

A TWO DIMENSIONAL, TWO FLUID  
MODEL FOR SODIUM BOILING  
IN LMBFR FUEL ASSEMBLIES

by

MARIO ROBERTO GRANZIERA

Engenheiro Mecanico, Universidade de São Paulo  
Mestre em Engenharia, Universidade de São Paulo

Submitted in Partial Fulfillment  
of the Requirements for the Degree of

DOCTOR OF PHILOSOPHY

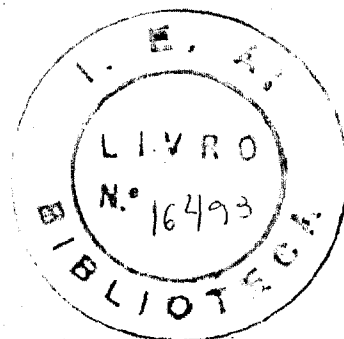
at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

MAY 1980

Massachusetts Institute of Technology

U. 1



Signature of Author.....  
Department of Nuclear Engineering, May, 1980

Certified by.....  
Thesis Supervisor

Accepted by.....  
Chairman, Department Committee

INSTITUTO DE PESQUISAS ENERGÉTICAS E NUCLEARES  
I. P. E. N.

A TWO DIMENSIONAL, TWO FLUID  
MODEL FOR SODIUM BOILING  
IN LMBFR FUEL ASSEMBLIES

by

MARIO ROBERTO GRANZIERA

Submitted to the Department of Nuclear Engineering  
on May 23, 1980 in partial fulfillment of the  
requirements for the degree of Doctor of Philosophy

ABSTRACT

A two dimensional numerical model for the simulation of sodium boiling transient was developed using the two fluid set of conservation equations. A semiimplicit numerical differencing scheme capable of handling the problems associated with the ill-posedness implied by the complex characteristic roots of the two fluid problem was used, which took advantage of the dumping effect of the exchange terms.

Of particular interest in the development of the model was the identification of the numerical problem caused by the strong disparity between the axial and radial dimensions of fuel assemblies. A solution to this problem was found which used the particular geometry of fuel assemblies to accelerate the convergence of the iterative technique used in the model.

The most important feature of the model was its ability to simulate severe conditions of sodium boiling, in particular flow reversal, which was shown in the tests performed with the model.

Three sodium boiling experiments were simulated with the model, with good agreement between the experimental results and the model predictions.

Thesis Supervisor: Mujid S. Kazimi

Assistant Professor of Nuclear Engineering

*Mujid S. Kazimi*

ACKNOWLEDGEMENTS

I wish to express my gratitude to Professor Mujid Kazimi who in the critical moments of this work gave me the help and advice much needed. To Doctor William Hinkle, who was always willing to give me his assistance.

Bruce Stewart introduced me to the concepts of Numerical Analysis and shared with me his extensive experience in the field. To him, my sincerest thanks.

I also wish to thank my colleagues in the Sodium Boiling Project, Michel, Mike, Greg, Rick, Andrei and Lou, for their assistance during these years we worked together.

The Comissao Nacional de Energia Nuclear provided the financial support which made possible this work.

## TABLE OF CONTENTS

Abstract	2
List of Figures	7
List of Tables	10
Nomenclature	11
Chapter 1: Introduction	13
1.1 LMFBR Safety Analysis	14
1.2 Characteristics of Numerical Models for Sodium Boiling	23
1.2.1 Dimensionality	23
1.2.2 Boundary Conditions: Pressure Vs. Inlet Velocity	26
1.2.3 Two Fluid Model	28
Chapter 2: The Conservation Equations and the Numerical Method	31
2.1 The Mass, Momentum and Energy Equations Averaged over a Control Volume	31
2.1.1 The Mass Equation	31
2.1.2 The Momentum Equation	34
2.1.3 The Energy Equation	46

2.2	The Finite Difference Equations	54
2.3	The Numerical Scheme	71
2.4	The Pressure Problem	86
2.5	Stability Analysis of the Numerical Method	93
Chapter 3: The Constitutive Equations and Functions of State		107
3.1	The Sodium Functions of State and Transport Properties	107
3.1.1	Saturation Temperature	107
3.1.2	Vapor Density	109
3.1.3	Liquid Density	110
3.1.4	Internal Energies	111
3.1.5	Transport Properties	112
3.2	Mass Exchange Rate	116
3.3	Momentum Exchange	126
3.4	Energy Exchange	131
3.4.1	Fuel Pin Heat Conduction	131
3.4.2	Fuel Pin Material Properties	137
3.4.3	Convective Heat Transfer Coefficient	139
3.4.4	Fuel Assembly Structure Model	145
3.4.5	Interphase Heat Transfer	145
Chapter 4: Experimental Tests Simulation		148
4.1	The SLSF P3A Experiment	148
4.2	One Dimensional Analysis of the P3A Experiment	164
4.3	The W1 Experiment	170
4.4	The GR19 Experiment	199

Chapter 5: Conclusions and Recommendations	207
5.1 Conclusions	207
5.2 Recommendations	208
References	211
Appendix A: NATOF-2D Input Data Manual	216
Appendix B: NATOF-2D Code Structure Description	228
Appendix C: NATOF-2D I/O Examples	237
Appendix D: NATOF-2D Program Listing	256

## LIST OF FIGURES

<u>No.</u>		
1.1	Possible Accident Paths and Lines of Assurance for a Potential CDA	17
1.2	Key Events and Potential Accident Paths for Unprotected Loss of Flow Accident	18
1.3	Key Events and Potential Accident Paths for Loss of Pipe Integrity Accident	19
1.4	Key Events and Potential Accident Paths for Unprotected Transient Overpower Accident	20
1.5	Key Events and Potential Accident Paths for Inadequate Natural Circulation Decay Heat	21
1.6	Key Events and Potential Accident Paths for Local Subassembly Accident	22
2.1	A Typical Cell Arrangement	59
2.2	Position Evaluation of Variables	60
2.3	Different Positions for the Radial Velocity	67
3.1	Condensation Coefficient as a Function of Pressure	119
3.2	Bubbly Flow Representation	121
3.3	Low Void Fraction Bubbly Flow Representation	121
3.4	Suppression Factor Vs. Reynolds Number	141
3.5	Reynolds Number Factor	142

4.1	Pin Number Location	156
4.2	P3A: Mass Flow Rate Vs. Time	157
4.3	P3A: Experimental Mass Flow Rate	158
4.4	P3A: Temperature Vs. Time, Central Channel	159
4.5	P3A: Temperature Vs. Time; Edge Channel	160
4.6	P3A: Axial Temperature Profile	161
4.7	P A: Radial Temperature Profile	162
4.8	Void Fraction Maps for the PBA Experiment	163
4.9	P3A-1D: Temperature Vs. Time	166
4.10	P3A-1D: Temperature Vs. Time	167
4.11	P3A-1D: Axial Temperature Profile	168
4.12	P3A: Comparison Between 1D and 2D; Mass Flow Rate	169
4.13	Typical Boiling Window Flow Decay for the W1 Test	177
4.14	W1: Temperature and Mass Flow Rate for Sequence 5	178
4.15	W1: Temperature and Mass Flow Rate for Sequence 6	179
4.16	W1: Temperature and Mass Flow Rate for Sequence 6a	180
4.17	W1: Axial Temperature Profile for Sequence 6a	181
4.18	W1: Temperature and Mass Flow Rate for Sequence 7	182
4.19	W1: Temperature and Mass Flow Rate for Sequence 7a	183
4.20	W1: Axial Temperature Profile for Sequence 7a	184
4.21	W1: Temperature and Mass Flow Rate for Sequence 7a'	185
4.22	W1: Axial Temperature Profile for Sequence 7a'	186
4.23	W1: Void Maps for Sequence 7a'	187
4.24	W1: Temperature and Mass Flow Rate for Sequence 7b'	188
4.25	W1: Axial Temperature Profile for Sequence 7b'	189

4.26	W1: Void Maps for Sequence 7b'	190
4.27	W1: Temperature and Mass Flow Rate for Sequence 3	191
4.28	W1: Axial Temperature Profile for Sequence 3	192
4.29	W1: Temperature and Mass Flow Rate for Sequence 4	193
4.30	W1: Axial Temperature Profile for Sequence 4	194
4.31	W1: Void Maps for Sequence 4	195
4.32	Temperature and Mass Flow Rate for 217-Pin Bundle Under Sequence 7b' Conditions	196
4.33	Axial Temperature Profile for 217-Pin Bundle Under Sequence 7b' Conditions	197
4.34	Void Maps for 217-Pin Bundle Under Sequence 7b' Conditions	198
4.35	GR19: Temperature Profile for .311 Kg/sec Mass Flow Rate	202
4.36	GR19: Temperature Profile for .265 Kg/sec Mass Flow Rate	203
4.37	GR19: Temperature Profile for .260 Kg/sec Mass Flow Rate	204
4.38	GR19: Quality Contours for .265 Kg/sec Mass Flow Rate	205
4.39	GR19: Quality Contours for .260 Kg/sec Mass Flow Rate	206
A.1	Cell Arrangement in the r-z Plane	217
A.2	Fuel Pin Numbering	218
A.3	Cell Arrangement for Fuel Pin Heat Conduction	219
B.1	NATOF-2D Subroutine Structure	229

## LIST OF TABLES

No.

1.1	Sodium Boiling Issues	16
3.1	Units Used in this Work and the Correspondent Usual Ones	108
4.1	SLSF-P3A Test Bundle Data	152
4.2	Assumed Non-Uniform Radial Power Distribution in P3A Test Bundle	154
4.3	Event Sequence Times of the P3A Experiment	155
4.4	W1 Test Bundle Data	173
4.5	Boiling Window Matrix for the W1 Experiment	175
4.6	Design Data for the GR19 Experiment	200

## NOMENCLATURE

$\alpha$	Void Fraction
$\rho$	Density
$e$	Internal Energy
$s$	Mass Exchange Rate
$P$	Pressure
$U$	Velocity
$f$	Friction Force
$g$	Gravity Acceleration
$V$	Volume
$A$	Area
$M$	Momentum Exchange Rate
$Q$	Heat Exchange Rate
$D$	Fuel Pin Diameter
$\Delta r$	Radial Mesh Spacing
$\Delta z$	Axial Mesh Spacing
$\Delta t$	Time Step Size

## SUPERSCRIPTS

N	Time Level
K	Iteration Level

## INTEGRALS

$\int_V dV$	Integral over the Volume Occupied by the Fluid Alone (Fuel Pin and Structure Excluded)
$\int_A dA$	Surface Integral
$\oint_A dA$	Integral Over a Closed Surface

## AVERAGES

$$X = \frac{1}{V} \int_V X(r,z) dV = \text{Volume Averaged Quantity}$$

$$X_A = \frac{1}{A} \int_V X(r,z) dA = \text{Surface Averaged Quantity}$$

## I INTRODUCTION

The growing public concern about the nuclear industry places an increasingly large emphasis on the safety aspects of nuclear reactor design. In particular, commercial size liquid metal cooled fast breeder reactors (LMFBR) with its large amount of plutonium fuel, combined with its inherent safety problems, namely the potentially positive void coefficient of reactivity, and the high chemical reactivity of the liquid metal coolant with air and water, must be designed, constructed and operated with large safety margins to assure that the public risk will be acceptably low.

In order to accomplish the stringent requirements of safety, designers must have a thorough understanding of the phenomena occurring in all possible reactor situations and the adequate analytical tools to correctly predict the reactor behavior in all possible situations.

The objective of this work is to provide an analytical model capable of predicting the transient sodium boiling in LMFBR fuel assemblies under realistic conditions. In order to situate the model proposed in this work in the broad field of sodium boiling a review of LMFBR safety analysis and a general description of the accidents of principal concern will be presented, followed by a review of the present status of analytical models currently available.

### 1.1 LMFBR Safety Analysis

The U. S. Fast Breeder Reactor Safety Program approach is to provide four levels of protection, which are aimed at reducing both the probability and consequences of a Core Disruptive Accident (CDA) [1]. These levels of protection are referred to as Lines of Assurance (LOA). Figure 1.1 illustrates the possible accident paths for a potential core disruptive accident.

- The first line of assurance aims at reducing the probability of occurrence of a serious accident. The emphasis is placed on quality assurance, inservice inspection and monitoring at the level of construction and operation, and at the level of design on providing a multilevel redundant plant protection system, which can quickly respond to faults and place the reactor in a safe shutdown condition without damage to the core [1,2].

The second line of assurance assumes that in spite of the measures taken in the first line, low probability but mechanistically possible events involving failure of the first line systems will occur [1]. The strategy in this line is to provide the reactor design with features which make the system respond inherently to accidents in a way which tends to maintain reactivity control and coolability, containing the damage to a limited number of fuel assemblies.

The third and fourth lines of assurance aim at limiting the consequences of a serious accident. It is assumed that the first two lines have failed and two subsequent events form a potential sequence leading to core disruption and release of radioactivity to the environment. The objective of these last lines is to make the consequences of a core disruption accident sufficiently limited by the plant containment capability which combined with the low probability of occurrence of the failure of the first and second lines of assurance makes the risk to the public acceptably small[1,2].

