AXI-SYMMETRIC TEMPERATURE DISTRIBUTION IN CYLINDROCONICAL BREWHOUSE FERMENTERS FROM FINITE ELEMENT ANALYSIS

Thesis

Submitted to

The School of Engineering of the

UNIVERSITY OF DAYTON

In Partial Fulfillment of the Requirements for

The Degree

Master of Science in Chemical Engineering

by

Bradley R. Bernas

University of Dayton

Dayton, Ohio

August 1999

AXI-SYMMETRIC TEMPERATURE DISTRIBUTION IN CYLINDROCONICAL BREWHOUSE FERMENTERS FROM FINITE ELEMENT ANALYSIS

APPROVED BY:

Ronald A. Servais, D.Sc., P.E.
Professor of Chemical Engineering
University of Dayton, School of Engineering
Committee Chairman

Kevin Myers, D.Sc., P.E.
Professor of Chemical Engineering
University of Dayton, School of Engineering
Committee Member

Sarwan Sandhu, Ph.D.
Professor of Chemical Engineering
University of Dayton, School of Engineering
Committee Member

Harold J. Brandon, D. Sc., P.E.
Professor of Mechanical Engineering
Washington University in Saint Louis
School of Engineering & Applied Science
Committee Member

Donald L. Moon, Ph.D.
Associate Dean
Graduate Engineering Programs and Research
University of Dayton, School of Engineering

Blake Cherrington, Ph.D., P.E.
Dean
University of Dayton, School of Engineering

ABSTRACT

AXI-SYMMETRIC TEMPERATURE DISTRIBUTION IN CYLINDROCONICAL BREWHOUSE FERMENTERS FROM FINITE ELEMENT ANALYSIS

Name: Bernas, Bradley Raymond
University of Dayton, 1999

Advisor: Ronald A. Servais, D.Sc., P.E.

Finite element analysis and a two-equation turbulence model were employed to study the thermal history of a brewhouse Uni-Tank cooling operation. The model system assumed a geometry that typified most brewery tank designs. A theoretical measurement position for the tank mean temperature is presented.

ACKNOWLEDGEMENTS

I am grateful to Dr. Ronald A. Servais, my advisor, for his confidence in my ability to design, set up, and execute this work. His advice, experience, guidance, and patience were invaluable in bringing this effort to conclusion.

I wish to express my sincere thanks and appreciation to Dr. Harold Brandon, Washington University in Saint Louis, MO, for sharing his expertise during the project concept, design, and analysis phases. His contributions to the success of this work cannot be overstated.

Appreciation for Mr. Jim Larson, Siebel Institute of Technology, Chicago, IL, and Mr. Chad Kelly, Enerfab Incorporated, Cincinnati, OH, for their excellent advice, provision of scientific data, and review of the data and project concept work is also hereby given.

I also would like to thank my wife, Julia, my parents, Raymond and Marijean, and my sister, Melissa, for their many years of support.

CONTENTS

CHAPTER

ILLUSTRATIONS

FIGURE PAGE

FIGURE PAGE

FIGURE PAGE

FIGURE PAGE

TABLES

NOMENCLATURE

A	Momentum convection term (shorthand notation)
B	Buoyancy term (shorthand notation)
C	Pressure dissipation term (shorthand notation)
C_p	Heat capacity, or specific heat
C_p^+	Mean heat capacity across the laminar sublayer
C_{pw}	Fluid heat capacity at the wall
C_μ	Turbulence two-equation model constant
D	Energy convection term (shorthand notation)
$\mathbf{F}$	Solution vectors for the velocity, pressure, or temperature
F	Force balance (shorthand notation)
G	Energy balance (shorthand notation)
g_0	Gravitational acceleration
h	Film heat transfer coefficient
i	Horizontal directional heading
j	Vertical directional heading
K	Viscous dissipation term (shorthand notation)
k	Thermal conductivity
k_0	Reference thermal conductivity
k_t	Turbulent thermal conductivity
L	Energy dissipation term (shorthand notation)
l	Characteristic length
M	Mass term (shorthand notation)
M_w	Mass flux through a laminar sublayer element
N	Capacitance term (shorthand notation)
N_{Br}	Brinkmann Number
N_{Eu}	Euler Number
N_{Gr}	Grashof Number
N_{Pr}	Prandtl Number
N_{Ra}	Rayleigh Number
N_{Re}	Reynolds Number
p	Pressure
$\mathbf{P}$	The pressure column vector
P	Pressure term (shorthand notation)
Q	Collected matrix terms
q_w	Heat flux across the wall
$\mathbf{R}$	Residual values for the velocity, pressure, or temperature

r	Radial direction, radius
t	Time
T	Temperature
$\mathbf{T}$	The temperature column vector
T^+	Mean temperature across the laminar sublayer
T_0	Reference temperature
T_w	Wall temperature
T_∞	Alternate representation for wall temperature
t_i	Instantaneous time value
u	Fluid velocity
U	Overall heat transfer coefficient
u^*	Frictional field velocity, proportional to y^+
u^+	Mean velocity across the laminar sublayer
$\mathbf{U}_{i,j}$	The velocity column vector in the *i* or *j* direction
u_w	Fluid velocity at the wall
x	Arbitrary position or length (horizontal)
y	Arbitrary position or length (vertical)
y^+	Dimensionless distance in the boundary layer
z	Axial direction
Δt	Change in time (also *dt or* ∂t)
β_T	Coefficient of volume expansion
ε	Turbulent dissipation
ϕ	Dimensionless time
φ	Velocity shape function for a quadrilateral element
κ	Turbulent kinetic energy
λ	Penalty function constant or pressure dissipation
μ	Dynamic viscosity
μ_t	Turbulent viscosity
μ_0	Reference viscosity
θ	Angular direction
ρ	Instantaneous fluid density
ρ_0	Reference fluid density
σ_t	Turbulence two-equation model constant
τ_{ij}	Turbulent shear stress
υ	Temperature shape function for a quadrilateral element
ω	Relaxation constant of the algebraic solver
ψ	Pressure shape function for a quadrilateral element
f	Buckingham-π function
$\partial^2/\partial n^2$	Second derivative form for an arbitrary variable, n
$\partial/\partial n$	First derivative form for an arbitrary variable, n

CHAPTER I

INTRODUCTION

Brewers pride themselves in providing a consistent, quality product to the consumer. As product curators, they employ the latest biological, engineering, and raw materials technologies to ensure brand equity while remaining competitive. Of the many processes under the brewer's control, one unit operation of particular interest within the brewery is post-fermentation cooling of finished beer before aging. During this procedure, warm fermented beer is chilled and stored within a narrow temperature range away from light under anaerobic conditions. The brewer's specific objectives are cooling the beer quickly to cease fermentation, and starting the aging process at a single average temperature to ensure proper development of aroma and flavor.

With increasing scale, post-fermentation cooling has proven to be a difficult design and measurement problem for both brewers and tank design engineers, who have developed only empirical guidelines to assist their control efforts. If this transformation could be studied theoretically, a further understanding of system boundary, temperature, and time relationships would be gained, resulting in better process design, product consistency, and higher quality. Therefore, this work was undertaken using the latest computational tools to define a theoretical basis for the complex flow fields generated during cooling *en route* to beer aging.

CHAPTER II

OVERVIEW

Problem Description

The engineering objective of this work is to develop and present a computational fluid dynamics (CFD) scheme for solving an otherwise intractable analytical fluid problem. The results will be compared with actual data collected from a field system of same design. The validated model will reproducibly simulate a beer cooling unit operation and provide information for a preferred location of a mean temperature sampling point for the fluid.

The system under study for this work is known as a Uni-Tank, which in brewing is a discipline-specific term meaning a vessel in which both fermentation and aging of beer will occur. These tanks are cylindrical in geometry with a dished head at the top of the vessel, and a conical section fused to the bottom of the cylinder. Commercial equipment is normally specified by the ratio of the cylinder height to the diameter, or aspect ratio. This is shown schematically in Figure 1. A summary of the brewing process has been provided in Appendix A.

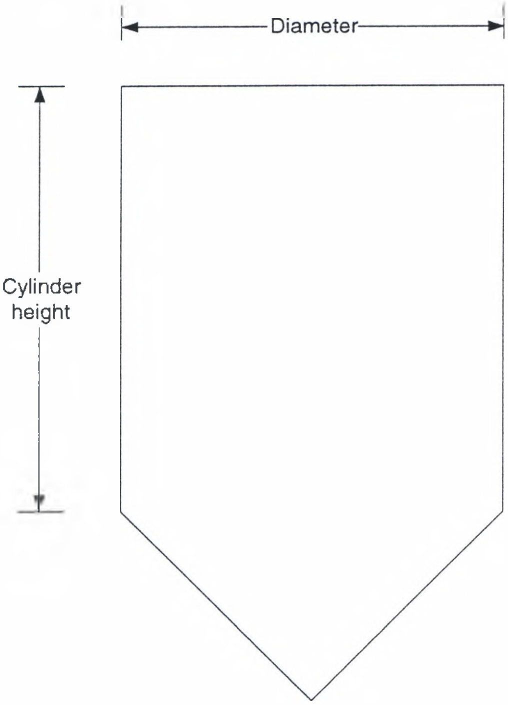


Figure 1. Dimensions for the Uni-Tank Aspect Ratio

At the beginning of a lager Uni-Tank process, sweetened aqueous grain extract (wort) is blended with a yeast culture in an empty and clean tank at an initial temperature of 53-57 °F (12-14 °C). The wort [*wert*] is transformed into beer by the yeast according to strict fermentation specifications. After conversion, the beer is chilled in one or two stages from 55-65 °F (13-18 °C) to a final temperature near the freezing point of water by an applied cooling surface. Buoyancy forces develop from strong thermal gradients within the fluid, resulting in a slowly recirculating flow in the vessel. Yeast kinetics will cease during this event, and the biomass will settle into the tank cone for subsequent removal. After the target temperature is reached, the beer is stored for a specific time as defined by the brewer.

The convective nature of the mixing is complicated by a flow inversion during the middle of the unit operation (~40 °F), where the density no longer increases with decreasing temperature. Below the inversion point, the fluid density will decrease monotonically as additional heat is removed from the beer. The initial recirculation pattern will stagnate, then reverse direction for the remaining portion of the process cycle.

To study these flow phenomena, an algebraic technique known as finite element analysis (FEA) is employed to solve purely mathematical statements of the fluid motion. These statements are known as the Navier-Stokes equations, which express mass, momentum, and energy conservation laws *via* differential equations. Proper application of the equations requires fluid property data, system boundary information, and a physical understanding of the fluid flow in the system. Criteria for model verification includes direct comparison of the theoretical system with actual experimental data collected separately.

Modeling Versus Experimentation

Tank design and fabrication requires simultaneous consideration of such factors as available real estate, brewing philosophy, construction site logistics, end user specifications, equipment lead times, government regulations, labor costs, raw material pricing fluctuations, and processing objectives. Many combinations of these variables exist, as demonstrated by the vastly differing system geometries at plants and small breweries worldwide. This level of variability is particularly cumbersome for the tank designer whose objective is to reliably meet performance criteria at the first design attempt. Today, an engineer's research cache contains two main tools for testing new, existing, or retrofit designs: computational modeling and experimental modeling.

Employment of computational models provides several potential advantages over experimental scale modeling. First, any parameter used in simulation efforts can be arbitrarily adjusted for design optimization. By performing parametric studies, single variable or multivariate relationships between dependent variables can be studied quickly with a minimum setup time between runs. Second, for larger research efforts, the cost of computational modeling becomes inversely related to the number of experimental runs. The opposite is true for scale modeling, which requires refabrication of equipment or custom preparation of test fluids for each new run. Third, computational experiments can be run at any time, without regard to scheduling or staffing in a research and development laboratory. This testing option has become considerably more pragmatic as CFD software can now be run on desktop PCs. Lastly, numerical techniques provide a form of non-invasive sampling of the system fluid domain.

Planning for potential problems before using CFD ensures a better chance for success. One significant consideration is software suitability to the modeling purpose. A variety of CFD codes exists, each with a particular strength and weakness. Looking beyond the needs of current research will avoid future disappointments and wasted resources. Another issue concerning CFD software is the tendency to believe results without experimental data to back the conclusions. Numerical output cannot be accepted blindly without some form of model validation. Thus, a scaled physical model representing an optimally configured CFD model must be constructed at some time. However, numerical analysis will reduce the possible design list considerably, saving both time and capital expenditures. A summary of the main advantages and disadvantages of both computational and experimental modeling has been included in Table 1.

Table 1.

Comparison of Computational and Experimental Methods

Computation		Experimentation	
Advantages	Disadvantages	Advantages	Disadvantages
• System variables can be studied parametrically • Many boundary conditions can be studied • Fluid properties can be readily changed • Non-invasive sampling • Inexpensive for long term studies • Superiority for turbulence experimentation • Assumptions easily tested • Experiments can be run anytime	• No physical system • May be too costly for single runs • Users require training to be productive • Results may falsely represent the real system • May use incorrect physical data • Risk of unnecessary complexity • Data for analysis or turbulence modeling may not be available	• Gives physically meaningful data • Experiments can be set up and run quickly • May be more cost effective for small systems with simple fluid mechanics • Data collection is straightforward	• Must coordinate lab time • Experimenters must be present • Costly for study at very large scale or for multiple systems; hard to parameterize • Mechanical breakdown • Risk of fluctuations in conditions while experiments are conducted • Results may be affected by sampling technique

CHAPTER III

LITERATURE REVIEW

Processing of Fermented Beer

Practical constraints challenge the successful and timely processing of beer during post-fermentation cooling. For example, no flow assistance can be used in the tank since: *(a)* exposed edges, internal fittings, or crevices such as tank fillets or baffles may host microbiological contamination, and *(b)* impeller agitation will sweep settled biomass into the beer, ultimately affecting the clarity, filterability, and sensory characteristics. For these reasons, Uni-Tanks are generally unagitated vessels with smooth internal contact surfaces. Cooling is provided by heat transfer jackets constructed around the external shell of the tank.

In large breweries, cooling generally occurs in a one- or two-step event called “crash cooling”. If a first crash cool is completed, the warm fermented beer is chilled to ~40 °F, deactivating any live yeast and flocculating proteins in the product. During a second crash cool, the temperature is further reduced to 28-34 °F. It is also acceptable to perform both crash cools in one step. In this work, only the second crash cooling sequence is studied.

Developments Leading to the Uni-Tank Process

Before the early 1900s, brewers employed many geometries for the production of their beer (1): closed and open top cubes, horizontal closed cylinders, open top spheres, open top vats, slanted bottom closed tanks, and vertical cylinders with dished heads. Vertical cross-sections of some tanks even resembled that of a loaf of bread (square sides and bottom, semi-circular top). Many vessels were additionally fitted with cooling jackets, recirculation pumps, or gas injection equipment in an effort to produce greater quantities of consistent quality beer. In the mid-1930s, the Nathan group of Germany introduced a cylindroconical vessel (called a Nathan tank) that could be used for fermentation or aging (2). This style of tank gave many advantages to the brewer and engineer over previous designs, and rectified many of the processing trade-offs that hampered quality and productivity. A few of these advantages were:

- The vertically oriented vessel gave a significantly higher output of product per square foot in the plant, requiring less real estate to meet productivity targets.
- Fermentation and aging were completed in a single vessel that fewer personnel operated safely and consistently.
- Yeast was recovered through the conical bottom of the tank, which reduced opportunity for microbiological contamination.
- Because of their inherent efficiency, the fermentation vessels were built to contain volumes greater than 3000 barrels (93,000 gallons or 3600 hL).

- Waste was minimized as no beer was lost from wort production through fermentation and aging. In addition, CO_2 evolved from the process was collected and reused for carbonation adjustment.
- The closed tanks were cleaned automatically using a clean in place (CIP) method that prevented contamination by wild yeast strains or other pathogens.

Attempts at Understanding Uni-Tank Fluid Flow

After Uni-Tanks gained acceptance in the brewing industry, effort was directed toward controlling the internal flow patterns for reasons of process optimization. The first published evidence of fluid mechanical understanding occurred in 1938 when Bishop (3) noted the mixing characteristics of beer yeast suspensions during fermenter cooling. Few additional articles were published until a landmark paper was presented by Delente, Akin, Krabbe, and Ladenburg (4) in 1968. In their trials, cross-sectional cutaway tanks closed with Plexiglas® were constructed, and beer was prepared in the vessels to monitor convective flow behavior. The study concluded that cylindroconical vessels with cooling jackets gave better heat transfer, higher productivity, and simpler yeast removal while the amount of personnel needed to operate such equipment was reduced. Although today it is accepted that the experimental results were inaccurate due to irregular geometry, the work was a major conceptual step forward with respect to convective flow visualization within brewery vessels.

By the mid-1970s, competitive pressures among the world's largest commercial breweries drove the development of equipment that maximized the economy of scale. During this time, strict control of the Uni-Tank process became the object of considerable industry focus. Though beer could now be produced consistently in commercial quantities (5), there remained a large gap between theoretical and practical knowledge of the vessel fluid mechanics.

Knudsen, in 1977 (1), constructed small scale glass and stainless steel models to observe internal convective flow patterns during brewing without the inherent disadvantages of the Delente study. Knudsen's results were similar to that of Delente, and were more dimensionally consistent with vessels of the period.

In 1988, a thorough engineering approach to Uni-Tank design was submitted by Larson and Brandon (6), and a preliminary description of turbulent behavior during cooling was described. In addition, the effects of tank geometry, beer temperature, cooling jacket temperature, and cooling jacket positioning were summarized. Time/temperature data were collected by adjustable immersion thermoprobes placed at several locations throughout the tank, and boundary layer data was collected at the wall *via* syringe (7). It was proposed by the authors that the best location for an average temperature measurement was at a fixed distance above the junction of the cone and cylindrical parts of the vessel. A follow-up to this study was presented in 1995 (8) by Reuther, Brandon, Raasch, and Raabe. In this work, a 1/10th scale model of an actual vessel was constructed to simulate the flow behavior of a commercial tank. Their dimensionless analysis results were in excellent agreement. Temperature/time data were collected according to the study performed by Larson and Brandon (6).

Recently, the first published attempt at non-intrusive, computational modeling of internal flow during Uni-Tank crash cooling was presented by Ishiguro, Mizutani, and Kuwahara, where the Navier-Stokes equations were solved using a multi-directional finite difference method and a supercomputer (9). Experimental temperature/time data were collected from a test tank and showed excellent correlation with the CFD results. No specific recommendation was made for the placement of an average temperature thermoprobe within the vessel.

Finite Element Analysis

Definition and Background

FEA is a method of evaluating analytically unsolvable differential equations by substituting simpler but approximate algebraic expressions. Early FEA algorithms were developed between 1930 and 1960, and were primarily applied to structural and aerospace engineering problems. The method was a convergence of similar ideas and numerical thinking by mathematicians, physicists, and engineers of the period. The first demonstration of the complete method as it is known today was outlined in a paper by Clough (10), in which material elasticity phenomena were described using mathematics. The first algebraic solution algorithms were developed by Courant (11), who employed numerically discrete triangular parcels and energy minimization techniques to study torsional stresses. Other practitioners of the art modified the basic method to include both irregular domains and boundaries. Through these efforts, the initial utility and validity of the method were established, resulting in rapid literature growth through current times. Many introductory and advanced texts are now available to the interested reader covering topics such as turbulence modeling, algebraic solution methods, and convergence techniques.

Application of Finite Element Analysis

Engineers are called upon to understand and explain transient fluid systems for process design or control reasons. While attempting to characterize a process, the engineer will recall the Navier-Stokes equations from a textbook, invoke a few simplifying assumptions, assume initial and boundary conditions, then set the simplified equations for solution. What they quickly realize is a unique solution to the problem does exist, but cannot be readily determined in closed form using current analytical differential equation techniques.

Fortunately, FEA was developed to handle such a situation. The method estimates pointwise average values within the system by assembling a *finite* set of small subregions called *elements*. This process is known as *discretization*. Treated as such, the properties of each discrete element can be evaluated iteratively using algebraic techniques, and the ensemble of results can be integrated to approximate true system values for any variable under study. FEA solutions are by definition never exact, but with proper convergence criteria, highly accurate results can be obtained. As vast computational power has proliferated at greatly reduced cost, the method is now widely available, simple to use, and within the reach of almost any scientist committed to learning the technique and software.

Basics of Finite Element Analysis

After the proper engineering questions have been posed and a suitable equation describing the fluid situation has been developed, the next step is to prepare the system for FEA implementation. A general understanding of how the method handles fluid systems will act as an avoidance mechanism for improper system definition. A rudimentary outline of the method is presented in the following paragraphs.

First, assume that a fluid domain and its associated physical characteristics behave as a continuum with an infinite number of potential solutions. Second, subdivide the fluid domain into discrete elements, and further assume that these elements can be flexible, slightly distorted, or bent. Next, incorporate a suitable reduction of the Navier-Stokes equations in differential form, and collect system boundary data and primitive fluid properties. Insert all known quantities into the reduced equations. What remains is a set of dependent variables that cannot be known in advance everywhere in the fluid. Normally, the unknowns will include the velocity, u in any direction, the temperature, T, and the pressure, p. In addition, these relationships may be a function of time, *t*.

The last step is to identify the shape of the element, and thus, a set of interpolation functions that will assist in the evaluation of the unknowns. Although many shapes such as triangles, pyramids, cubes, and tetrahedrons exist to fit any arbitrary domain, the quadrilateral has been demonstrated as mathematically robust for simple, two-dimensional geometries. The basic form of the quadrilateral element is shown in Figure 2:

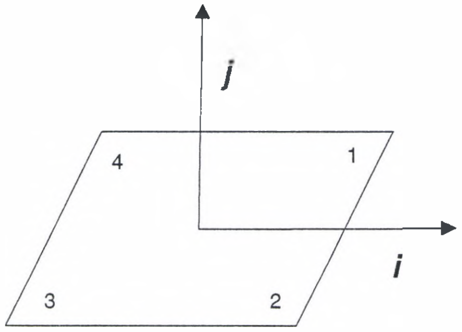


Figure 2. Schematic of a Simple Four Node Quadrilateral Element

where *i* and *j* are two-dimensional vectors and the numbered corners represent nodes. The unknown properties of the quadrilateral element are estimated using interpolation functions (also known as shape or basis functions). If φ, ψ, and υ represent interpolation functions for velocity, pressure, and temperature, respectively, the quadrilateral interpolation functions are defined as follows:

φ = {Mean of all four nodal velocities in the *i* or *j* direction}

ψ = {Mean of all four nodal pressures}

υ = {Mean of all four nodal temperatures} where fluid properties are evaluated at each corner of the quadrilateral and equally weighted by averaging.

Thus far, a system is defined with physical properties, primitive variables, a reduced fluid equation, and a desired element shape with associated interpolation functions. A method is now needed to assemble the information and estimate the unknown velocities, pressures, and temperatures.

Allow all of the nodal unknowns in the entire domain to be represented by the vector quantities:

$\mathbf{U}_{i,j}$ The velocity vectors of the nodes in the *i* or *j* direction

P The pressure vectors of the nodes

T The temperature vectors of the nodes

Then, a general FEA statement of the solution for an element is represented by:

$$u_{i,j}(x,t) = \varphi \mathbf{U}_{i,j}(t)$$

$$p(x,t) = \psi \mathbf{P}(t)$$

$$T(x,t) = \upsilon \mathbf{T}(t)$$

Simply put, the preceding equations state that within an element, values for velocity ($u_{i,j}$), pressure (p), and temperature (T) at some position (x) and some time (t) are approximated by averaging all of the nodal properties in the element ($\varphi\mathbf{U}_{i,j}$,$\psi\mathbf{P}$, $\upsilon\mathbf{T}$) at the same time (t). Since the velocity ($\mathbf{U}_{i,j}$), pressure (**P**), and temperature (**T**) vectors are unknown, the best chance for solution is by approximating to a small acceptable deviation, known as a *residual*.

Symbolically, these approximate expressions are:

$$\mathbf{f}_1(\varphi,\psi,\upsilon,\mathbf{U}_{i,j},\mathbf{P},\mathbf{T}) = \mathbf{R}_1$$

$$\mathbf{f}_2(\varphi,\mathbf{U}_{i,j}) = \mathbf{R}_2$$

$$\mathbf{f}_3(\varphi,\upsilon,\mathbf{U}_{i,j},\mathbf{T}) = \mathbf{R}_3$$

where $\mathbf{R}_1$, $\mathbf{R}_2$, and $\mathbf{R}_3$ are residual vectors from the calculations. The above expressions imply a functional relationship between the velocity, pressure, and temperature variables, and represent a basic statement for the momentum, mass, and energy conservation equations, respectively. The quantities $\mathbf{f}_1$, $\mathbf{f}_2$, and $\mathbf{f}_3$ represent unknown solution vectors for the system.

To achieve solution convergence, the residuals must be as small as possible. This is done through a mathematical procedure called *orthogonality*, where the product of the residual vectors $\mathbf{R}_1$, $\mathbf{R}_2$, and $\mathbf{R}_3$ and the solving functions $\mathbf{f}_1$, $\mathbf{f}_2$, and $\mathbf{f}_3$ are forced to a very small value *via* multiplication of $\mathbf{R}_i$ by the inverse vector $\mathbf{f}_i$. When the $\mathbf{f}_i\mathbf{R}_i$ products reach a predefined minimum value, known as the *error tolerance*, the system is sufficiently approximated for that time step in the process. After the tolerance condition has been satisfied, the system will move forward in time, and the entire process is repeated until all calculations are completed. In this manner, estimates for the resultant vectors $\mathbf{U}_{i,j}$, $\mathbf{P}$, and $\mathbf{T}$, and in turn, approximate values of $u_{i,j}$, T, and p are obtained. During the solution of a transient problem, all solutions are obtained simultaneously for each element and each degree of freedom until the error tolerance is satisfied. These solutions are integrated to give approximations for the unknowns, or are used as initial values for another iterative sequence.

Therefore, the mathematical objective of FEA is to minimize the difference between the true value of the system and an approximate solution using known physical properties, shape functions, and vector manipulation to achieve this end.

The final basic step in FEA is assembly of the algebraic matrix system used to solve the conservation equations at each node. To demonstrate how the algorithm works, a simplified symbolic explanation is now presented.

Let the momentum conservation law be restated as (12):

$$M(u) + A(u) + K(u,T) - P(c) + B(T) = F(T)$$

where:

M	Mass terms from the conservation law
A	Momentum convection terms from the conservation law
u	Unknown velocities
K	Viscous dissipation term from the conservation law
T	Unknown temperature
P	Unknown pressure term
c	Pressure dissipation term
B	Buoyancy term from the conservation law
F	Force balance or momentum driving force

Using similar conventions, let the law of conservation of energy be represented by:

$$N(T)+ D(u,T) + L(T) = G(u,T)$$

where:

N	Capacitance term from the conservation law
T	Unknown temperature
D	Energy convection term from the conservation law
u	Unknown velocities
L	Energy dissipation term from the conservation law
G	Energy balance or thermal driving force

If no mass is generated or accumulated due to kinetics, the matrix form of the conservation equations is:

$$\begin{bmatrix} M & 0 \\ 0 & N \end{bmatrix} * \begin{bmatrix} u \\ T \end{bmatrix} + \begin{bmatrix} A(u)+K(u,T)+cP & B(T) \\ 0 & D(u)+L(T) \end{bmatrix} * \begin{bmatrix} u \\ T \end{bmatrix} = \begin{bmatrix} F(T) \\ G(u,T) \end{bmatrix}$$

By inspection, the matrix form states that the momentum and energy force balances $F(T)$ and $G(u,T)$ are a sum of:

- Density and heat capacity variations due to temperature and velocity fluctuations (terms: M, N, u, and T).
- Body, convective, dissipative, pressure, and viscous force variations due to temperature and velocity fluctuations (terms: A, K, P, B, D, L, u, and T).

Thus, the essential principles for understanding FEA are: *(a)* definition of a fluid system with simplified differentials, *(b)* selection of an element and interpolation method, *(c)* adoption of a residually based solution method with an error tolerance, and *(d)* conversion of the differentials to a linear algebraic statement.

Turbulence Modeling

In CFD, of which FEA is only one approach, there exist several methods for investigating turbulence. Even today, a complete theoretical and mechanistic understanding of the subject eludes engineers, mathematicians, and physicists. The current state of the art in turbulence modeling involves the selection of a representative system of averaging equations that will sufficiently approximate fluid attributes. While it is beyond the scope of this paper to summarize and evaluate all existing turbulence solution methods, two of the most commonly used in free convection problems will be briefly discussed.

Whether or not a system is considered turbulent, all conservation equations are upheld everywhere in the fluid. However, difficulty arises in solving turbulent systems when local velocities, pressures, and temperatures are randomly fluctuating at differing orders of magnitude. Within turbulent flow, three distinct regions are recognized: *(a)* a *laminar sublayer*, or non-turbulent film near a boundary, *(b)* a *transitional flow region* where fluid parcels are erratically exiting and re-entering turbulence and, *(c)* a *turbulent region*, where the rapid momentum and energy fluctuations are fully developed. Solving turbulent flow systems mathematically requires characterization of these fluctuations before treatment by CFD. The best approach to date is "time-smoothing" (13) quantities of interest, whereby assumed average values for conservation terms are bound by probability distributions of acceptable alternative values. Randomly selecting alternative values according to probability causes numerical fluctuations in flux or stress, and thus, simulates nature. However, additional conservation terms must be added to the differential or algebraic statement of the fluid system to balance the incidental forces.

The Prandtl Mixing Length Model

An early attempt at modeling turbulence was submitted by Prandtl (14), who compared eddy currents in fluids with the random motion of gas molecules. His proposal stated that if a turbulent eddy behaved in a similar manner to a free path of a gas, collisions would occur within a characteristic length scale. Furthermore, the length was partially dictated by geometry and material properties. Prandtl's statement for turbulent stresses is shown below:

$$\tau_{ij} \propto -\rho_0 l^2 \left(\frac{\partial u_i}{\partial j} \right)^2$$

where:

τ_{ij}	Total turbulent stress contribution
ρ_0	Reference fluid density
u_i	Velocity in the *i* direction
l	Characteristic length
i	Indicates the horizontal direction
j	Indicates the vertical direction

Prandtl's expression is termed a *zero-equation* model of turbulence in which no additional conservation terms are required for solution. The model calculates and re-uses a value for the turbulent stress while solving the conservation equations, and is accepted as a reasonable starting point for turbulence study. It adds no complexity to the conservation laws, and in free convective situations, may be equivalent to more sophisticated methods.

The κ-ε Turbulence Model

In an effort to improve upon the work of Prandtl, the *two-equation κ-ε* turbulence model proposed by Batchelor (15) has gained acceptance as a general purpose turbulence model within reasonable contexts. The original equation was based upon observations made separately by Kolmogoroff and Prandtl (16). Although CFD scholars have not fully assessed the method, it has gained much popularity for its ability to solve a broad spectrum of turbulent flows. In its most basic form, the model is represented by the following:

$$\tau_{ij} \propto \rho_0 \left(\frac{\kappa^2}{\varepsilon} \right)$$

where:

τ_{ij}	Total turbulent stress contribution
ρ_0	Reference fluid density
κ	Turbulent kinetic energy, estimated from turbulent velocities
ε	Viscous dissipation rate, estimated from turbulent velocities and viscosity data

Since both κ and ε depend upon an instantaneous velocity (and instantaneous density in the case of ε), additional transport terms are added to the conservation equations. Thus, the number of unknowns is expanded by two equations as u_{ij}, p, T, κ, and ε need to be solved *a priori* (hence the term “two-equation turbulence model”). The principal advantage of the method, despite the obvious expense computationally, is the energy and dissipation terms are not fixed through time. In effect, the method is sensitive to energy fluctuation and fluid environment, more closely representing real behavior.

CHAPTER IV

TECHNICAL APPROACH

System Description Using Primitive Variables

This section presents a traditional and effective method for determining influential primitive variables for convective systems. Assuming no previous expertise in convection, the engineer is typically faced with the task of simplifying the conservation laws to a shorter list of quantities that can be modeled or at least easily measured. An analysis technique known as the Buckingham-π method can be performed to better understand how the conservation equations are properly simplified for FEA.

Let the system be described by a set of primitive physical parameters:

$$\text{System} = f(l, g_0, u, \mu, k, \rho, C_p, T, p)$$

where:

l	Characteristic length standard for the system
g_0	Gravitational forces acting upon the system
u	Velocity
μ	Dynamic viscosity
k	Thermal conductivity
ρ	Fluid density
C_p	Heat capacity of the fluid
T	Temperature
p	Pressure

Note that neither the overall heat transfer coefficient, *U*, nor the film heat transfer coefficient, *h*, is listed in the group of terms. Since transfer coefficients are in part defined by surface areas, the terms are not primitive variables.

Of the nine preceding terms, all exist within the four fundamental dimensions of mass, length, time, and temperature. By rule, convection is therefore described by (9 variables – 4 dimensions) = 5 dimensionless factors known as π terms that will describe the bulk system. The five dimensionless factors are derived systematically by units canceling with respect to convective behavior. The objective is to envelop all primitive variables in at least one dimensionless quantity, thus relating all quantifiable physical changes in the system. The result of the analysis is presented in Table 2.

