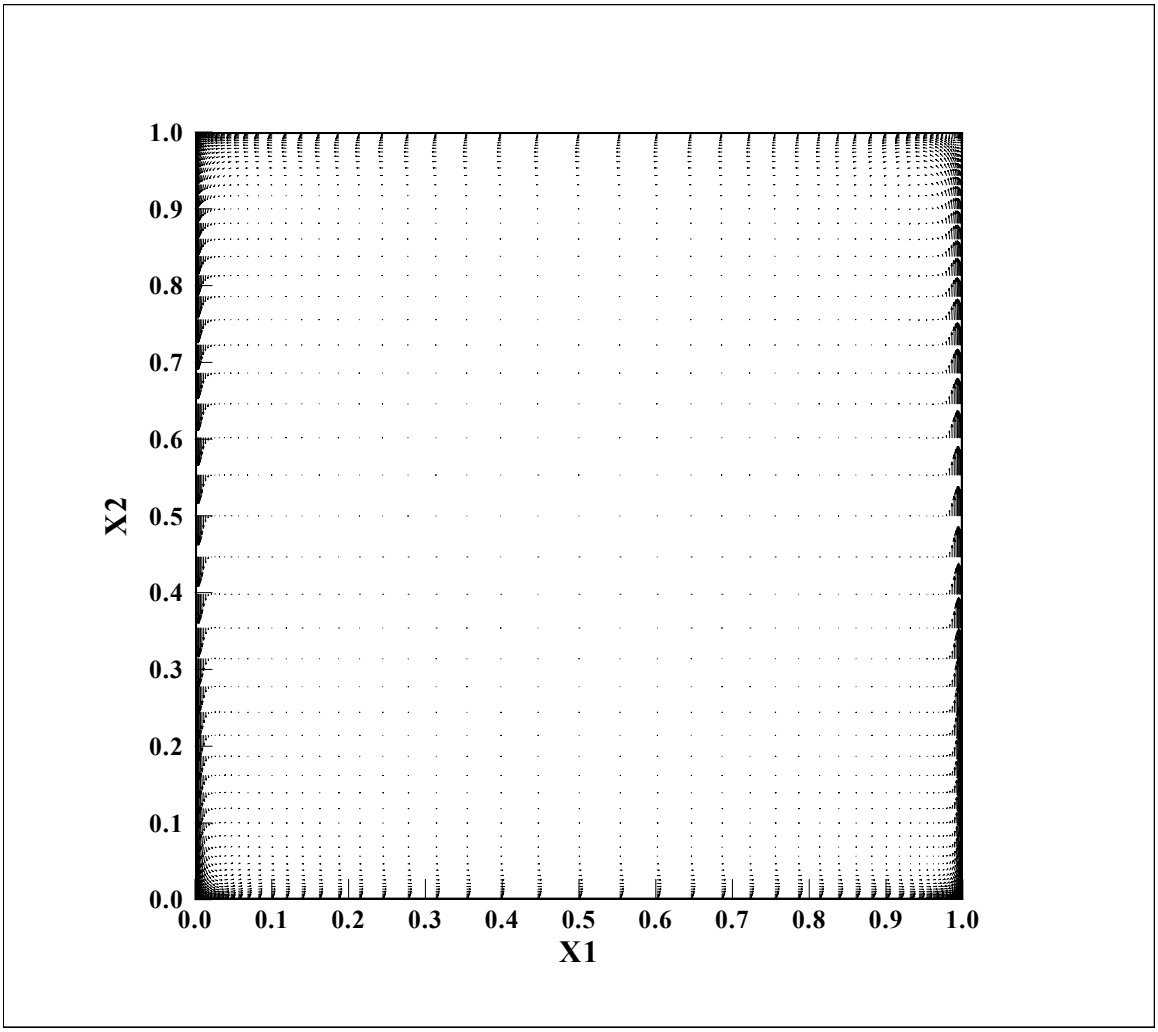

Figure 27: Liquid Hydrogen Analysis Velocity Vectors

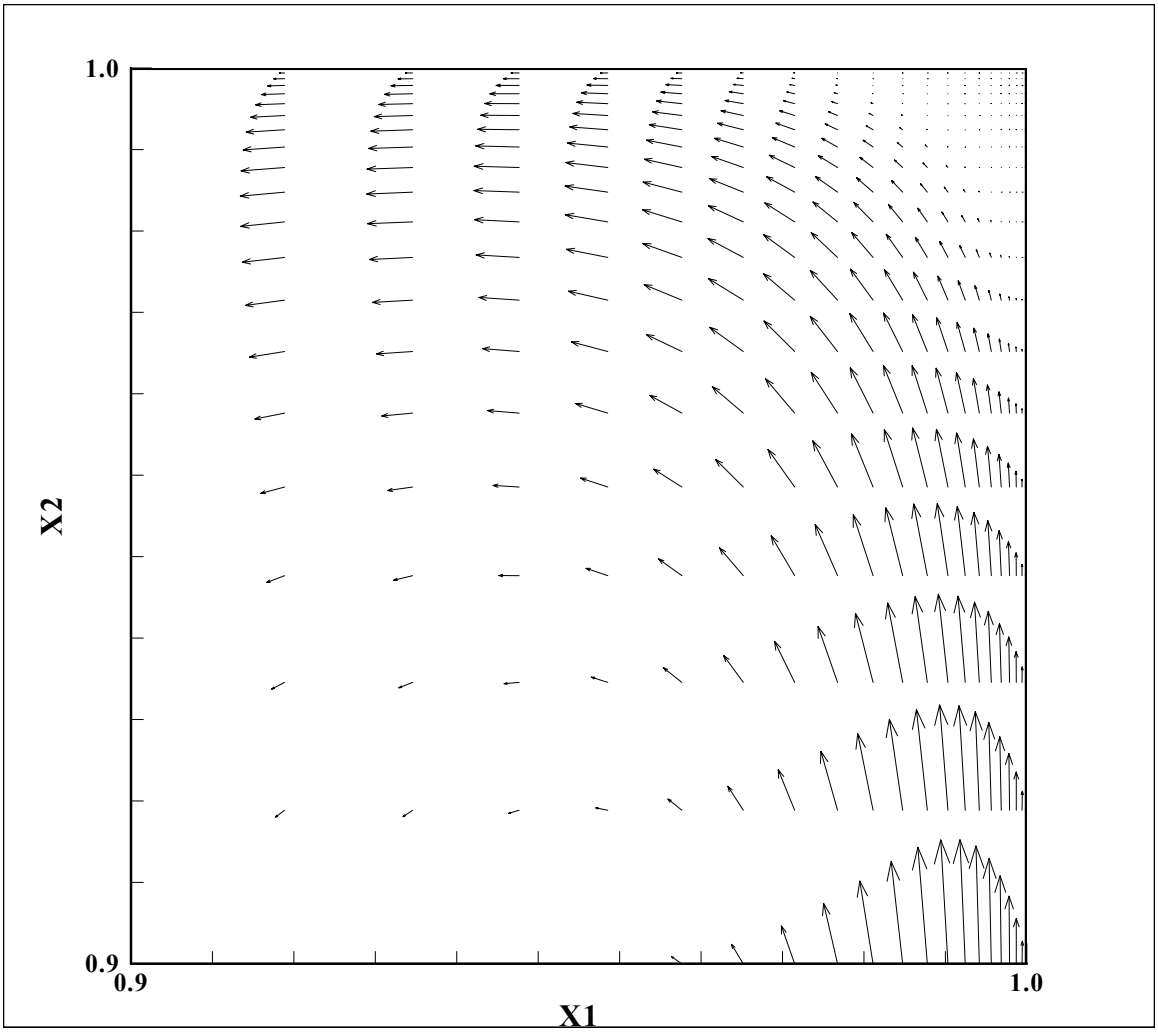

Figure 28: Liquid Hydrogen Analysis Velocity Vectors (Magnified)