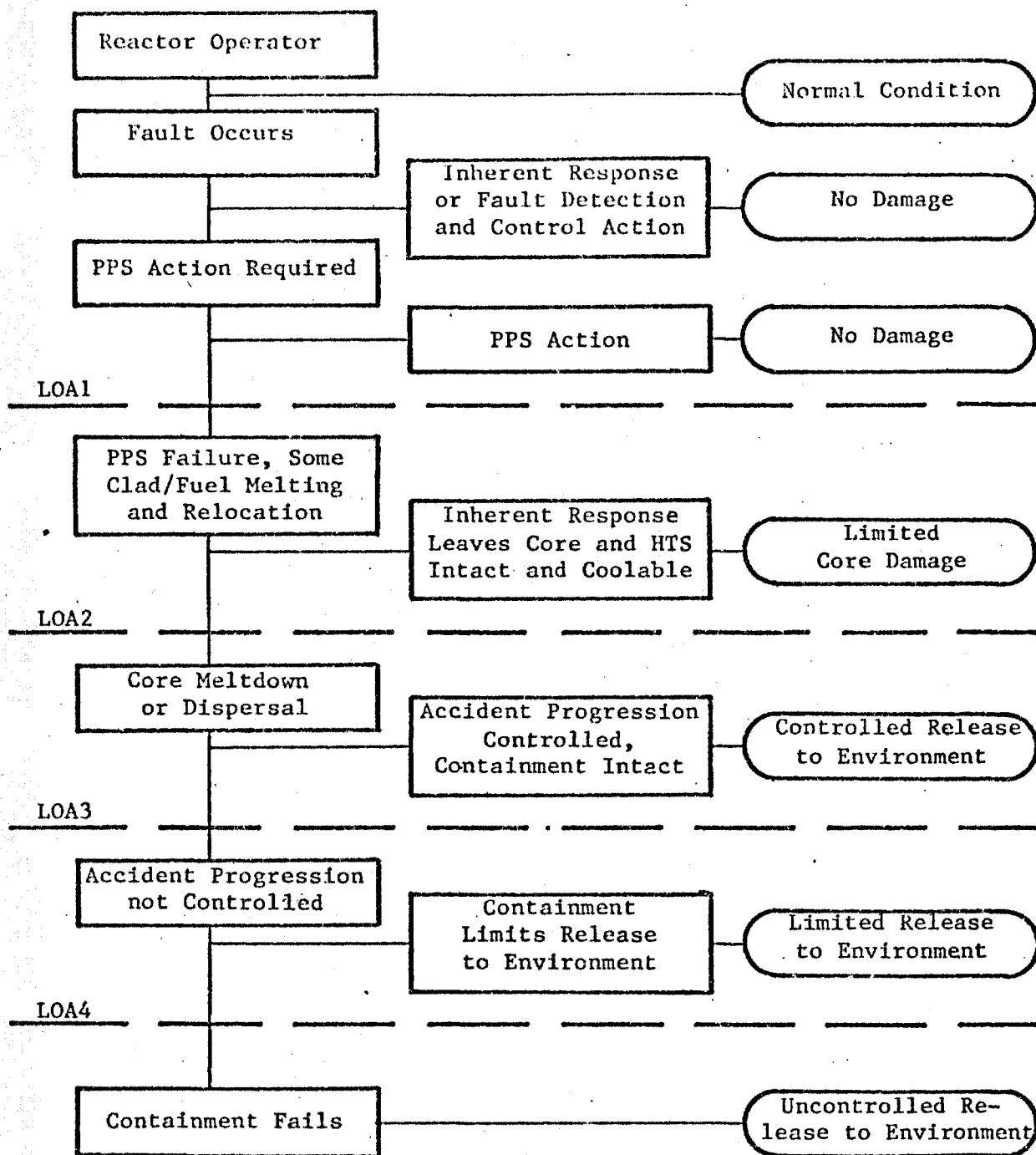
Some of the issues concerning the possible accident paths are still unresolved as are some of the phenomena involved in very low probability events.

In order to assess the importance of sodium boiling and two phase flow in the general picture of LMFBR safety analysis we reproduce from a compilation of the state of the art in sodium boiling by Hinkle [2] figures 1.2 through 1.6 illustrating the path of the most serious of the postulated accidents considered in LMFBR safety analysis. In all these accidents, the occurrence of sodium boiling and the stability of two phase flow assume a crucial role in determining the path, speed of events and final consequences. In table 1.1, also reproduced from reference [2], the technical issues which must be resolved related to sodium boiling are presented.

TABLE 1.1

Sodium Boiling Issues

Issue	Accidents
Effects of local blockage on single-phase flow and heat transfer — effects of location, size and composition; detectability.	Local Subassembly Accident
Stability of flow and heat transfer with local or bulk voiding due to fission gas release, molten fuel/coolant interaction and boiling — full power flow coastdown; flow and power decay following pipe rupture and scram; power increase/decrease at full flow and partial blockage; natural circulation at decay heat power level partial blockage at full power and flow	Unprotected Loss of Flow Accident Loss of Pipe Integrity Accident Unprotected Transient Overpower Accident Inadequate Natural Circulation Decay Heat Removal Accident Local Subassembly Accident
Transport of molten fuel by liquid and gas or vapor — effect of molten fuel/coolant interaction and voiding dynamics on tendency for fuel sweepout or relocation to form blockage and blockage propagation	Unprotected Loss of Flow Accident Unprotected Transient Overpower Accident Local Subassembly Accident
Reentry, rewetting and sustained cooling — effect of clad/fuel temperatures, molten fuel/clad and coolant interaction, extent of fuel/clad relocation and blockage	Unprotected Loss of Flow Accident Loss of Pipe Integrity Accident Unprotected Transient Overpower Accident Local Subassembly Accident



**Figure 1.1**  
**Possible Accident Paths and Lines of Assurance**  
**For a Potential CDA**  
**(From Reference 2)**

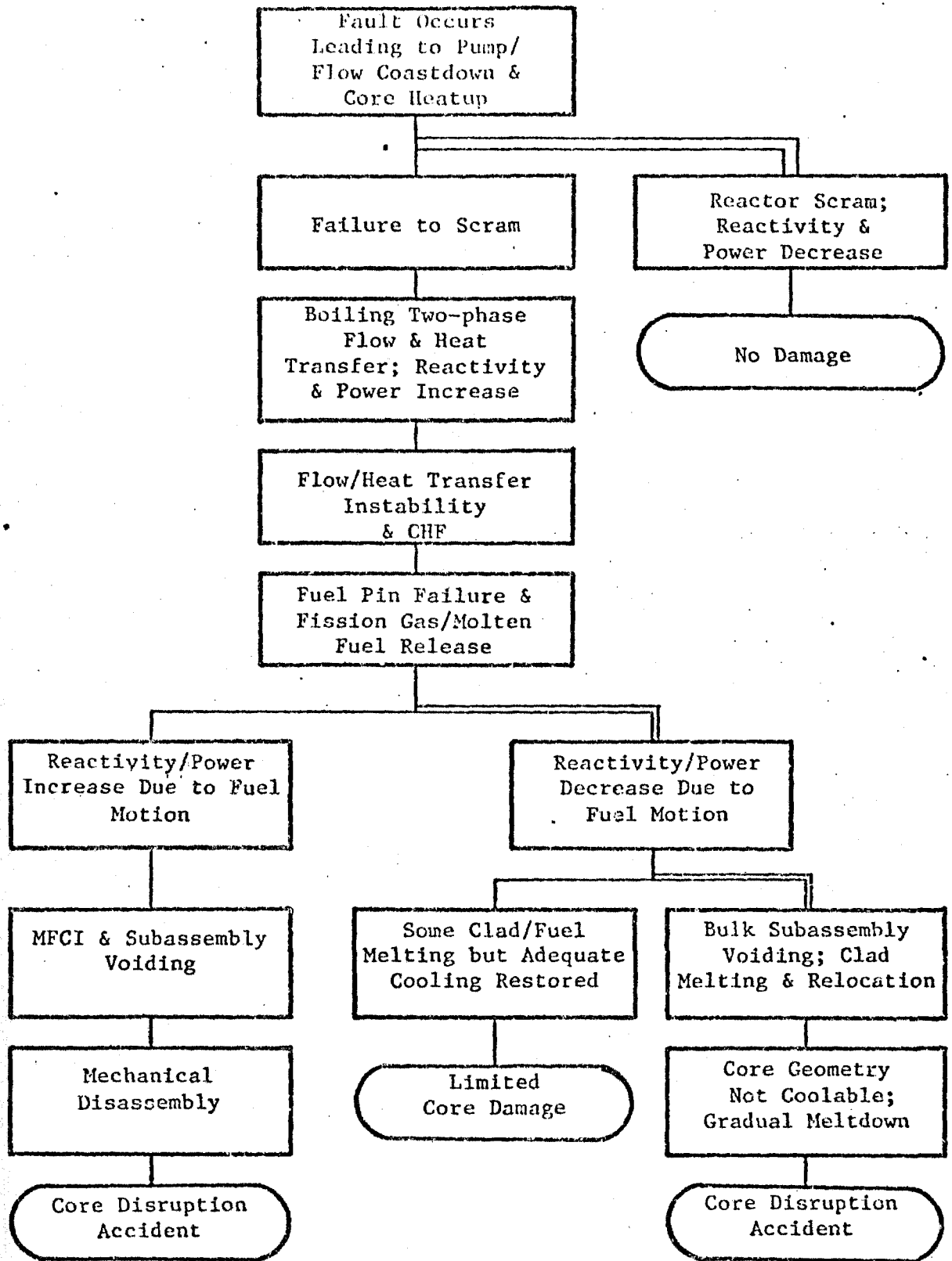


Figure 1.2

Key Events & Potential Accident Paths  
For Unprotected Loss of Flow Accident

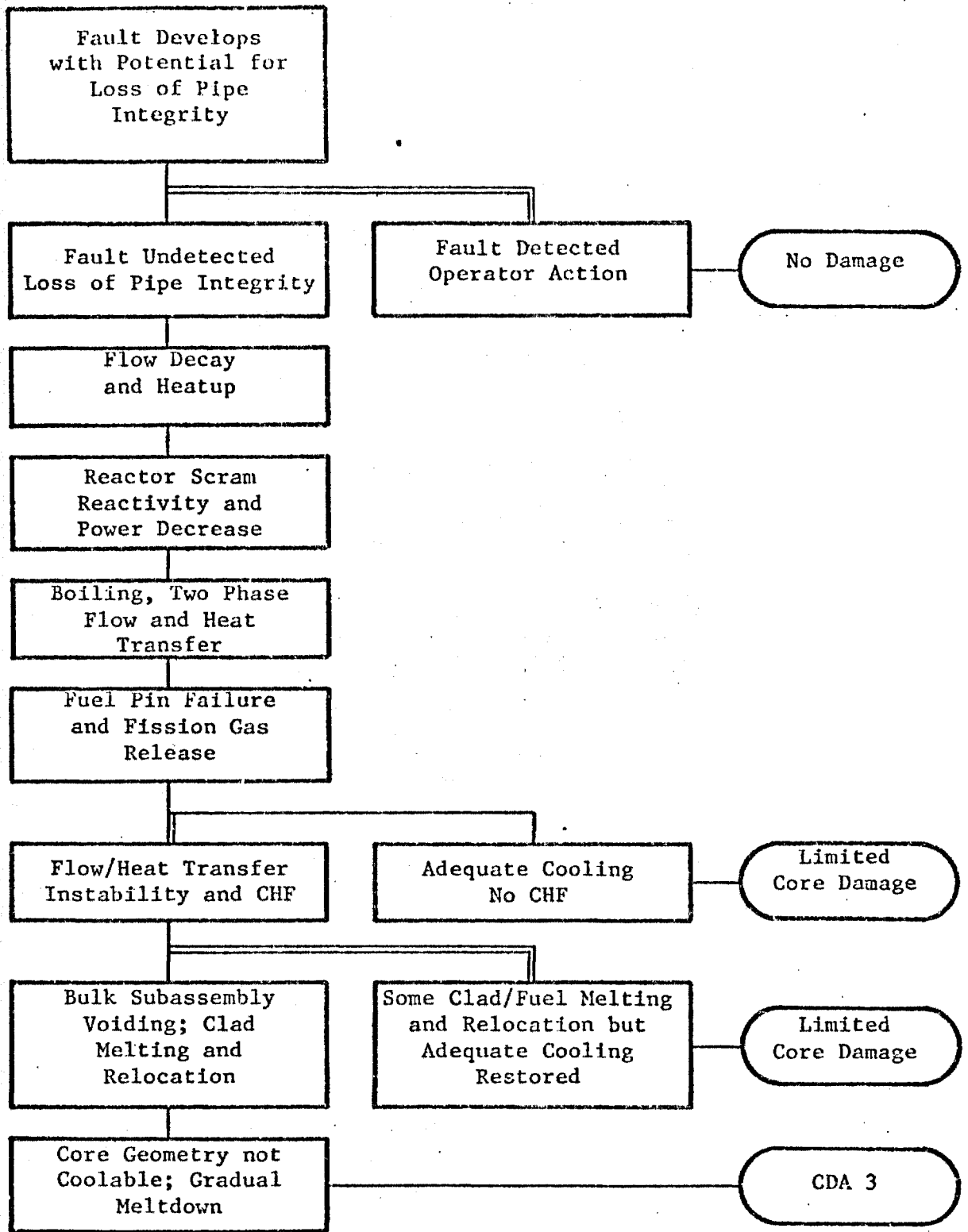


Figure 1.3

Key Events and Potential Accident Path  
 For Loss of Pipe Integrity Accident  
 (From Reference 2)

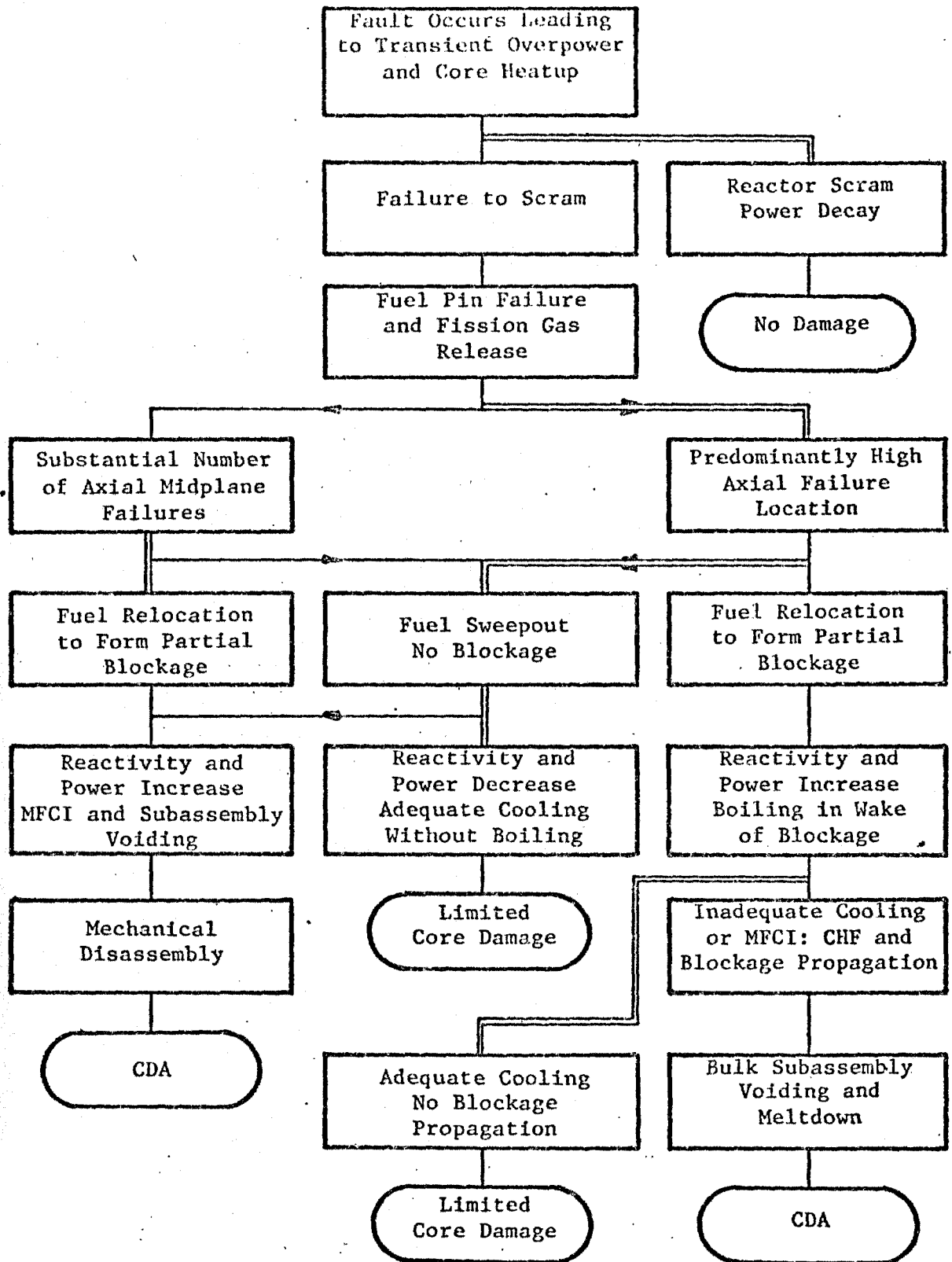


Figure 1.4  
 Key Events and Potential Accident Paths  
 For Unprotected Transient Overpower Accident  
 (From Reference 2)