Table 2.

Critical Convective System Variables (Buckingham-π Method)

Equation derivation	Solution	Result	Ratio
$\pi_1 = (\rho)^a \cdot (g_0)^b \cdot (l)^c \cdot (\mu)^d$	a: 2 b: 1 c: 3 d: -2	$N_{Gr} = \frac{\rho^2 g_0 l^3}{\mu^2}$ Grashof number	Buoyancy to viscous forces
$\pi_2 = (C_p)^a \cdot (k)^b \cdot (\mu)^c$	a: 1 b: -1 c: 1	$N_{Pr} = \frac{C_p \mu}{k}$ Prandtl number	Momentum to thermal diffusivity
$\pi_3 = (\mu)^a \cdot (u)^b \cdot (k)^c \cdot (T)^d$	a: 1 b: 2 c: -1 d: -1	$N_{Br} = \frac{\mu u^2}{kT}$ Brinkmann number	Viscous to conductive dissipation
$\pi_4 = (\mu)^a \cdot (u)^b \cdot (l)^c \cdot (\rho)^d$	a: -1 b: 1 c: 1 d: 1	$N_{Re} = \frac{ul\rho}{\mu}$ Reynolds number	Inertial to viscous forces
$\pi_5 = (p)^a \cdot (\rho)^b \cdot (u)^c$	a: 1 b: -1 c: -2	$N_{Eu} = \frac{p}{u^2 \rho}$ Euler number	Pressure to inertial forces

The available dimensionless numbers are reduced by considering system constraints. During Uni-Tank cooling, there is little variation in pressure from atmospheric, thus the Euler number can be disregarded. Velocity values within the fluid domain are low, (~10^{-3} m/s), so dimensionless quantities based upon these values lend little additional information to the analyst. Ruling out the influence of velocity eliminates the Reynolds and Brinkmann numbers as critical for system description. Only the Grashof and Prandtl numbers remain as essential system descriptors. In combination, the Prandtl and Grashof numbers form a single, convenient product known as the Rayleigh number:

$$N_{Ra} = N_{Pr} \cdot N_{Gr} = \frac{C_p \rho^2 g_0 l^3}{\mu k}$$

Thus, a turbulent convective system is characterized by only one dimensionless variable that dictates all the necessary primitives to define the system: *(a)* heat capacity, *(b)* density, *(c)* gravity, *(d)* distance, *(e)* viscosity, and *(f)* thermal conductivity. Temperature is implicitly contained in the density expression, and this relation will be developed in the following section.

When describing heat transfer with the Rayleigh number, it is accepted (17) that values of $N_{Ra} > 1.7 \times 10^3$ indicate at least weak convective forces are present in the system. Values of $N_{Ra} > 1.0 \times 10^9$ indicate that convection is turbulent with little or no conduction, while values of $N_{Ra} < 1.7 \times 10^3$ are purely conductive in nature.

Axi-Symmetric Momentum Conservation

With a dimensionless analysis completed, the conservation equations can be simplified and prepared for solution. In order to set proper direction for the derivations, consider the general shape of a Uni-Tank in the following figure. The system will be defined according to the cylindrical coordinate system.

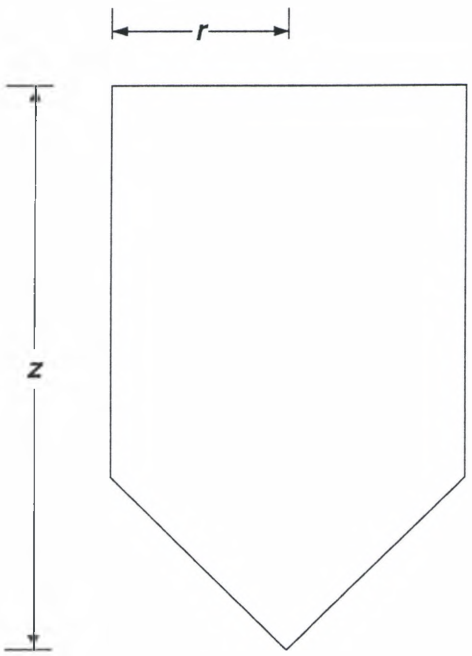


Figure 3. Definition of the Coordinate System

Under cooling circumstances, it is believed that a thin film of rapidly sinking flow is moving near the wall and a slower return flow is rising through the core of the tank. This pattern resembles a vertically stretched toroid, with a majority of fluid motion occurring in the radial and axial directions, but little or no motion in the angular direction, θ. Therefore, an axi-symmetric assumption is imposed, and the conservation equations are simplified to include only the *r* and *z* directions. Gravitational forces will act only in the axial direction, *z*. Referring to the text of Bird, Stewart, and Lightfoot (13), the full conservation of momentum equation in the radial direction is shown below:

$$\rho\left(\frac{\partial u_r}{\partial t} + u_r \frac{\partial u_r}{\partial r} + \frac{u_\theta}{r}\frac{\partial u_r}{\partial \theta} - \frac{u_\theta^2}{r} + u_z \frac{\partial u_r}{\partial z}\right) = -\frac{\partial p}{\partial r} + \mu\left\{\frac{\partial}{\partial r}\left[\frac{1}{r}\frac{\partial}{\partial r}(r u_r)\right] + \frac{1}{r^2}\frac{\partial^2 u_r}{\partial \theta^2} - \frac{2}{r^2}\frac{\partial u_\theta}{\partial \theta} + \frac{\partial^2 u_r}{\partial z^2}\right\} + \rho g_0$$

where:

ρ	Fluid density
u_r	Radial velocity component
u_z	Axial velocity component
u_θ	Angular velocity component
t	Time
r	Radial distance from the tank centerline
p	Pressure
μ	Dynamic viscosity
g_0	Gravitational forces acting upon the system

All terms acting in the θ-direction are canceled, and gravitational forces are also ignored. The basis equation set is reduced to:

$$\rho\left(\frac{\partial u_r}{\partial t} + u_r\frac{\partial u_r}{\partial r} + u_z\frac{\partial u_r}{\partial z}\right) = -\frac{\partial p}{\partial r} + \mu\left\{\frac{\partial}{\partial r}\left[\frac{1}{r}\frac{\partial}{\partial r}\left(ru_r\right)\right] + \frac{\partial^2 u_r}{\partial z^2}\right\}$$

Similarly, the momentum equation in the axial direction is simplified to:

$$\rho\left(\frac{\partial u_z}{\partial t} + u_r\frac{\partial u_z}{\partial r} + u_z\frac{\partial u_z}{\partial z}\right) = -\frac{\partial p}{\partial z} + \mu\left\{\frac{\partial}{\partial r}\left[\frac{1}{r}\frac{\partial}{\partial r}\left(ru_z\right)\right] + \frac{\partial^2 u_z}{\partial z^2}\right\} + \rho g_0$$

with body forces due to gravity included.

Convective Modification

To complete the derivation of transport equations for convective flow, it is necessary to incorporate buoyancy terms into the reduced Navier-Stokes equations. Assuming the Uni-Tank pressure will not change significantly during cooling, density differences due to pressure gradients approach zero and are a function of temperature only. This behavior is characteristic of an incompressible fluid. In such a fluid, a suitable representation of temperature induced density fluctuation is known as the Boussinesq approximation (12) which is represented as:

$$(\rho - \rho_0)g_0 = -\rho_0[\beta_T(T - T_0)]g_0$$

where:

ρ	Instantaneous density
ρ_0	Reference density
g_0	Gravitational force
β_T	Coefficient of thermal expansion
T	Instantaneous temperature
T_0	Reference temperature

The volume expansivity (β_T) is a functional relationship between density and temperature (13) at constant pressure:

$$\beta_T = -\frac{1}{\rho}\left(\frac{\partial \rho}{\partial T}\right)_p$$

Turbulent Modification

To model turbulence at $N_{Ra} > 10^9$, a two-equation κ–ε model will be implemented. Recall that employment of the two-equation turbulence model is based upon an additive/subtractive, fluctuating turbulent stress term. In this work, the model is represented by the following form:

$$\mu = \mu_0 + \mu_t = \mu_0 \pm \rho_0 C_\mu\left(\frac{\kappa^2}{\varepsilon}\right)$$

where:

μ	Dynamic viscosity
μ_t	Turbulent viscosity term to be added to or subtracted from the dynamic viscosity
C_μ	Correlation constant equal to 0.09 (12)
κ	Turbulent kinetic energy
ε	Viscous dissipation rate of turbulent kinetic energy
ρ_0	Reference density
μ_0	Reference dynamic viscosity

After substitution, the complete momentum ensemble describing the turbulent, buoyant behavior of fermenter cooling is described:

r-direction

$$\rho\left(\frac{\partial u_r}{\partial t}+u_r\frac{\partial u_r}{\partial r}+u_z\frac{\partial u_r}{\partial z}\right)=-\frac{\partial p}{\partial r}+\left[\mu_0\pm\rho_0 C_\mu\left(\frac{\kappa^2}{\varepsilon}\right)\right]\left\{\frac{\partial}{\partial r}\left[\frac{1}{r}\frac{\partial}{\partial r}(ru_r)\right]+\frac{\partial^2 u_r}{\partial z^2}\right\}$$

z-direction

$$\rho\left(\frac{\partial u_z}{\partial t}+u_r\frac{\partial u_z}{\partial r}+u_z\frac{\partial u_z}{\partial z}\right)=-\frac{\partial p}{\partial z}+\left[\mu_0\pm\rho_0 C_\mu\left(\frac{\kappa^2}{\varepsilon}\right)\right]\left\{\frac{\partial}{\partial r}\left[\frac{1}{r}\frac{\partial}{\partial r}(ru_z)\right]+\frac{\partial^2 u_z}{\partial z^2}\right\}+\rho_0\beta_T(T-T_0)g_0$$

Axi-Symmetric Energy Conservation

A similar geometric assumption to the preceding momentum conservation derivation produces the energy equation for the axi-symmetric system:

$$\rho C_p\left(\frac{\partial T}{\partial t}+u_r\frac{\partial T}{\partial r}+u_z\frac{\partial T}{\partial z}\right)=k\left[\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial T}{\partial r}\right)+\frac{\partial^2 T}{\partial z^2}\right]+2\mu\left[\left(\frac{\partial u_r}{\partial r}\right)^2+\left(\frac{\partial u_z}{\partial z}\right)^2\right]+\mu\left(\frac{\partial u_z}{\partial r}+\frac{\partial u_r}{\partial z}\right)^2$$

where:

ρ	Fluid density from the momentum conservation equations
u_r	Radial velocity component
u_z	Axial velocity component
C_p	Heat capacity at constant pressure
T	Temperature
r	Radial distance from the tank centerline
k	Thermal conductivity
μ	Dynamic viscosity

Since the instantaneous density is determined in the momentum equations, the system is coupled, and only turbulent term substitution is necessary:

$$\rho C_p\left(\frac{\partial T}{\partial t}+u_r\frac{\partial T}{\partial r}+u_z\frac{\partial T}{\partial z}\right)=k\left[\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial T}{\partial r}\right)+\frac{\partial^2 T}{\partial z^2}\right]+2\left[\mu_0\pm\rho_0 C_\mu\left(\frac{\kappa^2}{\varepsilon}\right)\right]\left[\left(\frac{\partial u_r}{\partial r}\right)^2+\left(\frac{\partial u_z}{\partial z}\right)^2\right]+$$

$$\left[\mu_0\pm\rho_0 C_\mu\left(\frac{\kappa^2}{\varepsilon}\right)\right]\left(\frac{\partial u_z}{\partial r}+\frac{\partial u_r}{\partial z}\right)^2$$

In addition to the κ–ε model for momentum, turbulent transport terms are required for the thermal conductivity and are described by:

$$k = k_0 + k_t = k_0 \pm \frac{C_p\left[\rho_0 C_\mu\left(\frac{\kappa^2}{\varepsilon}\right)\right]}{\sigma_t}$$

where:

k	Dynamic viscosity
k_t	Turbulent thermal conductivity term to be added to or subtracted from the reference thermal conductivity
C_p	Heat capacity
C_μ	Correlation constant equal to 0.09 (12)
σ_t	Correlation constant equal to 0.90 (12)
κ	Turbulent kinetic energy
ε	Viscous dissipation rate of turbulent kinetic energy
k_0	Reference thermal conductivity

Substitution of the above equation results in the full expression for the conservation of energy:

$$\rho C_p\left(\frac{\partial T}{\partial t} + u_r \frac{\partial T}{\partial r} + u_z \frac{\partial T}{\partial z}\right) = \left\{k_0 \pm \frac{C_p\left[\rho_0 C_\mu\left(\frac{\kappa^2}{\varepsilon}\right)\right]}{\sigma_t}\left[\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial T}{\partial r}\right) + \frac{\partial^2 T}{\partial z^2}\right]\right\} +$$

$$2\left[\mu_0 \pm \rho_0 C_\mu\left(\frac{\kappa^2}{\varepsilon}\right)\right]\left[\left(\frac{\partial u_r}{\partial r}\right)^2 + \left(\frac{\partial u_z}{\partial z}\right)^2\right] + \left[\mu_0 \pm \rho_0 C_\mu\left(\frac{\kappa^2}{\varepsilon}\right)\right]\left(\frac{\partial u_z}{\partial r} + \frac{\partial u_r}{\partial z}\right)^2$$

Solution Strategy

Fluid Properties

Engineering data for lager beer was obtained from the Siebel Institute of Technology in Chicago, IL, USA (18). The values of β_T were derived from the density/temperature profile of lager beer using data from the Institute. A first derivative of density with respect to temperature was determined, then multiplied by the inverse of the instantaneous density to give the coefficient of thermal expansion, β_T. Values for beer primitive variables are presented in Table 3.

Table 3.

Physical Constants Used for Uni-Tank Modeling

Physical constant	Temperature Dependence	Value
Dynamic viscosity, μ	Assumed insignificant	2.0E-03 Pa•s
Thermal conductivity, k	Assumed insignificant	5.4E-01 W/m•K
Heat capacity, C_p	Assumed insignificant	4.22E+03 J/Kg•K
Density, ρ	Slight	1.0098E+03 Kg/m^3
Thermal expansion, β_T	Slight	271 K: -1.0E-04 273 K: -6.8E-05 275 K: -3.3E-05 277 K: +2.7E-07 283 K: +1.0E-04

Software

To perform FEA, the program FIDAP is employed. The package name is an acronym for Fluid Dynamics Analysis Package, and is a commercially available software program from Fluent, Incorporated, of Lebanon, NH (USA). The CFD code, comprised of over three million lines, is written entirely in FORTRAN. Contained within FIDAP are several smaller modules that can be used for computer aided design (CAD) of the physical structure (FI-GEN), to enter physical data and a solution method (FI-PREP), to automatically assemble the algebraic matrix and generate the iterative solution (FISOLV), and to post-process the results or generate finished drawings (FIPOST). The selection of FIDAP as the code of choice is based upon product capabilities, provision of training and technical support for the educational institution, reasonable startup cost, and availability on the Microsoft Windows NT platform. In addition, FIDAP has a particular advantage in solving turbulent, convective systems from its strong combustion background. A brief description of FIDAP, including program capabilities, is provided in Appendix B.

Grid Generation and Meshing

When attempting to model a fluid system, one of the earliest considerations is the level of mesh complexity. It is critical to capture the essential structure of the problem, but not at the expense of speed and accuracy. As was learned during the execution of this effort, highly sophisticated meshing schemes with more nodes than necessary gave no more insight into the fluid motion than a faster, simpler nodal layout. No degree of meshing will compensate for a poor choice of boundary conditions, errant physical data, or a general lack of physical understanding of the system. Under these circumstances, modeling efforts will likely fail to meet expectations.

Meshing design for this work is based upon generally accepted fluid behavior in Uni-Tanks and the Buckingham-π results. Since the $N_{Ra} > 10^9$ during cooling, grading the mesh toward regions of turbulence is necessary to comprehend the boundary layer behavior. It is also prudent to study the influence of mesh density and node positioning within the model. For instance, a mesh that is too fine may cause a drastic increase in the computer memory required to arrive at a solution, while a mesh that is too coarse may lead to spurious results or solution divergence. A high concentration of nodes or extremely distorted elements could result in significant roundoff or truncation errors, significantly reducing the calculation efficiency and increasing the risk of failure.

Before CAD and meshing of the tank, dimensional information was obtained from the literature and private communications with the tank manufacturer (7,8,26). Specifications taken from the engineering drawings for both the large and small scale tanks are presented in Table 4. The small tank was rendered using the FI-GEN CAD package offered with FIDAP and meshed as outlined in Figure 4.

Table 4.

Engineering Dimensions Used for Uni-Tank Modeling

Tank	Height	Diameter	Cone Angle	Aspect Ratio	Volume
Large tank, commercial scale	4.87 m	4.64 m	45° to horizontal	1.05	100.1 m^3
Small tank, experimental scale	0.914 m	0.914 m	45° to horizontal	1.00	0.1001 m^3

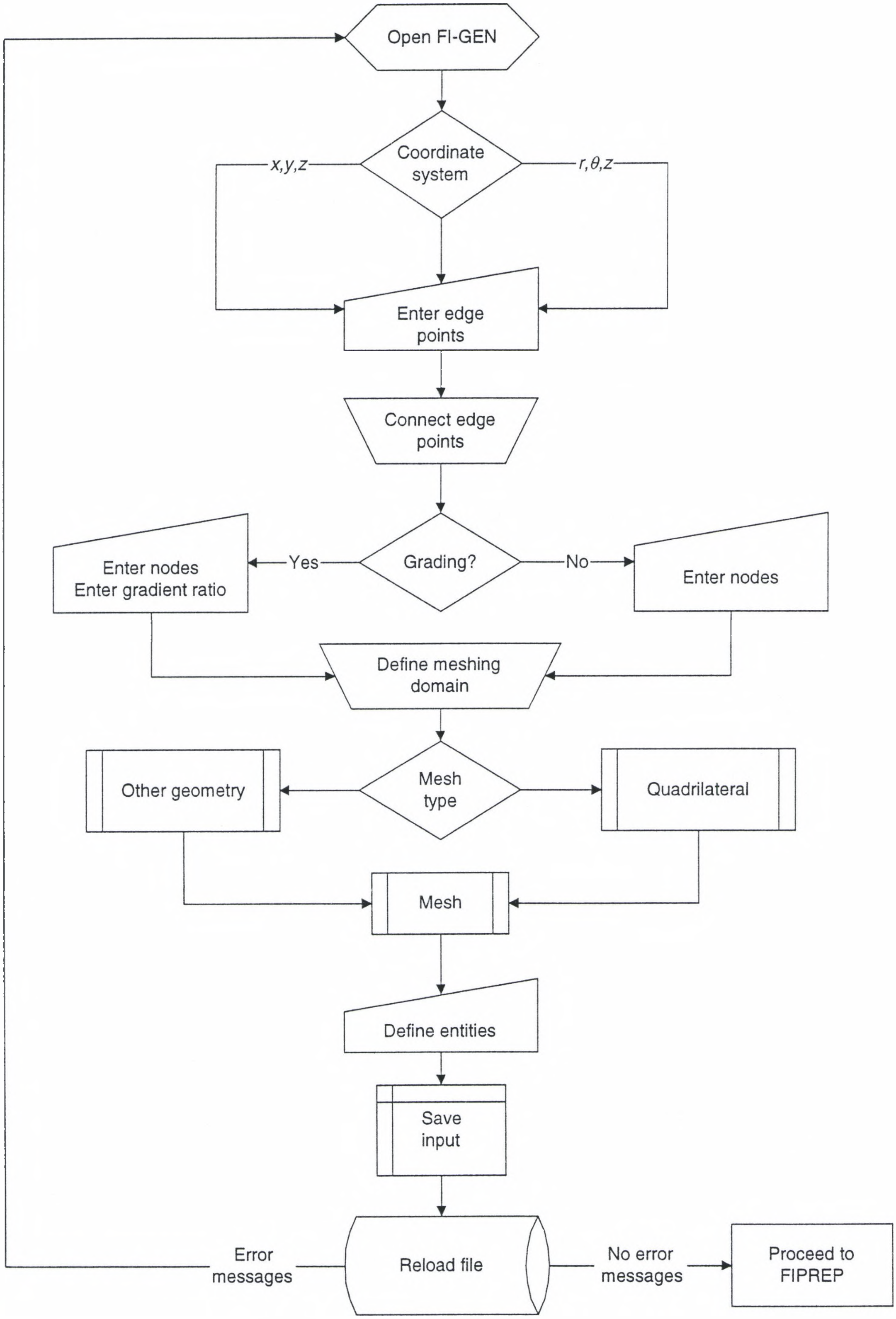


Figure 4. FI-GEN Meshing Flow Diagram

The finished CAD drawing, shown in Figure 5, is taken from the FI-GEN input files presented in Appendix B.

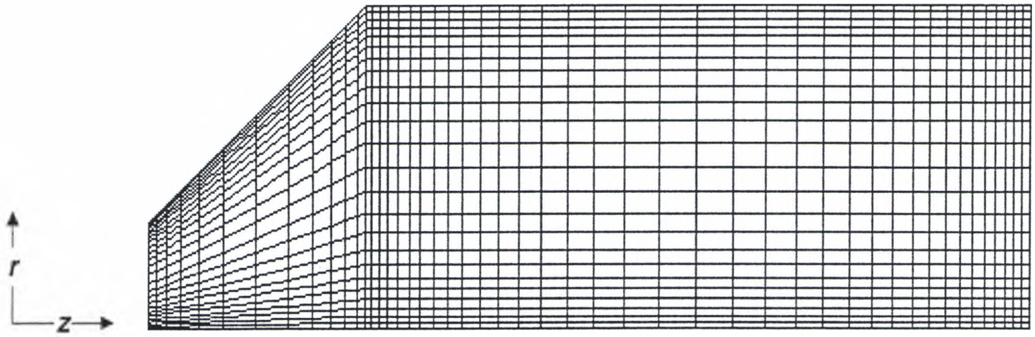


Figure 5. Axi-Symmetric Mesh of Model Tank Drawn in FI-GEN

A 90° right manipulation of the axi-symmetric tank is performed such that the gravity forces act in the –*z* direction. This is a convention dictated by the FIDAP code for proper analysis. The bottom triangular section of the fluid domain is disregarded, as it is assumed to contain sedimented yeast that does not participate in the flow field. The upper boundary of the domain is the fluid level inside the tank. Grading of the mesh toward natural boundaries, turbulent regions, and fluid surfaces within the tank is shown in Figure 5.

Once a suitable mesh is developed, boundary conditions, fluid data, and turbulence models are imposed on the geometry. In FIDAP, this is completed through a procedure termed "entity definition", whereby a set of system characteristics are attached to parts of the CAD drawing. For example, the flow field can be defined as a "fluid" entity to which all of the distinguishing properties of beer are attached. Another type is the "wall" entity, which can be used to assign temperatures and turbulence models to boundaries. Simply stated, entities are parts of the system upon which the fluid and boundary conditions act and interact. All Uni-Tank entities are shown in Figure 6. A dividing line is shown where the conical and cylindrical sections adjoin. Since both regions are defined as "fluid" entities, they will behave as a single, continuous domain.

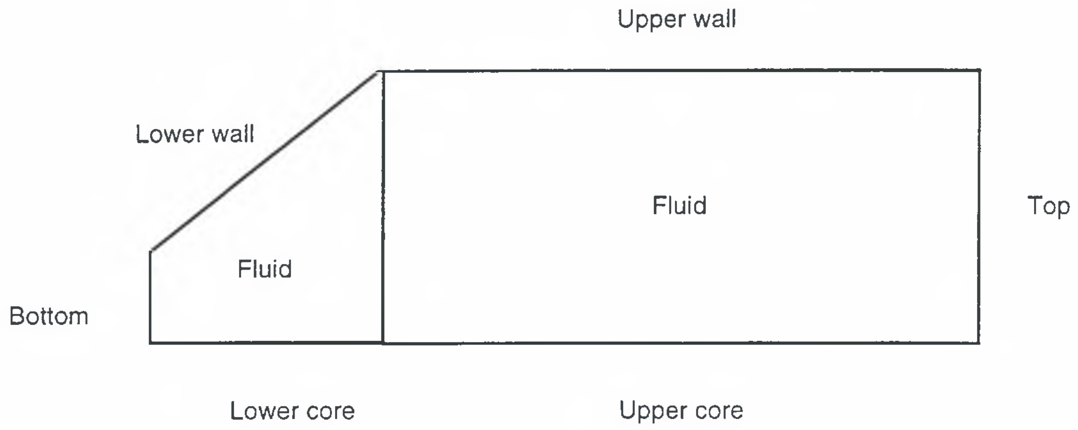


Figure 6. Fluid Entities Definition for Axi-Symmetric Analysis

Initial and Boundary Conditions

After the mesh was drawn and the entities were defined, all of the boundary, initial, physical, and system data listed in Table 3 were assigned as shown in Figures 7 and 8. The FIDAP command structure has been provided in Appendix B.

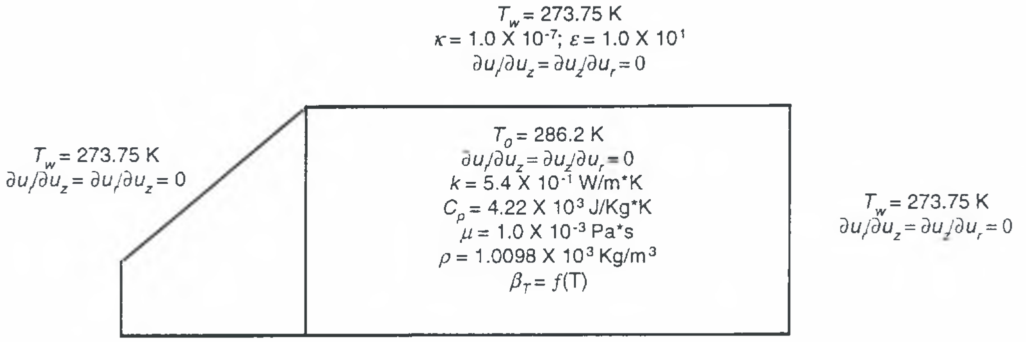

$T_w = 273.75$ K
$\kappa = 1.0 \text{ X } 10^{-7}$; $\varepsilon = 1.0 \text{ X } 10^{1}$
$\partial u_r/\partial u_z = \partial u_z/\partial u_r = 0$
$T_w = 273.75$ K
$\partial u_r/\partial u_z = \partial u_r/\partial u_z = 0$
$T_0 = 286.2$ K
$\partial u_r/\partial u_z = \partial u_z/\partial u_r = 0$
$k = 5.4 \text{ X } 10^{-1}$ W/m*K
$C_p = 4.22 \text{ X } 10^{3}$ J/Kg*K
$\mu = 1.0 \text{ X } 10^{-3}$ Pa*s
$\rho = 1.0098 \text{ X } 10^{3}$ Kg/m^3
$\beta_T = f(\mathrm{T})$
$T_w = 273.75$ K
$\partial u_r/\partial u_z = \partial u_z/\partial u_r = 0$

Figure 7. Initial Conditions Map of Uni-Tank System, $t = 0$

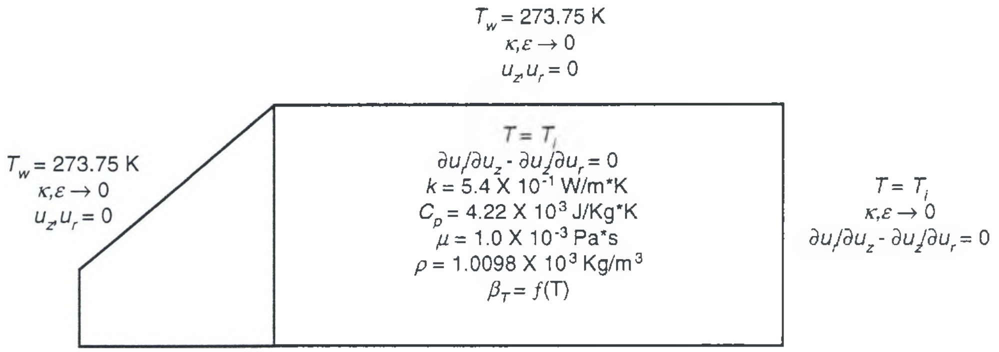

$T_w = 273.75$ K
$\kappa,\varepsilon \rightarrow 0$
$u_z, u_r = 0$
$T_w = 273.75$ K
$\kappa,\varepsilon \rightarrow 0$
$u_z, u_r = 0$
$T = T_i$
$\partial u_r/\partial u_z - \partial u_z/\partial u_r = 0$
$k = 5.4 \text{ X } 10^{-1}$ W/m*K
$C_p = 4.22 \text{ X } 10^{3}$ J/Kg*K
$\mu = 1.0 \text{ X } 10^{-3}$ Pa*s
$\rho = 1.0098 \text{ X } 10^{3}$ Kg/m^3
$\beta_T = f(\mathrm{T})$
$T = T_i$
$\kappa,\varepsilon \rightarrow 0$
$\partial u_r/\partial u_z - \partial u_z/\partial u_r = 0$

Figure 8. Boundary Conditions Map of Uni-Tank System, $t = t_i$

Note the axi-symmetric core ("upper core" and "lower core" entities) has imposed upon it no specific boundary conditions. In this fashion, the algebraic solver treats the surface as a continuum, with the virtual half of the tank assumed to behave as a mirror image of the visible half. The unbound entities on the symmetry line can attain any velocity or temperature.

At t=0, the cooling jackets ("upper wall" entities) are activated and cause a sharp thermal gradient across the wall. As the system continues cooling, the temperature of the wall remains constant, while the axi-symmetric core, fluid, and phase boundary are free to attain any velocity or temperature consistent with the system and convergence conditions. As thermal gradients become less pronounced, the turbulent contribution to the fluid flow is diminished until transitional or laminar flow is present in the vessel ($\kappa,\varepsilon \rightarrow 0$).

Pressure Modeling

Normally, it is assumed that water-like fluids, such as beer, are incompressible fluids. In the study of buoyant flows by CFD, even slight fluctuations in density with changing temperature must receive an accounting in the force balances of the conservation equations. Density changes at a fixed volume generate a pressure force. In convection, these forces tend to be trivial, and CFD practitioners have developed the *penalty function method* to represent small pressure changes that occur in Boussinesq fluids. The penalty function approximates pressure as follows:

$$p = -\lambda\left(\frac{\partial u}{\partial x}\right)$$

where:

p	Pressure
λ	Penalty constant, empirically derived
u	Velocity
x	Position in the fluid domain

In low velocity systems where the density changes are small, an analogy can be drawn between an incompressible fluid and incompressible elastic solid, making the penalty function, λ, analogous to a volume stress, or bulk modulus of elasticity (19). Convective velocities for the Uni-Tank system are on the order of

10^{-3} m/s, while pressure gradients within Boussinesq fluids can be six to nine orders of magnitude smaller. Solving large algebraic systems containing trace pressure values may trigger severe instability in the matrices during Gaussian elimination. By relating constitutively to velocity and position, all pressure terms in the conservation equations can be replaced by a velocity differential and a constant. To avoid ill-conditioning during FEA, the penalty technique is applied during solution of the momentum conservation laws, increasing computational efficiency during iteration with no appreciable loss in accuracy.

Experimental Details

Fluid Modeling Technique

Since the physical properties of beer closely resemble water, the heat capacity, thermal conductivity, and viscosity are assumed constant over the temperature range of interest. Beer density is calculated using the Boussinesq approximation, the result of which is plotted in Figure 9.

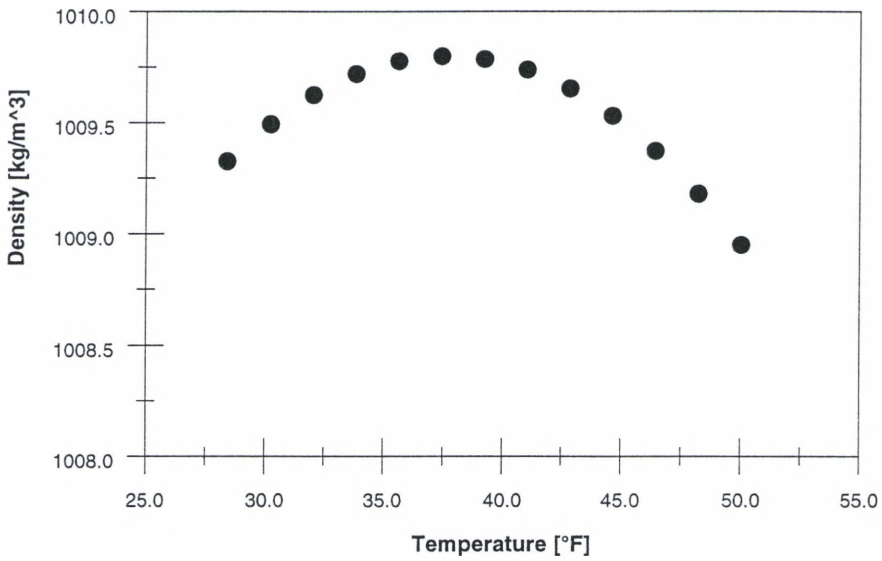


Figure 9. Lager Beer Density as a Function of Temperature

To calculate beer density during iteration, only the boundary and maximum expansion coefficients (β_T) are required by the solver. Densities for any given temperature are interpolated from the critical data by FIDAP. Additionally, the solver will be programmed to treat beer as an incompressible Newtonian fluid in a single-phase (as the CO_2 content of the fluid is < 3% by weight).