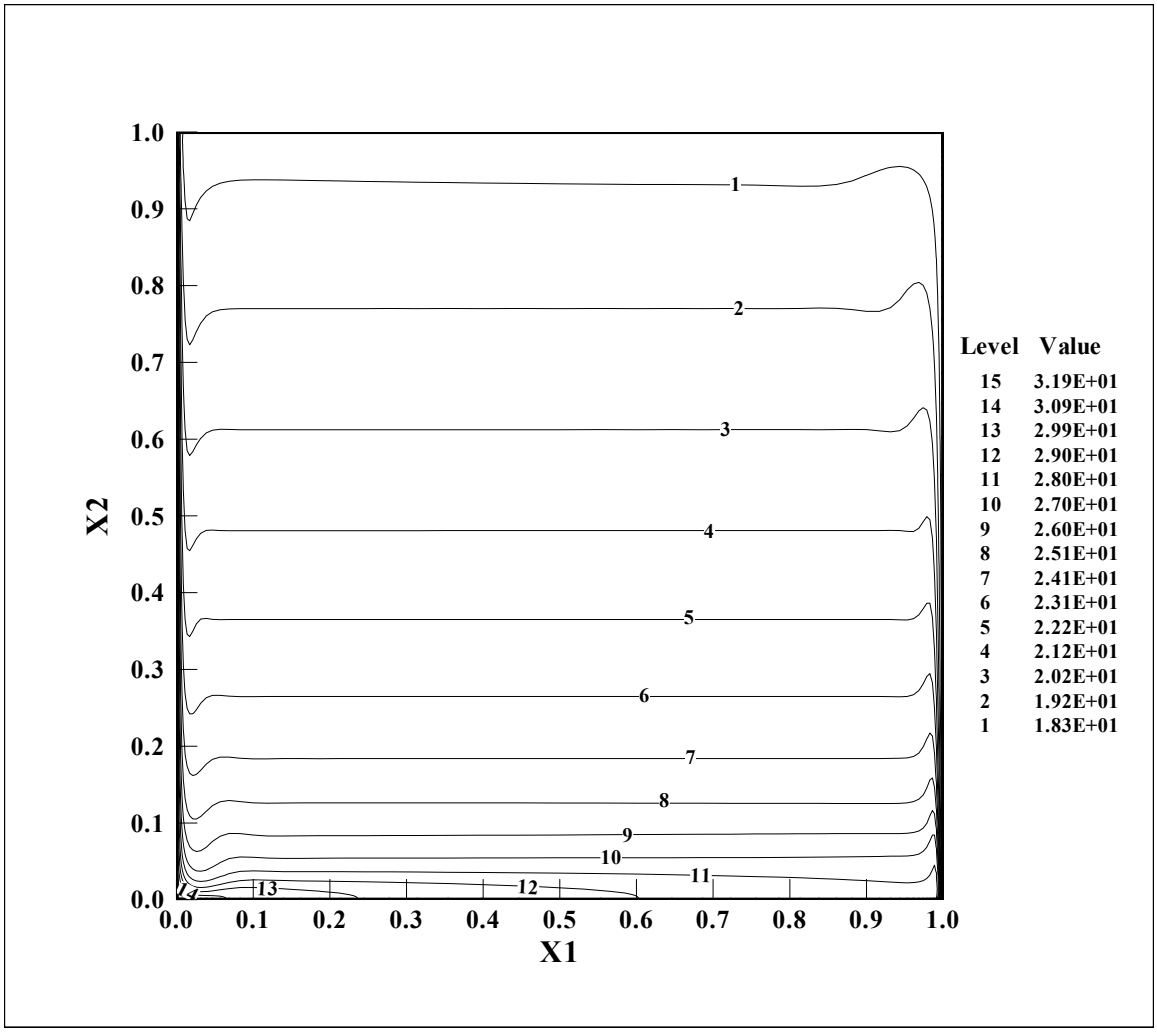

Figure 29: Liquid Hydrogen Analysis Density Contours

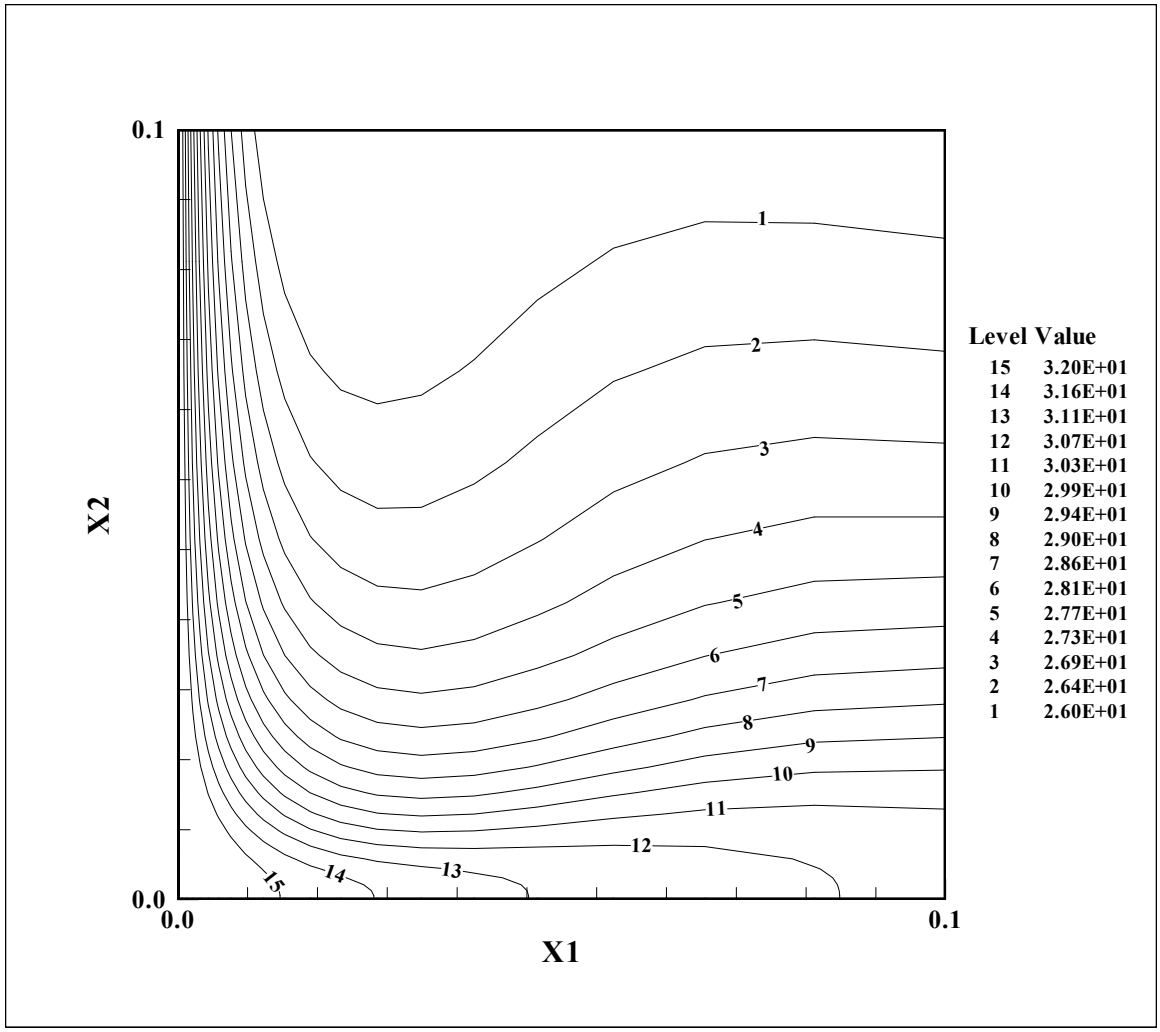

Figure 30: Liquid Hydrogen Analysis Density Contours (Magnified 1)