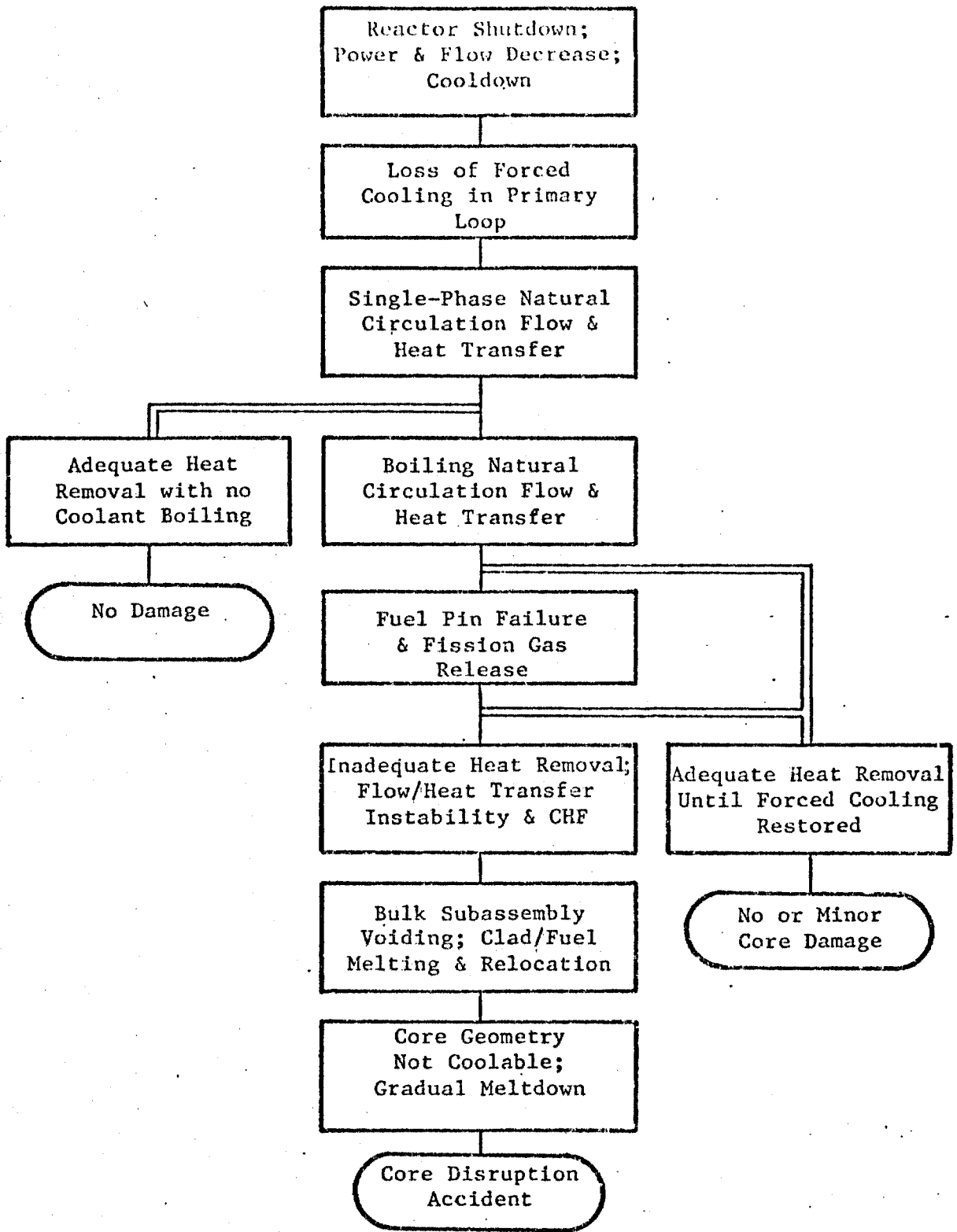


Figure 1.5

Key Events and Potential Accident Paths for  
Inadequate Natural Circulation Decay Heat Removal Accident

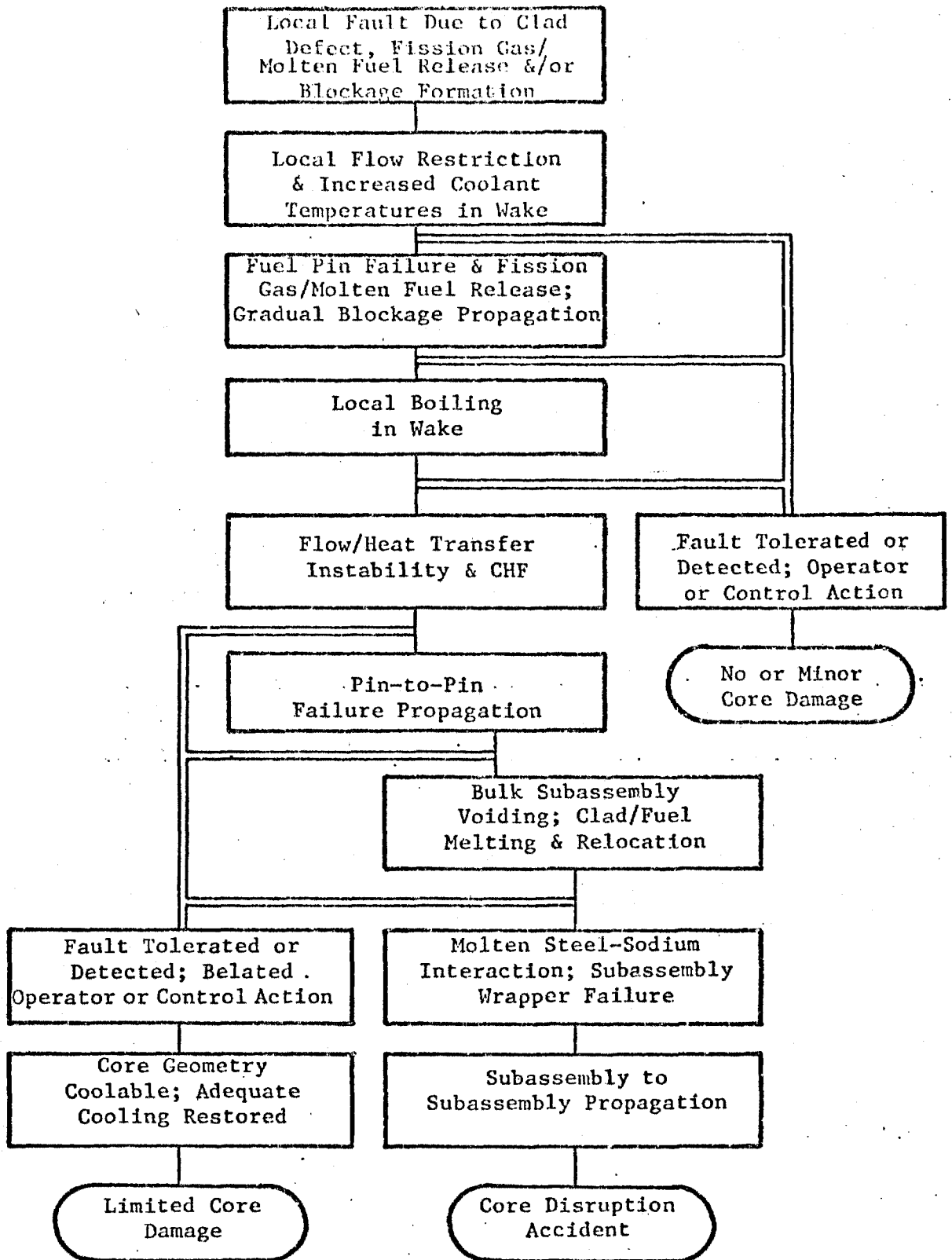


Figure 1.6

Key Events and Potential Accident Paths  
For Local Subassembly Accident

## 1.2 Characteristics of Numerical Models for Sodium Boiling

In the following paragraphs the most important characteristics of numerical models relevant to LMFBF fuel assembly fluid-dynamic analysis will be discussed, along with a comparison of the capabilities of the models presently available and the one proposed in this work.

### 1.2.1 Dimensionality

It is a well recognized fact that a non-flat radial temperature profile exists with steady-state conditions, as well as at the onset of boiling in loss of flow transients [3]. Calculations made for single phase flow with the COBRA III-C code [4] showed that a temperature difference as high as 450°F may exist between the central and peripheral channels in a typical FFTF fuel assembly (Figure 1.7). Obviously this temperature profile will force boiling to start in the central part of the fuel assembly and progress afterwards in the direction of the periphery. During this process, while part of the fuel assembly is under boiling, and the fuel pins in this region may eventually be suffering some damage, the periphery of the fuel assembly still maintains its coolability.

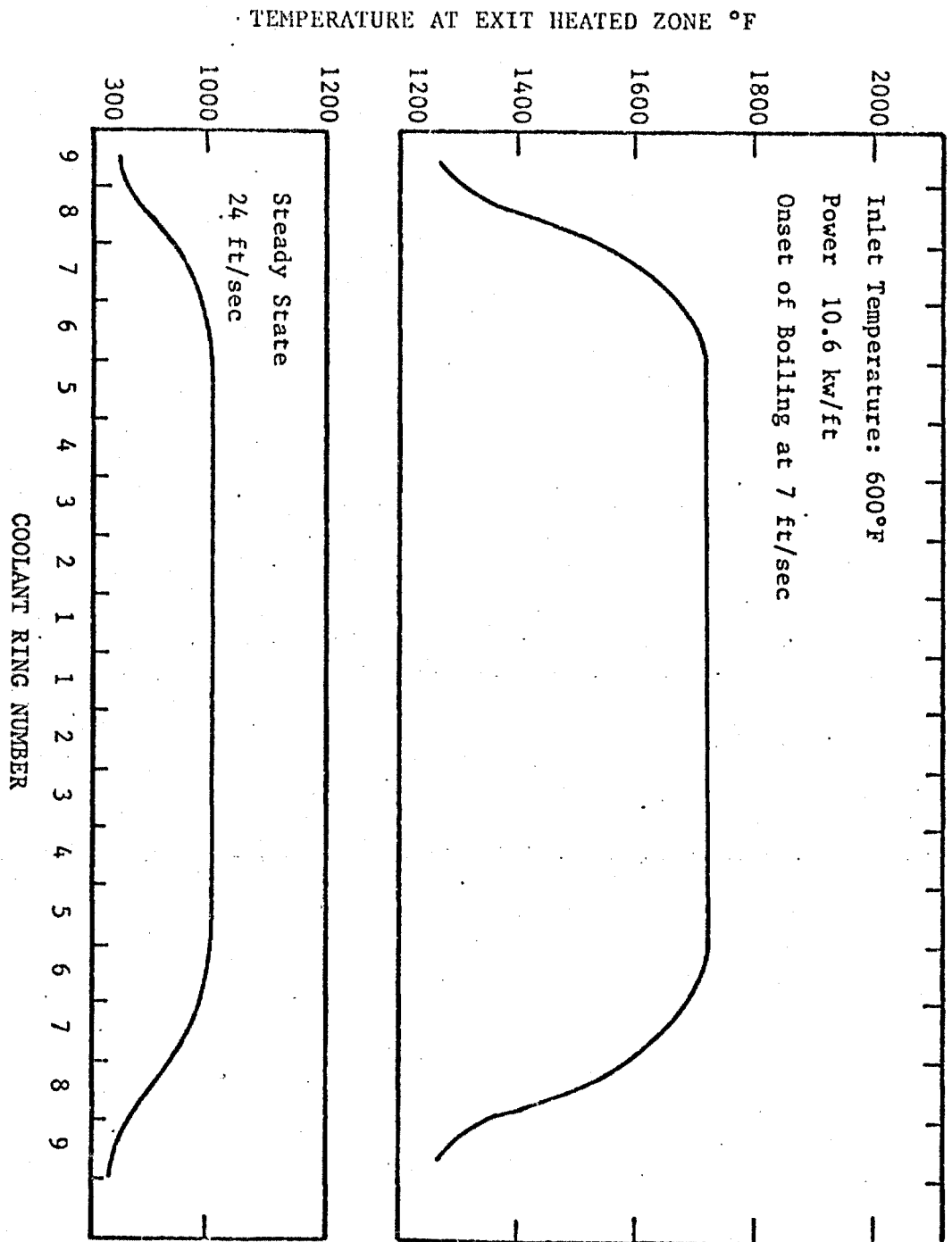


Figure 1.7

Radial Temperature Profile for 217 Pin Assembly

Here is a multidimensional effect the timing of which has a direct effect on the amount of damage resultant from the accident. Also, since this radial incoherence is an inherent design feature of all fuel assemblies, the radial void incoherence is expected to occur in every boiling transient.

Although this effect can be well represented by a two dimensional model, covering most of the transients of concern to LMFBR safety analysis, two cases present a non-radial symmetry, thus requiring a full three dimensional model for their representation, namely the transients with a non-uniform power profile and an asymmetrical flow blockage.

Considering the limited number of cases requiring a three dimensional model compared to those which can be analyzed with an axial-radial representation, and also considering the necessarily larger computational time required by a three dimensional model, it seems clear that a two dimensional model has definitive advantages.