Turbulence Assumptions

Since bulk beer convection is characterized by the Rayleigh number, a plot of N_{Ra} as a function of temperature can be used to estimate when turbulence modeling is necessary. From Figure 10, it is shown that a turbulence expression is required for the early and late stages of cooling, while capability of solving transitional behavior will be utilized near the inversion temperature.

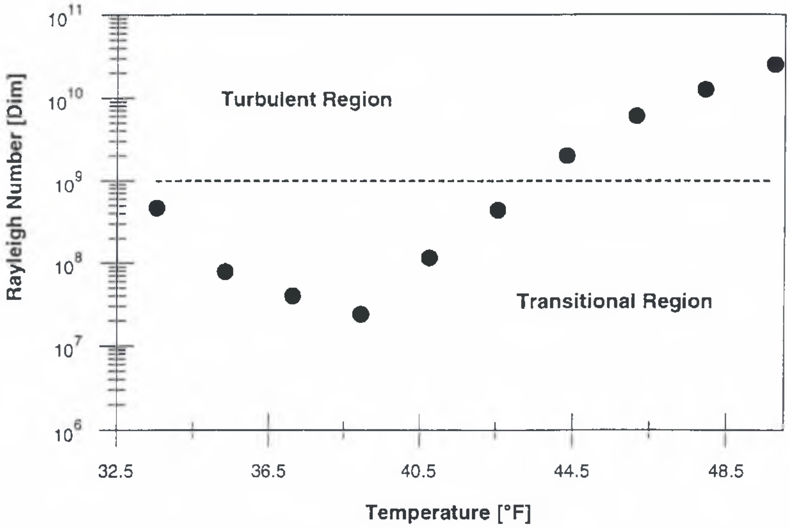


Figure 10. Rayleigh Number as a Function of Temperature

When using a two-equation κ–ε model in FIDAP, a lower bound known as a "clipping constant" can be assigned to deactivate the model as necessary. As the flow becomes transitional, values of κ and ε will become exceedingly small and there exists some risk of destabilizing the solution. A clipping procedure accomplishes stability by holding κ and ε constant at values small enough to deactivate the turbulence model, but large enough to avoid destabilization. Values of κ and $\varepsilon < 10^{-7}$ are considered non-turbulent, and are rounded to 10^{-7}.

Boundary Layer Modeling

When analyzing disturbed boundary flows, a significant consideration is how and where to position the first row of nodes away from the wall (12). The objective is to set the edge of the mesh within the turbulent boundary layer, but not deep enough to be inside the laminar sublayer. The reasoning behind this restriction is that two-equation models are not adept at resolving laminar flow. Such a situation results in unproductive over-iteration. If nodes are properly placed in the boundary layer, an additional near-wall model can effectively solve the laminar sublayer region from the wall to the first row of nodes. The challenge is determining where the two models will meet in the fluid domain.

Computational practitioners have developed a quantitative tool to aid their node placement efforts, as:

$$y^{+} = \frac{y\sqrt{\rho\tau_{ij}}}{\mu}$$

where:

y^{+}	Distance in the boundary layer region
y	Distance from the wall
ρ	Fluid density
τ_{ij}	Turbulent shear stress
μ	Fluid viscosity

In general, y^{+} values of 30-100 empirically indicate the mesh is appropriately established in the turbulent stream. Adjusting the mesh will affect distance calculation, giving the experimenter direct feedback for refinement. Typically, a short CFD run is performed and then the y^{+} value is checked. After mesh adjustment, the procedure is repeated until the y^{+} value is within aforementioned range.

Laminar Sublayer Modeling

Calculation of no-slip laminar flow is completed by fitting “special elements” between physical boundaries and the first row of nodes placed in the turbulent stream. Near-wall flows are resolved by assuming a local equilibrium state of motion in parallel to the wall. This resembles no-slip Couette flow where a fluid moves between a stationary plate (the wall) and one moving plate (the turbulent field). Using the Couette assumption as a basis, the mean properties of the laminar sublayer can be calculated as follows:

$$u^{+} = \frac{(u - u_{w})}{u\,*}$$

$$T^{+} = \frac{(T_{w} - T\,)u\,*\,\rho C_{p}}{q_{w}}$$

$$C_{p}{}^{+} = \frac{(C_{pw} - C_{p})u\,*\,\rho}{m_{w}}$$

where:

u^{+}	Mean velocity across the laminar sublayer
u	Fluid velocity
u_{w}	Fluid velocity at the wall (ideally $u_{w} = 0$)
T^{+}	Mean temperature across the laminar sublayer
C_{p}^{+}	Mean heat capacity across the laminar sublayer
ρ	Fluid density
C_{p}	Fluid heat capacity
C_{pw}	Fluid heat capacity at the wall
q_{w}	Flux across the wall (calculated by FIDAP)
m_{w}	Mass flux across the sublayer (calculated by FIDAP)
u^{*}	Frictional field velocity, defined by:

$$u^{*} = \sqrt{\frac{\tau_{ij}}{\rho}}$$

with ρ and τ_{ij} as defined in the y^{+} calculations.

Surface Roughness

Commercial Uni-Tanks are fabricated of stainless steel. Beer is mildly acidic, with a typical *p*H range of 3-6 depending upon the grain bill and water source used for the production of wort. The preferred metal surface finish is grade 2B, which is defined as a variance of less than ±0.13 μm (5 X 10^{-6} inches) between a surface peak or valley from the mean surface height (20). The significance of this grade of finish is twofold: *(a)* the small surface allows for an ideally smooth computational assumption at the wall, and *(b)* the wall surface will not retain yeast cells or beer pathogens, which have particle diameters of > 3-5 μm and >0.45 μm, respectively (21). Thus, no special accommodation for surface roughness is made during computation since the material surface characteristics do not interfere with boundary layer flow patterns.

Calculation Stability

Turbulent FEA calculations tend to be naturally unstable. Besides averaging *via* two-equation methods, another way to dampen radical fluctuations in transient flows is by retaining some historical information from prior iterations. This is accomplished by a procedure called *relaxation*, which acts to mitigate fluctuating numerical values during calculation. A general scheme for relaxation is as follows:

$$\text{New value} = \text{Calculated value } (1-\omega) + \text{Previous value } (\omega)$$

where ω is a predetermined constant of relaxation with a value less than unity. From the preceding equation, it is shown that relaxation techniques influence calculations by carrying previous results into subsequent calculations. Although this occurs at some expense of calculation speed, the stabilizing aspects of the method are virtually mandatory for modeling transient behavior properly. This is especially true for the early stages of turbulent analysis where the flow is in disarray. At these times, carryover as high as 70% (ω=0.70) is needed to keep the solver from diverging. As the flow organizes and the N_{Ra} drops below 10^9, carryover values no higher than 5% (ω=0.05) are needed to keep iterations stable. FIDAP can be set to automatically adjust the value of ω *in situ* by examining convergence tendencies from previous iterations and assigning a new, "best guess" value.

Time Stepping Algorithm

In transient systems, the matrix assembly that contains the conservation equations, unknown variables, physical data, and numerical stabilization techniques is proceeding through "virtual" time. Movement through this dimension requires a means of handling time-based rates of change in the system.

Assume the quantity *Q* represents the entire matrix statement of the conservation laws. Since time rates of change can be represented by differentials, let the definition of the time derivative be:

$$Q'(t) = lim_{\Delta t \longrightarrow 0} \frac{[Q(t+\Delta t)+Q(t)]}{\Delta t}$$

where:

Q'(t)	Estimated matrix value Q at the future time *t*
Q	Matrix value Q at the current time
t	Time
Δt	Change in time

Let the change in time (Δt) be sufficiently small, such that:

$$Q'(t) \cong \frac{[Q(t+\Delta t)+Q(t)]}{\Delta t}$$

Then, solving for the quantity $Q(t+\Delta t)$ will give an estimate of changes in *Q* at the next time step ($t+\Delta t$). The above derivation is known as Euler's Method, which rapidly computes time derivatives. FIDAP manages the accuracy and expediency of Euler's Method by adjusting time step sizes as residuals and error tolerances allow.

Expected Analysis Outputs

Time Normalization

The purpose of experimental modeling and CFD is to reasonably explain a very large system with a very small or virtual one. A problem one encounters when modeling is how to relate time based data among several scales. The best way to align data of this nature is through a process of normalization. Specifically, the conversion of time to dimensionless form will eliminate the impact of vessel proportions upon the data. This results in the engineer being able to directly overlay and compare experimental results visually.

Brandon's (8) 1/10th scale Uni-Tank results proved this comparison was successful for any two geometrically consistent systems. For example, the scale of the commercial tank was ten times that of his dimensional model, and the commercial process took ten times as long to reach the same temperature objective. A direct comparison of Brandon's results with the FEA output can be made by transposing temperature/time data to temperature/dimensionless time quantities, according to the following conversion:

Dimensionless time (ϕ) = Elapsed time / Total time

Grid Resolution

For this work, the meshed system is grid resolved when: *(a)* values of y^+ between 30 to 100 are upheld in the boundary layer, indicating proper placement of the first row of nodes, *(b)* the algebraic system converges reproducibly to the same solution, despite further diffusion or concentration of the mesh, and *(c)* increasing the total number of nodes gives no further solution accuracy. Demonstrating grid resolution and reproducibility of the mesh is the single best way to establish credibility for the model and its results.

Comparison With Real Systems

CFD output should agree with experimental data collected from a real system of same design. Post-processing and visual inspection of the flow fields should be performed to ensure that the FEA results are not coincidental artifacts.

CHAPTER V

RESULTS

Uni-Tank Cooling

Figure 11 compares the large and small scale experimental data of Brandon (8) with the results obtained from two-equation (κ–ε) FEA. The commercial and model tank volumes were 100 m^3 and 0.1 m^3, respectively. The CFD results for this work were collected using the model tank dimensions of Brandon. Geometric specifications are listed in Table 4 for both tanks. Initial and boundary conditions for the model are given in Figures 7 and 8.

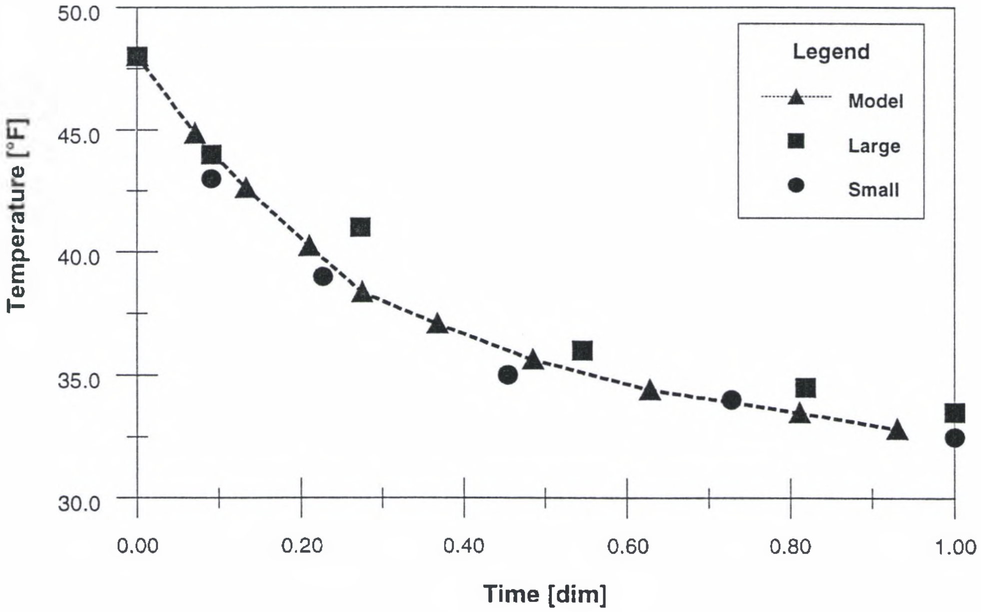


Figure 11. Temperature/Time Data for the Uni-Tank Cooling Cycle (κ-ε)

A departure analysis of the temperature/time data for both the scaled and virtual tank was completed after FEA. Comparison of the data sets with each other and with the commercial tank gave the following conclusions: *(a)* the 1/10th scale data (8) was within 4.0% of the commercial Uni-Tank, and *(b)* the FEA results obtained were within 2% of small scale data, and within 3% of the large Uni-Tank results.

Figure 11 reflects this agreement. It is interesting to note that the FEA results qualitatively fit between the large and small scale data on the plot. The CFD model is constructed from drawings of the small tank. Additional streamline function, temperature, vector, and velocity profile results are presented in Appendix C for review and consideration. The graphics were generated by the FIDAP post-processor (FIPOST), and visual inspection of the output resembles previous work by Ishiguro (9) and Knudsen (5). Particular attention is directed to streamline and vector plots in Appendix C at times $\phi = 0.19$ and 0.25, which demonstrate the ability of FIDAP to simulate a density inversion during the beer cooling cycle.

After the two-equation approach proved satisfactory, the experiment was repeated using the Prandtl mixing length model. The results are presented in Figure 12.

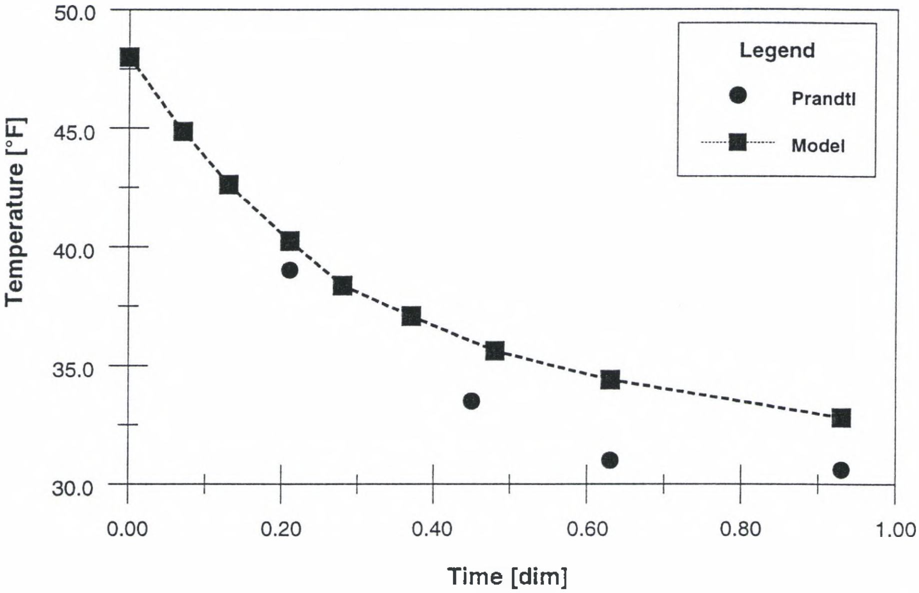


Figure 12. Temperature/Time Data for the Uni-Tank Cooling Cycle (Prandtl)

From the preceding figure, it is shown that Prandtl's model adequately represents the turbulent system during the early stages of cooling. This is demonstrated by a direct overlay of both sets of cooling data from ϕ=0.00-0.15. When ϕ>0.20, the $N_{Ra} < 10^9$ and the Prandtl model apparently does not recognize the transitional flow, resulting in an over cooling of the fluid.

Figure 13 summarizes the performance of the two-equation (κ–ε) and zero-equation (Prandtl) turbulence models. Values for the temperature dependent Rayleigh number are also added to indicate where the turbulent models should be activated and deactivated. The turbulent/transitional flow boundary is indicated by a solid line at $N_{Ra} = 10^9$. From the figure, a relationship between Prandtl model performance and turbulence boundaries is evident.

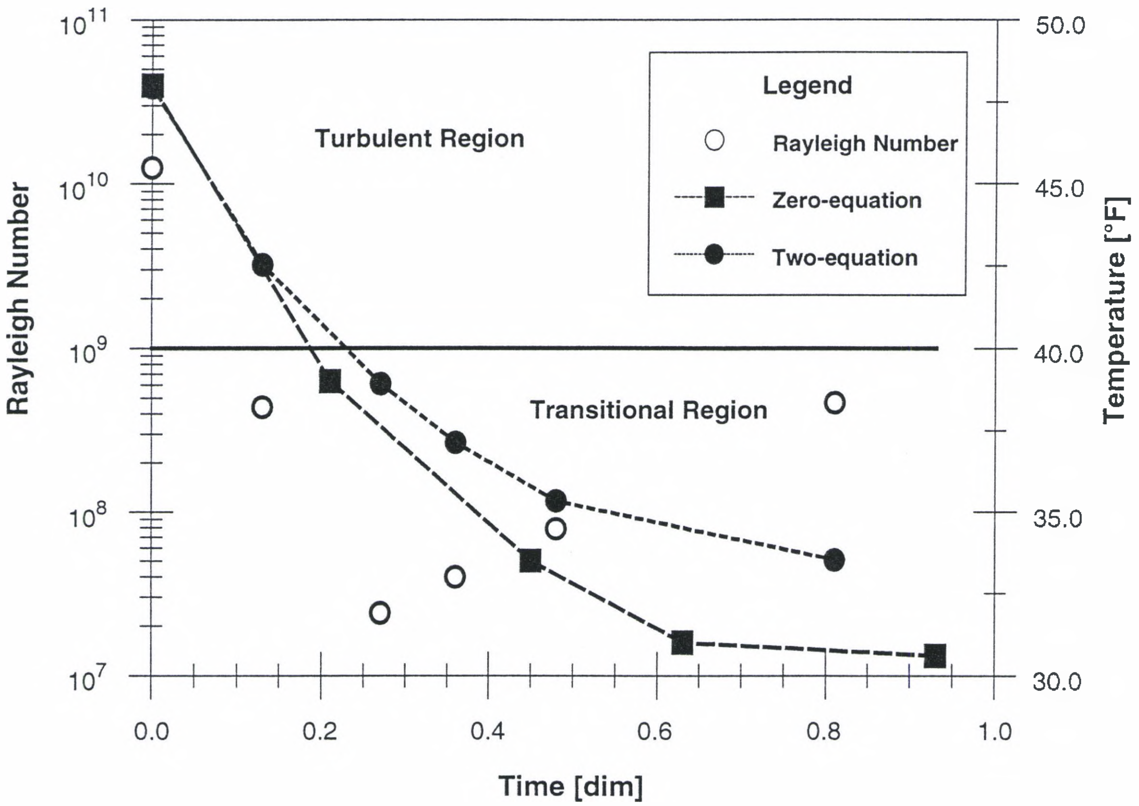


Figure 13. Rayleigh Number Effect Upon Turbulence Model Performance

Further trials did not resolve the undesirable tendencies of the mixing length method. Although the Prandtl model predicts an errant temperature profile, it does adequately simulate the density inversion and the qualitative motion of the fluid. With some *in situ* characteristic length adjustment, the Prandtl technique could be a potential method of solution for this engineering problem.

Grid Convergence

While testing for grid convergence, the objective was to maximize calculation stability and minimize computation time while using the coarsest mesh possible.

A review of Figure 5 shows the mesh is graded toward walls and expected areas of turbulence for reasons of efficient mesh coverage. In FIDAP, mesh gradients are specified by ratios that indicate the fractional distance from one node to the next relative to a fixed point. For instance, a gradient ratio of 5:1 means that the first node is placed at one-fifth the distance from a point of origin to the boundary edge. Then, the next node is placed at one-fifth the distance from the first node to the edge of the boundary, and so on until the boundary is reached and the required number of nodes are used.

For the virtual tank, mesh grading was completed by bisecting the axi-symmetric drawing with respect to both length and width, resulting in two arbitrary centerlines for each section. Recall that the fluid entity is divided into upper and lower regions. For the upper region, a 5:1 gradient ratio was directed axially from the bisection line to both the top and bottom of the region. Similarly, a 4:1 gradient ratio from the centerline to both the wall and the line of axial symmetry was completed. For the lower conical portion, both the radial and axial directions were graded in a 4:1 ratio from the centerline.

After the required number of nodes was situated with adequate grading, FEA proceeded smoothly and reproducibly at y^+ values between 2-15. The dimensionless boundary layer values are summarized as a function of time in

Figure 14. The low predicted values for y^+ indicated the turbulent boundary layer in the Uni-Tank was exceptionally thin.

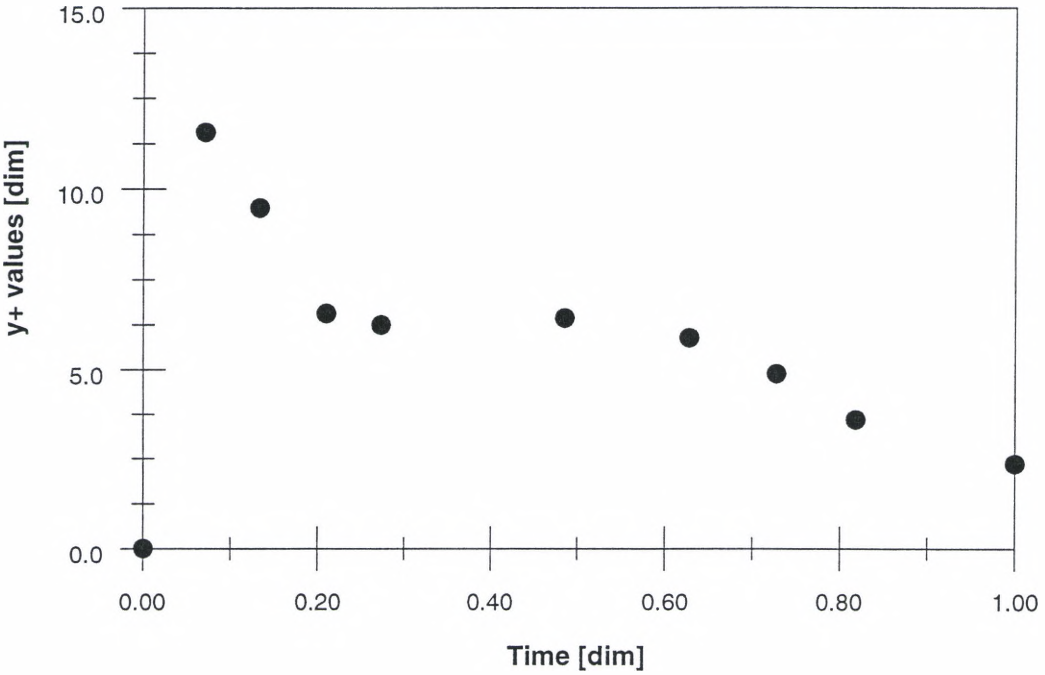


Figure 14. Predicted y^+/time Data for the Uni-Tank Cooling Cycle

In practice, undesirable flow behavior was observed when $y^+ > 25$. What typically occurred was the fluid remained stationary until $\phi \sim 0.10$, then a sudden burst of fluid motion set forth rapid tank recirculation. The action was similar to that of a viscoelastic fluid with a yield stress. Interestingly, this early behavior did not prevent the model from recovering and converging to the correct final temperature, and provided an example of coincidental success. The phenomenon was eliminated by concentrating nodes toward the wall.

Lastly, after the difference between the actual and computed temperature was below 10%, further concentration of the mesh at a fixed gradient ratio was continued at intervals of 5-10% until the solution did not change. It was learned that a mesh constructed of 1152 nodes was required to converge reproducibly. The successful mesh was comprised of 48 nodes in the axial (*z*) and 24 nodes in the radial (*r*) direction. In Figure 15 is presented a comparison of the final mesh configuration (indicated with a line) with a sub-optimal refinement effort. The inadequate mesh was composed of 864 nodes, as a 36 by 24 system. Gradient ratios were the same for both cases. Further refinements of the successful mesh, such as a 54 by 24 system (1296 nodes) or a 54 by 36 system (1944 nodes) were not plotted since the results were redundancies of the 48 by 24 system.

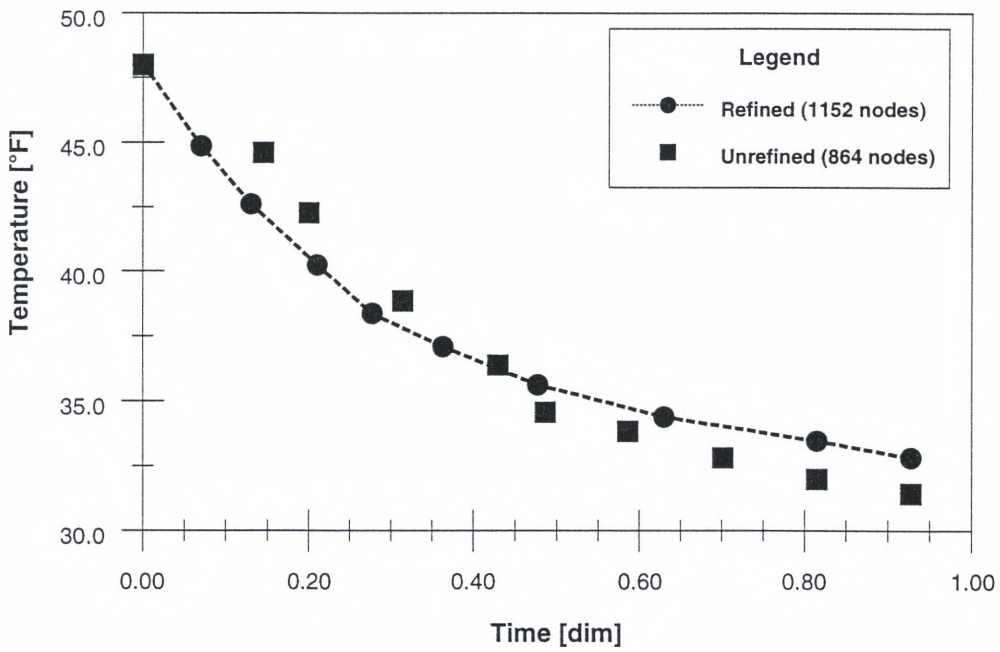


Figure 15. Mesh Refinement Effect Upon FEA Performance

Time Stepping Methodology

A semi-log plot of the time stepping scheme that produces a satisfactory solution is shown in Figure 16. Raw time step data generated by FIDAP is collected in Appendix C. When a time step is too large, the solver automatically halves the step and repeats the iteration. This is reflected by “sawtooth” configurations in some plots located in the appendix.

It is easy to bracket the proper time stepping sequence using FIDAP. If the time stepping is too coarse, the solver diverges quickly and produces error messages due to excessive truncation. When time steps are too fine, roundoff errors predominate and the solver will not converge. When time steps are within FIDAPs acceptable range, the solver smoothly self-adjusts according to the tolerance value and generally converges reproducibly.

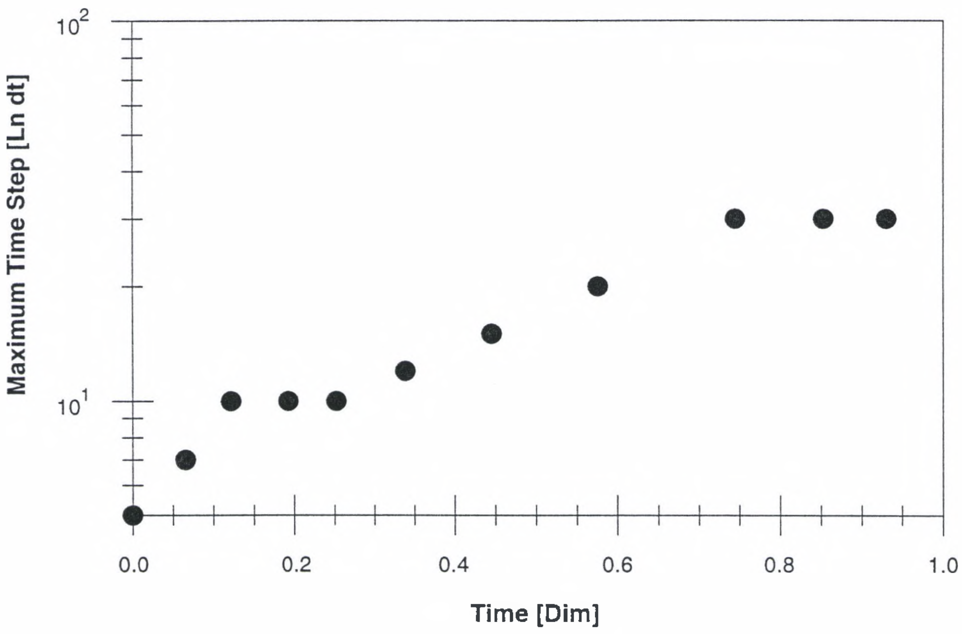


Figure 16. Time Stepping Methodology

Starting at t=0, the first time step taken is 0.005 seconds. The time step size is increased in multiples no greater than 1.001(Δt)-1.01(Δt) of the previous iteration until the turbulence diminishes. As the system enters transitional flow, time steps of 1.25(Δt) are taken until an upper limit of 15 seconds per time step is reached. Time steps > 20 seconds tend to cause a significant amount of time step halving, which makes experimental runs longer. Reproducible divergence is obtained when fixed time step experiments are run, demonstrating the mandatory use of flexible time stepping techniques.

Mean Temperature Measurement

Figure 17 shows the result of a departure analysis from the mean temperature for the tank cooling cycle. The highlighted points in the figure represent a positive or negative departure from the mean no greater than 0.52 °F, and are the largest single cluster of nodes detected by FEA exhibiting this property. Although no "perfect" point to measure the mean temperature exists, the indicated region is the closest possible location to an ideal. With the exception of the wall boundary layer where thermal gradients are pronounced, the majority of the tank is within 1-2 °F of the mean temperature. A summary of the node numbers highlighted in the figure and the average departure per node (in °F) over the entire cooling cycle is provided in Appendix D.

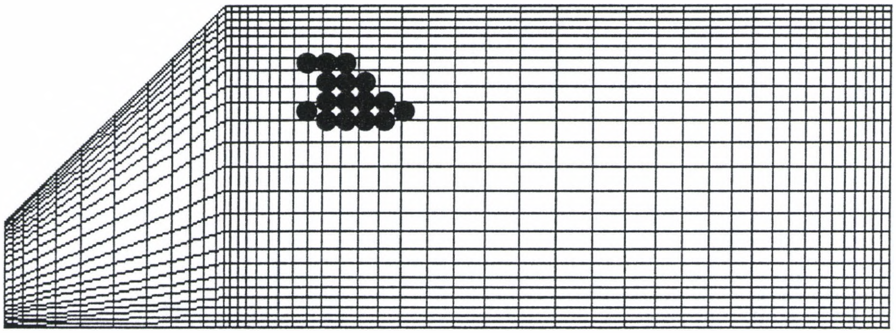


Figure 17. Elements with Lowest Departure from Mean Temperature

From Brandon (8), it is suggested that a temperature probe be located at ~25% of the distance of the tank straight side, originating at the junction of the cone and the cylinder. FEA shows that positioning a thermoprobe in this fashion yields a mean departure of ~0.55 °F for the entire unit operation, including the stagnant inversion phase. The significance of this conclusion is that experimentally based recommendations regarding single point temperature measurement are valid for this geometry. In addition, sampling within this volume of consistent thermal and velocity behavior diminishes any errors in probe placement and will not adversely affect the measurement of the mean temperature.

Fluid Velocity During Uni-Tank Cooling

Fluid velocities are relatively low in free convection, and this system is no exception. At ~40 °F, the onset of transitional flow slows the velocity field dramatically. This indicates why cooling beer through a density inversion is problematic to brewers and engineers. The onset of inversion and change to transitional flow occur simultaneously, without any recourse to mechanically mix the stagnating fluid. Referring to Figure 18, the turbulent portion of the cooling cycle is complete within the first 20% of the total time necessary to complete the operation (ϕ = 0.00-0.20). Any notable mixing beyond this point is probably achieved by conduction, density inversion, and weak buoyancy effects.