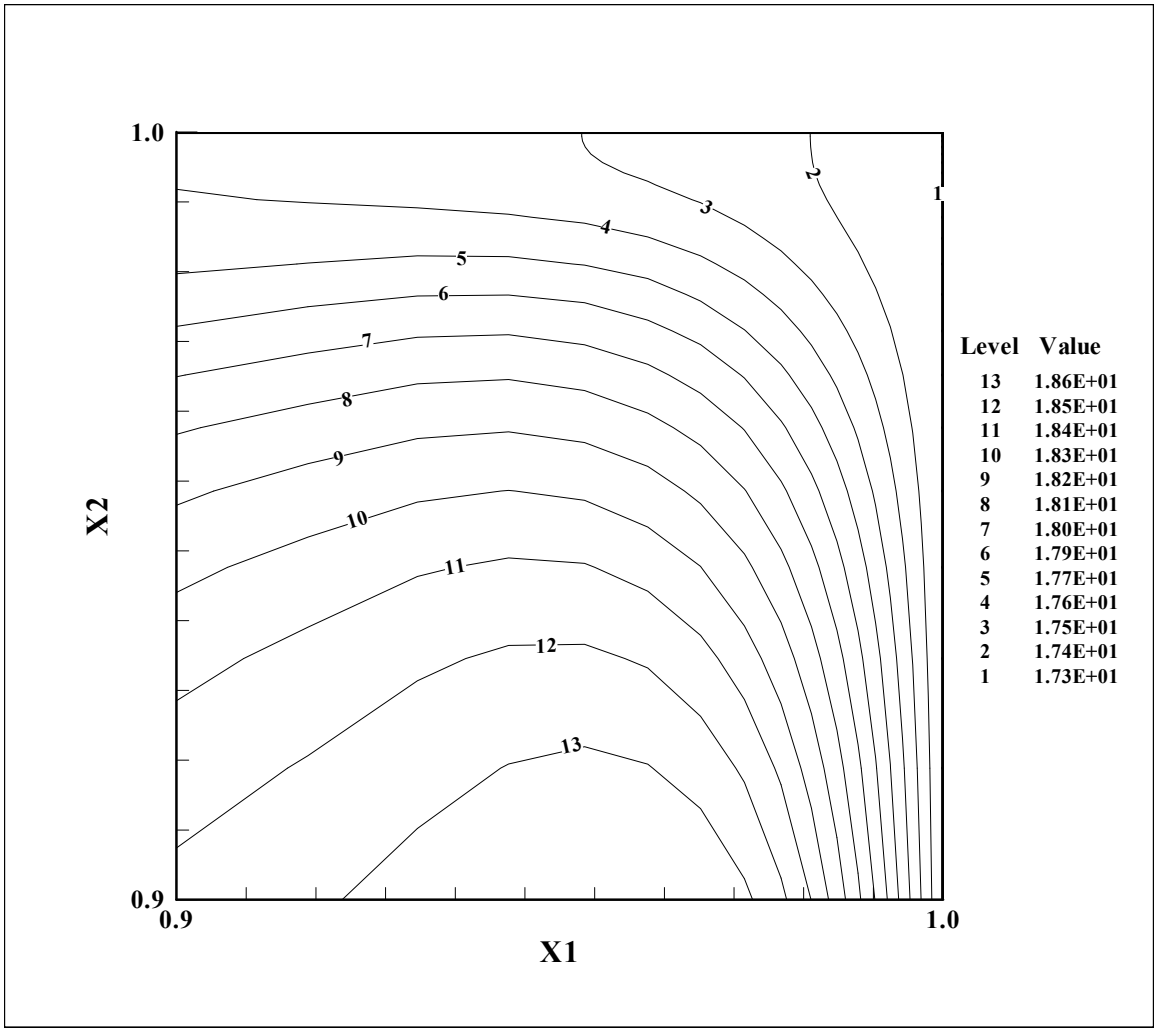

Figure 31: Liquid Hydrogen Analysis Density Contours (Magnified 2)