As for the present situation in computational modeling, the only existing numerical model which can claim success in applications to sodium boiling is the SAS code, which is a one dimensional code [5]. Other codes such as the HEV-2D [6], COMMIX [7], BACHUS [8], to mention only a few of them, have encountered some difficulties in representing sodium boiling up to the point of flow reversal. Therefore a new code with two dimensional capability seems to be well situated.

### 1.2.2 Boundary Conditions: Pressure Vs. Inlet Velocity

The boundary conditions applied to the problem are strongly related to the numerical solutions used in the model. In this way, the marching technique, where the solution of the fluid-dynamic equations is obtained successively in planes along the direction of the main flow, can only operate with inlet velocity boundary condition, whereas the simultaneous pressure matrix inversion can work with both kinds of boundary conditions.

The advantages of the marching technique are its numerical stability for arbitrary large time steps combined with its relatively quick and straightforward numerical procedure. Its limitations lie in the assumptions necessary to the validity of the marching method: it requires that the flow be predominantly in one direction and always in that direction, making it impossible to analyze any kind of flow reversal. Also certain transport terms in the transverse momentum equations are ignored, whereas there are some doubts on the validity of these assumptions [9].

The simultaneous pressure matrix inversion method avoids the limitations of the marching technique at the price of using a smaller time step and a more laborious numerical solution. In this method, the solution of the fluid-dynamic equations is performed, at each time step, simultaneously for all mesh cells of the problem. In

general, this simultaneous solution can be reduced to a pressure matrix inversion. In this way, upstream propagations can be accounted for, and flow reversal transients can in principle be analyzed. The method does not impose any limitation on the number of conservation equations, therefore the choice of any model, from homogeneous equilibrium to the full two-fluid model is allowed.

The disadvantages of the method are that because of the large number of unknowns involved in the matrix inversion, a fully implicit differencing scheme becomes practically impossible, and a semi-implicit method, with its consequent limitation in time step size, becomes practically the only option.

Another problem with this method arises when used in conjunction with a multidimensional model. When the conservation equations are reduced to a pressure problem, the resultant pressure matrix becomes only marginally diagonally dominant, the diagonal dominance being provided only by the compressibility terms, which in some cases may be very small. In these situations the usual techniques of matrix inversion fail to produce a solution in a reasonable computational time, and special procedures must be introduced. Indeed, the ability of the model proposed in this work to produce results in a reasonable amount of computational time owes much to the special technique devised for this matrix inversion which is presented in section 2.4.

### 1.2.3 Two Fluid Model

In the early years of two phase flow modeling, much attention has been given to the homogeneous equilibrium model. This model describes the two phase flow in terms of average quantities, such as the density and velocity. In this way, these quantities are defined to represent an homogeneous mixture of the two phases (or two fluids).

There are situations during reactor core transients where the assumptions required for this modeling depart from reality, namely when either phase does not stay close to saturation conditions and more importantly when the phase velocities differ substantially. Attempts to circumvent these limitations were made with the introduction of semi-empirical correlations to describe the unequal phase velocities, the so-called slip correlations, and to allow non-saturation conditions. Because of the semi-empirical nature of these correlations, their accuracy is limited to the range of variables for which they were developed, and their generalization is restricted.

A new approach to overcome the limitations of the homogeneous equilibrium model was attempted with the drift flux model. This model stays in between the homogeneous equilibrium and two-fluid models in terms of the number of conservation equations employed. Although some variations on the particular set of equations composing the model exist, in general the drift flux model represents the two phase flow with a set of three mixture conservation equations plus two equations

for one of the pairs mass-momentum, mass-energy or energy-momentum for one of the phases.

In this model the sophistication in the direction of being closer to first principles is increased over the homogeneous mixture model, and so are the complications and size of the numerical solution technique. Indeed, there are some doubts about the computational time advantage of the drift flux model over the two-fluid model.

The two-fluid model represents the fluid flow with two complete, separate sets of conservation equations, treating individually the properties of both phases. Its clear advantage is that no assumption is made on the relationship between the properties of the two phases, and the most general situations can in principle be represented. The model requires constitutive expressions for the interaction between phases, namely the exchange of mass, momentum and energy. Unlike the slip correlation this constitutive expressions do not depend on circumstantial conditions of the particular flow situation, but on the physical principles of the transport phenomena involved.

Much work has to be done in the field of the constitutive relations required by the two fluid model, and the work presented here cannot claim to represent accurately the two phase flow phenomena without first solving the problems present in this area. Nonetheless, the model presented here can serve as a valuable tool in developing and testing the much needed constitutive relations for sodium two phase

flow. This subject will be readdressed in chapter 5, when the recommendations for future developments related to this model will be presented.

A final word has to be said about the controversial issue of the complex characteristic roots of the two fluid model, and its consequences to the stability of its numerical solution. Although this subject will be addressed at length in section 2.5, for the moment it is sufficient to point out the inadequacy of using the techniques of partial differential equations and numerical analysis developed for linear systems in analyzing the thermo-fluido dynamic equations, which are non-linear. Therefore, any conclusion on well-posedness of the two fluid model and the stability of its numerical solution has to be drawn from an analysis which takes into account all the characteristics of the model, in particular the damping effect of the interphase exchange terms.

The discussion of the porous body versus the subchannel approach is deliberately omitted here. It is our belief that this issue does not play any important role in the numerical treatment of reactor fluid flow. Indeed, it is possible to extend the porous body model to the limit of very small mesh cells or lump together subchannels to form larger ones. The two concepts overlap completely and no relevant distinction between them can be made in the numerical aspects of code development.

## II. THE CONSERVATION EQUATIONS AND THE NUMERICAL METHOD

### 2.1 The Mass, Momentum and Energy Equations Averaged over a Control Volume.

In this chapter the derivation of the differential and difference form of the conservation equations will be given. First all assumptions built into the model will be detailed, providing a clear picture of its limitations and range of validity. Secondly, the precise meaning of each term in the set of equations will be established. As will be seen later, this is particularly important with terms describing the geometry of the interacting cells.

For the sake of compactness, and to avoid being monotonous, details will be given for the equations of the vapor phase, mentioning only the final form of the equations of the liquid phase. This will cause no lack of understanding, since the two fluid model is completely symmetric with respect to the liquid and vapor phases.

#### 2.1.1 The Mass Equation

The mass equation has the form:

$$\begin{aligned} \frac{\partial}{\partial t} \int_V \alpha \rho_v dV + \int_{A_{z+}} - \int_{A_{z-}} \alpha \rho_v U_{vz} dA + \int_{A_{r+}} - \int_{A_{r-}} \alpha \rho_v U_{vr} dA = \\ = \int_V (S_e - S_c) dV \end{aligned} \quad (2.1)$$

The density as well as its first time derivative are assumed independent of position within the volume occupied by each phase separately and the void fraction is also assumed independent of position within the control volume. It follows that:

$$\int_V \alpha \rho_v dV = \langle \rho_v \rangle \int_V \alpha dV = \langle \rho_v \rangle \langle \alpha \rangle V \quad (2.2)$$

$$\int_A \alpha \rho_v U_v dA = \langle \alpha \rho_v U_v \rangle A \quad (2.2)$$

$$\int_V (s_e - s_c) dV = (\langle s_e \rangle - \langle s_c \rangle) V \quad (2.3)$$

We substitute these equations into our original mass equation, and we get

$$\begin{aligned} \frac{\partial}{\partial t} \left[ \langle \alpha \rangle \langle \rho_v \rangle \right] + \frac{A_z}{V} \left[ \langle \alpha \rho_v U_{vz} \rangle_{A_{z+}} - \langle \alpha \rho_v U_{vz} \rangle_{A_{z-}} \right. \\ \left. + \frac{A_{r+}}{V} \langle \alpha \rho_v U_{vr} \rangle_{A_{r+}} - \frac{A_{r-}}{V} \langle \alpha \rho_v U_{vr} \rangle_{A_{r-}} \right] = \langle s_e \rangle - \langle s_c \rangle \quad (2.5) \end{aligned}$$

In the above equation we have introduced the areas  $A_z$  and  $A_r$  bounding our control volume. The axial cross sectional area  $A_z$  poses no problem, since in the particular geometry of fuel assemblies of interest for LMFBR it is a constant throughout the axial length. For the radial cross sectional area however the same is not true. Here we have a highly position dependent area, not only in the macro scale, i.e., from one control volume to another, but also in the particular position with respect to the fuel pin rows chosen for this area.

So far this position can be chosen arbitrarily. Later it will be seen that for the averaged radial velocities in the momentum equations to be compatible with those in the mass and energy equations we must impose a precise value for this radial cross sectional area. The choice of this position is postponed until we have developed the momentum equations.

Finally it can be easily inferred that the liquid mass equation will undergo the same steps and present a similar form:

$$\begin{aligned}
 & \frac{\partial}{\partial t} \left[ (1 - \langle \alpha \rangle) \rho_l \right] + \frac{A_z}{V} \left[ \langle (1 - \alpha) \rho_l U_{lz} \rangle_{A_{z+}} \right. \\
 & - \langle (1 - \alpha) \rho_l U_{lz} \rangle_{A_{z-}} + \frac{A_{r+}}{V} \langle (1 - \alpha) \rho_l U_{lr} \rangle_{A_{r+}} \\
 & \left. - \frac{A_{r-}}{V} \langle (1 - \alpha) \rho_l U_{lr} \rangle_{A_{r-}} \right] = \langle S_c \rangle - \langle S_e \rangle \quad (2.6)
 \end{aligned}$$

### 2.1.2 The Momentum Equations

Following the same procedure used with the mass equation, the momentum equations in a control volume form are:

#### Axial Direction

$$\frac{\partial}{\partial t} \int_V \alpha \rho_v U_{vz} dV + \int_{A_{z+}} - \int_{A_{z-}} \alpha \rho_v U_{vz}^2 dA + \int_{A_{r+}} - \int_{A_{r-}} \alpha \rho_v U_{vz} U_{vr} dA$$

$$= \oint_{A_v} P \cdot \hat{k} \cdot \hat{n} dA = - \int_{A_w} f_{vz} dA - \int_V \alpha \rho_v g dV + \int_V M_{vz} dV \quad (2.7)$$

#### Radial Direction

$$\frac{\partial}{\partial t} \int_V \alpha \rho_v U_{vr} dV + \int_{A_{z+}} - \int_{A_{z-}} \alpha \rho_v U_{vz} U_{vr} dA + \int_{A_{r+}} - \int_{A_{r-}} \alpha \rho_v U_{vr}^2 dA$$

$$= \oint_{A_v} P \cdot \hat{r} \cdot \hat{n} dA = - \int_{A_w} f_{vr} dA - \int_V M_{vr} dV \quad (2.8)$$

To obtain the momentum equations in non-conservative form the mass equation (2.1) multiplied by  $\langle U_{vz} \rangle$  and  $\langle U_{vr} \rangle$  is subtracted

from equations 2.7 and 2.8 respectively:

$$\begin{aligned}
 & \frac{\partial}{\partial t} \int_V \alpha \rho_v U_{vz} dV - \langle U_{vz} \rangle \frac{\partial}{\partial t} \int_V \alpha \rho_v dV + \int_{A_{z+}} - \int_{A_{z-}} \alpha \rho_v U_{vz}^2 dA \\
 & - \langle U_{vz} \rangle \int_{A_{z+}} - \int_{A_{z-}} \alpha \rho_v U_{vz} dA + \int_{A_{r+}} - \int_{A_{r-}} \alpha \rho_v U_{vz} U_{vr} dA \\
 & \dot{=} \langle U_{vz} \rangle \int_{A_{r+}} - \int_{A_{r-}} \alpha \rho_v U_{vr} dA - \int_{A_v} P \hat{k} \cdot \hat{n} dA = \\
 & = - \int_{A_w} f_{vz} dA - g \int_V \alpha \rho_v dV + \int_V M_{vz} dV - \langle U_{vz} \rangle \int_V (s_e - s_c) dV \quad (2.9)
 \end{aligned}$$

$$\begin{aligned}
 & \frac{\partial}{\partial t} \int_V \alpha \rho_v U_{vr} dV - \langle U_{vr} \rangle \frac{\partial}{\partial t} \int_V \alpha \rho_v dV + \int_{A_{z+}} - \int_{A_{z-}} \alpha \rho_v U_{vz} U_{vr} dA \\
 & - \langle U_{vr} \rangle \int_{A_{z+}} - \int_{A_{z-}} \alpha \rho_v U_{vz} dA + \int_{A_{r+}} - \int_{A_{r-}} \alpha \rho_v U_{vr}^2 dA \\
 & - \langle U_{vr} \rangle \int_{A_{r+}} - \int_{A_{r-}} \alpha \rho_v U_{vr} dA - \int_{A_v} P \hat{r} \cdot \hat{n} dA = \\
 & = - \int_{A_w} f_{vr} dA + \int_V M_{vr} dV - \langle U_{vr} \rangle \int_V (s_e - s_c) dV \quad (2.10)
 \end{aligned}$$

With the previously stated assumption of position independence of the density and its time derivative the first pair of terms in both equations become:

$$\begin{aligned} & \frac{\partial}{\partial t} \int_V \alpha \rho_v U_{vz} dV - \langle U_{vz} \rangle \frac{\partial}{\partial t} \int_V \alpha \rho_v dV = \\ & = \int_V \left[ \alpha \rho_v \frac{\partial U_{vz}}{\partial t} + (U_{vz} - \langle U_{vz} \rangle) \frac{\partial \alpha \rho_v}{\partial t} \right] dV \\ & = \langle \alpha \times \rho_v \rangle \frac{\partial}{\partial t} \langle U_{vz} \rangle V \end{aligned} \quad (2.11)$$

and

$$\frac{\partial}{\partial t} \int_V \alpha \rho_v U_{vr} dV - \langle U_{vr} \rangle \frac{\partial}{\partial t} \int_V \alpha \rho_v dV = \langle \alpha \times \rho_v \rangle \frac{\partial}{\partial t} \langle U_{vr} \rangle V \quad (2.12)$$

Next consider the convective terms. We define:

$$\langle U_{vz} \rangle_{A_z} = \left[ \int_{A_z} \alpha \rho_v U_{vz}^2 dA \right] \cdot \left[ \int_{A_z} \alpha \rho_v U_{vz} dA \right]^{-1} \quad (2.13)$$

$$\langle U_{vr} \rangle_{A_z} = \left[ \int_{A_z} \alpha \rho_v U_{vz} U_{vr} dA \right] \cdot \left[ \int_{A_z} \alpha \rho_v U_{vz} dA \right]^{-1} \quad (2.14)$$

$$\langle \alpha \rho_v U_{vz} \rangle_{A_z} = \frac{1}{A_z} \int_{A_z} \alpha \rho_v U_{vz} dA \quad (2.15)$$

$$\langle \alpha \rho_v U_{vr} \rangle_{A_z} = \frac{1}{A_z} \int_{A_z} \alpha \rho_v U_{vr} dA \quad (2.16)$$

Assume  $U_{vz}$  is position independent in each axial cross sectional area  $A_z$ . Also assume that  $U_{vz}$  and  $U_{vr}$  are axially variable in such a way that:

$$\langle U_{vz} \rangle = \frac{1}{2} ( \langle U_{vz} \rangle_{A_{z+}} + \langle U_{vz} \rangle_{A_{z-}} ) \quad (2.17)$$

$$\langle U_{vr} \rangle = \frac{1}{2} ( \langle U_{vr} \rangle_{A_{z+}} + \langle U_{vr} \rangle_{A_{z-}} ) \quad (2.18)$$

or in other words, that these velocities have a linear axial variation.