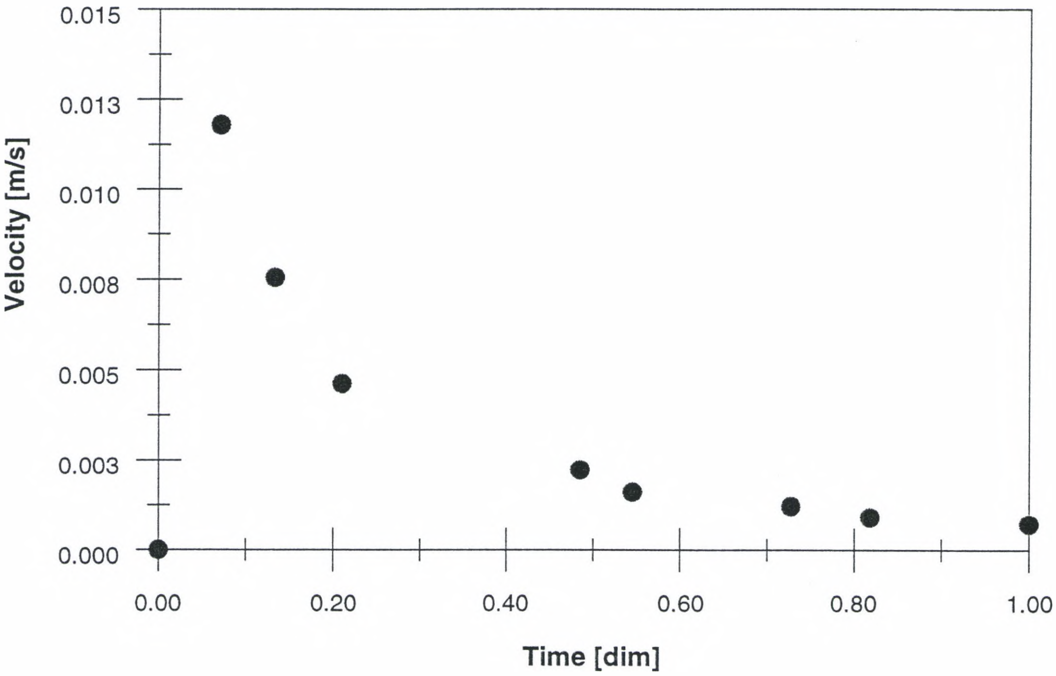


Figure 18. Velocity/Time Data for the Uni-Tank Cooling Cycle

CHAPTER VI

SUMMARY

Finite element analysis of a Uni-Tank cooling cycle was performed and compared with field data presented in the literature. The quantitative temperature/time and qualitative flow results were in agreement with experiment and provided validation for two-equation turbulence modeling of transient free convective flow. In addition, a single-point measurement position mentioned in the literature was substantiated by finite element analysis. Approximation and stability theory used to arrive at an acceptable solution was presented to the reader for employment and guidance for new problems and modeling efforts.

CHAPTER VII

SUGGESTED FUTURE WORK

Below are suggested tasks for analysis to build upon the results of this study:

- **Rechecking the Prandtl model by *in situ* adjustment of the assumed mixing length.** In the turbulent regime, the performance of Prandtl's method is equivalent to the two-equation method, but does not perform well in transitional flow. Though the $\kappa\text{–}\varepsilon$ method works for modeling small Uni-Tanks, the amount of CPU resources required to solve commercial scale vessels is beyond the capabilities of current PC platforms. Prandtl's model is advantageous since the total amount of transport equations solved is reduced, freeing up memory resources for larger nodal systems.
- **Alternate geometries.** Although most Uni-Tanks are constructed at an aspect ratio of unity, there exist many other configurations. Ratios of twice unity are common in the United States, while in Europe Uni-Tanks can reach aspect ratios of five times unity. Some brewers even operate at ratios below unity in shallow, nearly flat coned Uni-Tanks. Flow fields under these physical circumstances may be quite different than those reported herein.

- **Parametric study of primitive variables.** Though all of the primitive variables with the exception of density are assumed constant in the range of interest, this is not truly the case. Some simple effect testing would be desirable to develop a truly robust general FEA method for Uni-Tanks.
- **Cooling jacket configuration.** One of the great outstanding questions in Uni-Tank design is the proper location and size of cooling jackets. Although any credible estimation of proper design is beyond the scope of this work, one needs only to review the flow fields in Appendix C to learn where the mixing, and thus the opportunity for optimization, is the best. The data suggests that before inversion, cooling jackets should at least be activated at the top of fluid surface and the junction of the cone and cylinder. During inversion, cooling the entire cylinder would best remove heat from the beer. After inversion, the top and bottom jackets would again direct flow properly.
- **Cone angle and cone cooling.** Another interesting endeavor would be the study of mixing effects from cone angles and cone cooling. This may be one way to intensify the turbulent flow field and achieve a faster rate of tank turnover. As the yeast settles into the cone, a low level of biological kinetics is still occurring, which gradually heats the cone to a few degrees warmer than the fluid. Cone cooling may be a way to enhance the momentum of the flow field.

REFERENCES

1. Knudsen, F. B., *MBAA Technical Quarterly*, **15**(3), pp. 132-139 (1978).

2. Shardlow, P.J., Thompson, C.C., *The Brewer's Digest*, p. 76-80, August 1971.

3. Bishop, L. R., *Journal of the Institute of Brewing*, **44**, pp. 69-73, (1938).

4. Delente, J., Akin, C., Krabbe, E., Ladenburg, K., *MBAA Technical Quarterly*, **5**(4), pp. 228-234 (1968).

5. Knudsen, F.B., Larson, J.W., *MBAA Technical Quarterly*, **11**(1), pp. 1-8 (1974).

6. Larson, J.W., Brandon, H.J., *MBAA Technical Quarterly*, **25**(2), pp. 41-46 (1988).

7. Brandon, H.J., *private communications*, The Washington University at Saint Louis, August 12, 1996.

8. Reuther, H., Brandon, H., Raasch, J., Raabe, D., *Monatsschrift für Brauwissenschaft*, **9/10**, pp. 310-317 (1995).

9. Ishiguro, T., Mizutani, S., Kuwahara, K., *MBAA Technical Quarterly*, **34**(3), pp. 164-170 (1997). Also see: Ishiguro, T., Mizutani, S., Kuwahara, K., "Numerical Analysis of Cooling Mechanisms in Cylindroconical Lager Tanks", *From the Annual Meeting of the Master Brewers Association of the Americas*, 1995.

10. Clough, R.W., "The Finite Element Method in Plane Stress Analysis", *Proceedings of the Second Conference on Electronic Computation*, ASCE, 1960.

11. Courant, R., *Bulletin of the American Mathematical Society*, **49** (1943).

12. *FIDAP 7.0 Theory Manual*, Fluent, Inc., Lebanon, NH, USA (1993).

13. Bird, R.B., Stewart, W.E., Lightfoot, E.N., *Transport Phenomena*, John Wiley & Sons, New York NY (1960), pp. 153-160.

14. Schlicting, H. Boundary Layer Theory, McGraw-Hill, New York, NY, USA (1955), Ch. XIX.

15. Batchelor, G.K., *The Theory of Homogeneous Turbulence*, Cambridge University Press, Cambridge, U.K. (1953).

16. Churchill, S.W., *Industrial and Engineering Chemistry Research*, **35**, pp. 3122-3140 (1996).

17. Kakaç, S., Shah, R.K., Aung, W., *Handbook of Single Phase Convective Heat Transfer*, John Wiley & Sons, New York, NY, USA (1987), pp. 108-119.

18. Larson, J. W., *private communications*, The Siebel Institute of Technology, January 27, 1998.

19. Heinrich, J.C., Marshall, R.S., *Computers and Fluids*, **9**, pp. 73-83 (1981).

20. Allison, W.F., *private communications*, Latrobe Brewing Company, February-March 1999; *See also* J&L Specialty Steel, Incorporated, Technical Data, http://members.aol.com/mckoolline/techdata.htm.

21. Gruss, J., Ecker, M., *private communications*, Latrobe Brewing Company, March 1999.

22. Noonan, G.J., *New Brewing Lager Beer*, Brewers Publications, Boulder, CO, USA (1986).

23. Daniels, R., *Designing Great Beers*, Brewers Publications, Boulder, CO, USA (1996).

24. McCabe, J., ed., *The Practical Brewer*, 3rd edition, Master Brewers Association of the Americas, Wauwatosa, WI, USA (1999).

25. De Clerck, J., *A Textbook of Brewing*, Chapman & Hall Ltd., London, UK (1957).

26. Kelly, C., *private communications*, ENERFAB Corporation, March 1998.

APPENDIX A

BREWING TECHNOLOGY

Beer Style

Of the many distinct styles, all beers can be classified under one of two general categories: a lager beer, which is fermented at temperatures ranging from 45-65 °F, or an ale, which is fermented at temperatures ranging from 60-75 °F. For this work, the physical data of a lager beer are used.

Beyond the preceding classifications, a large selection of ingredients and methods to prepare and age beer is known to brewers. Details of all methods and styles are well beyond the scope of this paper. Suggested readings are found in references 22-25.

The Brewing Process

A simple transformation diagram of a typical brewing operation is presented in Figure 19.

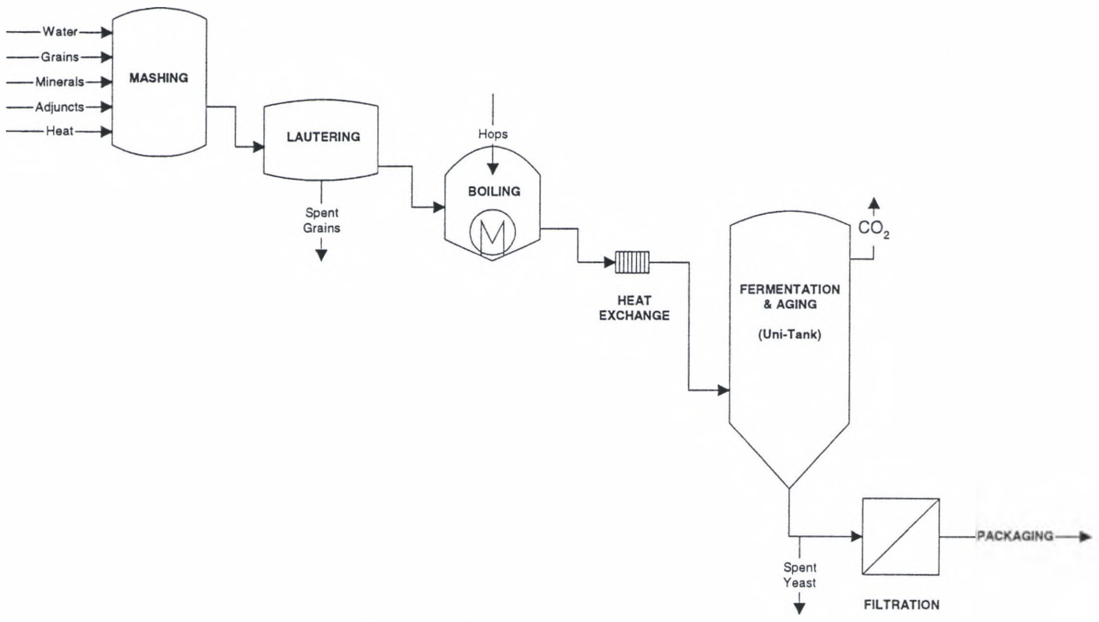


Figure 19. A Typical Brewing Process

Before processing, the entire system is cleaned and sanitized to eliminate microorganisms. At the starting point of the process, barley malt and possibly other grains (adjuncts) are suspended in heated water in a process called “mashing”. Control of the aqueous slurry temperature activates malt enzymes that will convert native starches to fermentable sugars. After filtration of the grain hulls by “lautering”, hops can be added, and the whole mixture is brought to a continuous boil. The liquid which results is formally called “wort”. After boiling is completed, an optional standing period can be completed to settle out fine particles in the wort. Afterward, the wort is rapidly chilled by passage through a heat exchanger. During cooling, yeast and trace amounts of oxygen are mixed with the sweetened solution. Fermentation will initiate and continue for the next five to ten days. After fermentation, the spent yeast is removed and the beer is chilled, then aged at a fixed temperature in an anaerobic atmosphere. After the beer reaches maturity, it can be filtered, or “brightened”, before being served to a customer or sent to a commercial packing line.

APPENDIX B

FIDAP

Capability Description

FIDAP is a commercially available fluid flow modeling and simulation program that can be run on all computer operating systems. The command structure is comprised of over three million lines of FORTRAN code, with the objective of providing a sophisticated, reliable, and simple to use implementation of FEA for the end user.

FIDAP is an integrated package that includes all functions necessary to draw, mesh, and process most fluid dynamics problems. Semantically, the modules are set up as follows:

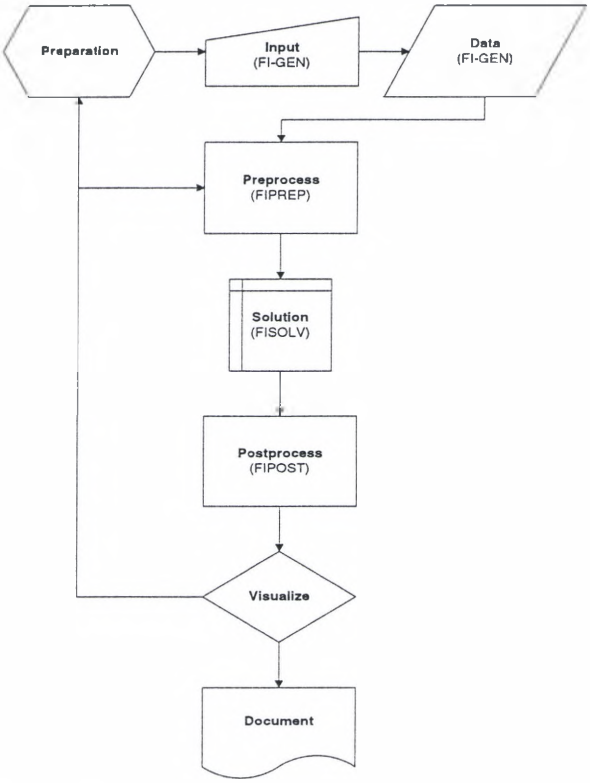


Figure 20. FIDAP Program Modular Structure

Problem Classes Addressed by FIDAP

The capabilities of FIDAP are far ranging, rendering it particularly useful for the research environment as well as industry. The program has been applied successfully in the areas of electronics, automotive design, metallurgy, HVAC, polymers, chemical kinetics, biomedicine, aerospace, mechanical engineering, heat exchange, and crystallization, to name a few. General classes of fluid flow problems that can be solved using FIDAP are summarized in Table 5.

Table 5.

Chart of FIDAP Capabilities

Fluid Situation	FIDAP Capabilities
Compressibility	• Incompressible, with or without Boussinesq approximation (pressure recovery models available, i.e., pressure, discretized) • Compressible, with ideal gas law or user defined equation of state
Flow	• Laminar • Turbulent, including the two-equation models of Prandtl, $\kappa-\varepsilon$, Wilcox κ-Ω, RNG $\kappa-\varepsilon$, anisotropic $\kappa-\varepsilon$; includes several eddy viscosity models and near wall modeling for viscous sublayer • Free surface • Periodic flows • Swirling flows • Moving surface fronts • Surface tension
Phase	• Single • Two phase flows, with complete mass, momentum and energy transfer between phases • Solidification with heat transfer • Melting with heat transfer
Viscosity	• Newtonian • Non-Newtonian, with all common models such as power law and Bingham included; user defined models are also accommodated • Visco-elastic
Media	• Porous flow, with Darcy's Law and updates
Transport	• Mass, both heterogeneous and homogeneous, with or without reaction kinetics • Momentum • Energy • Advection/Diffusion • Free and forced convection • Buoyancy • Conduction • Radiation, diffuse grey and non-grey body models

Time dependence	• Steady state or transient
Body forces	• Gravity • Centrifugal • Coriolis • Electromagnetic
Coordinate system	• Cartesian • Cylindrical • Spherical • Rotating frame
Boundary conditions	• Velocity • Temperature • Species concentrations • Kinetic energy and dissipation (turbulence) • Stresses • Fluxes • Constitutive properties • Constant or dependent variants of the above
Initial conditions	• Constitutive relations • Velocity • Temperature • Species
Solution techniques	• Successive substitution • Newton-Raphson • Modified Newton • Quasi-Newton • Segregated solver, particularly useful for large problems
Time integration	• Backward Euler • Forward Euler • Trapezoid Rule

Computer Requirements and Software Expenditures

The minimum requirements to run FIDAP 7.62, the version used for this work, are summarized in Table 5 for an IBM-compatible PC.

Table 6.

Requirements and Expenditures for FIDAP (as of 10/97)

PC Component	Requirement	Actual	Cost ($US)
Personal Computer	100% IBM compatible PC	Gateway, 100% IBM compatible	3800
Processor	Intel 80486	Intel Pentium II w	--
CPU	100 MHz	300 MHz	--
RAM	32 MB	128 MB	--
File System	NTFS	NTFS	--
Hard Disk Space	150 MB	6.4 GB	--
Other Drives	3.5" Floppy, CD-ROM	3.5" Floppy, CD-ROM	--
Mouse	Microsoft compatible	Microsoft compatible	--
Graphics Driver	Windows NT compatible	Windows NT compatible	--
Monitor Resolution	1024 X 768 pixels	1024 X 768 pixels	--
Monitor	15" Viewable	19" Viewable	--
Network Card	Ethernet 10BaseT	Ethernet 10BaseT	35
Software	• Windows NT, 3.51 • Hummingbird Exceed 5.0 • FIDAP 7.62, one year license • HPGL graphics converter	• Windows NT, 4.0 • Hummingbird Exceed 6.0 • FIDAP 7.62, one year license • Hijaak 4.5 Pro	145 500 2000 100
TOTAL COST			6580

Run times and Input Files

Below are the actual input file(s) used to complete this work. All runs were completed on a Windows NT version 4.0 platform with a Pentium II processor running at 300 MHz.

Input File

```
FI-GEN( ELEM = 1, POIN = 1, CURV = 1, SURF = 1, NODE = 0, MEDG = 1, MLOO = 1,
MFAC = 1, BEDG = 1, SPAV = 1, MSHE = 1, MSOL = 1, COOR = 1 )
WINDOW(CHANGE= 1, MATRIX )
   1.000000    .000000    .000000    .000000
    .000000   1.000000    .000000    .000000
    .000000    .000000   1.000000    .000000
    .000000    .000000    .000000   1.000000
  -10.00000   10.00000   -7.50000    7.50000   -7.50000    7.50000
WINDOW( CHAN = 1, MATR )
   1,    0,    0,    0
   0,    1,    0,    0
   0,    0,    1,    0
   0,    0,    0,    1
  -10,    10,  -7.5,   7.5,  -7.5,   7.5
PGRID( ON )
PGRID( SNAP )
POINT( ADD, COOR, WIND = 1 )

/Small tank model

/r:         0.4572 m (18")
/h:         0.9144 m (36")
/h':        0.4572 m (18")
/cone 45°
/h/D:       1.00

/ORIGINAL DIMENSIONS
/0,0,0
/0.45,0,0
/1.37,0,0
/1.37,0.45,0
/0.45,0.45,0
/0,0.01,0

/FILLED BOTTOM DIMENSIONS
0,0,0
0.30,0,0
1.22,0,0
1.22,0.45,0
0.30,0.45,0
0,0.15,0


POINT( SELE, ID, WIND = 1 )
   1
   2
CURVE( ADD )
POINT( SELE, ID, WIND = 1 )
   2
   3
CURVE( ADD )
POINT( SELE, ID, WIND = 1 )
   3
   4
CURVE( ADD )
POINT( SELE, ID, WIND = 1 )
   4
   5
CURVE( ADD )
POINT( SELE, ID, WIND = 1 )
   5
   2
CURVE( ADD )
POINT( SELE, ID, WIND = 1 )
   5
   6
CURVE( ADD )
POINT( SELE, ID, WIND = 1 )
   6
   1
CURVE( ADD )
CURVE( SELE, ID, WIND = 1 )
   4

MEDGE( ADD, LSTF, INTE = 36, RATI = 5, 2RAT = 5, PCEN = 0.5 )
CURVE( SELE, ID, WIND = 1 )
   2
MEDGE( ADD, LSTF, INTE = 36, RATI = 5, 2RAT = 5, PCEN = 0.5 )
CURVE( SELE, ID, WIND = 1 )
   1

MEDGE( ADD, LSTF, INTE = 12, RATI = 4, 2RAT = 4, PCEN = 0.5 )
CURVE( SELE, ID, WIND = 1 )
   6
MEDGE( ADD, LSTF, INTE = 12, RATI = 4, 2RAT = 4, PCEN = 0.5 )
CURVE( SELE, ID, WIND = 1 )
   5
MEDGE( ADD, LSTF, INTE = 24, RATI = 4, 2RAT = 4, PCEN = 0.5 )
/Small window
WINDOW(CHANGE= 1, MATRIX )
   1.000000    .000000    .000000    .000000
    .000000   1.000000    .000000    .000000
    .000000    .000000   1.000000    .000000
    .000000    .000000    .000000   1.000000
    -.03900    1.59900    -.38575     .84275   -1.63800    1.63800
  45.000000  45.000000  45.000000  45.000000
CURVE( SELE, ID, WIND = 1 )
   3
MEDGE( ADD, LSTF, INTE = 24, RATI = 4, 2RAT = 4, PCEN = 0.5 )
CURVE( SELE, ID, WIND = 1 )
   7
MEDGE( ADD, LSTF, INTE = 24, RATI = 4, 2RAT = 4, PCEN = 0.5 )
CURVE( SELE, ID, WIND = 1 )
   4
   3
   2
   5
MFACE( WIRE, EDG1 = 1, EDG2 = 1, EDG3 = 1, EDG4 = 1 )
CURVE( SELE, ID, WIND = 1 )
   6
   5
   1
   7
MFACE( WIRE, EDG1 = 1, EDG2 = 1, EDG3 = 1, EDG4 = 1 )
MFACE( SELE, ID, WIND = 1 )
   1
MFACE( MESH, MAP, ENTI = "fluid" )
MFACE( SELE, ID, WIND = 1 )
   2
```

```
MFACE( MESH, MAP, ENTI = "fluid" )
MEDGE( SELE, ID, WIND = 1 )
 7
ELEMENT( SETD, EDGE, NODE = 2 )
MEDGE( MESH, MAP, ENTI = "bottom" )
MEDGE( SELE, ID, WIND = 1 )
 4
MEDGE( MESH, MAP, ENTI = "lowerwall" )
MEDGE( SELE, ID, WIND = 1 )
 1
MEDGE( MESH, MAP, ENTI = "upperwall" )
MEDGE( SELE, ID, WIND = 1 )
 6
MEDGE( MESH, MAP, ENTI = "top" )
MEDGE( SELE, ID, WIND = 1 )
 2
MEDGE( MESH, MAP, ENTI = "uppercore" )
MEDGE( SELE, ID, WIND = 1 )
 3
MEDGE( MESH, MAP, ENTI = "lowercore" )
END( )

FIPREP()
PROBLEM(INCOMPRESSIBLE,TRANSIENT,NONLINEAR,NEWTONIAN,AXI-
SYMMETRIC,MOMENTUM,BUOYANCY,FIXED,TURBULENT,SINGLEPHASE)
PRESSURE(PENALTY)
SOLUTION(S.S.=15)
RELAXATION(RESIDUAL,MAXIMUM)
0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7
RELAXATION(MINIMUM)
0.05 0.05 0.05 0.05 0.05 0.05 0.05 0.05 0.05 0.05
EXTRAPOLATE(ON,AFTER=5,EVERY=5,ORDER=2,NOKE)

/run1
/TIMEINTEGRATION(BACKWARD,NSTEPS=101,TSTART=0,TEND=43200,DT=0.005,VARIABLE=0.0001,DTMAX=1,IN
CMAX=1.01)
/EXECUTION(NEWJOB)

/run2
/TIMEINTEGRATION(BACKWARD,NSTEPS=301,TSTART=0.274,TEND=43200,DT=0,VARIABLE=0.0001,DTMAX=5,IN
CMAX=1.01)
/TIMEINTEGRATION(BACKWARD,NSTEPS=301,TSTART=6.505,TEND=43200,DT=0,VARIABLE=0.0001,DTMAX=5,IN
CMAX=1.01)
/TIMEINTEGRATION(BACKWARD,NSTEPS=301,TSTART=128.6,TEND=43200,DT=0,VARIABLE=0.0001,DTMAX=5,IN
CMAX=1.01)
/TIMEINTEGRATION(BACKWARD,NSTEPS=301,TSTART=1315,TEND=43200,DT=0,VARIABLE=0.0001,DTMAX=5,IN
CMAX=1.01)
/TIMEINTEGRATION(BACKWARD,NSTEPS=401,TSTART=2820,TEND=43200,DT=0,VARIABLE=0.0001,DTMAX=7,IN
CMAX=1.001)
/TIMEINTEGRATION(BACKWARD,NSTEPS=401,TSTART=5268,TEND=43200,DT=0,VARIABLE=0.0001,DTMAX=10,I
NCMAX=1.001)
/TIMEINTEGRATION(BACKWARD,NSTEPS=401,TSTART=8325,TEND=43200,DT=0,VARIABLE=0.0001,DTMAX=10,I
NCMAX=1.001)
/TIMEINTEGRATION(BACKWARD,NSTEPS=401,TSTART=10890,TEND=43200,DT=0,VARIABLE=0.0001,DTMAX=10,I
NCMAX=1.001)
/TIMEINTEGRATION(BACKWARD,NSTEPS=401,TSTART=14610,TEND=43200,DT=0,VARIABLE=0.0001,DTMAX=12,I
NCMAX=1.001)
/TIMEINTEGRATION(BACKWARD,NSTEPS=401,TSTART=19230,TEND=43200,DT=0,VARIABLE=0.0001,DTMAX=15,I
NCMAX=1.001)
/TIMEINTEGRATION(BACKWARD,NSTEPS=401,TSTART=24890,TEND=43200,DT=0,VARIABLE=0.0001,DTMAX=20,I
NCMAX=1.001)
/TIMEINTEGRATION(BACKWARD,NSTEPS=401,TSTART=32150,TEND=43200,DT=0,VARIABLE=0.0001,DTMAX=30,I
NCMAX=1.001)
/TIMEINTEGRATION(BACKWARD,NSTEPS=401,TSTART=36870,TEND=43200,DT=0,VARIABLE=0.0001,DTMAX=30,I
NCMAX=1.001)
TIMEINTEGRATION(BACKWARD,NSTEPS=401,TSTART=40540,TEND=43200,DT=0,VARIABLE=0.0001,DTMAX=30,I
NCMAX=1.001)
EXECUTION(RESTART)

DATAPRINT(CONTROL)
```

```
POSTPROCESS(NBLOCKS=1)
0,36000,10

PRINTOUT(NBLOCKS=1)
0,36000,10

RENUMBER(PROFILE)

ENTITY(FLUID, NAME = "fluid")
ENTITY(WALL, NAME = "upperwall")
ENTITY(WALL, NAME = "top")
ENTITY(WALL, NAME = "bottom")
ENTITY(WALL, NAME = "lowerwall")
ENTITY(PLOT, NAME = "uppercore")
ENTITY(PLOT, NAME = "lowercore")

ICNODE(KINETIC,CONSTANT=1.0E-07,ENTITY = "upperwall")
ICNODE(DISSIPATION,CONSTANT=1.0E+01,ENTITY = "upperwall")
ICNODE(TEMPERATURE,CONSTANT = 282,ENTITY = "fluid" )

BCNODE(TEMPERATURE,CONSTANT = 272.75,ENTITY = "upperwall")
BCNODE(VELOCITY,ZERO, ENTITY = "upperwall")
BCNODE(VELOCITY,ZERO, ENTITY = "top")
BCNODE(VELOCITY,ZERO, ENTITY = "bottom")
BCNODE(VELOCITY,ZERO, ENTITY = "lowerwall")

GRAVITY(MAGN=9.81, THETA=-90)

/Water thermal conductivity:  0.588 W/m*K
CONDUCTIVITY(CONSTANT=0.54)

/Water specific heat: 4.2E+3 J/kg*K
SPECIFICHEAT(CONSTANT=4.22E+3)

/Water dynamic viscosity: 0.002 Pa*s
/VISCOSITY(CONSTANT=0.002,MIXLENGTH)
VISCOSITY(CONSTANT=0.001,TWO-EQUATION,CLIP=1.0E+5)
TURBOPTIONS(STANDARD)

DENSITY(CONSTANT=1009.8,TYP2,TEMPERATURE)
VOLUMEXPANSION(CURVE=7,TEMPERATURE, REFTEMP=276.29)
-1000, 271, 273, 275, 277, 283, 1000, -1.0E-4, -1.0E-4, -6.8E-5, -3.3E-5, 2.7E-7, 1E-4, 1E-4
END()
CREATE (FISO)
/DO NOT PUT AN END STATEMENT HERE OR THE PROGRAM WILL TERMINATE
```

Technical Support

All FIDAP licenses include the limited services of a technical support engineer. This service assists the understanding and implementation of program logic, command structure, computer compatibility, and program capability. Technical support is available through electronic messaging at univ@fluent.com or *via* telephone at (847) 491-0200.

APPENDIX C

GRAPHICAL OUTPUT

Graphical output was obtained from the FIDAP post-processor FIPOST. The output, given in HPGL (Hewlett- Packard Graphics Language) format, was converted to EMF (Enhanced MetaFile) format for insertion into this body of work. This format exhibited import superiority *via* testing of the following graphical formats: BMP (Windows Bitmap), MDD (MicroGraphix Design/Draw), JPG (JPEG), HPGL (Hewlett- Packard Graphics Language), WMF (Windows MetaFile), EPS (Encapsulated Post Script), PCT (MacIntosh PICT), TIF (Tagged Image File Format), TGA (Targa), GIF (CompuServe Graphic Image Format), and CGM (Computer Graphics Metafile). All images were rendered using the software program Hijaak 4.5 Pro.