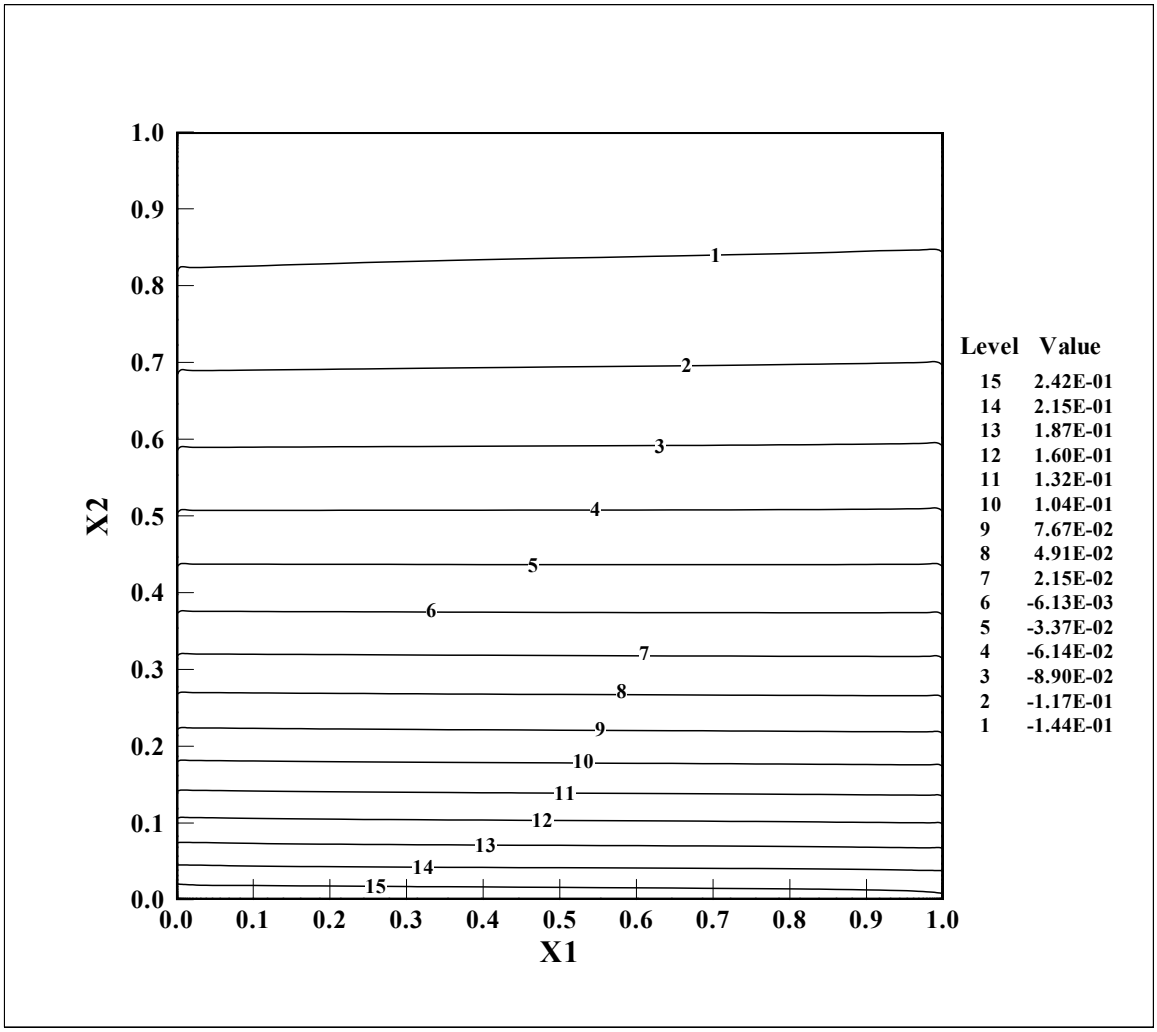

Figure 32: Liquid Hydrogen Analysis Pressure Contours

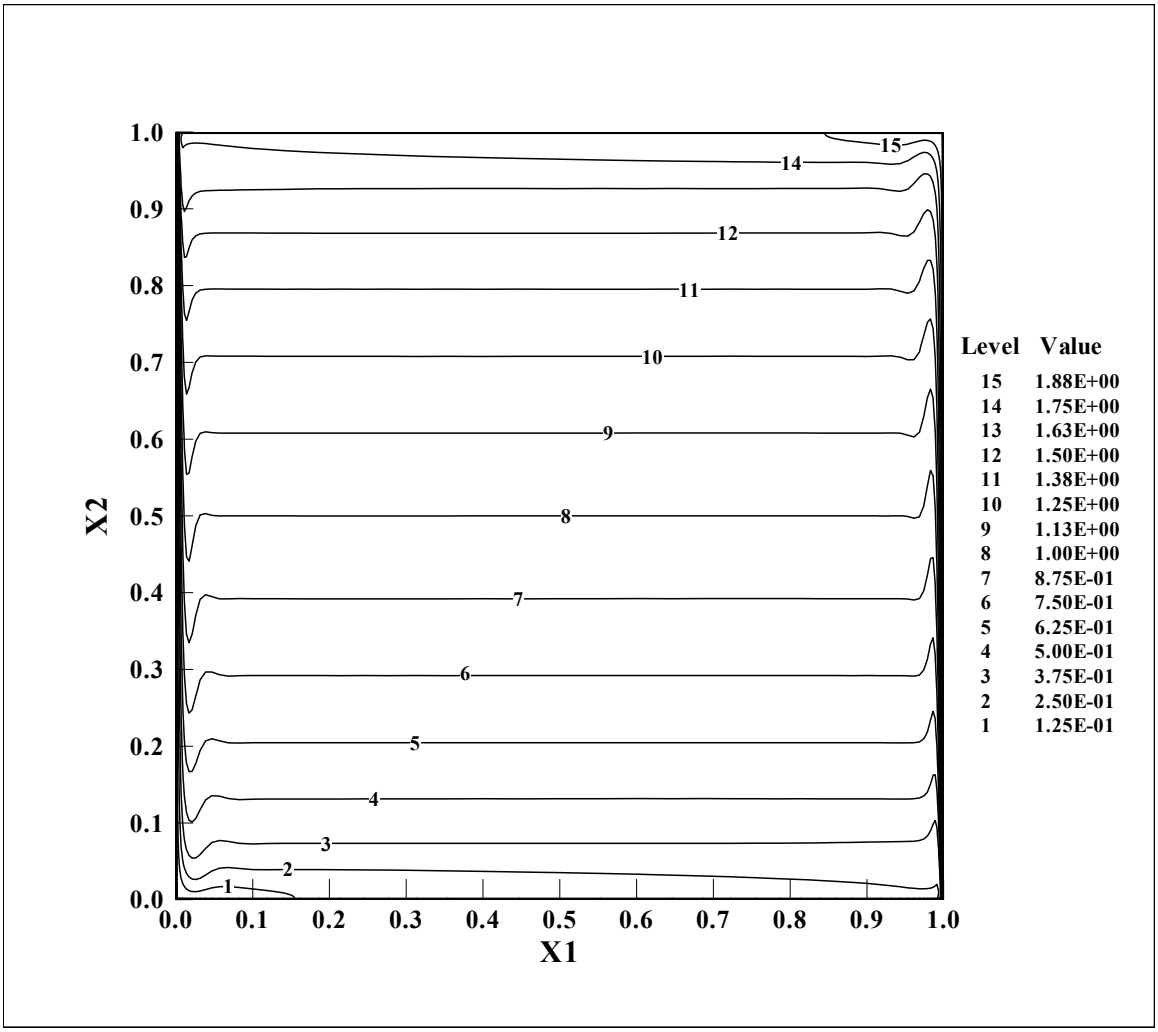