The axial convective terms in both momentum equations become:

$$\begin{aligned} & \int_{A_{z+}} - \int_{A_{z-}} \alpha \rho_v U_{vz}^2 dA - \langle U_{vz} \rangle \int_{A_{z+}} - \int_{A_{z-}} \alpha \rho_v U_{vz} dA \\ & = \langle \alpha \rangle \langle \rho_v \rangle \langle U_{vz} \rangle \left[ \langle U_{vz} \rangle_{A_{z+}} - \langle U_{vz} \rangle_{A_{z-}} \right] \end{aligned} \quad (2.19)$$

$$\begin{aligned}
& \int_{A_{z+}} - \int_{A_{z-}} \alpha \rho_v U_{vz} U_{vr} dA - \langle U_{vr} \rangle \int_{A_{z+}} - \int_{A_{z-}} \alpha \rho_v U_{vz} dA = \\
& = \langle \alpha \rangle \langle \rho_v \rangle \langle U_{vz} \rangle \left[ \langle U_{vr} \rangle_{A_{z+}} - \langle U_{vr} \rangle_{A_{z-}} \right] A_z \quad (2.20)
\end{aligned}$$

Using the same procedure to the r-convective terms, we define:

$$\langle U_{vr} \rangle_{A_r} = \left[ \int_{A_r} \alpha \rho_v U_{vr}^2 dA \right] \cdot \left[ \int_{A_r} \alpha \rho_v U_{vr} dA \right]^{-1} \quad (2.21)$$

$$\langle U_{vz} \rangle_{A_r} = \left[ \int_{A_r} \alpha \rho_v U_{vr} U_{vz} dA \right] \cdot \left[ \int_{A_r} \alpha \rho_v U_{vr} dA \right]^{-1} \quad (2.22)$$

Following the same procedure taken with the terms averaged over  $A_z$ , we must find an expression relating the properties averaged over the radial area and over the entire control volume. The linear variation of  $U_{vz}$  from  $A_{r-}$  to  $A_{r+}$  can be assumed without imposing simplifications beyond the level that has been used up till now. The same is also true when it is assumed that  $U_{vr}$  is constant over each radial area  $A_r$ . But due to the particular geometry of fuel assemblies under consideration, it will not be realistic to make a linear variation

of  $U_{vr}$  assumption for any arbitrary radial area, since due to the presence of fuel pins, this radial area varies drastically with radial position. Instead,  $A_{r+}$  and  $A_{r-}$  must be chosen such that:

$$\langle U_{vr} \rangle = \frac{1}{2} (\langle U_{vr} \rangle_{A_{r+}} + \langle U_{vr} \rangle_{A_{r-}}) \quad (2.23)$$

Introduce the quantities  $U_r^*(r)$  and  $A_r^*(r)$ :

$$\langle U_{vr} \rangle_{A_r(r)} \cdot A_r(r) = U_r^*(r) A_r^* \cdot r \quad (2.24)$$

where  $A_r^* \cdot r$  is the average linearly radial dependent area. This is equivalent to smearing the fuel pins over the fuel assembly to produce an equivalent homogeneous porous body. The criterion to find  $A_r^*$  is to require the integral of  $A_r^* \cdot r$  over one unit cell be equal to the volume occupied by the fluid:

$$V_{cell}^k = \int_{r_k}^{r_k + \xi} A_r^* \cdot r \, dr = A_r^* \left[ \frac{(r_k + \xi)^2 - r_k^2}{2} \right] \quad (2.25)$$

where  $\xi = p \cdot \frac{\sqrt{3}}{2}$

The volume  $V_{\text{cell}}^k$  is:

$$V_{\text{cell}}^k = \left( p^2 \frac{\sqrt{3}}{2} - A_{\text{pin}} \right) \left( \frac{2k+1}{2} \right) \Delta z$$

where  $A_{\text{pin}}$  includes the transverse area of fuel pin wire wrap and other structural materials present

$$r_k = k p \frac{\sqrt{3}}{2}$$

From equation (2.25) we get:

$$A_r^* = \Delta z \frac{\sqrt{3}}{2} - \frac{A_{\text{pin}}}{p} \quad (2.26)$$

We now can make the more acceptable assumption that  $\langle U_{vr} \rangle_{A_r} A_r(r) = U_r^*(r) A_r^* \cdot r = \text{constant}$ . It follows:

$$\begin{aligned} \langle U_{vr} \rangle &= \frac{1}{\langle \alpha \rangle V} \int_V \alpha U_{vr} dV = \frac{1}{V} \int dr \int_{A_r} U_{vr} dA \\ &= \frac{1}{V} \int_{r_-}^{r_+} \langle U_{vr} \rangle_{A_r} A_r(r) dr \\ &= U_r^*(r^*) \cdot A_r^* \cdot r^* \cdot \frac{(r_+ - r_-)}{V} \quad (2.27) \end{aligned}$$

where  $r^*$  is any value between  $r_-$  and  $r_+$ . Let us choose  $r^*$  such that

$$A_r^* r^* (r_+ - r_-) = V \quad (2.28)$$

but from equation 2.25 we have:

$$V = \int_{r_-}^{r_+} A_r^* r \, dr = A_r^* \frac{r_+^2 - r_-^2}{2}$$

so it follows

$$r^* = \frac{r_+ + r_-}{2} \quad (2.29)$$

Substituting for  $r^*$  in equation 2.27 we have

$$\langle U_{vr} \rangle = U_r^*(r^*) \quad (2.30)$$

but since we assumed  $U_r^*$  is a linear function of  $r$  this is equivalent to

$$\langle U_{vr} \rangle = \frac{U_r^*(r_+) + U_r^*(r_-)}{2} \quad (2.31)$$

Going back to equation 2.24 we have:

$$\langle U_{vr} \rangle = \frac{1}{2} \left[ \langle U_{vr} \rangle_{A_{r+}} \frac{A_{r+}}{A_r^* r_+} + \langle U_{vr} \rangle_{A_{r-}} \frac{A_{r-}}{A_r^* r_-} \right] \quad (2.32)$$

Finally, the desired criterion for choosing the radial cross sectional area such that the averaging procedures taken with the mass, energy and momentum equations be compatible follows immediately if  $r_+$  and  $r_-$  are chosen to satisfy:

$$\begin{aligned} A_{r_+} &= A^* r_+ \\ \text{and} \quad A_{r_-} &= A^* r_- \end{aligned} \quad (2.33)$$

then the desired equation 2.23 is obtained

$$\langle U_{vr} \rangle = \frac{\langle U_{vr} \rangle_{A_{r_+}} + \langle U_{vr} \rangle_{A_{r_-}}}{2}$$

with these considerations, after a few algebraic steps, the  $r$  convective terms in the momentum equations become:

$$\begin{aligned} & \int_{A_{r_+}} - \int_{A_{r_-}} \alpha \rho_v U_{vz} U_{vr} dA - \langle U_{vz} \rangle \int_{A_{r_+}} - \int_{A_{r_-}} \alpha \rho_v U_{vr} dA = \\ & = \langle \alpha \rangle \langle \rho_v \rangle \langle U_{vr} \rangle \left( \langle U_{vz} \rangle_{A_{r_+}} - \langle U_{vz} \rangle_{A_{r_-}} \right) \frac{(A_{r_+} + A_{r_-})}{2} \end{aligned} \quad (2.34)$$

$$\int_{A_{r_+}} - \int_{A_{r_-}} \alpha \rho_v U_{vr}^2 - \langle U_{vr} \rangle \int_{A_{r_+}} - \int_{A_{r_-}} \alpha \rho_v U_{vr} dA =$$

$$\langle \alpha \rangle \langle \rho_v \rangle \langle U_{vr} \rangle \left( \langle U_{vr} \rangle_{A_{r+}} - \langle U_{vr} \rangle_{A_{r-}} \right) \left( \frac{A_{r+} + A_{r-}}{2} \right) \quad (2.35)$$

The remaining terms of the momentum equations are obtained by simple averages:

$$\oint_{A_v} P \cdot \hat{k} \cdot \hat{n} \, dA = A_z \langle \alpha \rangle \left( \langle P \rangle_{A_{z+}} - \langle P \rangle_{A_{z-}} \right) \quad (2.36)$$

$$\oint_{A_v} P \cdot \hat{r} \cdot \hat{n} \, dA = \left( \frac{A_{r+} + A_{r-}}{2} \right) \langle \alpha \rangle \left( \langle P \rangle_{A_{r+}} - \langle P \rangle_{A_{r-}} \right) \quad (2.37)$$

$$\int_{A_w} f_{vz} \, dA = \langle f_{vz} \rangle_{A_w} A_w$$

$$\int_{A_w} f_{vr} \, dA = \langle f_{vr} \rangle_{A_w} A_w$$

$$\int_v M_{vz} \, dV - \langle U_{vz} \rangle \int_v (s_e - s_c) \, dV = \langle M'_{vz} \rangle V$$

$$\int_v M_{vr} \, dV - \langle U_{vr} \rangle \int_v (s_e - s_c) \, dV = \langle M'_{vr} \rangle V$$

Where the terms  $M'$  include both the momentum exchange between phases due to friction and mass exchange. These terms will be analysed in detail in Chapter 3, when discussing the constitutive equations.

Equations 2.9 and 2.10 can now be rewritten as:

$$\begin{aligned}
 & \langle \alpha \rangle \langle \rho_v \rangle \left[ \frac{\partial}{\partial t} \langle U_{vz} \rangle + \frac{A_z}{V} \langle U_{vz} \rangle \left( \langle U_{vz} \rangle_{A_{z+}} - \langle U_{vz} \rangle_{A_{z-}} \right) + \right. \\
 & \left. + \left( \frac{A_{r+} + A_{r-}}{2V} \right) \langle U_{vr} \rangle \left( \langle U_{vz} \rangle_{A_{r+}} - \langle U_{vz} \rangle_{A_{r-}} \right) \right] - \\
 & - \frac{A_z}{V} \langle \alpha \rangle \left( \langle P \rangle_{A_{z+}} - \langle P \rangle_{A_{z-}} \right) \\
 & = - \frac{A_w}{V} \langle f_{vz} \rangle - \langle \alpha \rangle \langle \rho_v \rangle g - \langle M'_{vz} \rangle \quad (2.38)
 \end{aligned}$$

$$\begin{aligned}
 & \langle \alpha \rangle \langle \rho_v \rangle \left[ \frac{\partial}{\partial t} \langle U_{vr} \rangle + \frac{A_z}{V} \langle U_{vz} \rangle \left( \langle U_{vr} \rangle_{A_{z+}} - \langle U_{vr} \rangle_{A_{z-}} \right) + \right. \\
 & \left. + \left( \frac{A_{r+} + A_{r-}}{2V} \right) \langle U_{vr} \rangle \left( \langle U_{vr} \rangle_{A_{r+}} - \langle U_{vr} \rangle_{A_{r-}} \right) \right] - \\
 & - \left( \frac{A_{r+} + A_{r-}}{2V} \right) \langle \alpha \rangle \left( \langle P \rangle_{A_{r+}} - \langle P \rangle_{A_{r-}} \right) = \frac{A_w}{V} \langle f_{vr} \rangle - \langle M'_{vr} \rangle \quad (2.39)
 \end{aligned}$$

Similarly for the liquid phase:

$$\begin{aligned}
 (1 - \langle \alpha \rangle) \langle \rho_\ell \rangle & \left[ \frac{\partial}{\partial t} \langle U_{\ell z} \rangle + \frac{A_z}{V} \langle U_z \rangle \left( \langle U_{\ell z} \rangle_{A_{z+}} - \langle U_{\ell z} \rangle_{A_{z-}} \right) + \right. \\
 & \left. + \left( \frac{A_{r+} + A_{r-}}{2V} \right) \langle U_{\ell r} \rangle \left( \langle U_{\ell z} \rangle_{A_{r+}} - \langle U_{\ell z} \rangle_{A_{r-}} \right) \right] - \\
 & - \frac{A_z}{2} (1 - \langle \alpha \rangle) \left( \langle P \rangle_{A_{z+}} - \langle P \rangle_{A_{z-}} \right) = \\
 & = \frac{A_w}{2} \langle f_{\ell z} \rangle - (1 - \langle \alpha \rangle) \langle \rho_\ell \rangle g - \langle M'_{\ell z} \rangle \quad (2.40)
 \end{aligned}$$

$$\begin{aligned}
 (1 - \langle \alpha \rangle) \langle \rho_\ell \rangle & \left[ \frac{\partial}{\partial t} \langle U_{\ell r} \rangle + \frac{A_z}{V} \langle U_{\ell z} \rangle \left( \langle U_{\ell r} \rangle_{A_{r+}} - \right. \right. \\
 & \left. \left. - \langle U_{\ell r} \rangle_{A_{r-}} \right) + \left( \frac{A_{r+} + A_{r-}}{2V} \right) \langle U_{\ell r} \rangle \left( \langle U_{\ell r} \rangle_{A_{r+}} - \langle U_{\ell r} \rangle_{A_{r-}} \right) \right] - \\
 & - \left( \frac{A_{r+} + A_{r-}}{2V} \right) (1 - \langle \alpha \rangle) \left( \langle P \rangle_{A_{r+}} - \langle P \rangle_{A_{r-}} \right) = \\
 & = \frac{A_w}{V} \langle f_{\ell r} \rangle - \langle M'_{\ell r} \rangle \quad (2.41)
 \end{aligned}$$