Using the FIPOST postprocessor module, the following commands were issued to the processing engine to give the plots in this Appendix:

```
device(HPGL)
color(off)
contour(speed, nosymbols)
contour(streamline, nosymbols)
contour(temperature, nosymbols)
vector(velocity)
step(increment)
```

Velocity Field Plots

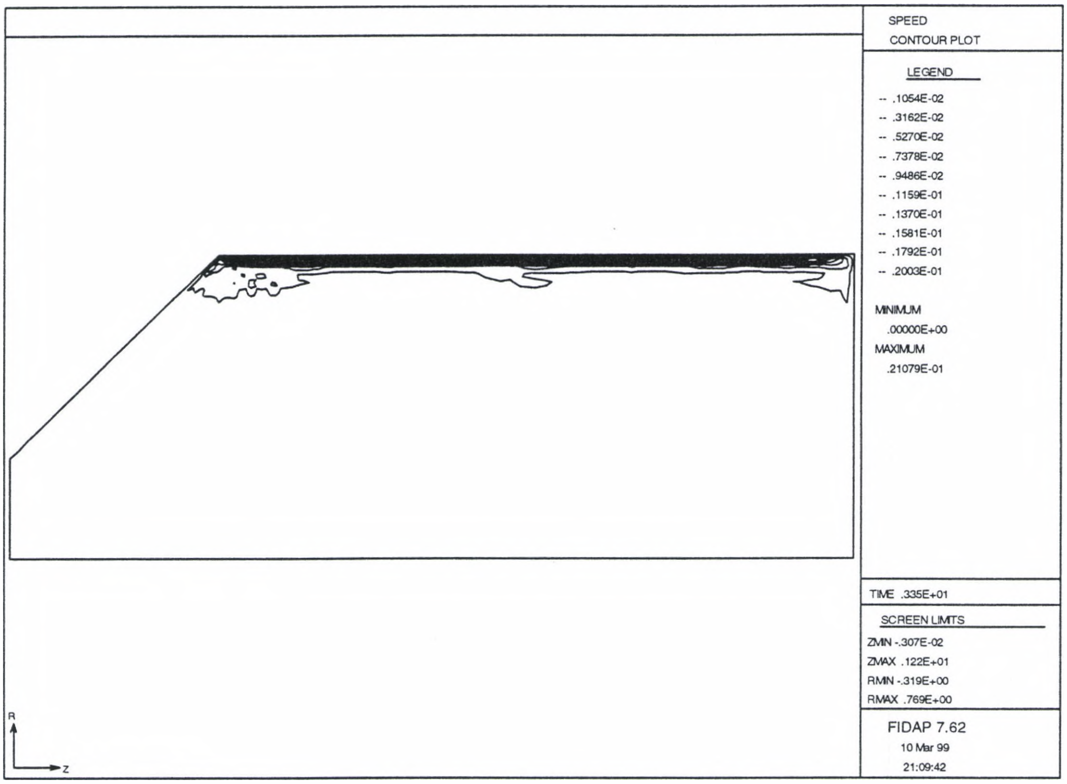


Figure 21. Velocity Field Plot, ϕ: 0.07

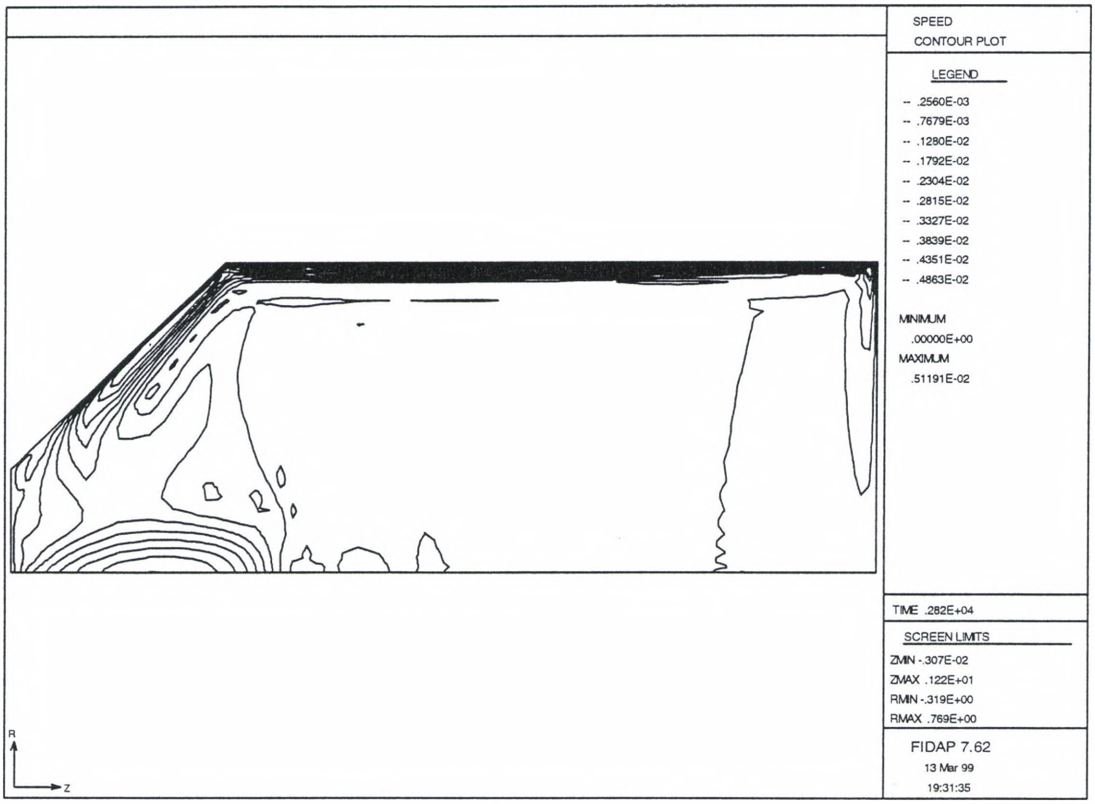


Figure 22. Velocity Field Plot, ϕ: 0.12

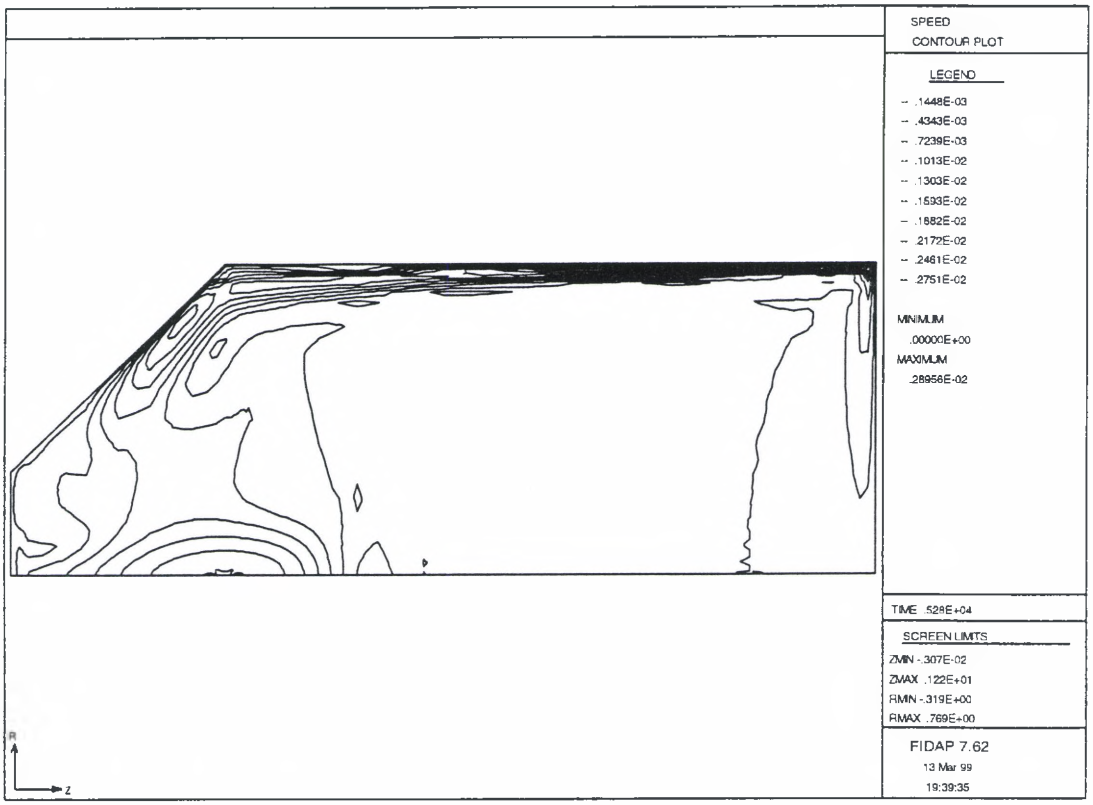


Figure 23. Velocity Field Plot, ϕ: 0.19

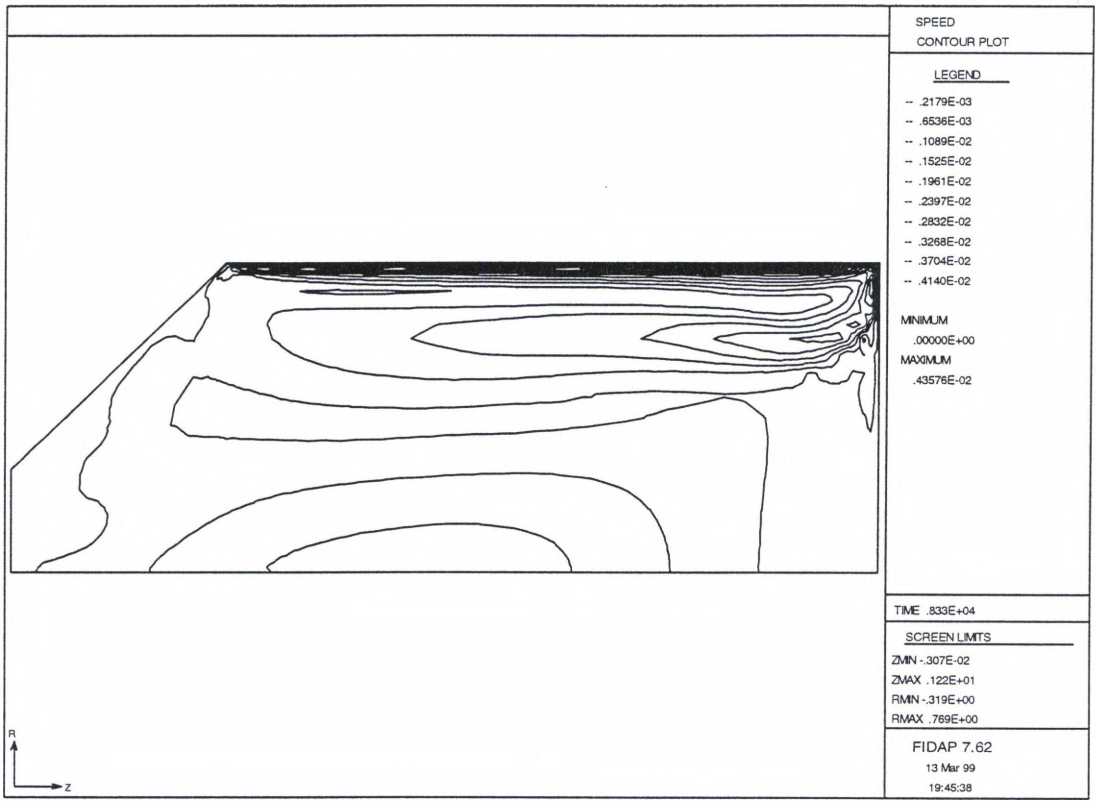


Figure 24. Velocity Field Plot, ϕ: 0.25

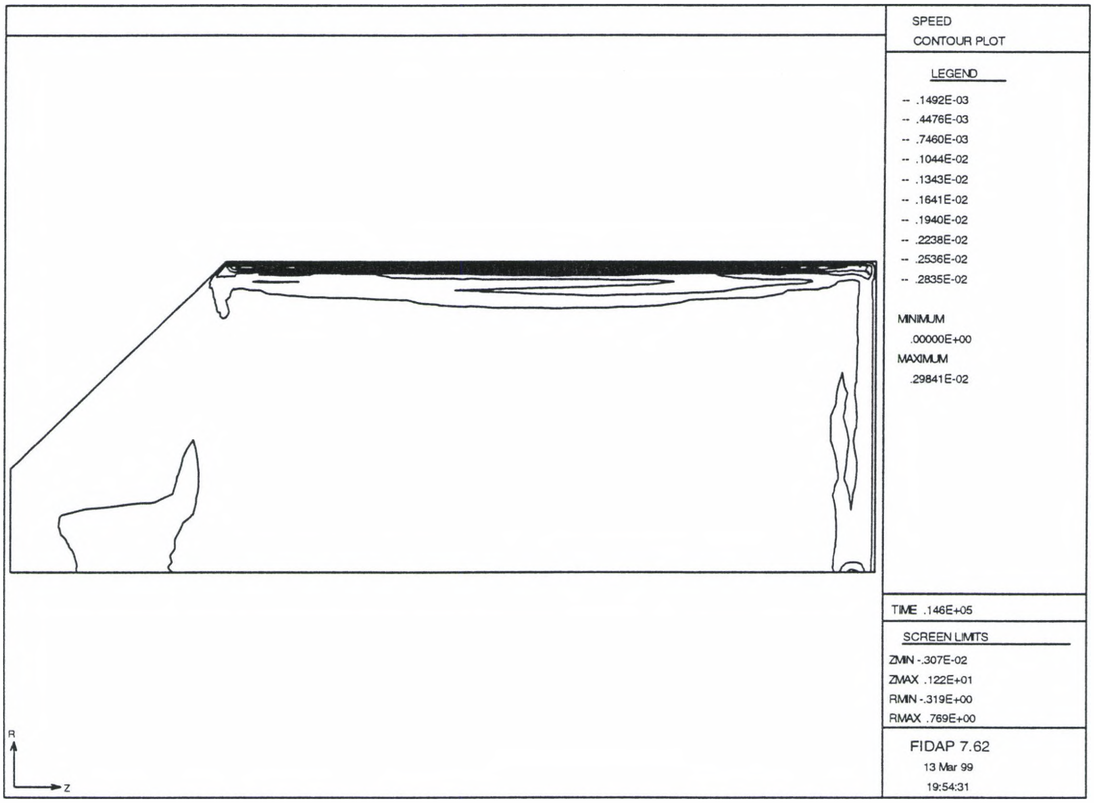


Figure 25. Velocity Field Plot, ϕ: 0.46

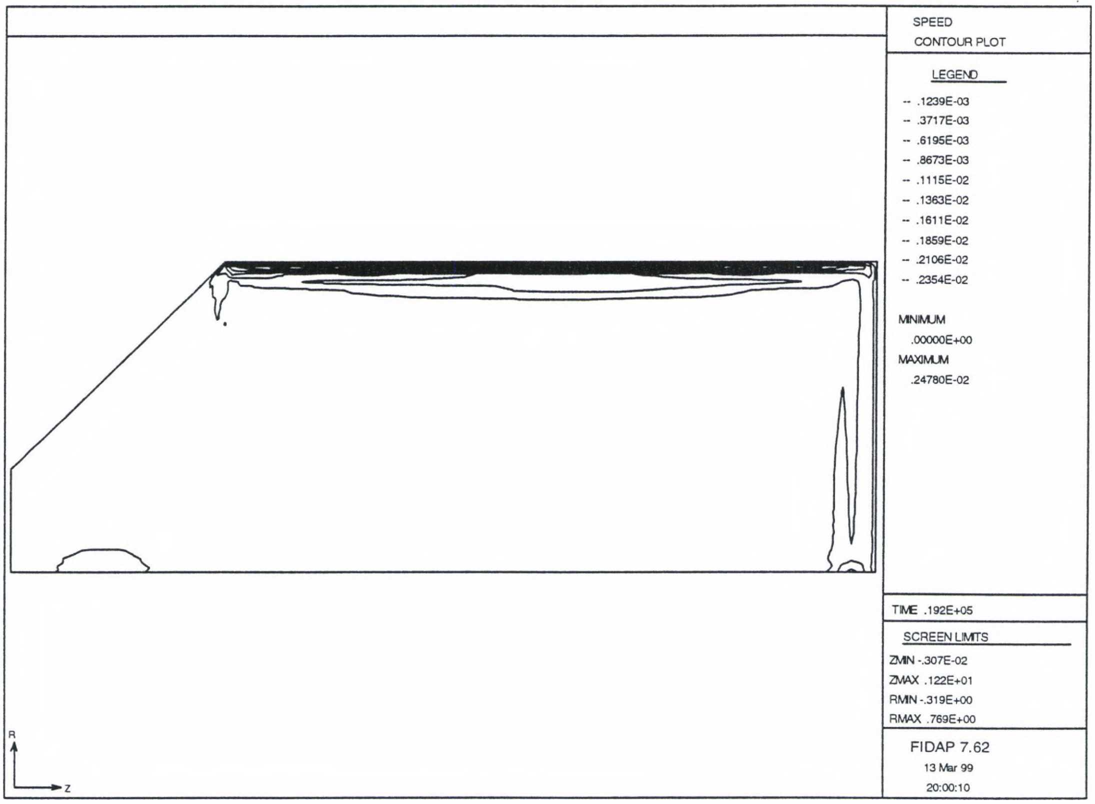


Figure 26. Velocity Field Plot, ϕ: 0.57

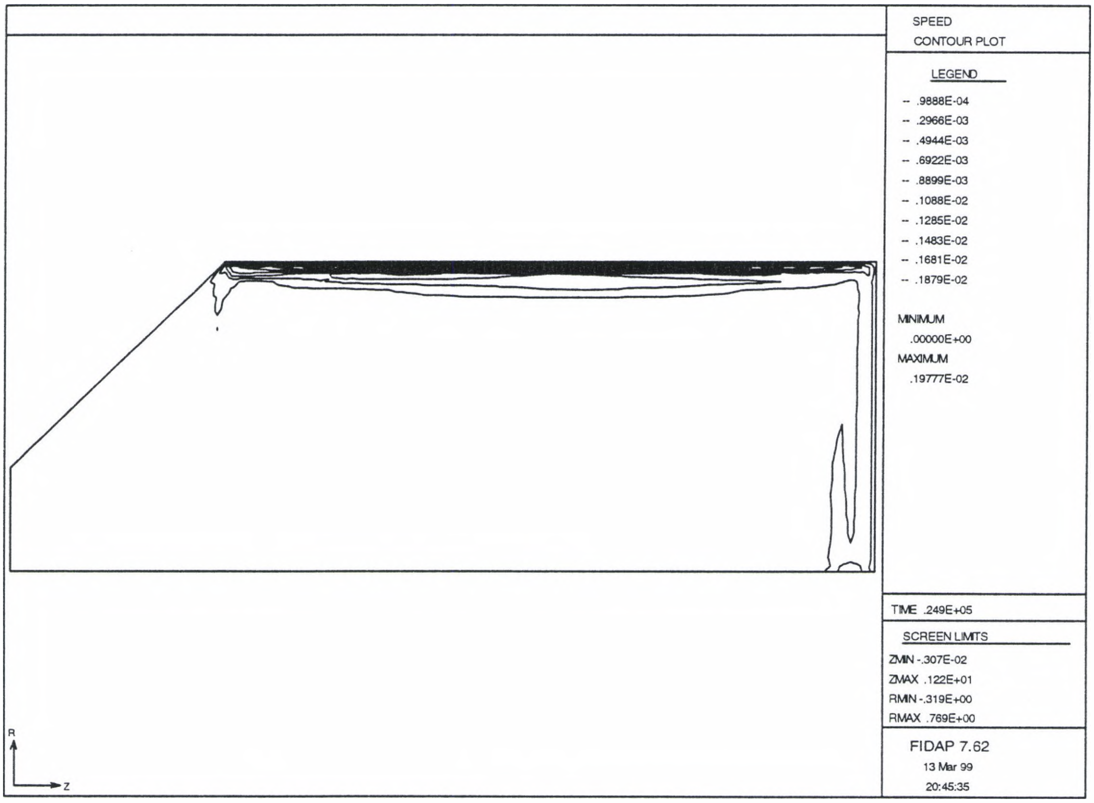
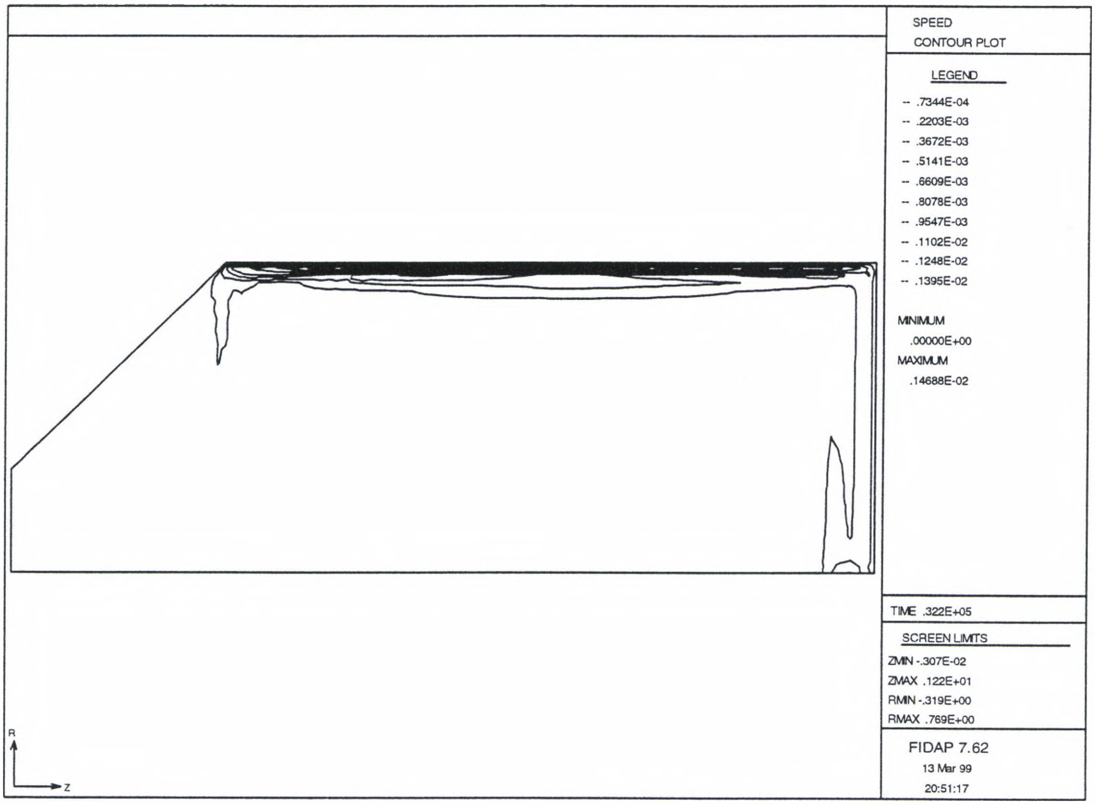


Figure 27. Velocity Field Plot, ϕ: 0.75



SPEED
CONTOUR PLOT

LEGEND

-- .7344E-04
-- .2203E-03
-- .3672E-03
-- .5141E-03
-- .6609E-03
-- .8078E-03
-- .9547E-03
-- .1102E-02
-- .1248E-02
-- .1395E-02