Figure 33: Liquid Hydrogen Analysis (INS) Temperature Contours

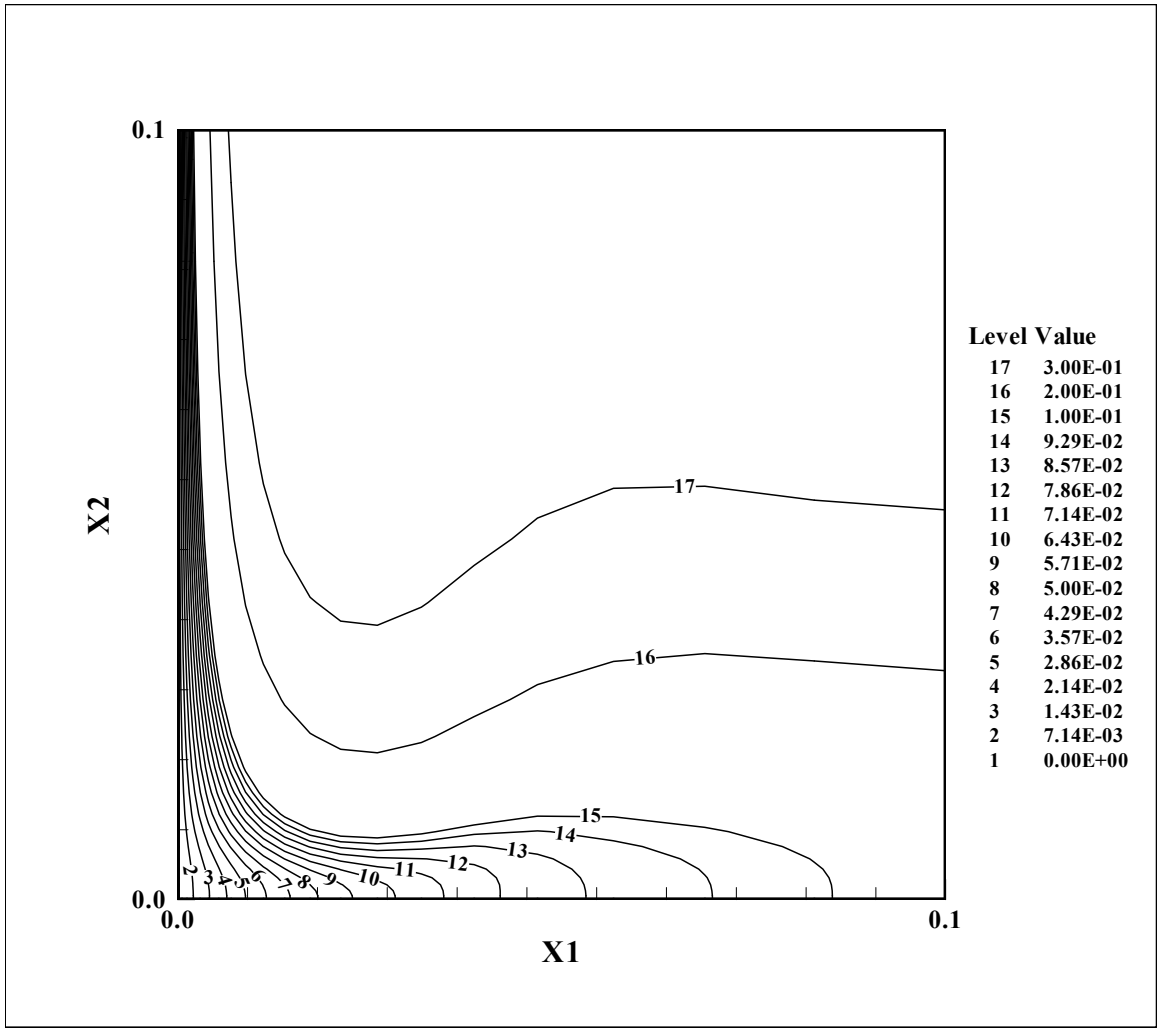

Figure 34: Liquid Hydrogen Analysis (INS) Temperature Contours (Magnified 1)