### 2.1.3 The Energy Equations

Again we start by writing the energy conservation equation in control volume form:

$$\begin{aligned}
 & \frac{\partial}{\partial t} \int_V \alpha \rho_v (e_v + \frac{1}{2} U_v^2) dV + \int_{A_{z+}} - \int_{A_{z-}} \alpha \rho_v U_{vz} (e_v + \frac{1}{2} U_v^2) dA + \\
 & + \int_{A_{r+}} - \int_{A_{r-}} \alpha \rho_v U_{vr} (e_v + \frac{1}{2} U_v^2) dA = \\
 & = \int_V Q_v dV - \int_V \alpha \rho_v g U_{vz} dV - \int_{A_w} \vec{U}_v \cdot \vec{f}_v dA + \int_{A_v} P \cdot \hat{n} \cdot \vec{U}_v dA - \\
 & - \int_V P \frac{\partial \alpha}{\partial t} dV \tag{2.42}
 \end{aligned}$$

Before proceeding with the averaging process, some algebraic manipulations will be made in order to eliminate the kinetic energy terms. Subtract from equation 2.42 equations 2.9 multiplied by  $\langle U_{vz} \rangle$  and 2.10 multiplied by  $\langle U_{vr} \rangle$ , and rearranging the result it follows

$$\frac{\partial}{\partial t} \int_V \alpha \rho_v e_v dV + \int_{A_{z+}} - \int_{A_{z-}} \alpha \rho_v U_{vz} e_v dA + \int_{A_{r+}} - \int_{A_{r-}} \alpha \rho_v U_{vr} e_v dA +$$

$$\begin{aligned}
& \int_V \left[ \frac{\partial}{\partial t} \alpha \rho_v \frac{1}{2} U_v^2 - \langle U_{vz} \rangle \frac{\partial}{\partial t} \alpha \rho_v U_{vz} - \langle U_{vr} \rangle \frac{\partial}{\partial t} \alpha \rho_v U_{vr} \right] dV + \\
& + \int_{A_{z+}} - \int_{A_{z-}} \alpha \rho_v \left[ U_{vz} \frac{1}{2} U_v^2 - \langle U_{vz} \rangle U_{vz}^2 - \langle U_{vr} \rangle U_{vz} U_{vr} \right] dA + \\
& + \int_{A_{r+}} - \int_{A_{r-}} \alpha \rho_v \left[ U_{vr} \frac{1}{2} U_v^2 - \langle U_{vz} \rangle U_{vz} U_{vr} - \langle U_{vr} \rangle U_{vr}^2 \right] dA = \\
& = \int_V Q_v dV - \int_V P \frac{\partial}{\partial t} dV + \oint_{A_v} \left[ P \hat{n} \cdot \vec{u}_v - P \hat{n} \cdot \hat{k} \langle U_{vz} \rangle - \right. \\
& \left. - P \hat{n} \cdot \hat{r} \langle U_{vr} \rangle \right] dA - \int_{A_w} \left[ \vec{U}_v \cdot \vec{f}_v - \langle U_{vz} \rangle f_{vz} - \langle U_{vr} \rangle f_{vr} \right] dA - \\
& \int_V \left[ \alpha \rho_v g U_{vz} - \langle U_{vz} \rangle \alpha \rho_v g \right] dV - \int_V \left[ \langle U_{vz} \rangle M_{vz} + \langle U_{vr} \rangle M_{vr} \right] dV
\end{aligned} \tag{2.43}$$

We will turn our attention to the terms involving the kinetic energy. To avoid the trouble of carrying over the whole expression, we will call:

$$\int_V \left[ \frac{\partial}{\partial t} \alpha \rho_v \frac{1}{2} U_v^2 - \langle U_{vz} \rangle \frac{\partial}{\partial t} \alpha \rho_v U_{vz} - \langle U_{vr} \rangle \frac{\partial}{\partial t} \alpha \rho_v U_{vr} \right] dV = 1E$$

$$\begin{aligned}
1E = & \int_V \left\{ \alpha \rho_v \left[ \frac{\partial}{\partial t} \frac{1}{2} U_v^2 - \langle U_{vz} \rangle \frac{\partial}{\partial t} U_{vz} - \langle U_{vr} \rangle \frac{\partial}{\partial t} U_{vr} \right] + \right. \\
& \left. + \frac{\partial \alpha \rho_v}{\partial t} \left[ \frac{1}{2} U_v^2 - \langle U_{vz} \rangle U_{vz} - \langle U_{vr} \rangle U_{vr} \right] \right\} dV \quad (2.44)
\end{aligned}$$

It must be assumed that the spatial variation of  $U_{vz}$  and  $U_{vr}$  around their mean values are small or in other words, if  $U_{vz}$  is written as:

$$U_{vz}(z, r) = \langle U_{vz} \rangle + \epsilon(z, r)$$

then

$$\langle U_{vz}^2 \rangle = \frac{1}{V} \int_V \left[ \langle U_{vz} \rangle + \epsilon(z, r) \right]^2 dV = \langle U_{vz} \rangle^2 + \frac{1}{V} \int_V \epsilon^2(z, r) dV$$

The requirement that  $\frac{1}{V} \int_V \epsilon^2 dV$  be small compared to  $\langle U_{vz} \rangle^2$ ,

would lead to:

$$\langle U_{vz}^2 \rangle \approx \langle U_{vz} \rangle^2 \quad (2.45)$$

if we recall that  $U_v^2 = U_{vz}^2 + U_{vr}^2$ , equation 2.44 becomes:

$$\begin{aligned}
1E = & V \langle \alpha \rangle \langle \rho_v \rangle \left[ \frac{\partial}{\partial t} \frac{1}{2} \left( \langle U_{vz}^2 \rangle + \langle U_{vr}^2 \rangle \right) - \right. \\
& \left. - \langle U_{vz} \rangle \frac{\partial}{\partial t} \langle U_{vz} \rangle - \langle U_{vr} \rangle \frac{\partial}{\partial t} \langle U_{vr} \rangle \right] + \\
& + \left[ \frac{1}{2} \left( \langle U_{vz}^2 \rangle + \langle U_{vr}^2 \rangle \right) - \langle U_{vz} \rangle^2 - \langle U_{vr} \rangle^2 \right] \frac{\partial}{\partial t} \int_V \alpha \rho_v dV
\end{aligned}$$

and in view of equation 2.45 this becomes:

$$1E = - \frac{1}{2} \left( \langle U_{vz}^2 \rangle + \langle U_{vr}^2 \rangle \right) \frac{\partial}{\partial t} \int_V \alpha \rho_v dV \quad (2.46)$$

the convective terms of the kinetic energy equation, which will be called 2E and 3E are:

$$2E = \int_{A_{z+}} - \int_{A_{z-}} dA \alpha \rho_v \left[ U_{vz} \frac{1}{2} U_v^2 - \langle U_{vz} \rangle U_{vz}^2 - \langle U_{vr} \rangle U_{vz} U_{vr} \right] \quad (2.47)$$

$$3E = \int_{A_{r+}} - \int_{A_{r-}} dA \alpha \rho_v \left[ U_{vr} \frac{1}{2} U_v^2 - \langle U_{vz} \rangle U_{vz} U_{vr} - \langle U_{vr} \rangle U_{vr}^2 \right] \quad (2.48)$$

Define:

$$\langle U_v^2 \rangle_{A_z} = \langle U_{vz}^2 \rangle_{A_z} + \langle U_{vr}^2 \rangle_{A_z} = \int_{A_z} \alpha \rho_v U_{vz} U_v^2 dA / \int_{A_z} \alpha \rho_v U_{vz} dA$$

and equations 2.47 and 2.48 become:

$$\begin{aligned}
 2E = & \left[ \frac{1}{2} \langle U_v^2 \rangle_{A_{z+}} - \langle U_{vz} \rangle \langle U_{vz} \rangle_{A_{z+}} - \right. \\
 & \left. - \langle U_{vr} \rangle \langle U_{vr} \rangle_{A_{z+}} \right] \int_{A_{z+}} \alpha \rho_v U_{vz} dA - \\
 & - \left[ \frac{1}{2} \langle U_v^2 \rangle_{A_{z-}} - \langle U_{vz} \rangle \langle U_{vz} \rangle_{A_{z-}} - \right. \\
 & \left. - \langle U_{vr} \rangle \langle U_{vr} \rangle_{A_{z-}} \right] \int_{A_{z-}} \alpha \rho_v U_{vz} dA \quad (2.49)
 \end{aligned}$$

$$\begin{aligned}
 3E = & \left[ \frac{1}{2} \langle U_v^2 \rangle_{A_{r+}} - \langle U_{vr} \rangle \langle U_{vr} \rangle_{A_{r+}} - \right. \\
 & \left. - \langle U_{vz} \rangle \langle U_{vz} \rangle_{A_{r+}} \right] \int_{A_{r+}} \alpha \rho_v U_{vr} dA - \\
 & - \left[ \frac{1}{2} \langle U_v^2 \rangle_{A_{r-}} - \langle U_{vr} \rangle \langle U_{vr} \rangle_{A_{r-}} - \right. \\
 & \left. - \langle U_{vz} \rangle \langle U_{vz} \rangle_{A_{r-}} \right] \int_{A_{r-}} \alpha \rho_v U_{vr} dA \quad (2.50)
 \end{aligned}$$

and from equations 2.17, 2.18 and 2.23 23 get:

$$2E = - \frac{1}{2} \left[ \langle U_{vz} \rangle_{A_{z+}} \langle U_{vz} \rangle_{A_{z-}} + \langle U_{vr} \rangle_{A_{z+}} \langle U_{vr} \rangle_{A_{z-}} \right] \int_{A_{z+}} - \int_{A_{z-}} \alpha \rho_v U_{vz} dA$$

$$3E = - \frac{1}{2} \left[ \langle U_{vr} \rangle_{A_{r+}} \langle U_{vr} \rangle_{A_{r-}} + \langle U_{vz} \rangle_{A_{r+}} \langle U_{vz} \rangle_{A_{r-}} \right] \int_{A_{r+}} - \int_{A_{r-}} \alpha \rho_v U_{vr} dA$$

In view of the previous assumption that deviation of the velocities from their averages are small we get

$$\langle U_{vz} \rangle_{A_{z+}} \langle U_{vz} \rangle_{A_{z-}} \approx \langle U_{vz}^2 \rangle \text{ etc, and it follows}$$

$$2E = - \frac{1}{2} \langle U_v^2 \rangle \int_{A_{z+}} - \int_{A_{z-}} \alpha \rho_v U_{vz} dA \quad (2.51)$$

$$3E = - \frac{1}{2} \langle U_v^2 \rangle \int_{A_{r+}} - \int_{A_{r-}} \alpha \rho_v U_{vr} dA \quad (2.52)$$

Combining the terms 1E, 2E and 3E, and recalling equation 2.1, we have:

$$1E + 2E + 3E = - \frac{1}{2} (\langle S_e \rangle - \langle S_c \rangle) \langle U_v^2 \rangle \quad (2.53)$$

We proceed by noting that some terms in equation 2.43 will vanish upon the performance of the integrals. These terms are:

$$\int_V \left[ \alpha \rho_v g U_{vz} - \langle U_{vz} \rangle \alpha \rho_v g \right] dV = 0 \quad (2.54)$$

$$\oint_{A_v} \left[ P \hat{n} \cdot \vec{U}_v - P \hat{n} \cdot \hat{k} \langle U_{vz} \rangle - P \hat{n} \cdot \hat{r} \langle U_{vr} \rangle \right] dA = 0 \quad (2.55)$$

$$\int_{A_w} \left[ \vec{U}_v \cdot \vec{f}_v - \langle U_{vz} \rangle f_{vz} - \langle U_{vr} \rangle f_{vr} \right] dA = 0 \quad (2.56)$$

The next step would be to define average properties and obtain the final form of the energy equation. Since this procedure is completely similar to that used for the mass equation, only the resultant energy equations are presented here:

$$\begin{aligned} \frac{\partial}{\partial t} \left[ \langle \alpha \rangle \langle \rho_v \rangle \langle e_v \rangle \right] + \frac{A_z}{V} \left[ \langle \alpha \rho_v e_v U_{vz} \rangle_{A_{z+}} - \langle \alpha \rho_v e_v U_{vz} \rangle_{A_{z-}} \right] \\ + \frac{A_{r+}}{V} \langle \alpha \rho_v e_v U_{vr} \rangle_{A_{r+}} - \frac{A_{r-}}{V} \langle \alpha \rho_v e_v U_{vr} \rangle_{A_{r-}} \\ = \langle Q_v \rangle - \langle P \rangle \frac{D\alpha}{Dt} - \langle Q_{lv} \rangle \end{aligned} \quad (2.57)$$

where the energy exchange between phases has been grouped under the term  $Q_{lv}$ .

For the liquid phase the energy equation is:

$$\begin{aligned}
 & \frac{\partial}{\partial t} \left[ (1 - \langle \alpha \rangle) \langle \rho_l \rangle \langle e_l \rangle \right] + \frac{A_z}{V} \left[ \langle (1 - \alpha) \rho_l e_l U_{lz} \rangle_{A_{z+}} - \langle (1 - \alpha) \rho_l e_l U_{lz} \rangle_{A_{z-}} \right] \\
 & + \frac{A_{r+}}{V} \langle (1 - \alpha) \rho_l e_l U_{lr} \rangle_{A_{r+}} - \frac{A_{r-}}{V} \langle (1 - \alpha) \rho_l e_l U_{lr} \rangle_{A_{r-}} = \\
 & = \langle Q_l \rangle + \langle P \rangle \frac{D\alpha}{Dt} + \langle Q_{lv} \rangle \quad (2.58)
 \end{aligned}$$

## 2.2 The Finite Difference Equations

Having established the properly averaged differential equations, the next step is to approximate the conservation equations by a set of algebraic equations suitable for the numerical solution. Before choosing any particular scheme, it is appropriate to discuss in general terms the various applicable finite difference approaches and identify the kind of problems we expect to solve with our model.

For the spacial discretization very little can be said in general: the idea to be followed is that one can find the best spacial differenciation to suit a particular time discretization.

There are three broad categories concerning the time level at which the variables are to be evaluated: fully implicit, fully explicit and semi-implicit (or semi-explicit). Associated with each of these categories there is a stability criterion which will relate the time step size with the characteristic roots of the set of equations.

A fully implicit method is the one in which all spacial derivatives, as well as all the exchange terms are evaluated at the new time level. With this time discretization it is possible in general to find a spacial arrangement which makes the whole method unconditionally stable, thus enabling the problem to be solved with a time step as large (or as small) as desired. As an exemple, a one dimensional mass equation in this scheme is:

$$\frac{\alpha_i^{n+1} \rho_{vi}^{n+1} - \alpha_i^n \rho_{vi}^n}{\Delta t} + \frac{(\alpha \rho_v U_v)_{i+\frac{1}{2}}^{n+1} - (\alpha \rho_v U_v)_{i-\frac{1}{2}}^{n+1}}{\Delta z} = S_i^{n+1}$$

Note that the convective terms are evaluated at different locations than the other terms.