MINIMUM
.00000E+00
MAXIMUM
.14688E-02

TIME .322E+05

SCREEN LIMITS
ZMIN -.307E-02
ZMAX .122E+01
RMIN -.319E+00
RMAX .769E+00

FIDAP 7.62
13 Mar 99
20:51:17

Figure 28. Velocity Field Plot, ϕ: 0.85

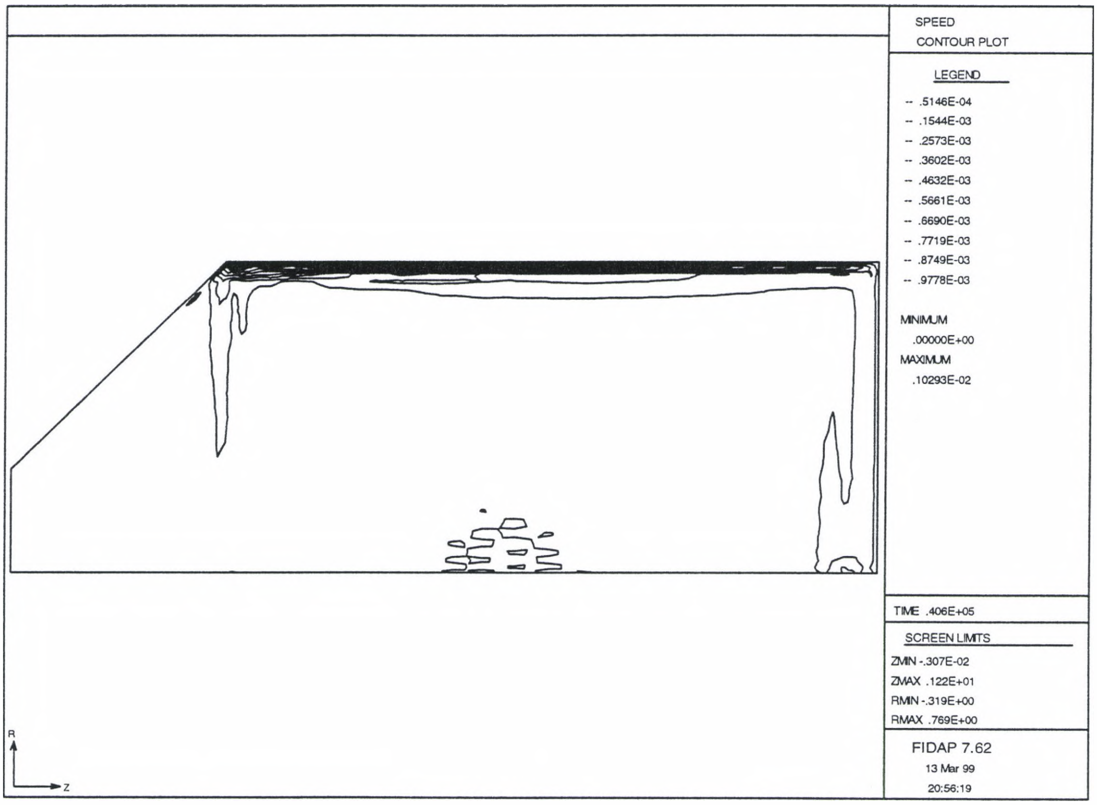


Figure 29. Velocity Field Plot, ϕ: 0.93

Streamline Plots

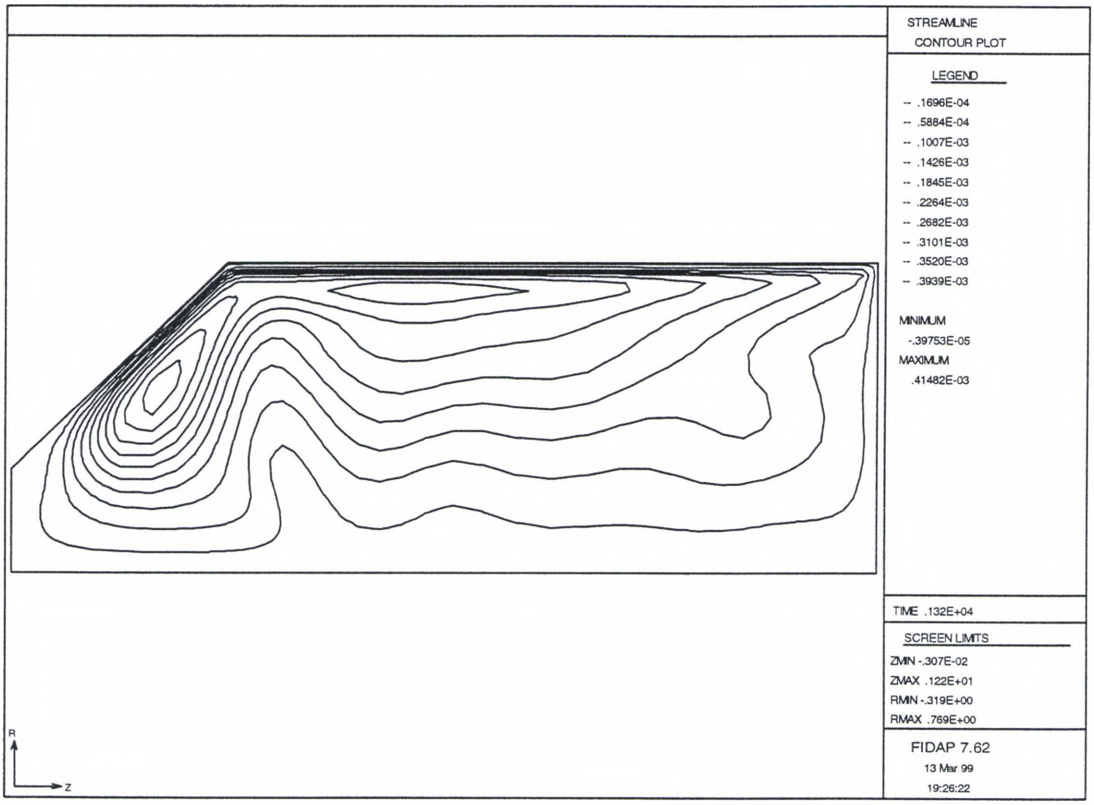


Figure 30. Streamline Plot, ϕ: 0.07

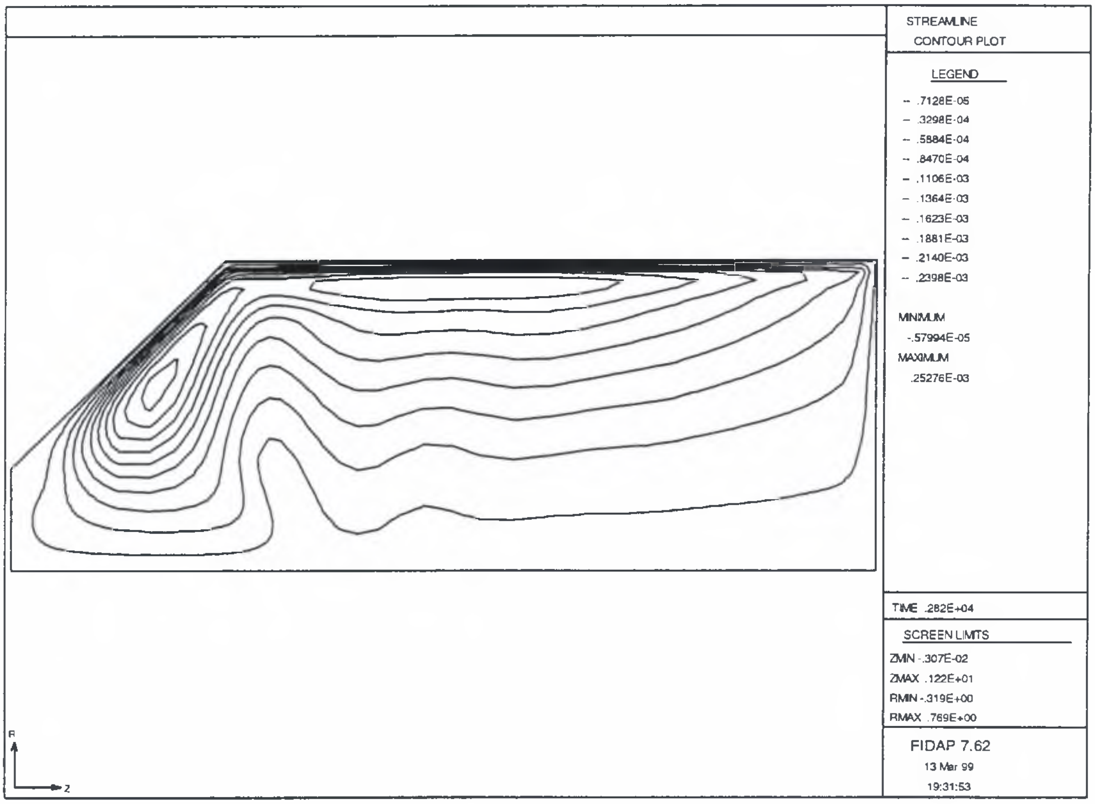


Figure 31. Streamline Plot, ϕ: 0.12

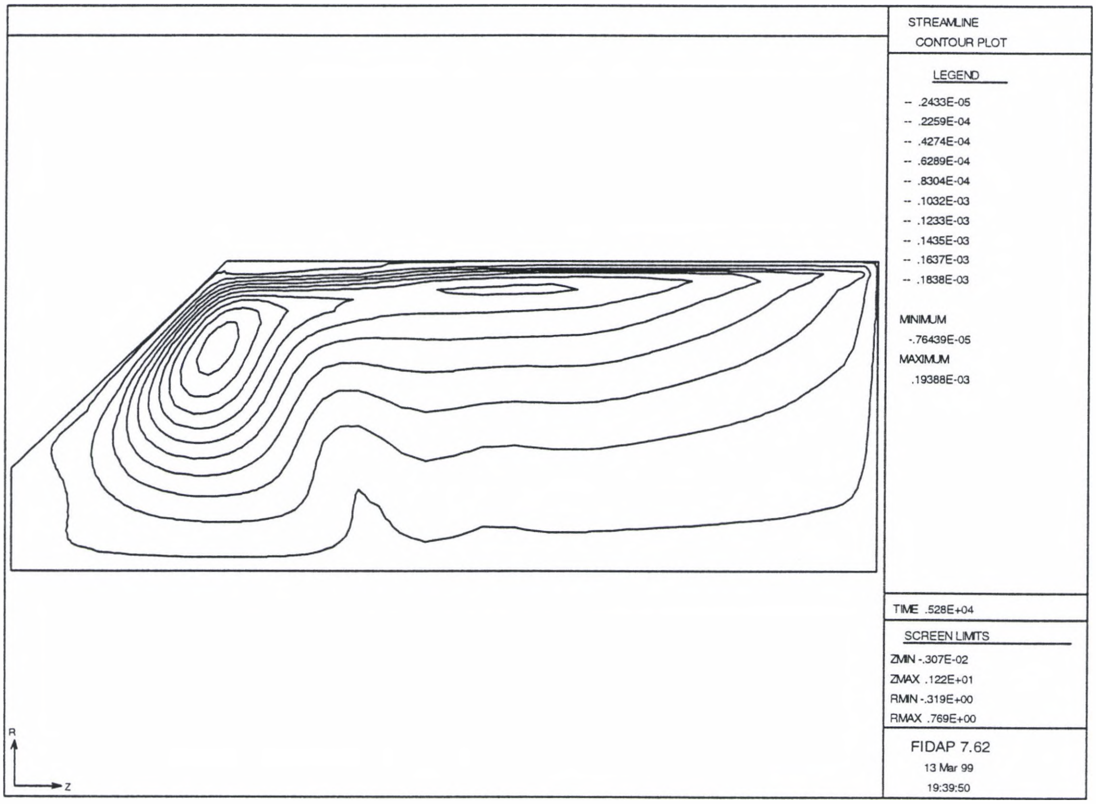


Figure 32. Streamline Plot, ϕ: 0.19

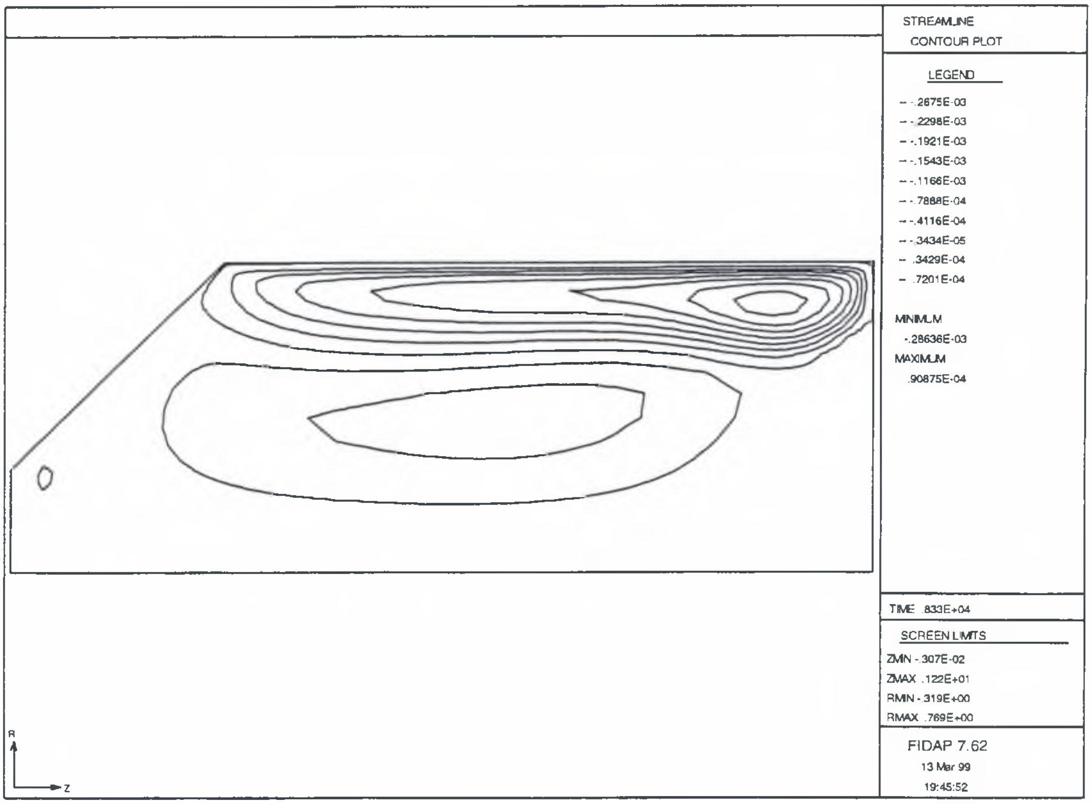


Figure 33. Streamline Plot, ϕ: 0.25

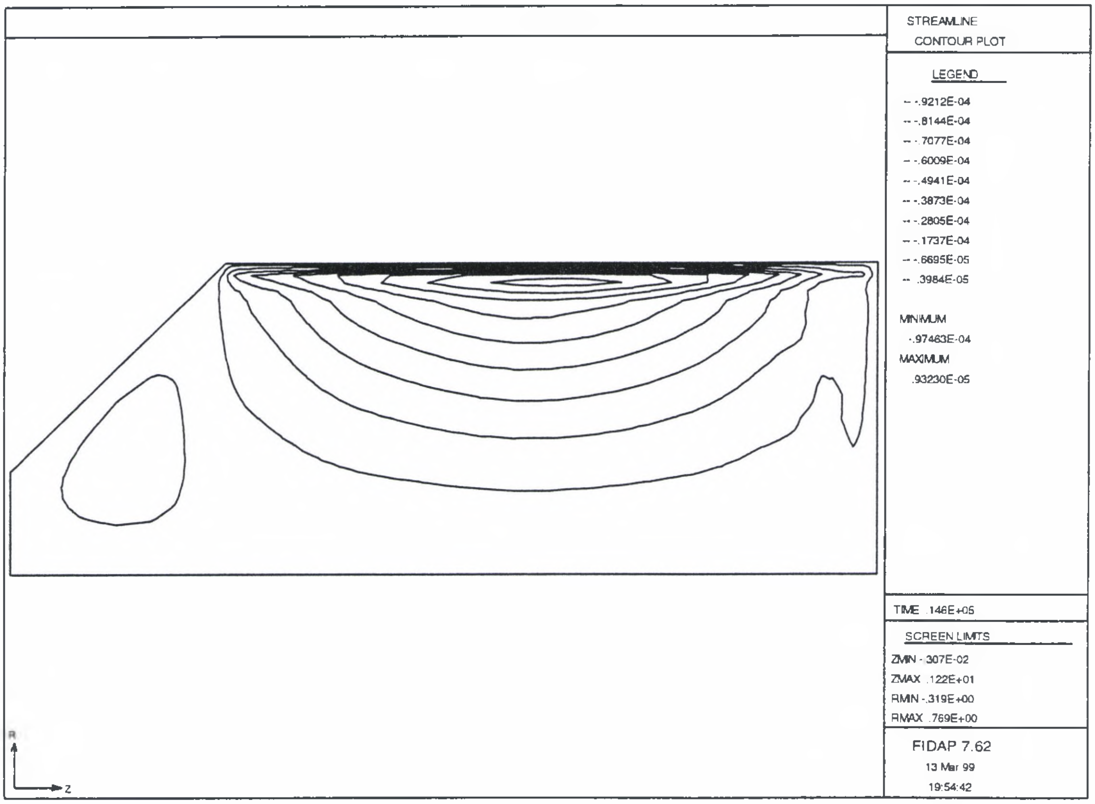


Figure 34. Streamline Plot, ϕ: 0.46

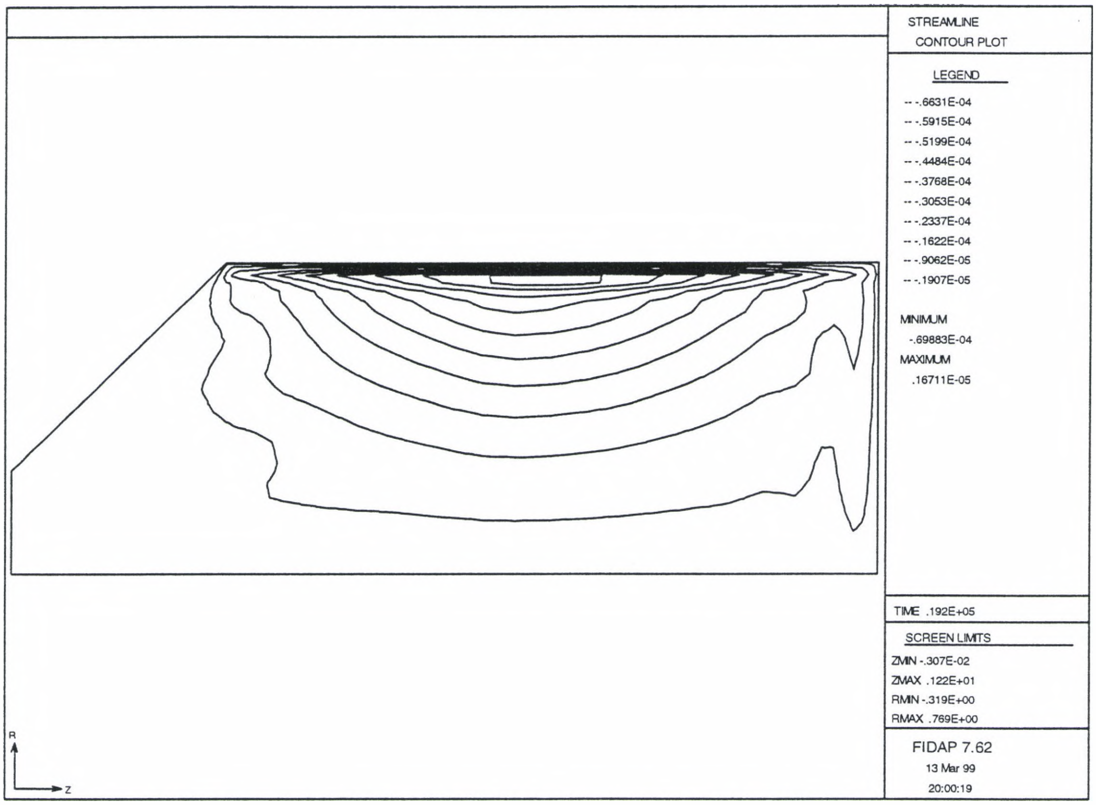


Figure 35. Streamline Plot, ϕ: 0.57

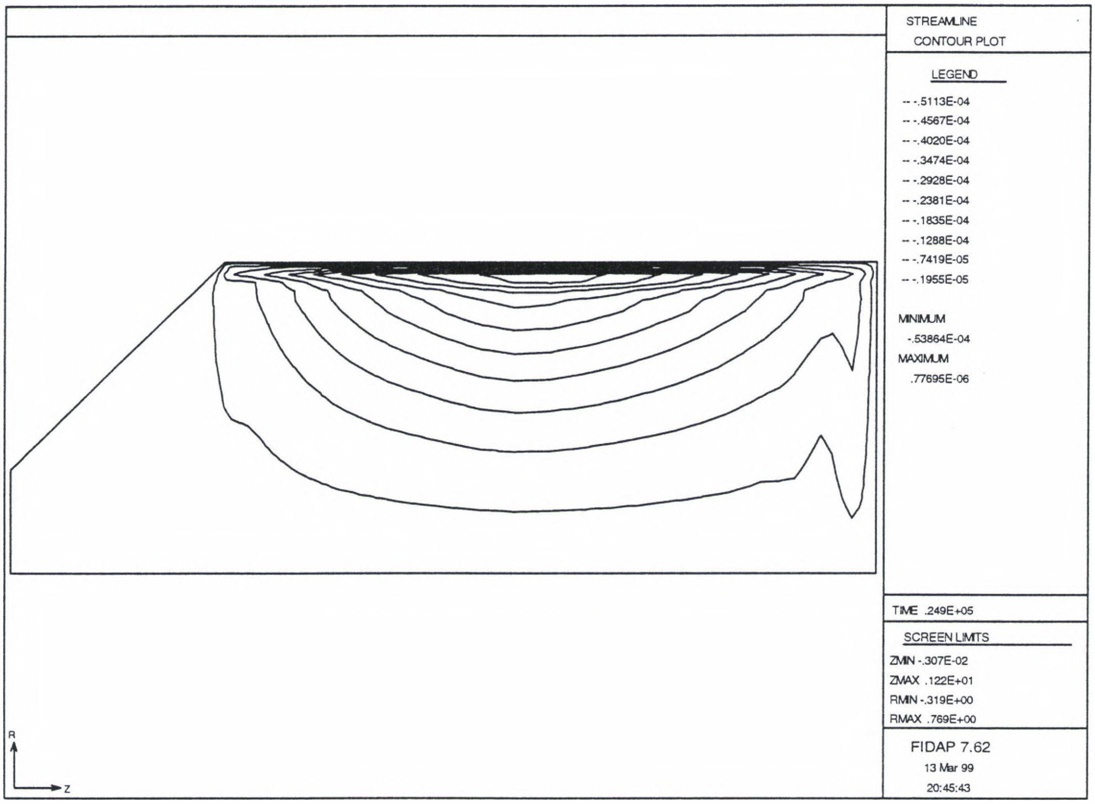


Figure 36. Streamline Plot, ϕ: 0.75

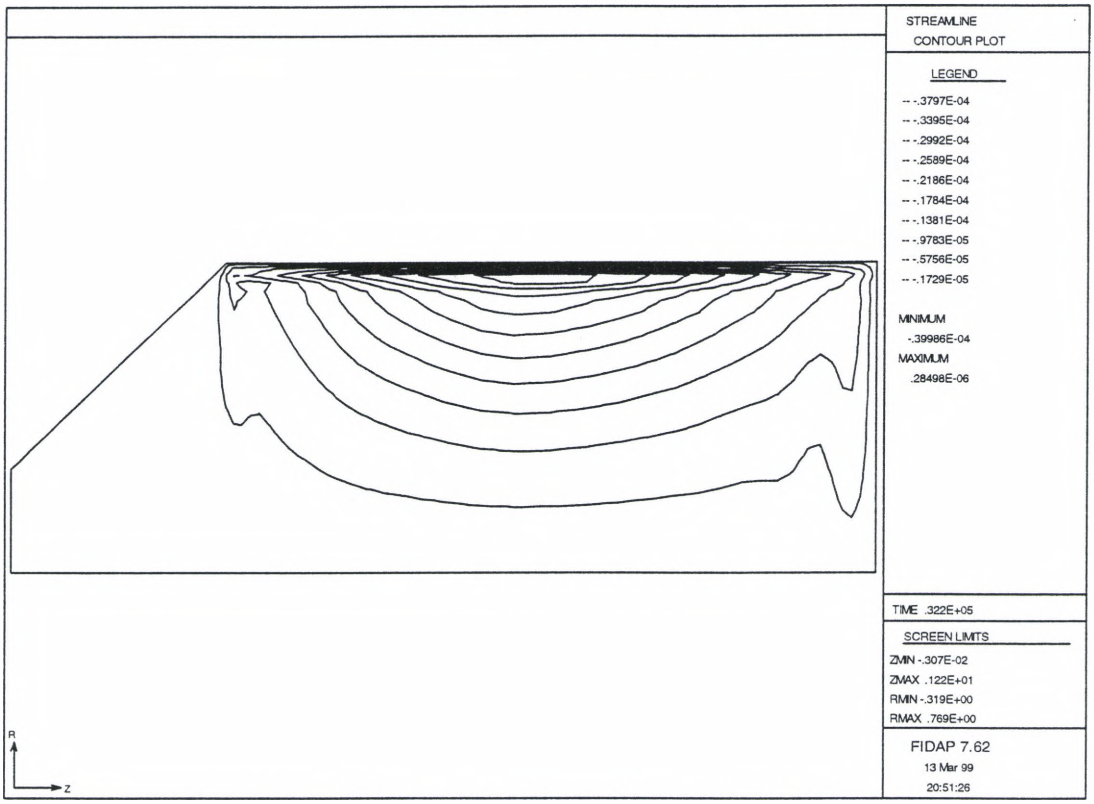


Figure 37. Streamline Plot, ϕ: 0.85

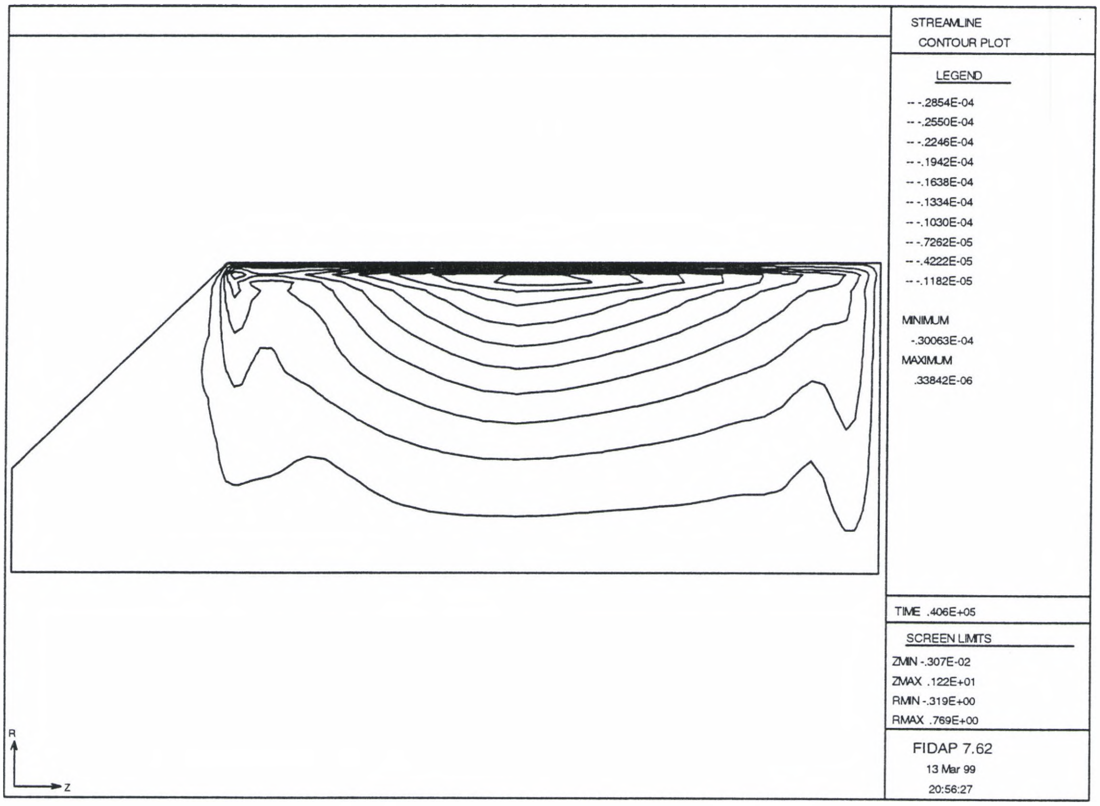


Figure 38. Streamline Plot, ϕ: 0.93

Temperature Field Plots

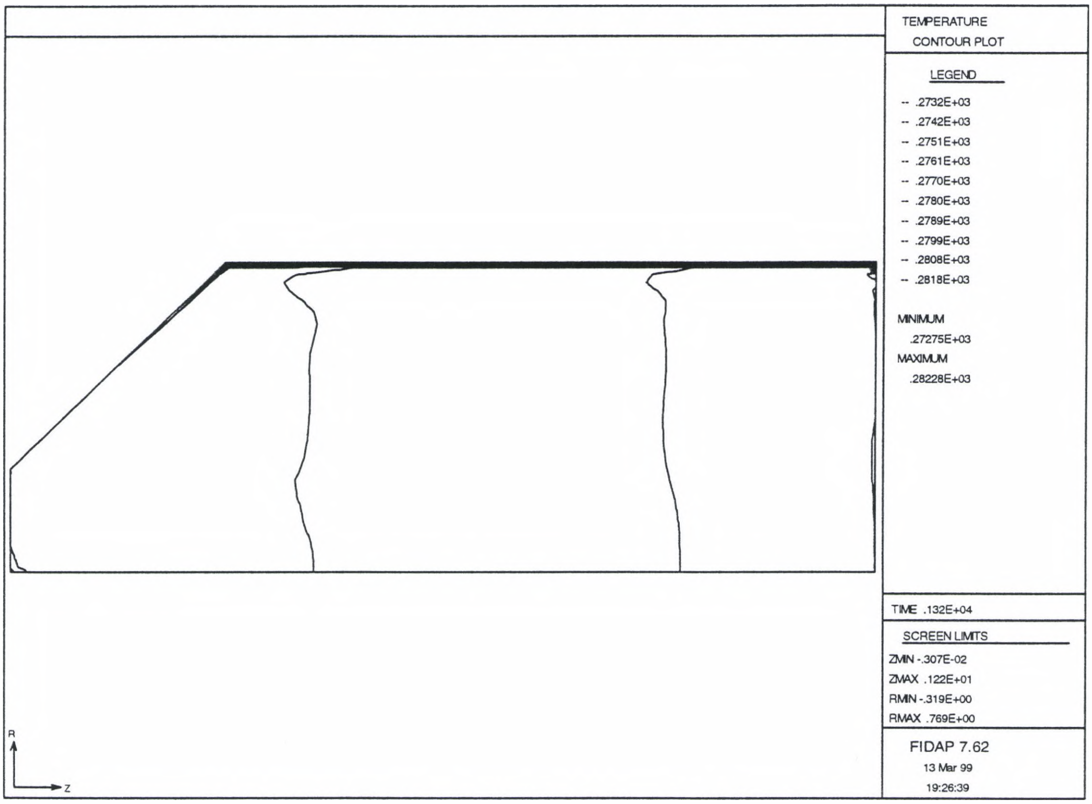


Figure 39. Temperature Plot, ϕ: 0.07

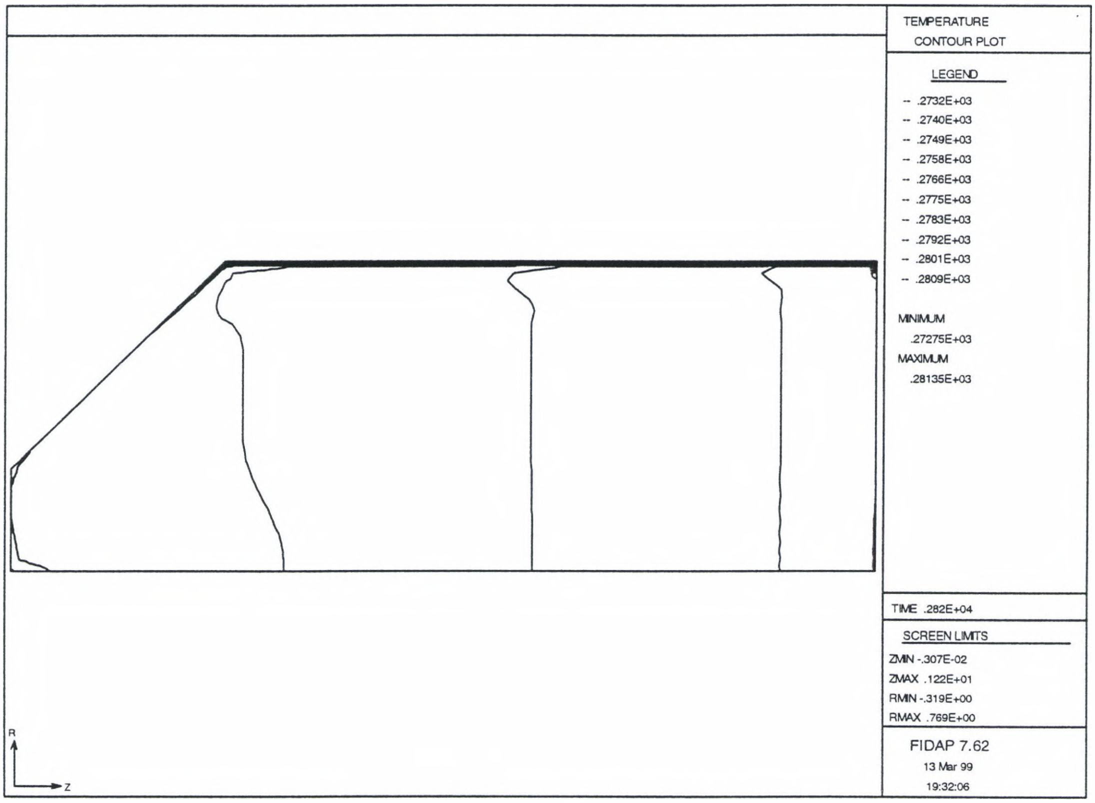


Figure 40. Temperature Plot, ϕ: 0.12

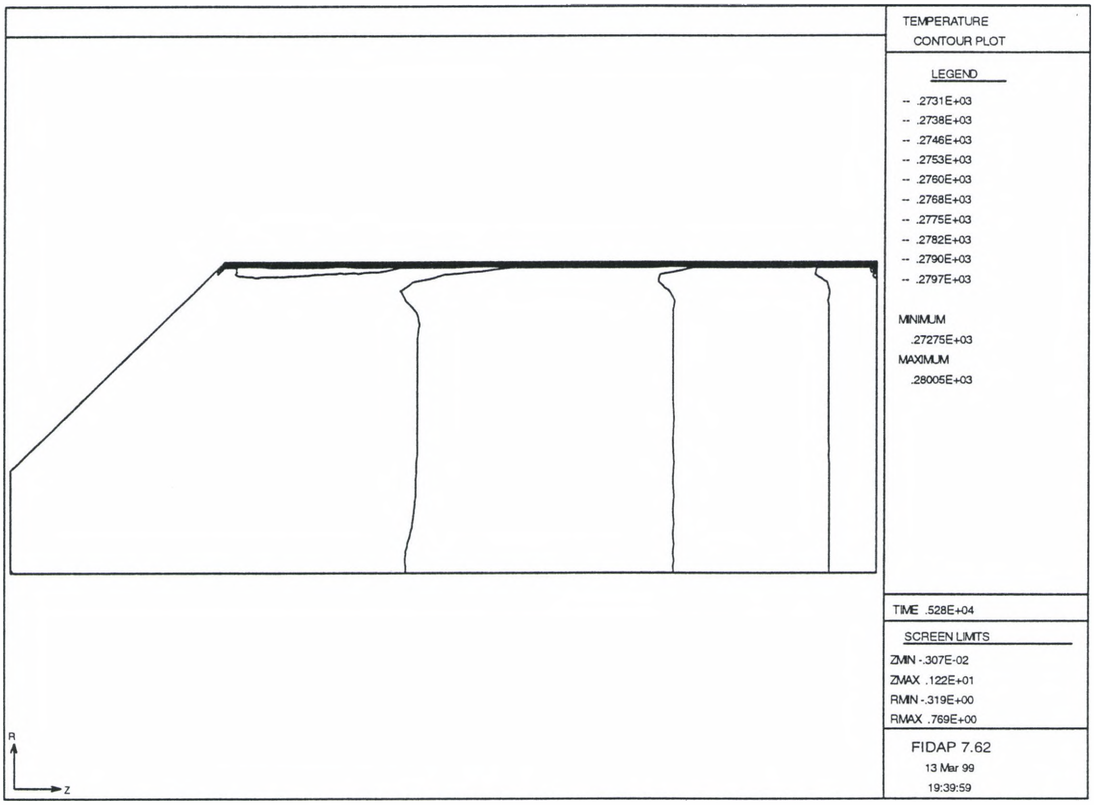


Figure 41. Temperature Plot, ϕ: 0.19

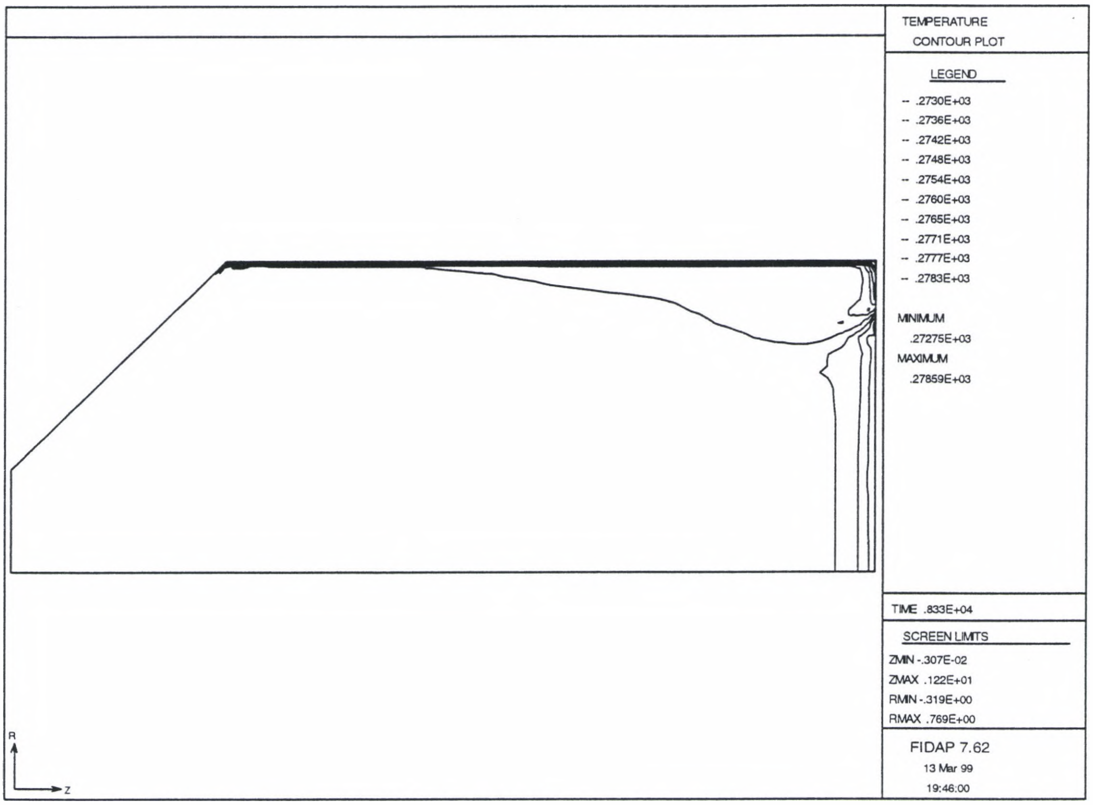


Figure 42. Temperature Plot, ϕ: 0.25

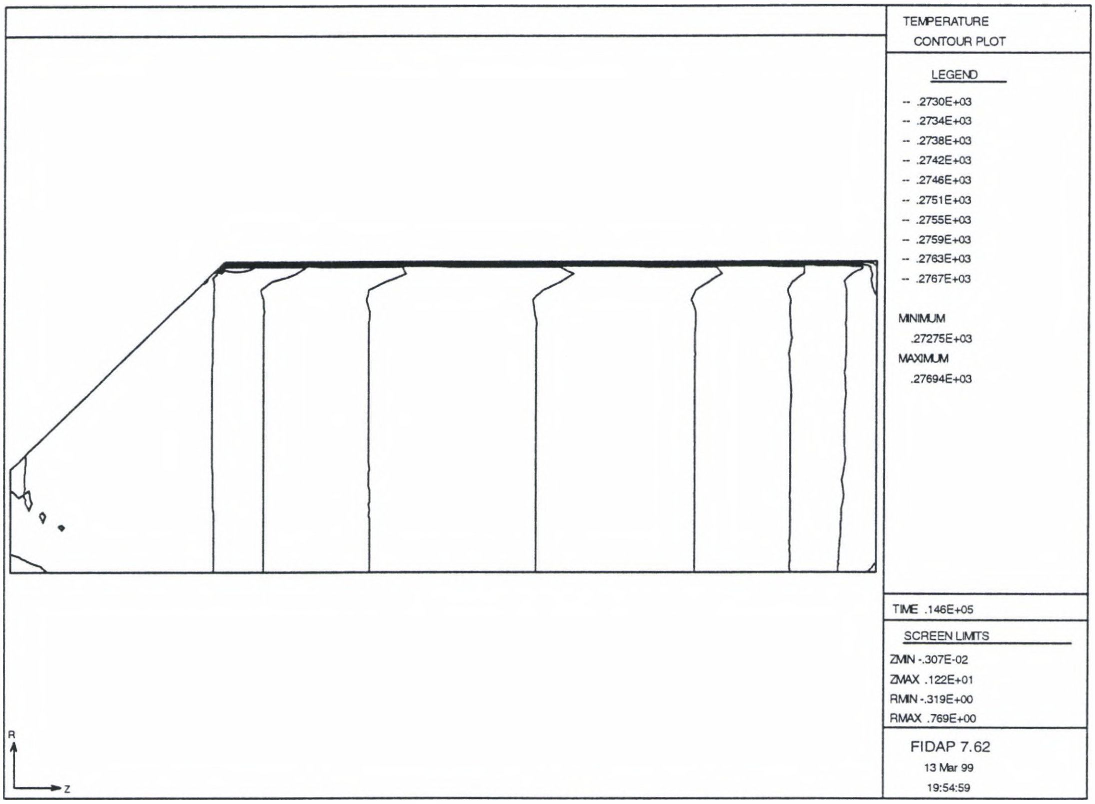


Figure 43. Temperature Plot, ϕ: 0.46

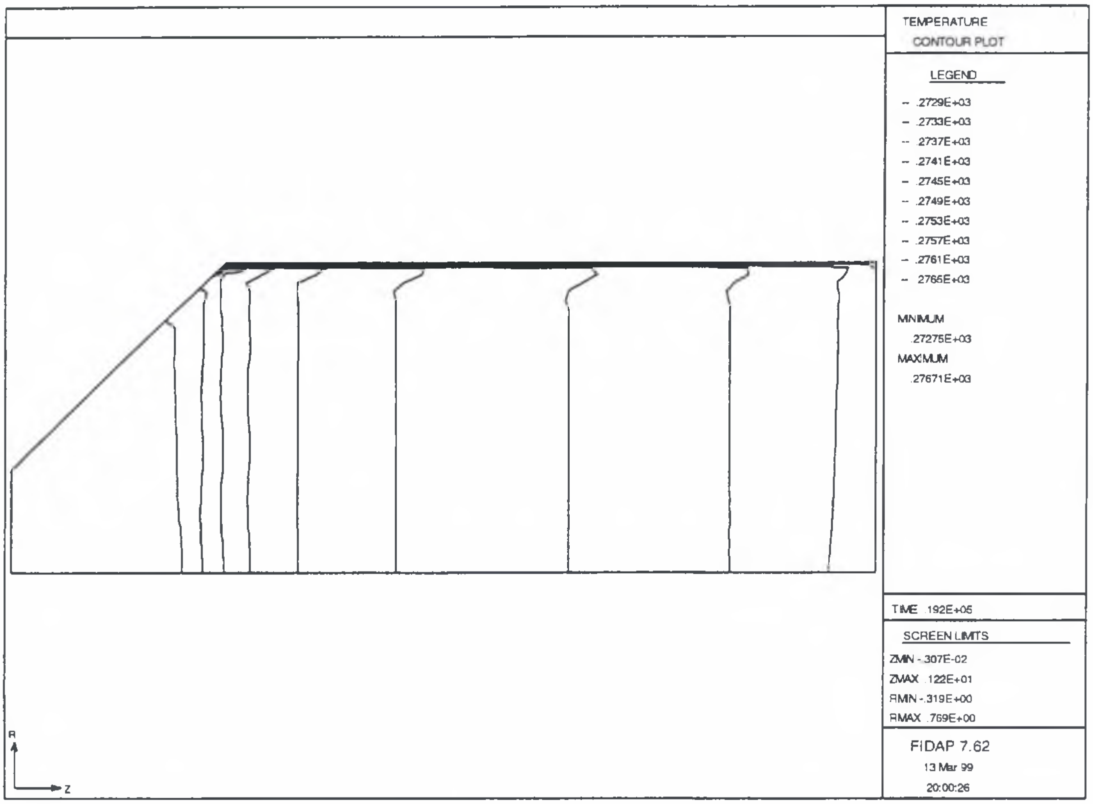


Figure 44. Temperature Plot, ϕ: 0.57

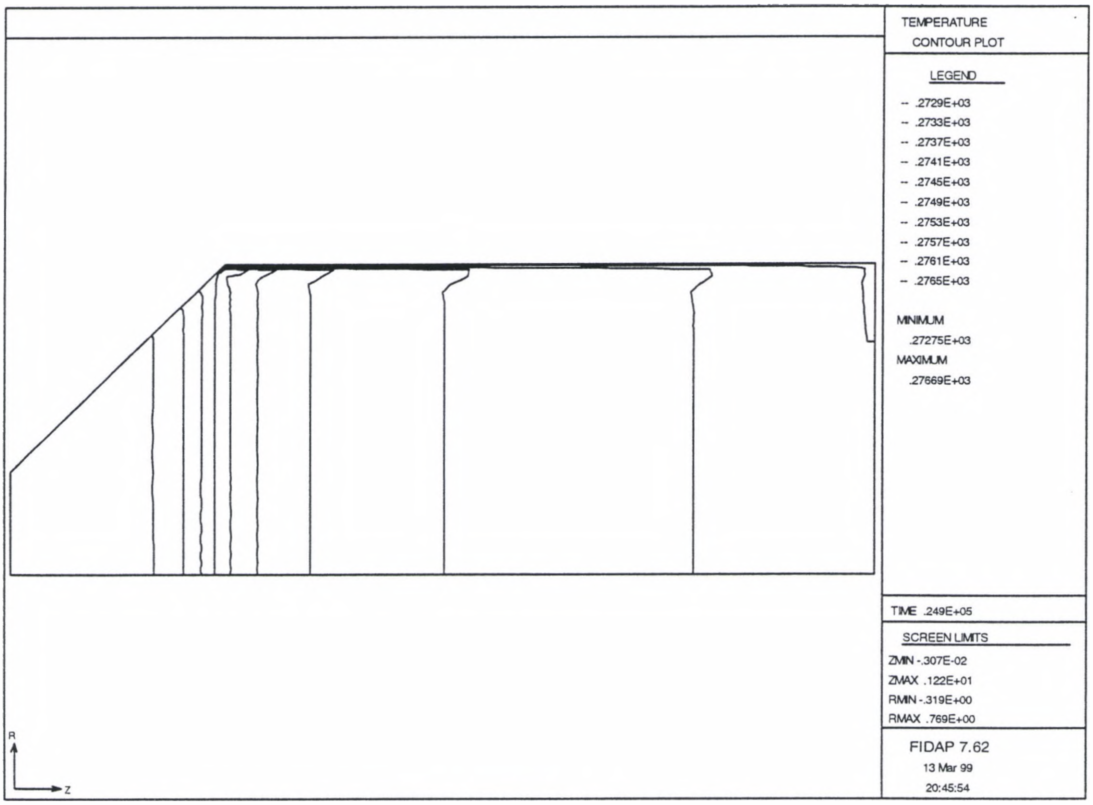


Figure 45. Temperature Plot, ϕ: 0.75

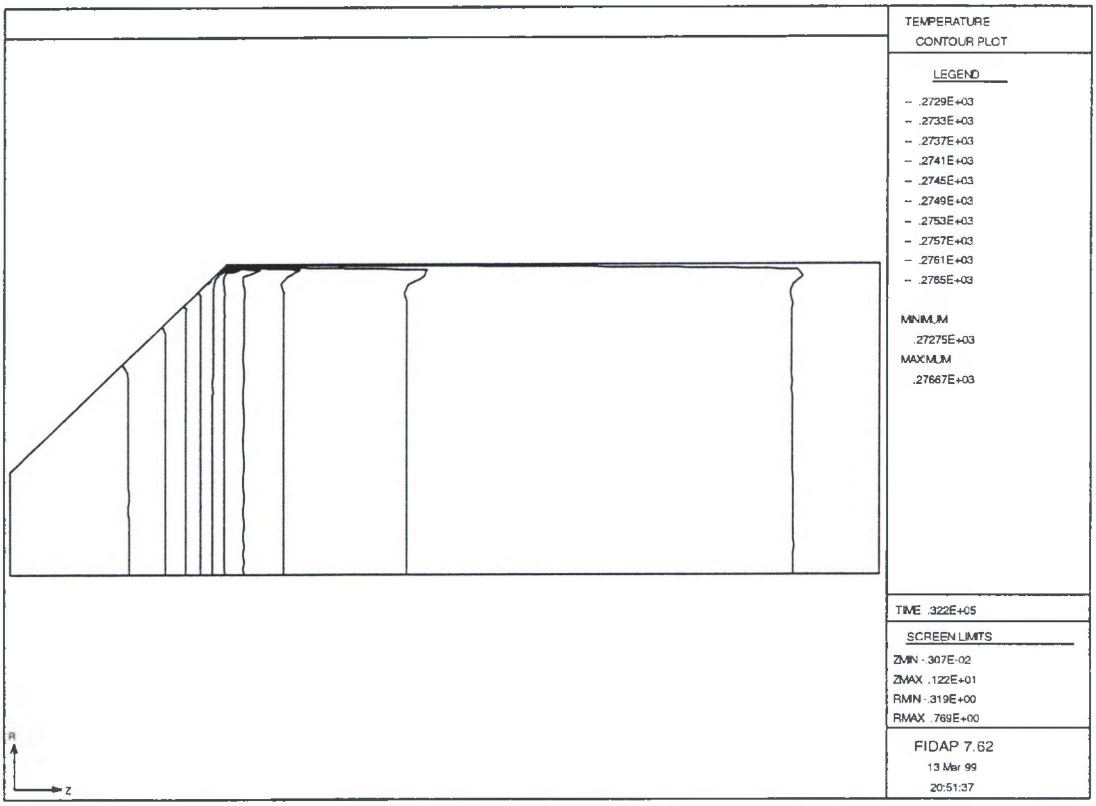


Figure 46. Temperature Plot, ϕ: 0.85

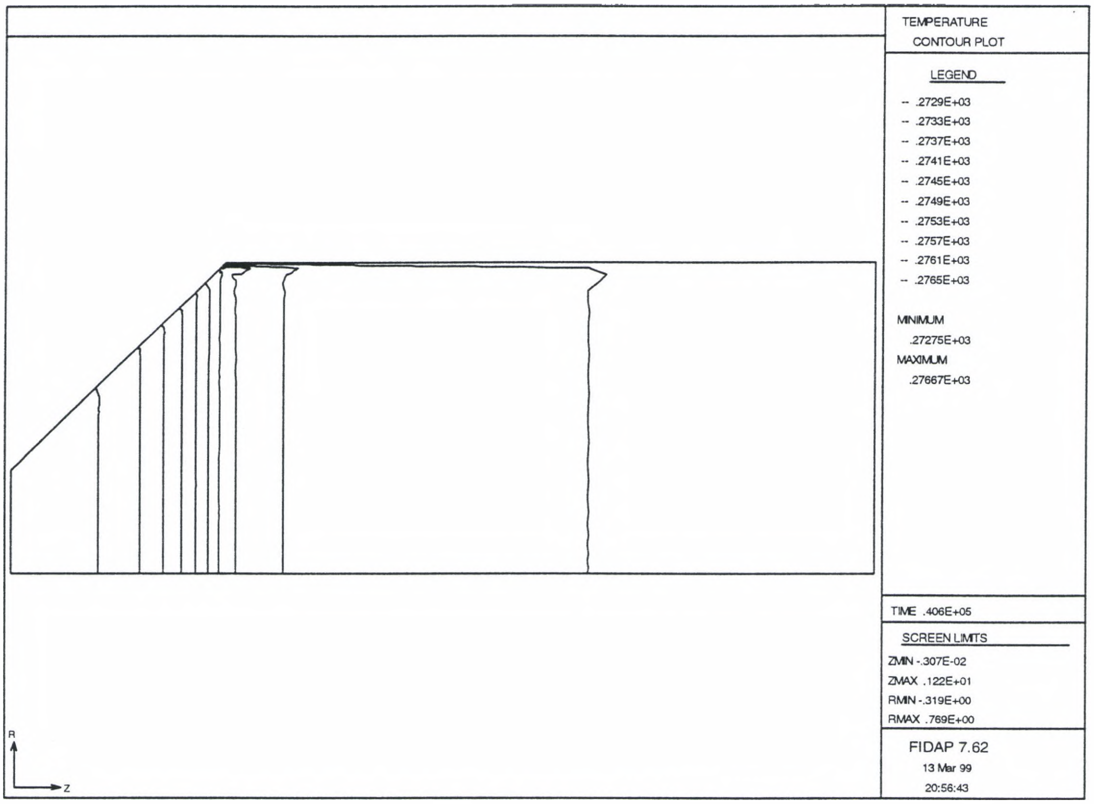


Figure 47. Temperature Plot, ϕ: 0.93

Vector Field Plots

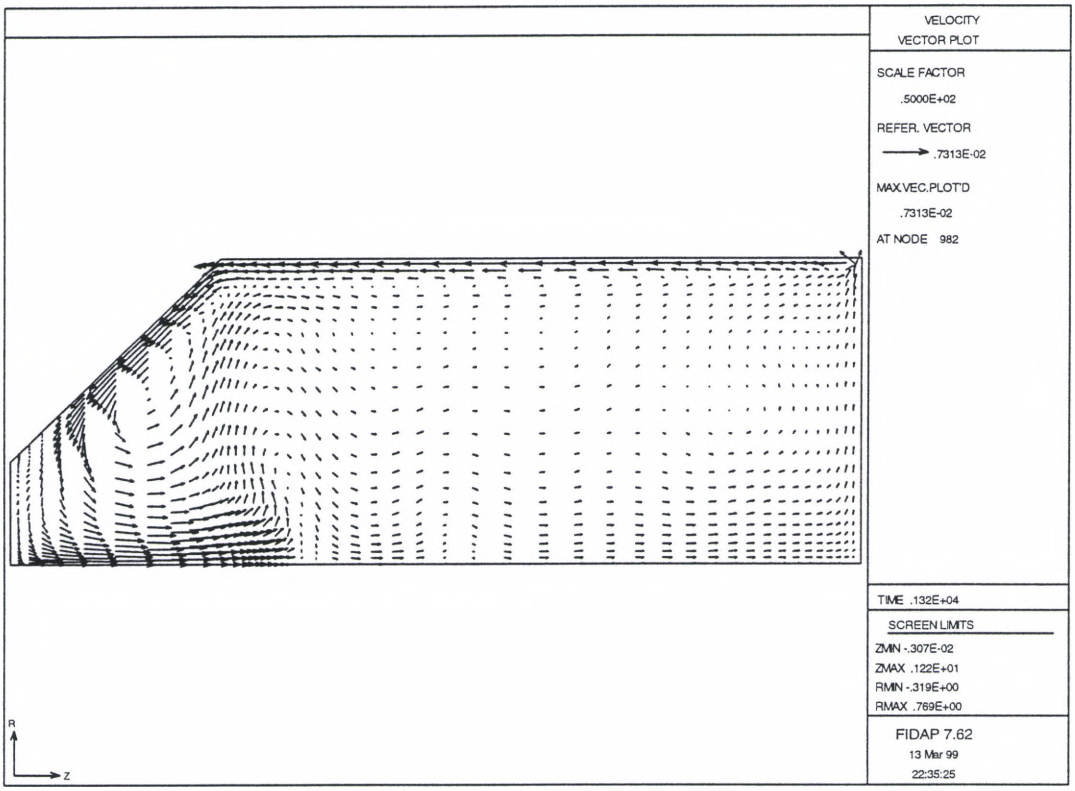


Figure 48. Vector Plot, ϕ: 0.07

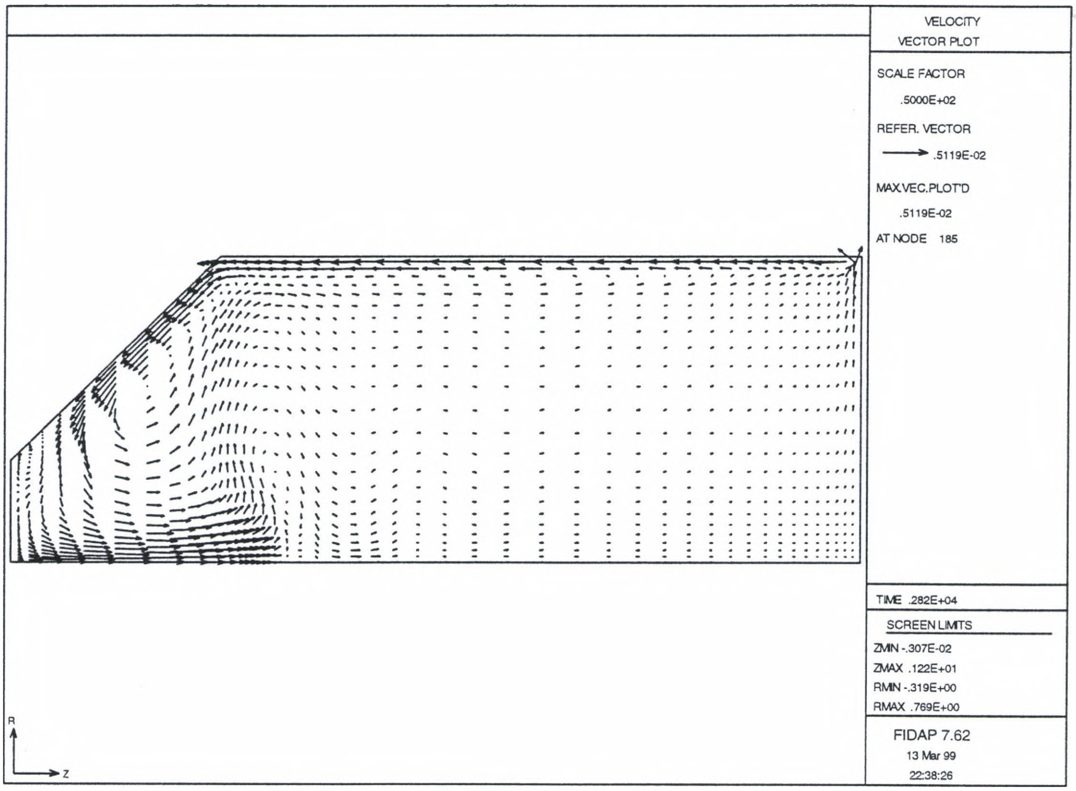


Figure 49. Vector Plot, ϕ: 0.12

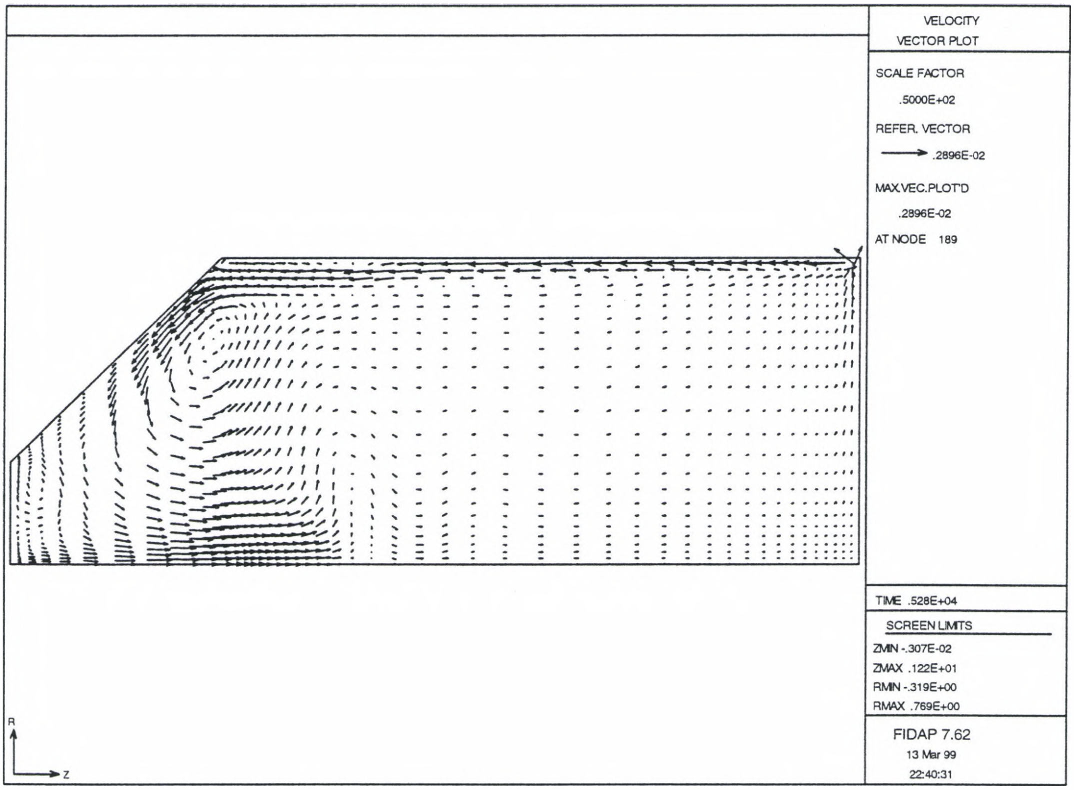


Figure 50. Vector Plot, ϕ: 0.19

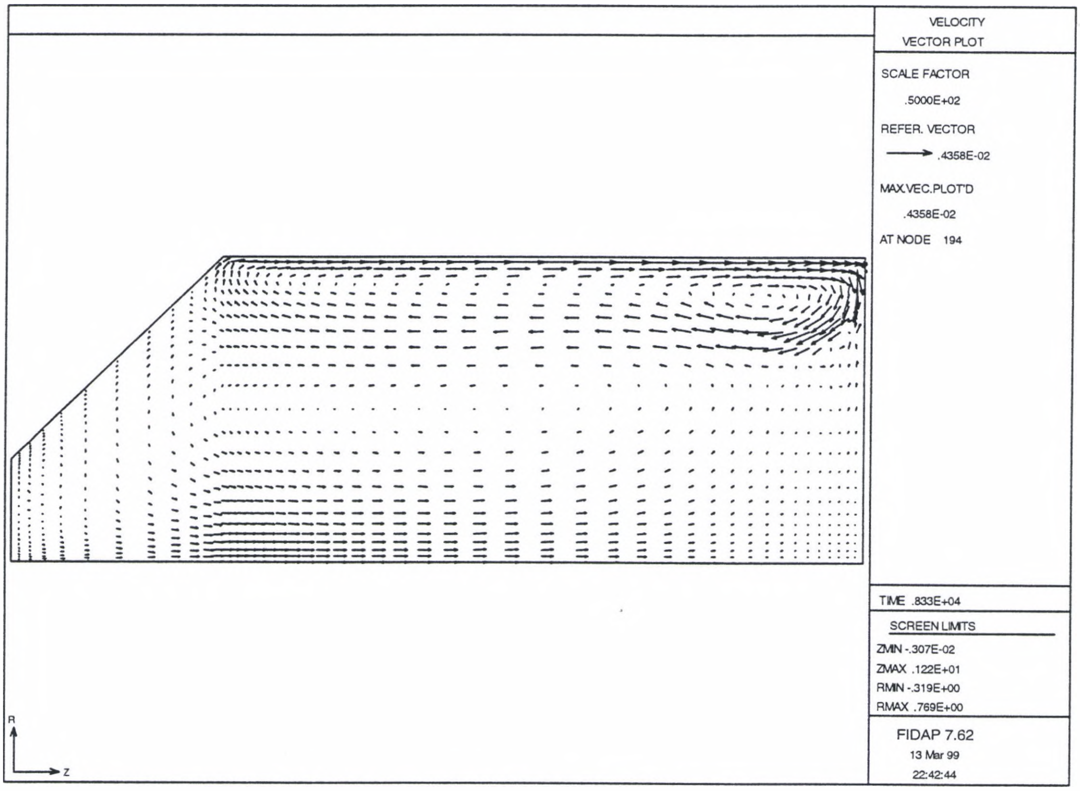


Figure 51. Vector Plot, ϕ: 0.25

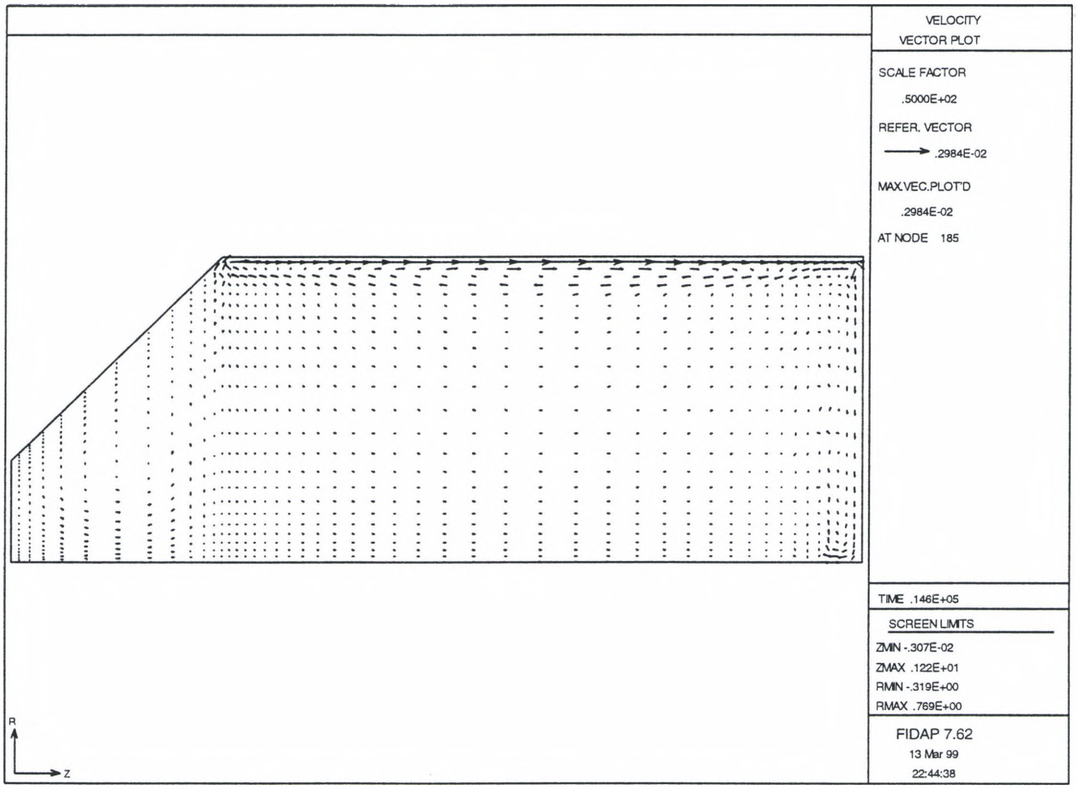


Figure 52. Vector Plot, ϕ: 0.46

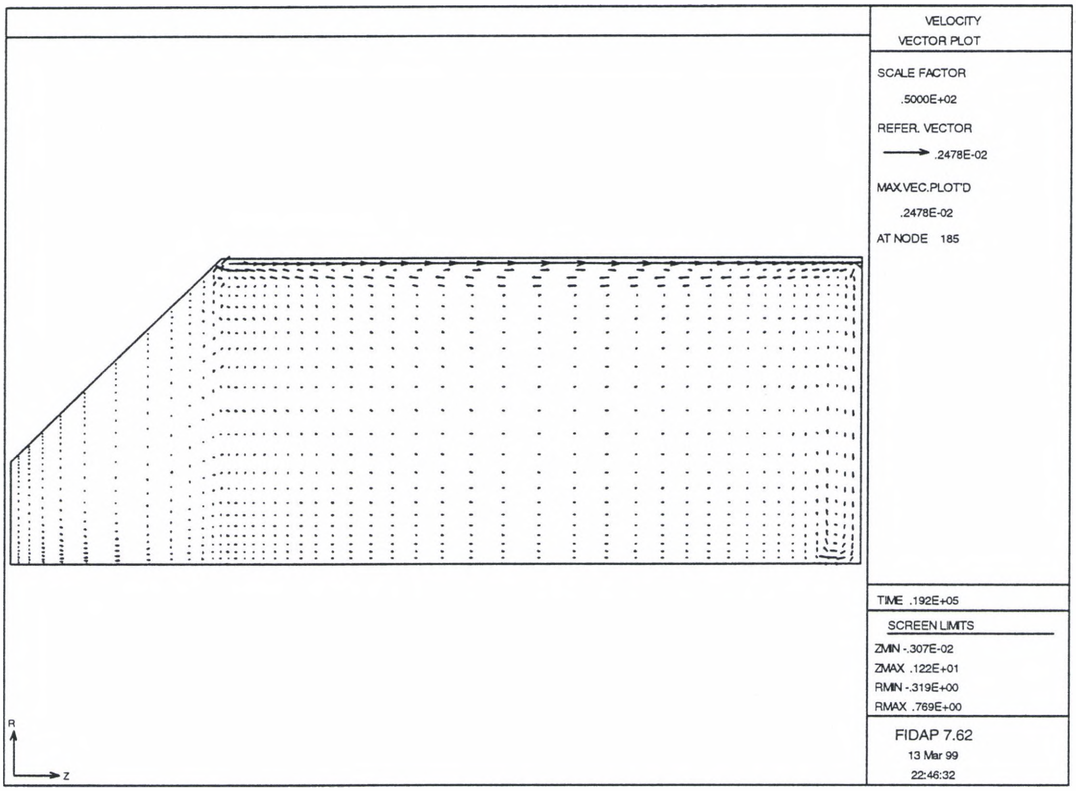


Figure 53. Vector Plot, ϕ: 0.57

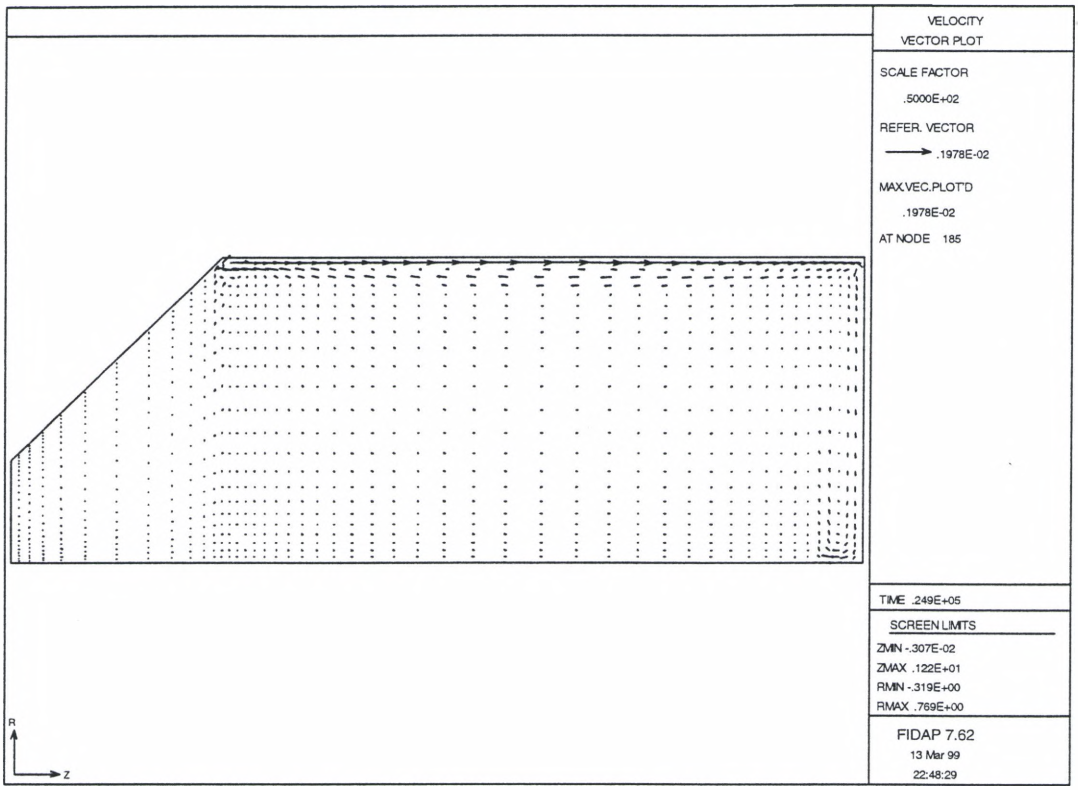


Figure 54. Vector Plot, ϕ: 0.75

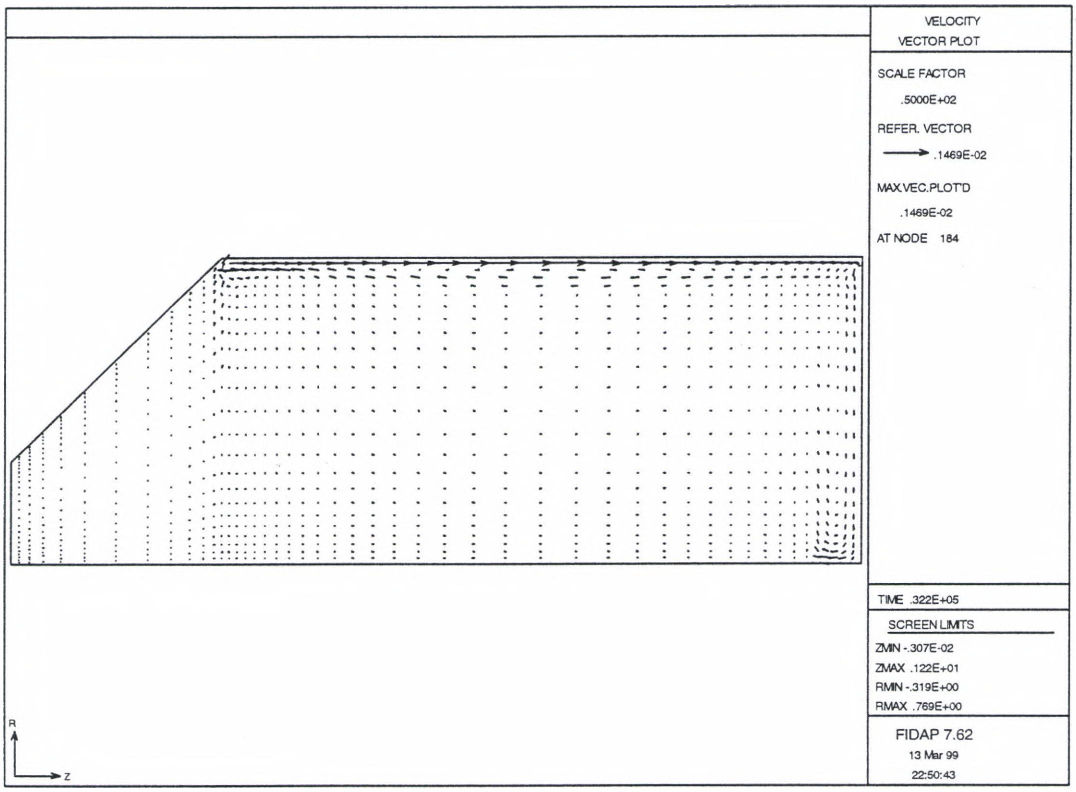


Figure 55. Vector Plot, ϕ: 0.85

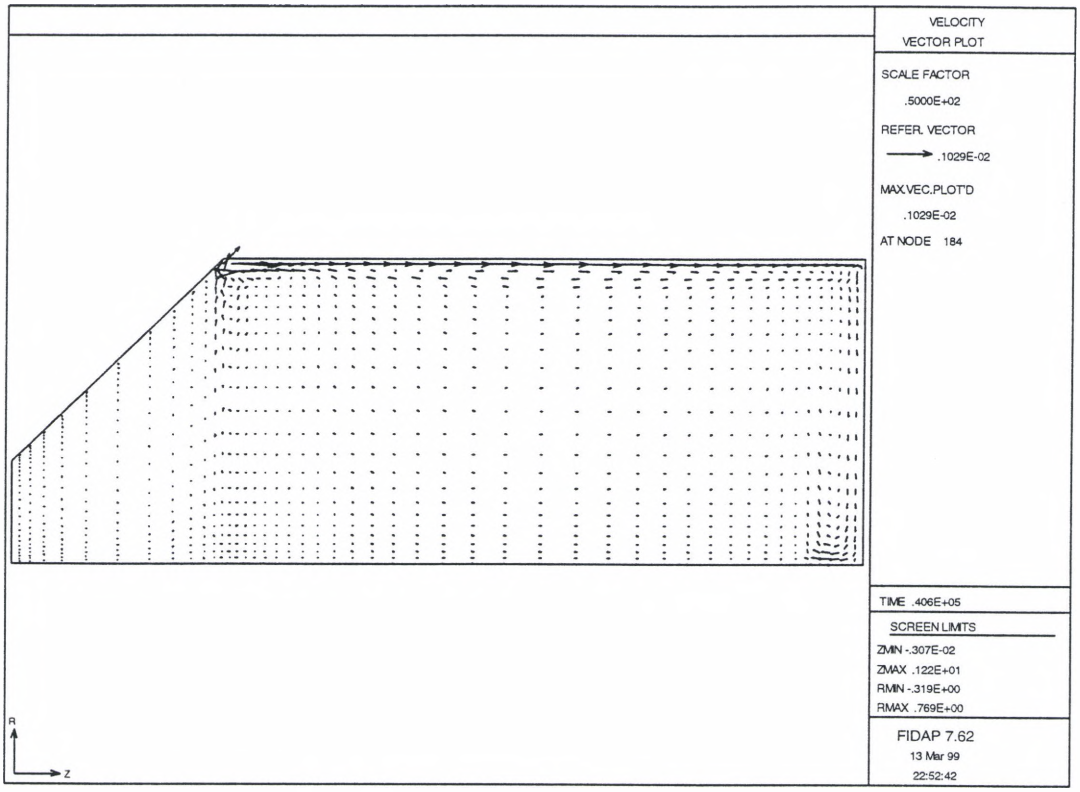


Figure 56. Vector Plot, ϕ: 0.93

Time Step History Plots

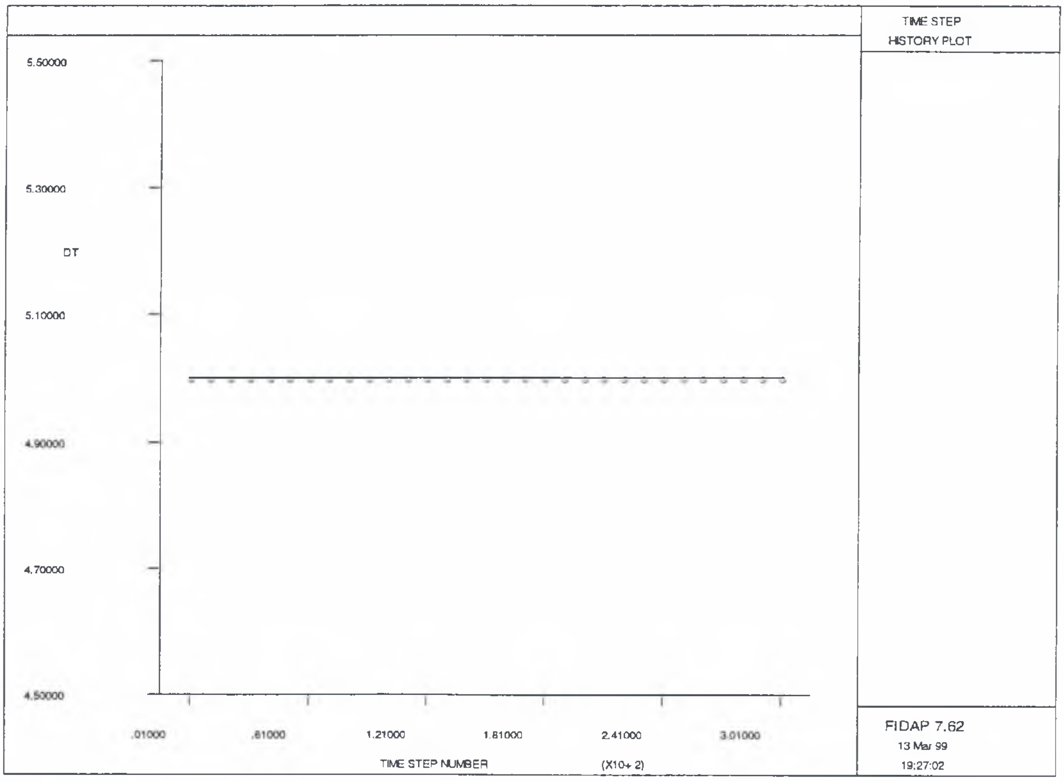


Figure 57. Time History Plot, ϕ: 0.07

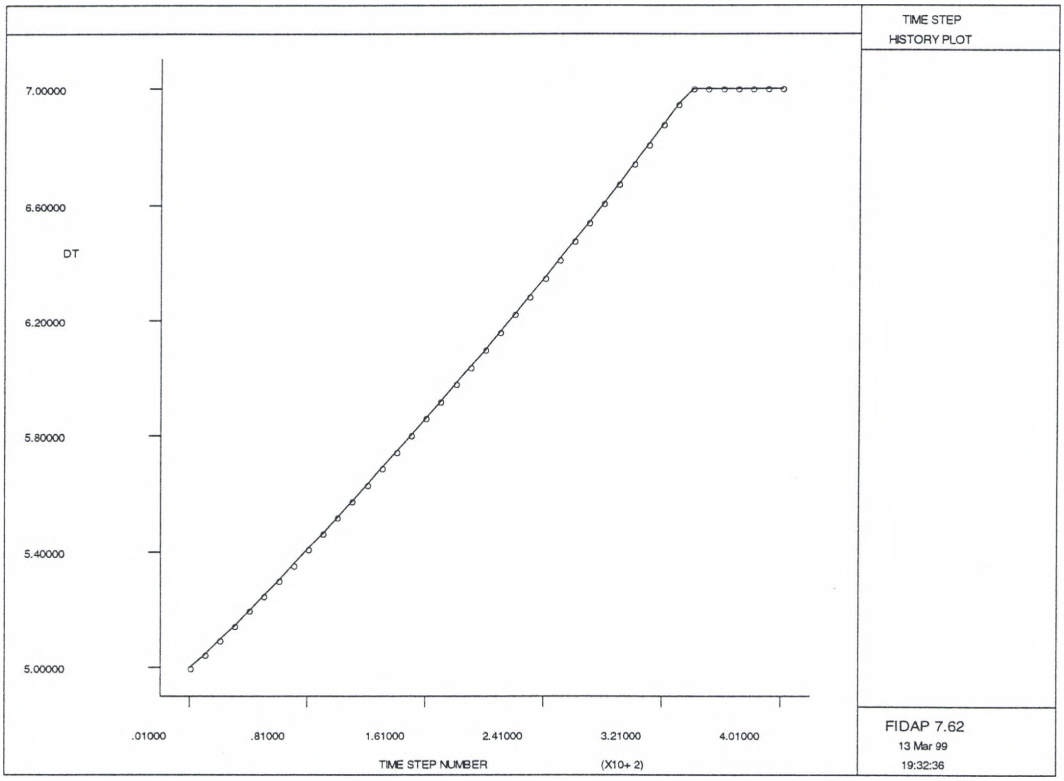


Figure 58. Time History Plot, ϕ: 0.12

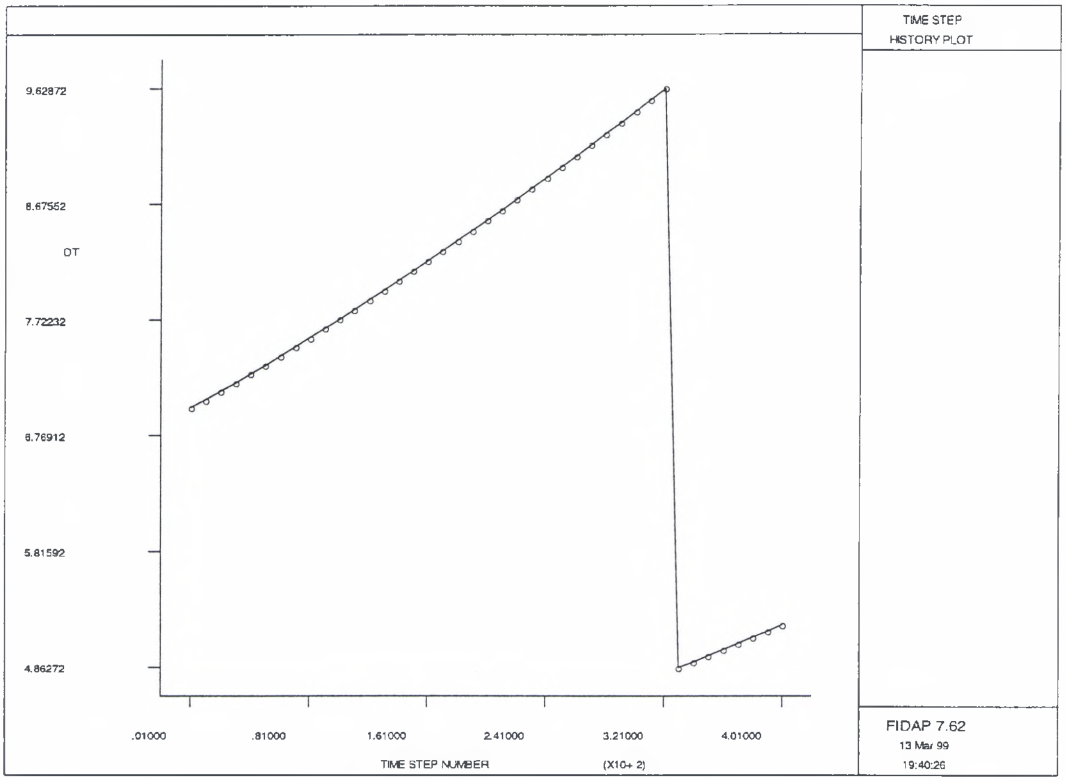


Figure 59. Time History Plot, ϕ: 0.19

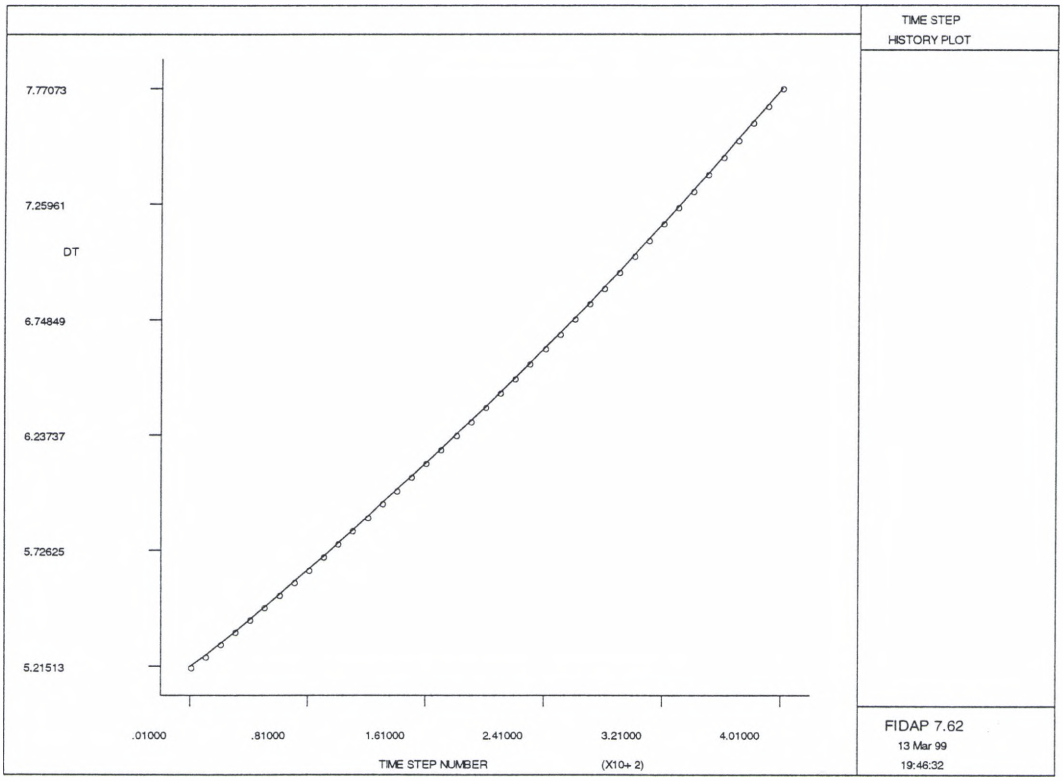


Figure 60. Time History Plot, ϕ: 0.25

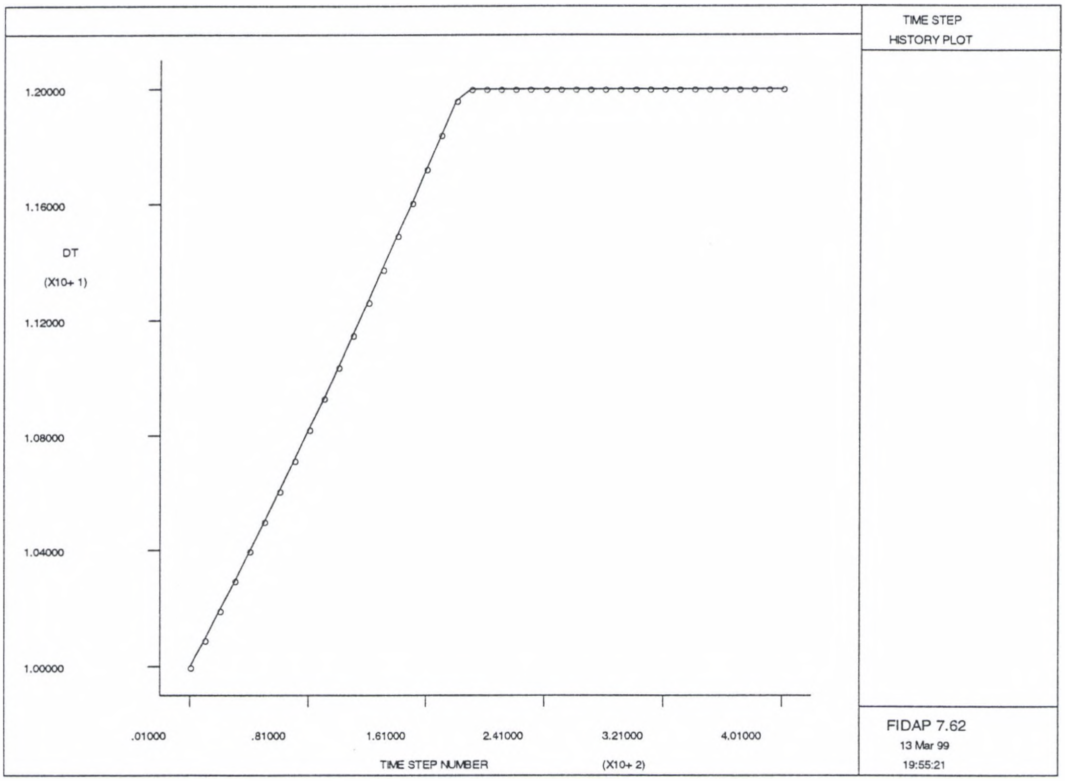


Figure 61. Time History Plot, ϕ: 0.46

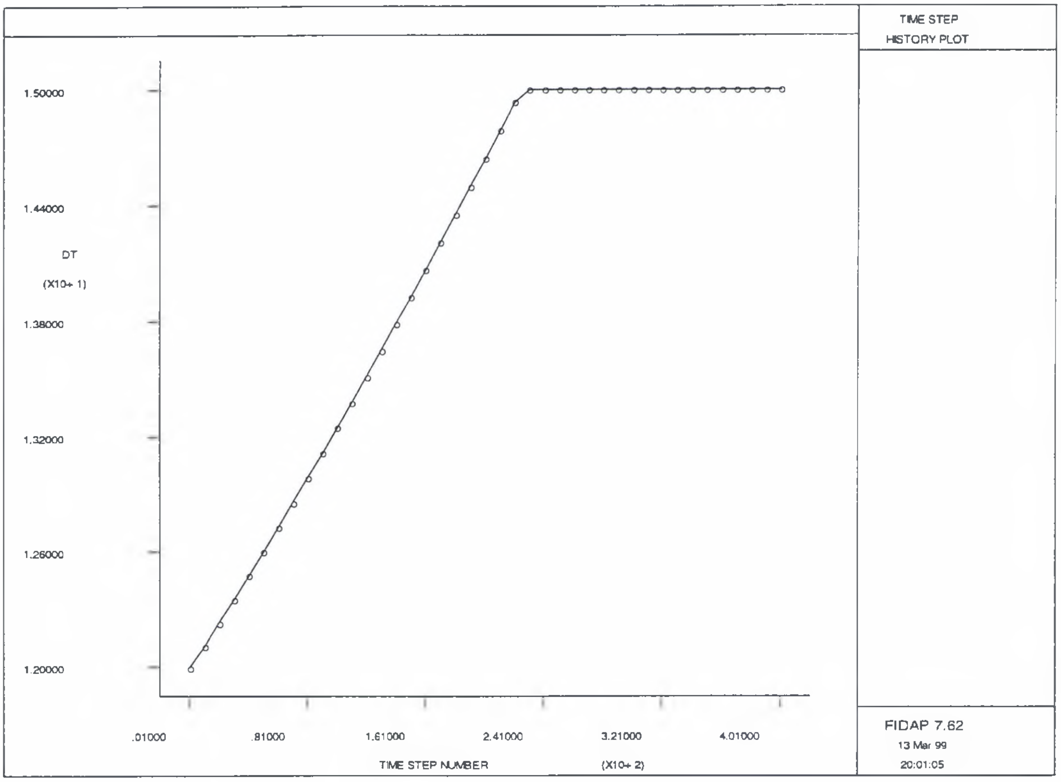


Figure 62. Time History Plot, ϕ: 0.57

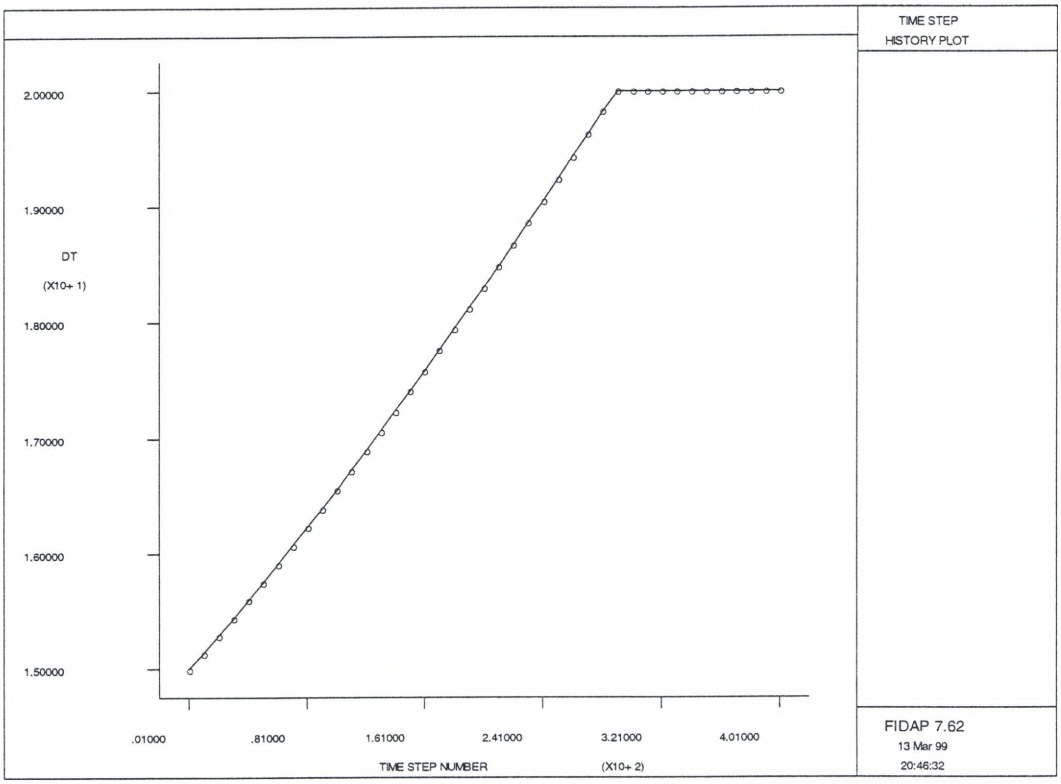


Figure 63. Time History Plot, ϕ: 0.75

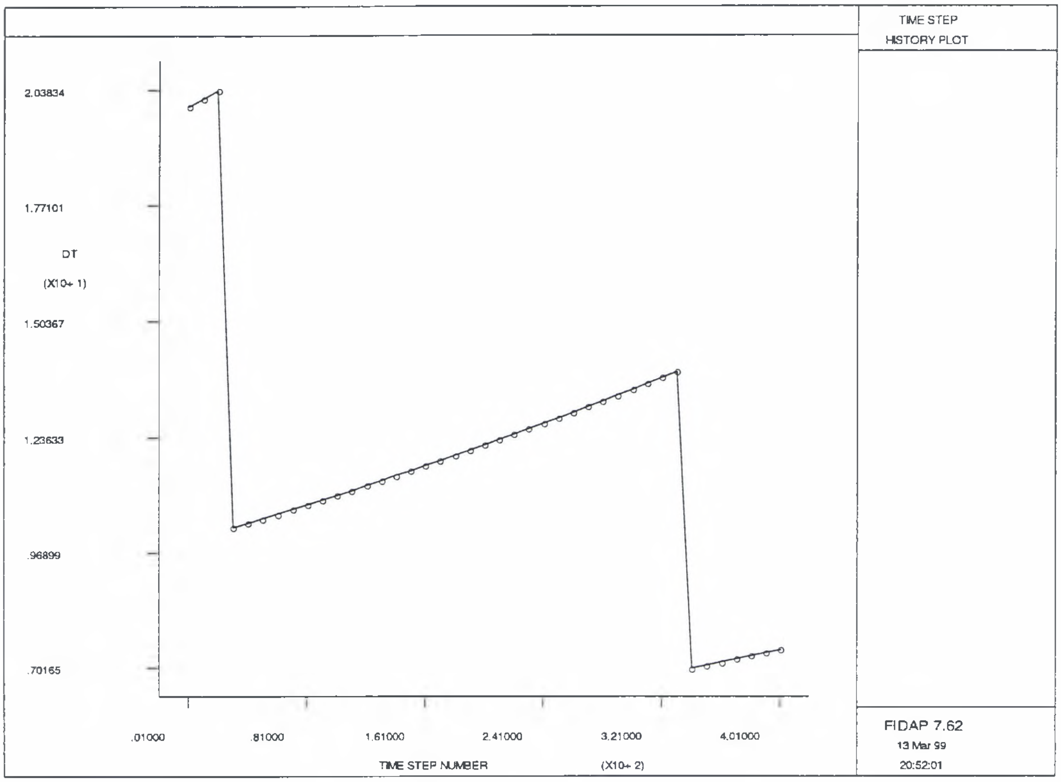


Figure 64. Time History Plot, ϕ: 0.85

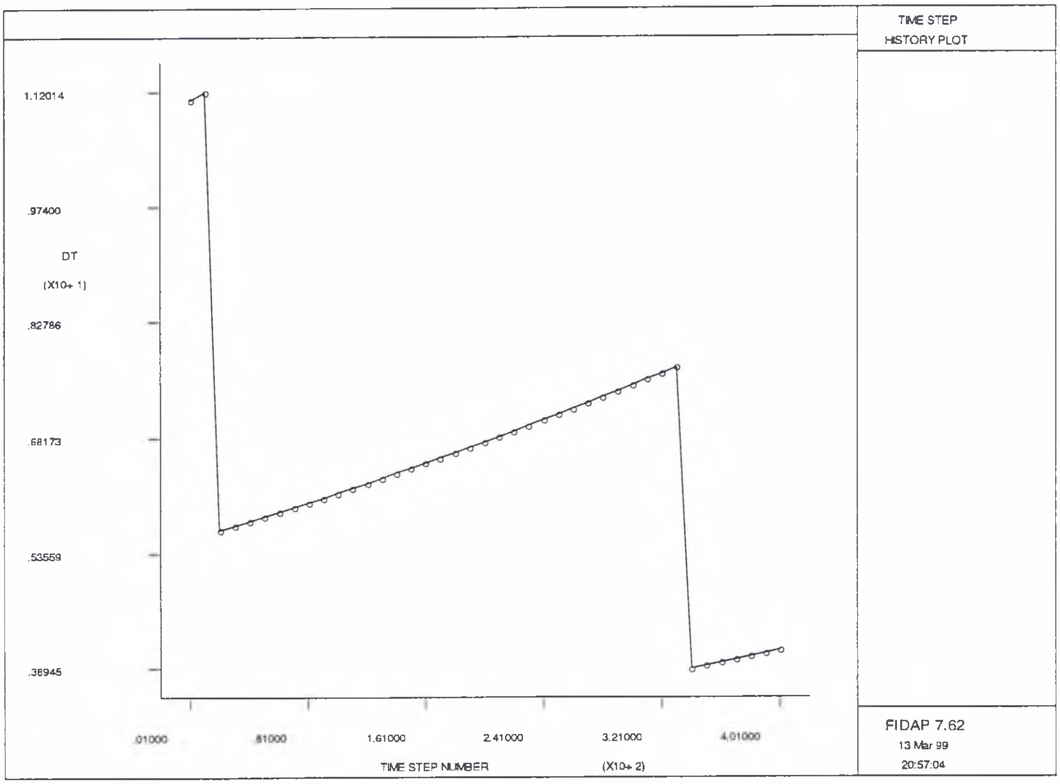


Figure 65. Time History Plot, ϕ: 0.93

APPENDIX D

NODAL DEPARTURE DATA

Table 7.

Nodal Departure Data for the Virtual Uni-Tank

Node	Departure [°F]
345	0.464
346	0.464
347	0.480
451	0.481
452	0.483
453	0.492
454	0.520
523	0.500
524	0.521
525	0.543
526	0.554
527	0.564
559	0.528
560	0.541
561	0.554
562	0.563
AVERAGE	0.516

APPENDIX E

TROUBLESHOOTING

Some basic guidelines and how-to's when handling turbulent modeling with FIDAP are summarized:

- **The simulation has diverged for κ and ε, but all other variables appear to converge properly.** This generally happens when poor initial guesses for the turbulent kinetic energy and dissipation terms are made. The best method of working around the problem, since no guidelines exist, is to let the number of sub-iterations and run times guide the user to a workable solution. For instance, run a mini-experiment with combinations of different magnitudes for κ and ε, searching for a pairing that converges rapidly with the fewest iterations possible. Select a fixed time goal, say, the first ten seconds of Uni-Tank cooling, and run the combinations of κ and ε. It is usually obvious by run times when the user is selecting good initial guesses.