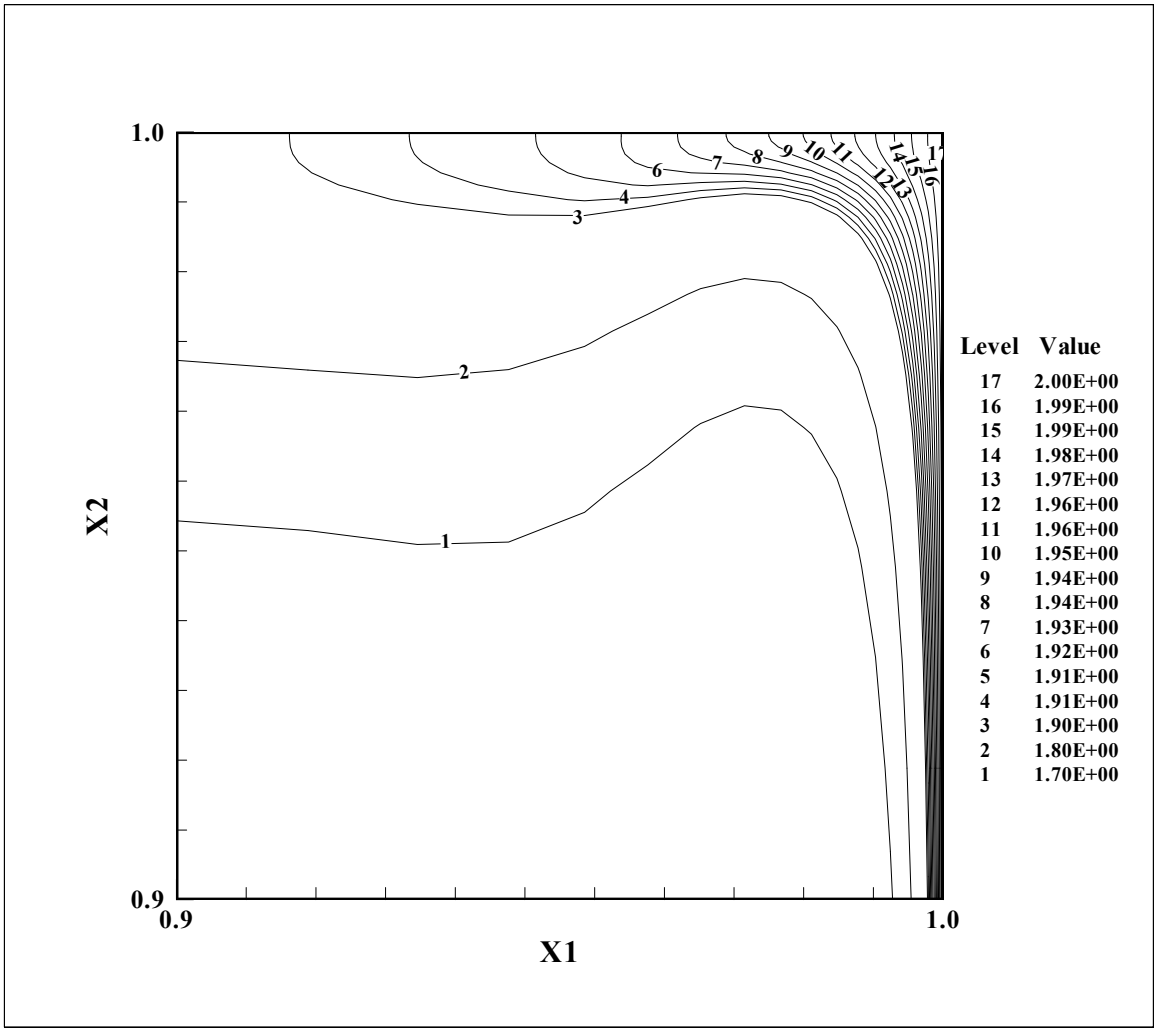

Figure 35: Liquid Hydrogen Analysis (INS) Temperature Contours (Magnified 2)