Since they are evaluated at the new time level, they are unknown, which means the solution at one particular cell is coupled to the solution at its neighbors, thus requiring the numerical solution to be made simultaneously in all locations and all variables. This poses a very complex matrix inversion problem, using relatively large computational times.

On the other side is the fully explicit method, in which the spacial derivatives and the exchange terms are evaluated at the old time level. The mass equation would look like:

$$\frac{(\alpha \rho_v)_i^{n+1} - (\alpha \rho_v)_i^n}{\Delta t} + \frac{(\alpha \rho_v U_v)_{i+\frac{1}{2}}^n - (\alpha \rho_v U_v)_{i-\frac{1}{2}}^n}{\Delta z} = S_i^n$$

We can see in this case the terms which are evaluated at locations other than the cell  $i$  are in the old time level, and so they are known. The solution at each cell is independent of its neighbors and the numerical solution of the set of equations will be relatively simple. The penalty for that simple solution is that the stability criterion for this category is severe, requiring in general very small time step sizes. Typically it would require that a pressure or temperature perturbation travel no farther than one mesh space in one time step, or in mathematical words it would require:

$$\Delta t < \frac{\Delta z}{c}$$

where  $c$  is a sonic speed. Typical values are on the order of  $10^{-2}$  to  $10^{-1}$  m for the mesh spacing and  $10^3$  m/sec for the sonic speed. Thus, we can expect to be limited to time step sizes of the order of  $10^{-4}$  or  $10^{-5}$  seconds with this method.

In between these two extremes, the semi implicit methods are those in which some terms are treated implicitly while others explicitly. If liquid convection is treated explicitly, the time step restriction for this class of schemes is the convective limit

$$\Delta t < \frac{\Delta z}{v}$$

where  $v$  is the phase velocity.

The general idea behind this category is to devise a particular balance between implicit and explicit terms which would make the solution of the particular set of equations simple compared to that of the fully implicit method, combined with a less restrictive stability criterion compared to the fully explicit method.

Here a very large number of possibilities exist, and a general analysis would prove to be of little value since what might be the best solution for a particular problem may not be a good one for another.

Thus, instead of a general study of semi-implicit methods we will just analyse the particular scheme used in our model and show the motivation for its choice. One important consideration for this choice is the time scale of the phenomena to be analysed with the model.

A typical loss of flow two-phase transient lasts from onset of boiling to flow reversal for about one second. Therefore a sufficient detailed description of this transient requires that the solution scheme

produces information with a time interval of about  $10^{-1}$  to  $10^{-2}$  second. In this kind of transient we can expect to have axial velocities on the order of 10 meter per second. Thus using a axial mesh spacing on the order of  $10^{-1}$  meter, the convective limit  $\Delta z/v$  characteristic of the semi-implicit method will be of the same order of magnitude of the time interval in which we want information, and a method with such time step limitation would fit perfectly our purpose.

Much longer simulations can be expected in the case of natural convection decay heat removal. But in this class of phenomena the phase velocities would be much smaller, and again in this case, a time step restriction connected to the phase velocity is of the same order of magnitude of the required information interval.

Therefore, with the semi-implicit method we take advantage of a simpler solution of the fluid flow equations, with smaller number of operations performed per time step without increasing the number of time steps required to cover the whole transient.

After this brief outline of the general features of numerical methods, we proceed with a detailed description of the particular scheme adopted for the model, explaining how this particularly fits our set of equations and insures the stability of the method.

We start by dividing the fuel assembly to be simulated into a two dimensional  $r - z$  grid. To allow flexibility of application and a more efficient allocation of time and memory space, this division is made to accept variable mesh spacing in both directions, with the sole restriction that at each radial or axial level the mesh spacing corre-

sponding to that direction remains the same for all cells in that level. With this restriction, each cell, except the boundaries cells, will have only one neighbor at each of its four sides. Figure 2.1 shows a typical arrangement of cells.

All unknowns of the problem, with the exception of the velocities, are evaluated at the center of the mesh cells, while the velocities are evaluated at the faces of these cells. Figure 2.2 shows a typical mesh cell where this is illustrated. This figure also shows the subscript convention used in the difference equations. In this convention subscripts  $i$  and  $j$  indicate position in the center of a cell along the axial and radial axis respectively, while subscripts  $i + \frac{1}{2}$  and  $j + \frac{1}{2}$  indicates position at the faces of the cells corresponding to the  $z$  and  $r$  directions respectively.

Superscripts are used to indicate the time level in which the variables are evaluated. Superscript  $n$  indicates evaluation at the old time level, thus corresponding to a known quantity, and  $n + 1$  indicates a variable in the new time level, to be determined in this step. The exchange terms, which are in general function of both new and old time variables do not carry any superscript. They will be discussed at length in Chapter 3.

With these conventions established, the difference form of the mass and energy equations, which are differentiated about the center of the mesh cells are:

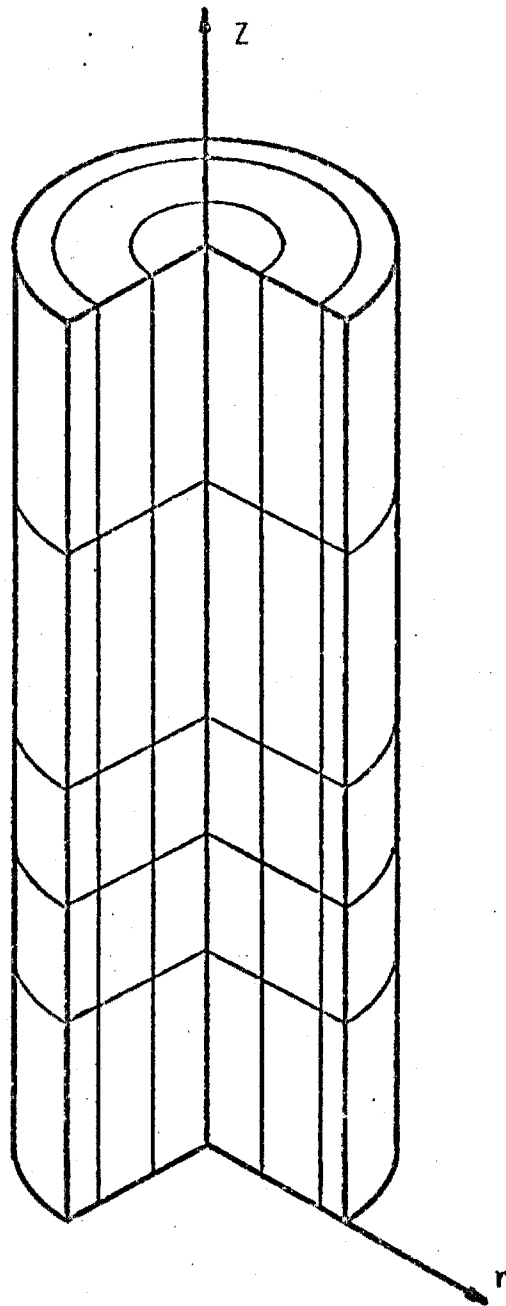


Figure 2.1 A Typical Cell Arrangement

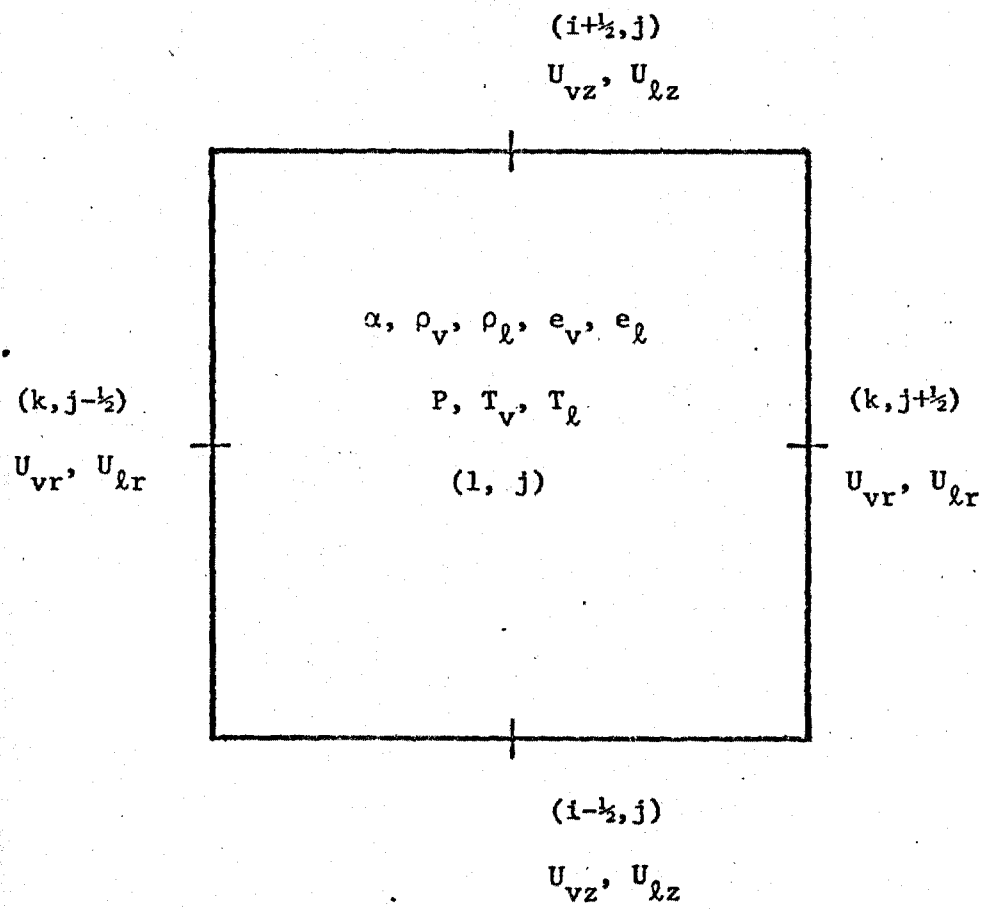


Figure 2.2 Position Evaluation of Variables

Vapor Mass:

$$\begin{aligned} & \frac{(\alpha^{n+1} \rho_v^{n+1} - \alpha^n \rho_v^n)_{i,j}}{\Delta t} + \frac{(\alpha^n \rho_v^n U_{vz}^{n+1})_{i+\frac{1}{2},j} - (\alpha^n \rho_v^n U_{vz}^{n+1})_{i-\frac{1}{2},j}}{\Delta z_i} + \\ & + \left( A_r/V \alpha^n \rho_v^n U_{vr}^{n+1} \right)_{i,j+\frac{1}{2}} - \left( A_r/V \alpha^n \rho_v^n U_{vr}^{n+1} \right)_{i,j-\frac{1}{2}} = S_e - S_c \end{aligned} \quad (2.2.1)$$

Vapor Energy:

$$\begin{aligned} & \frac{(\alpha^{n+1} \rho_v^{n+1} e_v^{n+1} - \alpha^n \rho_v^n e_v^n)_{i,j}}{\Delta t} + \frac{(\alpha^n \rho_v^n e_v^n U_{vz}^{n+1})_{i+\frac{1}{2},j}}{\Delta z_i} - \\ & - \frac{(\alpha^n \rho_v^n e_v^n U_{vz}^{n+1})_{i-\frac{1}{2},j}}{\Delta z_i} + \left( A_r/V \alpha^n \rho_v^n e_v^n U_{vr}^{n+1} \right)_{i,j+\frac{1}{2}} - \\ & - \left( A_r/V \alpha^n \rho_v^n e_v^n U_{vr}^{n+1} \right)_{i,j-\frac{1}{2}} + P_{ij}^n \left[ \frac{(\alpha^{n+1} - \rho^n)_{i,j}}{\Delta t} + \right. \\ & + \frac{(\alpha^n U_{vz}^{n+1})_{i+\frac{1}{2},j} - (\alpha^n \rho U_{vz}^{n+1})_{i-\frac{1}{2},j}}{\Delta z_i} + \left. \left( A_r/V \alpha^n U_{vr}^{n+1} \right)_{i,j+\frac{1}{2}} - \right. \\ & \left. - \left( A_r/V \alpha^n U_{vr}^{n+1} \right)_{i,j-\frac{1}{2}} \right] = Q_{wv} + Q_{\rho v} \end{aligned} \quad (2.2.2)$$

Liquid Mass:

$$\begin{aligned}
 & \frac{\left[ (1-\alpha^{n+1}) \rho_{\ell}^{n+1} - (1-\alpha^n) \rho_{\ell}^n \right]_{i,j}}{\Delta t} + \frac{\left[ (1-\alpha^n) \rho_{\ell}^n U_{\ell z}^{n+1} \right]_{i+\frac{1}{2},j} - \left[ (1-\alpha^n) \rho_{\ell}^n U_{\ell z}^{n+1} \right]_{i-\frac{1}{2},j}}{\Delta z_i} + \\
 & + \left[ A_r/V (1-\alpha^n) \rho_{\ell}^n U_{\ell r}^{n+1} \right]_{i,j+\frac{1}{2}} - \left[ A_r/V (1-\alpha^n) \rho_{\ell}^n U_{\ell r}^{n+1} \right]_{i,j-\frac{1}{2}} = -S_e + S_c
 \end{aligned} \tag{2.2.3}$$

Liquid Energy:

$$\begin{aligned}
 & \frac{\left[ (1-\alpha^{n+1}) \rho_{\ell}^{n+1} e_{\ell}^{n+1} - (1-\alpha^n) \rho_{\ell}^n e_{\ell}^n \right]}{\Delta t} + \frac{\left[ (1-\alpha^n) \rho_{\ell}^n e_{\ell}^n U_{\ell z}^{n+1} \right]_{i+\frac{1}{2},j}}{\Delta z_i} - \\
 & - \left[ (1-\alpha^n) \rho_{\ell}^n e_{\ell}^n U_{\ell z}^{n+1} \right]_{i-\frac{1}{2},j} + \left[ A_r/V (1-\alpha^n) \rho_{\ell}^n e_{\ell}^n U_{\ell r}^{n+1} \right]_{i,j+\frac{1}{2}} - \\
 & - \left[ A_r/V (1-\alpha^n) \rho_{\ell}^n e_{\ell}^n U_{\ell r}^{n+1} \right]_{i,j-\frac{1}{2}} + P_{ij}^n \left\{ -\frac{(\alpha^{n+1} - \alpha^n)_{ij}}{\Delta t} + \right. \\
 & + \frac{\left[ (1-\alpha^n) U_{\ell z}^{n+1} \right]_{i+\frac{1}{2},j} - \left[ (1-\alpha^n) U_{\ell z}^{n+1} \right]_{i-\frac{1}{2},j}}{\Delta z_i} + \left. \left[ A_r/V (1-\alpha^n) U_{\ell r}^{n+1} \right]_{i,j+\frac{1}{2}} \right. \\
 & \left. - \left[ A_r/V (1-\alpha^n) U_{\ell r}^{n+1} \right]_{i,j-\frac{1}{2}} \right\} = Q_{w\ell} - Q_{\ell v}
 \end{aligned} \tag{2.2.4}$$