- **General solution divergence.** The two main causes are improper time stepping and mesh concentration near walls or in corners. To differentiate between the two, examine the FIDAP output files (FDSTAT or FISTAT). If a text message like "solution not converging" is present, reduce the time step size. If mesh refinement is the problem, the program will cut off suddenly

during the analysis, with some reference to a "velocity, temperature, or pressure term being out of range in element ____." If possible, open FIPOST, and plot the mesh. Next, using the mesh command (See the FIPOST manual), enter the number of the element to view its location. If the divergent element is along a wall, increase the grading toward the wall. If the divergent element is toward the middle of the domain, either shift the grading toward the middle of the domain, or increase the mesh density. If grading is shifted toward the wall and divergence in an interior element results, increase the mesh density. Generally, as the modeling technique improves, the mesh can be made sparser.

- **Physical properties "out of range".** FIDAP recommends that physical properties be extrapolated outside the range of interest by ±10%. This presents an advantage in terms of making sub-iterations locally stable by giving a wide radius of convergence. Difficulties arise, particularly in the case of beer, when the density model is extrapolated beyond the freezing point and beyond the upper temperature limit. As the solver extrapolates into non-physical temperature regions, unreasonably high or low values of a physical property may be used by the solver as an initial guess. During this work, use of extrapolated density values above or below the temperature range of interest was disastrous, and delayed the completion of this work. The method that worked successfully for density was holding out-of-range values fixed instead of maintaining monotonic curvature. For instance, once the temperature exceeded the maximum temperature of interest, the solver was directed to hold the density constant regardless of the temperature estimate. The same was done for the minimum. If this procedure is not followed, overmixing and overcooling of the system was virtually guaranteed. This will be manifested as proper fluid behavior with respect to all variables except temperature, which will drop too rapidly.

- **Not enough memory resources to run the simulation.** Transient FEA on the Windows NT platform using FIDAP consumes a tremendous amount of computer HD memory and RAM. To note, over 1 GB of temporary memory (as a paging file) beyond the base RAM of 128 MB was assigned to the 300 MHz unit for error-free processing. All of the FIDAP text output for this work consumed > 3 GB of permanent storage. Thus, it is easy to envision that while studying transient flows, some expertise in resource management is warranted. There are four main causes of insufficient memory: algebraic solver, number of nodes, page file size, and post-processing of the results. *Algebraic solver.* Besides transience, the method of solution can also play a role in memory usage. Implicit solvers such as the Newton methods and successive substitution (12) solve the conservation laws *a priori* such that no information is written to disk. When many sub-iterations are required to converge, the temporary storage requirement rises dramatically. Researchers at FIDAP have developed a segregated iterative solver that uses significantly less memory, but takes longer to converge. It may be useful to investigate the method if the problem is physically large, or requires > 3000 nodes to mesh.