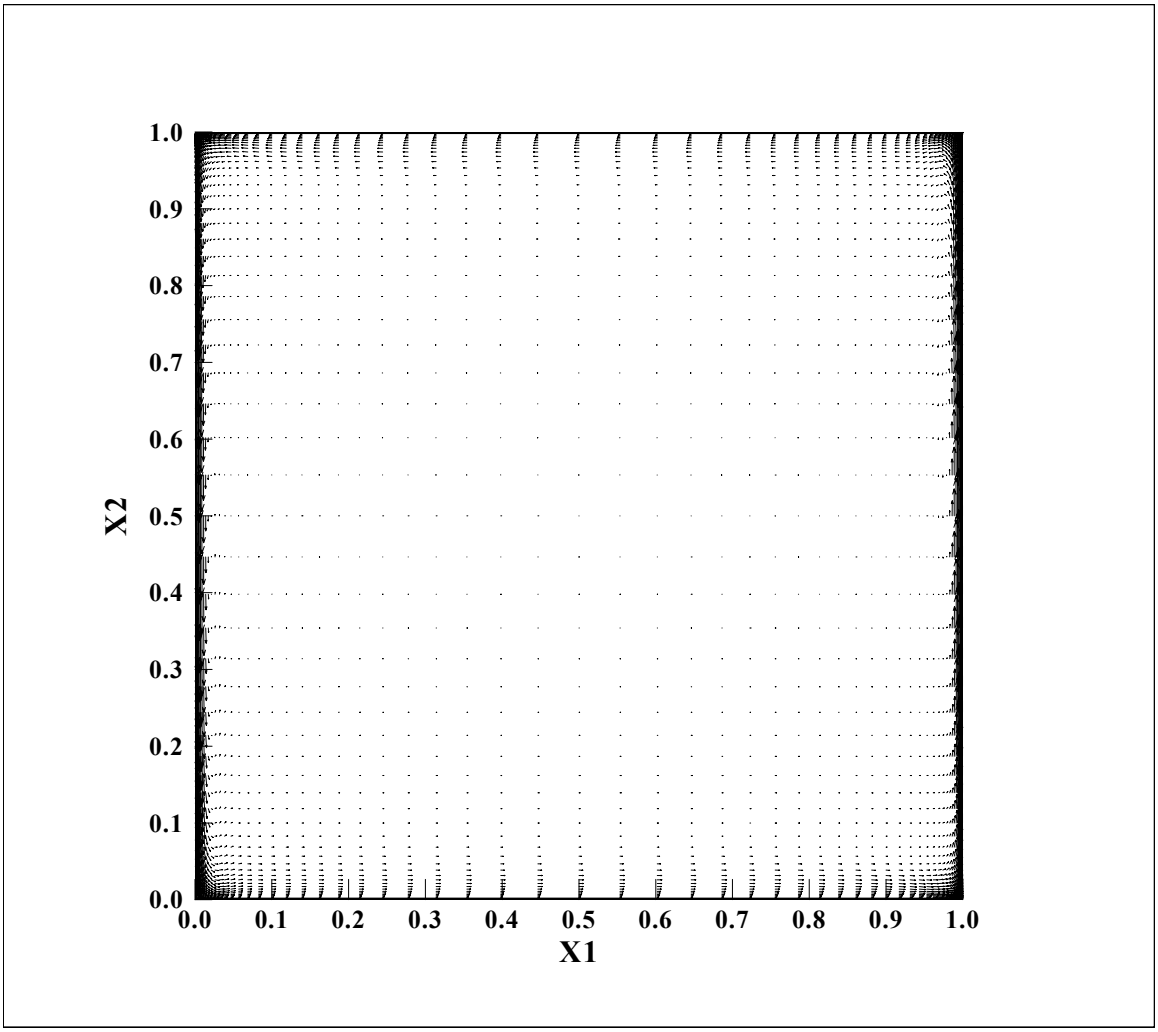

Figure 36: Liquid Hydrogen Analysis (INS) Velocity Vectors

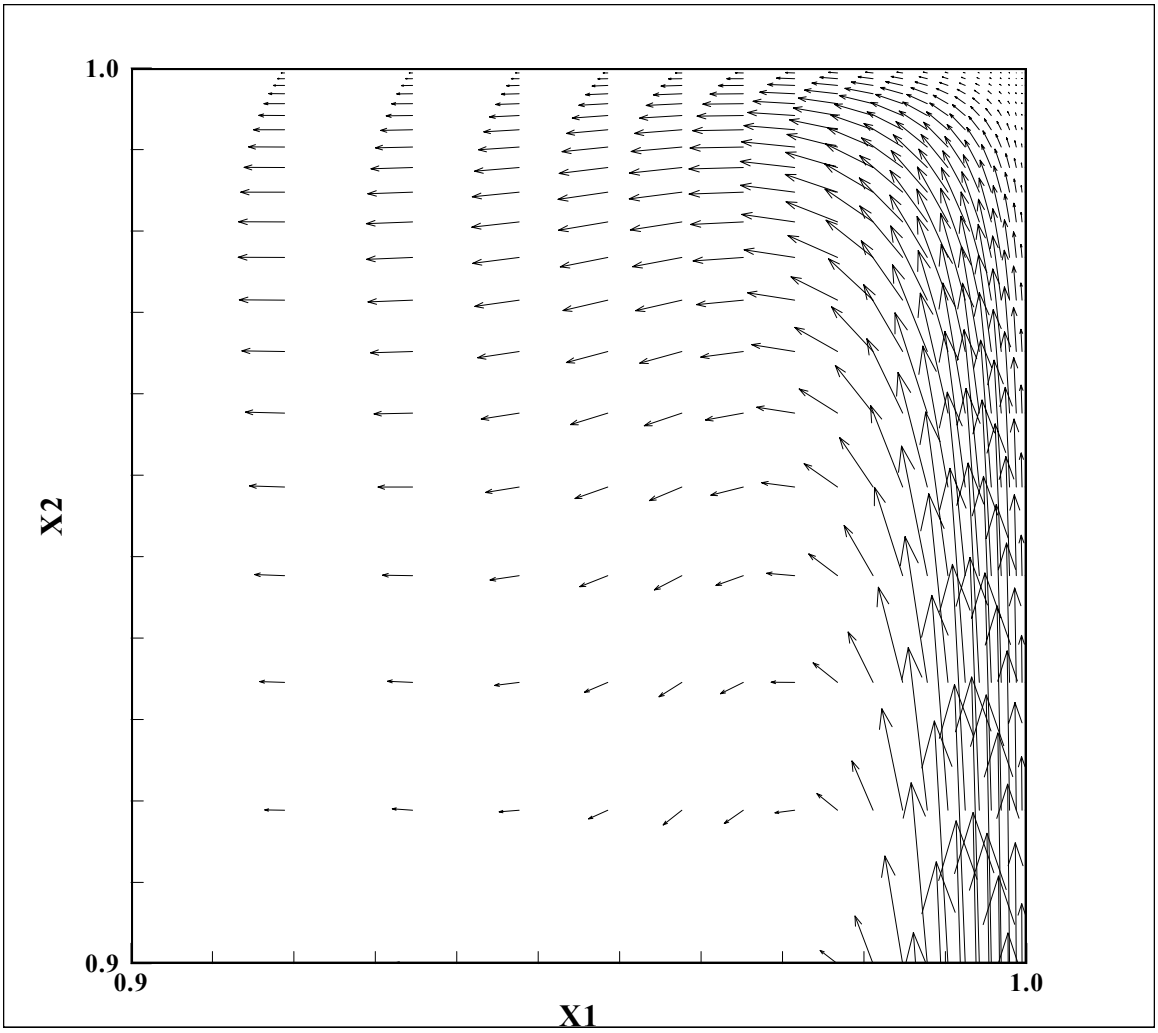

Figure 37: Liquid Hydrogen Analysis (INS) Velocity Vectors (Magnified)