Number of nodes. For processing, FIDAP determines the size and composition of the matrix, then attempts to load the entire system of equations into *core memory*. If the matrix is larger than the amount of core memory, it will spill over into other temporary memory locations to make up the difference. Communication between iterations slows drastically. There are a couple of ways to alleviate the problem: *(a)* reduce the number of nodes to bring the entire system into core memory. To check if the entire matrix is in core, review the following statements in the FDSTAT output file located in the working directory after computation:

```
================================================================================
FIDAP  ( 7.62 )                        07 Dec 98  AT   20:18:10

THIS SOFTWARE IS A LICENSED PRODUCT OF FLUID DYNAMICS INTERNATIONAL, INC. AND
MAY ONLY BE USED ACCORDING TO THE TERMS OF THAT LICENSE ON THE SYSTEM
IDENTIFIED IN THE LICENSE AGREEMENT. COPYRIGHT (C) 1990 BY FLUID DYNAMICS
INTERNATIONAL, INC.  ALL RIGHTS RESERVED.
================================================================================
PROBLEM DEFINITION AND CONTROL INFORMATION INPUT, NO. OF ERRORS =     0
 PROBLEM DEFINITION IS:
  AXI-SYMMETRIC  INCOMPRESSIBLE  TRANSIENT      TURBULENT
  NONLINEAR      NEWTONIAN       MOMENTUM       SINGLEPHASE
  FIXED MESH     BUOYANCY
 EXTRAPOLATION IS ON
 RELAXATION METHOD IS RESIDUAL
 UPWINDING METHOD IS STREAMLINE
TIME FUNCTION INFORMATION INPUT,                    NO. OF ERRORS =    0
NODAL DATA INPUT,                                   NO. OF ERRORS =    0
CONSTRAINED D.O.F.S INPUT,                          NO. OF ERRORS =    0
VOLUMETRIC FORCES INPUT,                            NO. OF ERRORS =    0
COORDINATE SYSTEM DATA INPUT,                       NO. OF ERRORS =    0
INITIAL CONDITIONS INPUT,                           NO. OF ERRORS =    0
PROBLEM ELEMENT INFORMATION INPUT COMPLETED
DISTANCE DATA OR NORMAL-TANGENTIAL B.C.S STORED
MATRIX PROFILE STRUCTURE COMPUTED

TOTAL SYSTEM DATA
   NON SYMMETRIC MATRIX BLOCK STRUCTURE
      NUMBER OF EQUATIONS ............ (NQNS)   =                5611
      NUMBER OF MATRIX ELEMENTS ......... (NEMN)   =             1394469
      MEAN HALF BANDWIDTH ............ (MINBN)  =                124
      MAXIMUM HALF BANDWIDTH ........... (MAXBN)  =              140
      STORAGE FOR ONE BLOCK (INTEGER WORDS) . . . (MEMORN) =     2788944
      NUMBER OF BLOCKS .............. (NBLOKS) =                 1
      MAX. TOTAL STORAGE AVAILABLE (INTEGER WORDS) (MTOT)  =     24000000
      ADD. MEMORY FOR GLOBAL MATRIX TO BE IN CORE  (MCMCIN) =           0
       (0 = IN CORE)

FILE STRUCTURE DATA
-------------------
UNIT  NO. LOGICAL  LOGICAL REC.  NO. PHYSICAL  PHYSICAL REC.    KBYTES
        RECORDS       LENGTH       RECORDS        LENGTH
```

```
32      500     1332      46      1332      239
33       32     5642      32     65536       39

TOTAL CORE MEMORY REQUIRED FOR INPUT    PHASE =    86698
TOTAL CORE MEMORY REQUIRED FOR SOLUTION PHASE =  2877009
```

Observe the statement "ADD. MEMORY FOR GLOBAL MATRIX TO BE IN CORE (MCMCIN) = 0." Any value other than zero means the calculations will be slowed by data swapping. The user should become familiar with the content of the output files for troubleshooting purposes, *(b)* alternatively, review the FISOLVMEM sections of the FIDAP manuals (12), and select an alternative value for the base memory requirement. Any memory increase will be system dependent for each PC.

Page file size. FIDAP requires a large quantity of memory to store and solve the conservation equation in a matrix. When attempting to model engineering problems of this nature, the user should set the OS paging file to at least 512 MB. Space of 1 GB or higher is preferred.

Iterations between post-processing. After every iteration and sub-iteration, FIDAP will process the data and text stamp the results into an output file in permanent memory on the HD. This results in text-only output files > 20 MB. To reduce consumption, the user is guided to the POSTPROCESS(NBLOCKS=XX) and PRINTOUT(NBLOCKS=XX) commands in FIDAP (12).