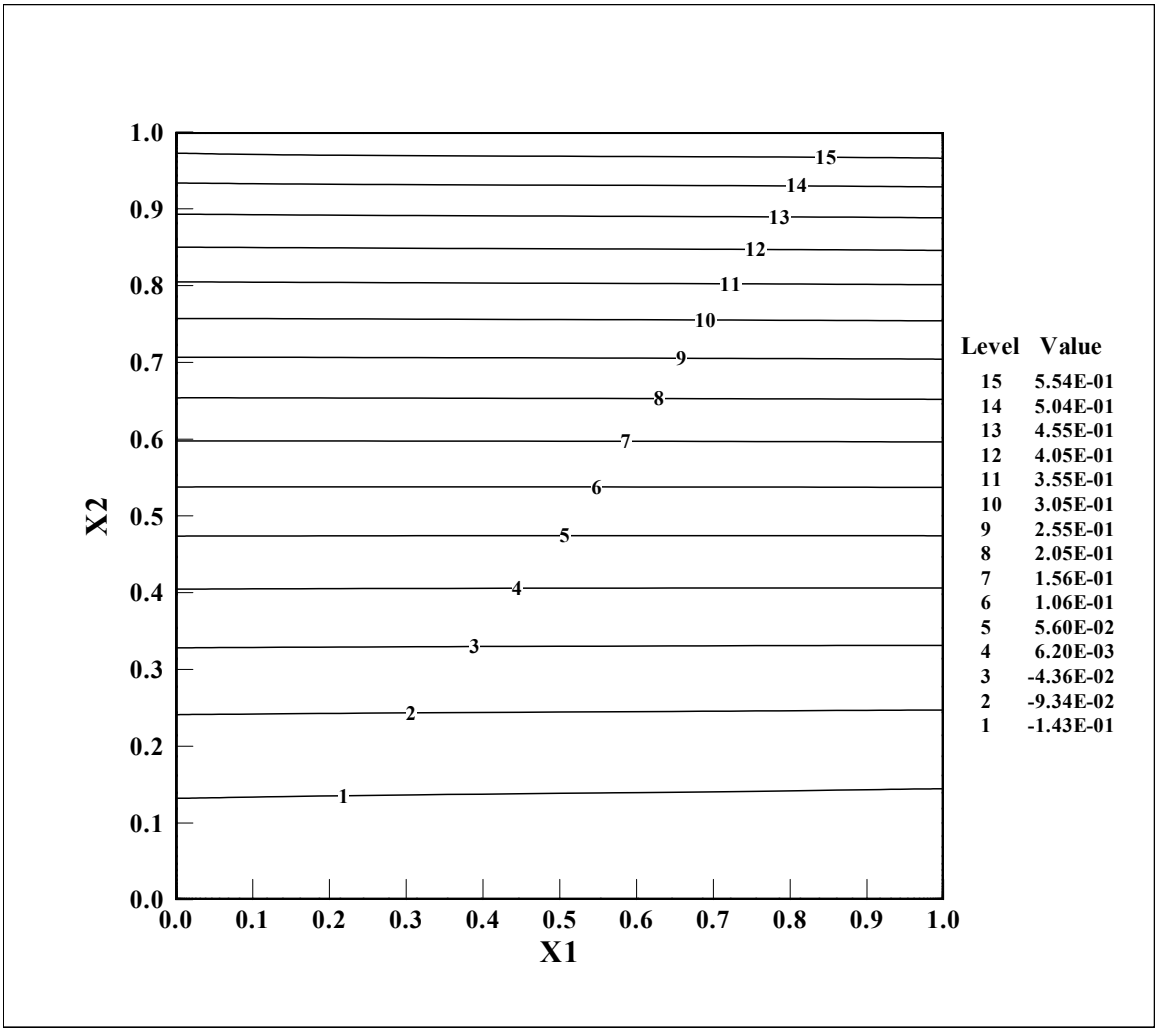

Figure 38: Liquid Hydrogen Analysis (INS) Pressure Contours

APPENDIX D: ANIMATIONS

The material listed in this appendix consists of a series of animations stored in electronic ("avi") file format. In order to view the animations, software capable of reading "avi" format media files must be installed on the PC. To launch an animation within Adobe Acrobat, simply move the cursor over the highlighted filename (i.e. "vdensity1.avi") and click the mouse button. The animation should launch in a separate window. To stop the animation, press the computer's "Esc" key. Alternately, the animations may be viewed directly by using the computer's file system navigator (i.e., "Windows Explorer"), and double click on the name of the animation file of choice. The following is a list of the animation files included with this document:

Animation 1: Time Evolution of Liquid Hydrogen Temperature (Variable-Density Algorithm). (file "vdensity1.avi")

Animation 2: Time Evolution of Liquid Hydrogen Temperature (INS Algorithm). (file "ins1.avi")

Animation 3: Time Evolution of Liquid Hydrogen Temperature and Velocity Fields at Lower Left Corner (Variable-Density Algorithm). (file "vdensity2.avi")

Animation 4: Time Evolution of Liquid Hydrogen Temperature and Velocity Fields at Lower Left Corner (INS Algorithm). (file "ins2.avi")

Animation 5: Time Evolution of Liquid Hydrogen Temperature and Velocity Fields at Upper Right Corner (INS Algorithm). (file "vdensity3.avi")

Animation 6: Time Evolution of Liquid Hydrogen Temperature and Velocity Fields at Upper Right Corner (INS Algorithm). (file "ins3.avi")

VITA

Mark William Yambert was born in Knoxville, Tennessee on May 7, 1962. He was raised in Johnson City, Norris, and Nashville Tennessee. He graduated from John Overton High School in in Nashville in 1980. After high school he attended the Georgia Institute of Technology in Atlanta, Georgia. In 1985 he graduated, with highest honors, from the Georgia Institute of Technology with a B.S. Degree in Engineering Science.

After graduating from college, he worked from 1985-1987 as an Aircraft Structural engineer with Lockheed Georgia Aircraft Company in Marietta, Georgia. Since 1987 he has been employed as a member of the Computational Sciences and Engineering Division at Oak Ridge National Laboratory in Oak Ridge, Tennessee.