Some variables in the above equations are used in a location other than the place where they are primarily defined (see figure 2.2.2). For instance, the void fraction  $\alpha$ , which is a cell-centered quantity, appears in the convective terms of all four equations located in the cell's faces. So in order to make these equations completely determined we must establish a rule to transport the value of these variables from the center to the faces of the cells. In our model we have used a relationship known as donor-cell differencing. Later on, in section 2.4 we will see that this scheme has an important effect on the stability of the method. To illustrate how this technique works, let a general variable  $X$  stand for any cell centered quantity such as  $\alpha$ ,  $\rho_v$ ,  $\rho_l$ ,  $e_v$ ,  $e_l$ . The face centered value  $X_{i+\frac{1}{2}}$  will be given by:

$$X_{i+\frac{1}{2}} = \begin{cases} X_i & \text{if } U_{zi+\frac{1}{2}} \geq 0 \\ X_{i+1} & \text{if } U_{zi+\frac{1}{2}} < 0 \end{cases} \quad (2.2.5)$$

In the above rule we have used the axial direction as an example. A similar rule is used to dislocate the variables in the radial direction. The final ambiguity to be removed is in the evaluation of the void fraction. Though it is obvious which of the phase velocities we should use to evaluate the densities and internal energies, this choice is not clear when we refer to the void fraction, which appears in both the liquid and vapor equations. To remove this ambiguity we state that the velocity to be used in the decision of equation 2.2.5 is the one corresponding to the equation in which the variable will appear. Thus for the vapor equations the void fraction

will be calculated using the vapor velocity in equation 2.2.5, while for the liquid equation, the liquid velocity will serve in the decision.

As a final remark, we note that the donor cell rule is used only to locate quantities at time level  $n$ , never at time level  $n+1$ , thus the velocity used in the decision is always a known quantity. Further more, the rule of equation 2.2.5 places a cell centered variable  $x$  in the cell's face where the velocity used in the decision is defined, so that we will never have ambiguity in this decision.

We now turn our attention to the momentum equations. Here a difference with respect to the mass and energy equations should be noted since the velocities are primarily defined at the faces of the mesh cells. Let those faces be the reference points for the differencing of the momentum equations.

Then the vapor momentum equations are:

$$\begin{aligned}
 (\alpha \rho_v)^n_{i+\frac{1}{2},j} & \left[ \frac{(U_{vz}^{n+1} - U_{vz}^n)_{i+\frac{1}{2},j}}{\Delta t} + U_{vz}^n_{i+\frac{1}{2},j} \frac{(\Delta_z U_{vz}^n)_{i+\frac{1}{2},j}}{\Delta z} + \right. \\
 & \left. + U_{vr}^n_{i+\frac{1}{2},j} \frac{(\Delta_r U_{vz}^n)_{i+\frac{1}{2},j}}{\Delta r} \right] + \alpha_{i+\frac{1}{2},j}^n \frac{(P_{i+1,j}^{n+1} - P_{ij}^{n+1})}{\Delta z_{i+\frac{1}{2}}} + \\
 & + (\alpha \rho_v)^n_{i+\frac{1}{2},j} g = - (M_{wzv} + M_{lvz})_{i+\frac{1}{2},j} \quad (2.2.6)
 \end{aligned}$$

$$\begin{aligned}
& (\alpha \rho_v)^n_{i+\frac{1}{2},j} \left[ \frac{(U_{vr}^{n+1} - U_{vr}^n)_{i,j+\frac{1}{2}}}{\Delta t} + U_{vz}^n_{i,j+\frac{1}{2}} \frac{(\Delta_z U_{vr}^n)_{i,j+\frac{1}{2}}}{\Delta z} + \right. \\
& \left. + U_{vr}^n_{i,j+\frac{1}{2}} \frac{(\Delta_r U_{vr}^n)_{i,j+\frac{1}{2}}}{\Delta r} \right] + \rho_{i,j+\frac{1}{2}}^n \frac{(P_{i,j+1}^{n+1} - P_{ij}^{n+1})}{\Delta r_{j+\frac{1}{2}}} = \\
& = - (M_{wrv} + M_{lvr})_{i,j+\frac{1}{2}} \tag{2.2.7}
\end{aligned}$$

And the liquid momentum equations:

$$\begin{aligned}
& \left[ (1-\alpha) \rho_l \right]_{i+\frac{1}{2},j}^n \left[ \frac{(U_{lz}^{n+1} - U_{lz}^n)_{i+\frac{1}{2},j}}{\Delta t} + U_{lz}^n_{i+\frac{1}{2},j} \frac{(\Delta_z U_{lz}^n)_{i+\frac{1}{2},j}}{\Delta z} + \right. \\
& \left. + U_{lr}^n_{i+\frac{1}{2},j} \frac{(\Delta_r U_z^n)_{i+\frac{1}{2},j}}{\Delta r} \right] + (1-\alpha)_{i+\frac{1}{2},j}^n \frac{(P_{i+1,j}^{n+1} - P_{i,j}^{n+1})}{\Delta z_{i+\frac{1}{2}}} + \\
& \left[ (1-\alpha) \rho_l \right]_{i+\frac{1}{2},j}^n g = - (M_{wzl} - M_{lvz})_{i+\frac{1}{2},j} \tag{2.2.8}
\end{aligned}$$

$$\left[ (1-\alpha) \rho_l \right]_{i,j+\frac{1}{2}}^n \left[ \frac{(U_{lr}^{n+1} - U_{lr}^n)_{i,j+\frac{1}{2}}}{\Delta t} + U_{lz}^n_{i,j+\frac{1}{2}} \frac{(\Delta_z U_{lr}^n)_{i,j+\frac{1}{2}}}{\Delta z} + \right.$$

$$\begin{aligned}
& + U_{lr}^n \left[ \frac{(\Delta_r U_r^n)_{i,j+\frac{1}{2}}}{\Delta r} \right] + (1-\alpha_{ij+\frac{1}{2}}^n) \frac{(P_{i,j+1}^{n+1} - P_{i,j}^{n+1})}{\Delta r_{j+\frac{1}{2}}} = \\
& = - (M_{wrl} - M_{lvr})_{i,j+\frac{1}{2}} \tag{2.2.9}
\end{aligned}$$

Again in the momentum equations some variables are used at a location different from where they were primarily defined. The question is how are these quantities evaluated? First, consider the void fraction  $\alpha$  and the densities  $\rho_v$  and  $\rho_l$ . Contrary to the mass and energy equations, these quantities do not appear in the momentum equation as difference terms. Thus, they do not influence the stability of the method the way they did in equations 2.2.1 through 2.2.4, and we can use a simple averaging rule such as:

$$X_{i+\frac{1}{2}} = \frac{X_{i+1} \Delta Z_{i+1} + X_i \Delta Z_i}{\Delta Z_{i+1} + \Delta Z_i} \tag{2.2.10}$$

where X stands for the void fraction  $\alpha$  and the two densities  $\rho_v$  and  $\rho_l$ . A similar rule is used to transfer the variables to the faces  $j+\frac{1}{2}$  in the radial direction, with  $\Delta r$  replacing  $\Delta Z$ .

We next consider the velocities appearing in our momentum equations. First we look at the velocities  $U_{vr\ i+\frac{1}{2},j}$ ,  $U_{lr\ i+\frac{1}{2},j}$ ,  $U_{vz\ i,j+\frac{1}{2}}$ ,  $U_{lz\ i,j+\frac{1}{2}}$ . Figure 2.3 shows as an example the position of

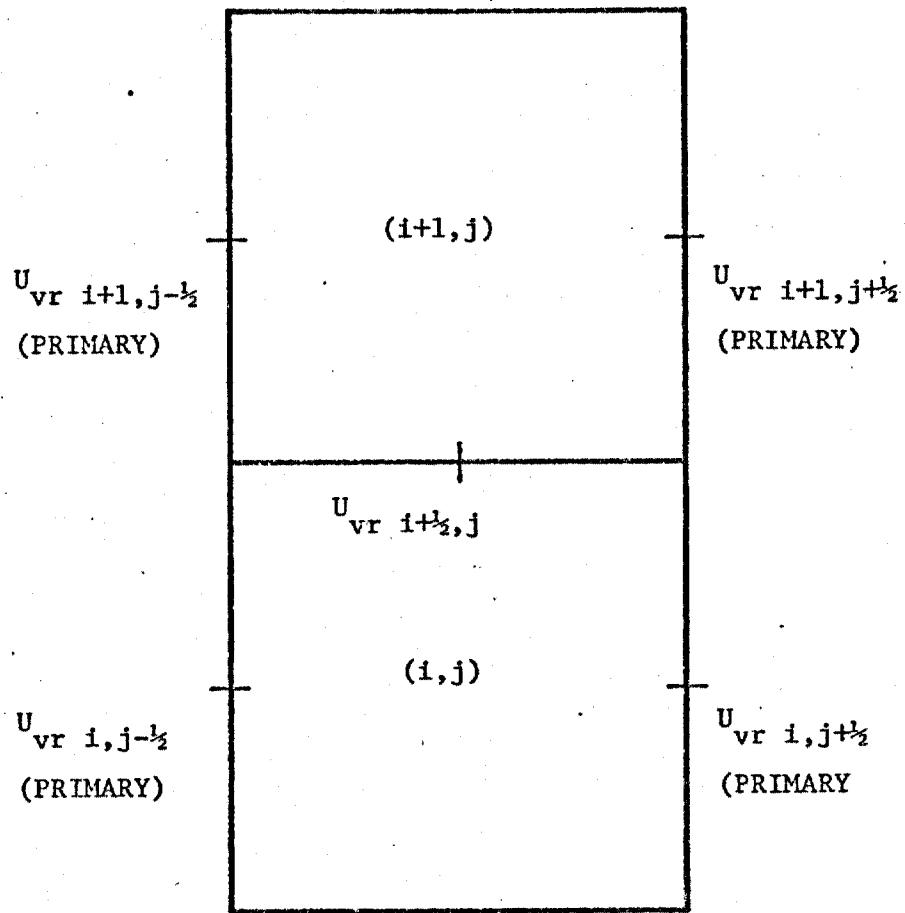


Figure 2.3 Different Positions for the Radial Velocity

$U_{vr\ i+1/2,j}$  compared with the location where  $U_{vr}$  is primarily defined. Again these velocities do not appear as difference terms and a simple averaging procedure can be used without compromising the stability of the method. Thus we define:

$$U_{vz\ i,j+1/2} = \frac{1}{4} \left[ U_{vz\ i+1/2,j} + U_{vz\ i-1/2,j} + U_{vz\ i+1/2,j+1} + U_{vz\ i-1/2,j+1} \right] \quad (2.2.11)$$

$$U_{vr\ i+1/2,j} = \frac{1}{4} \left[ U_{vr\ i,j+1/2} + U_{vr\ i,j-1/2} + U_{vr\ i+1,j+1/2} + U_{vr\ i+1,j-1/2} \right] \quad (2.2.12)$$

and a similar pair of relationships for the liquid phase.

Finally those velocities appearing in the difference terms must be evaluated. Here a simple averaging procedure would lead to a differencing scheme unstable. Therefore these velocities are evaluated with the donor cell technique. In this way, the expressions for the difference terms are:

$$\left( \frac{\Delta_z U_{vz}}{\Delta Z} \right)_{i+1/2,j} = \begin{cases} \frac{U_{vz\ i+3/2,j} - U_{vz\ i+1/2,j}}{\Delta Z_{i+1}} & \text{if } U_{vz\ i+1/2,j} < 0 \\ \frac{U_{vz\ i+1/2,j} - U_{vz\ i-1/2,j}}{\Delta Z_i} & \text{if } U_{vz\ i+1/2,j} \geq 0 \end{cases} \quad (2.2.13)$$

$$\left( \frac{\Delta_r U_{vz}}{\Delta r} \right)_{i+\frac{1}{2},j} = \begin{cases} \frac{U_{vz\ i+\frac{1}{2},j+1} - U_{vz\ i+\frac{1}{2},j}}{\Delta r_{j-\frac{1}{2}}} & \text{if } U_{vr\ i+\frac{1}{2},j} < 0 \\ \frac{U_{vz\ i+\frac{1}{2},j} - U_{vz\ i+\frac{1}{2},j-1}}{\Delta r_{j-\frac{1}{2}}} & \text{if } U_{vr\ i+\frac{1}{2},j} \geq 0 \end{cases} \quad (2.2.14)$$

$$\left( \frac{\Delta_z U_{vr}}{\Delta z} \right)_{i,j+\frac{1}{2}} = \begin{cases} \frac{U_{vr\ i+1,j+\frac{1}{2}} - U_{vr\ i,j+\frac{1}{2}}}{\Delta z_{i+\frac{1}{2}}} & \text{if } U_{vz\ i,j+\frac{1}{2}} < 0 \\ \frac{U_{vr\ i,j+\frac{1}{2}} - U_{vr\ i-1,j+\frac{1}{2}}}{\Delta z_{i-\frac{1}{2}}} & \text{if } U_{vz\ i,j+\frac{1}{2}} \geq 0 \end{cases} \quad (2.2.15)$$

$$\left( \frac{\Delta_r U_{vr}}{\Delta r} \right)_{i,j+\frac{1}{2}} = \begin{cases} \frac{U_{vr\ i,j+\frac{3}{2}} - U_{vr\ i,j+\frac{1}{2}}}{\Delta r_{j+1}} & \text{if } U_{vr\ i,j+\frac{1}{2}} < 0 \\ \frac{U_{vr\ i,j+\frac{1}{2}} - U_{vr\ i,j-\frac{1}{2}}}{\Delta r_j} & \text{if } U_{vr\ i,j+\frac{1}{2}} \geq 0 \end{cases} \quad (2.2.16)$$

where the mesh spacings  $\Delta z_{i+\frac{1}{2}}$  and  $\Delta r_{j+\frac{1}{2}}$  appearing in the above expressions are defined as:

$$\Delta z_{i+\frac{1}{2}} = \frac{\Delta z_{i+1} + \Delta z_i}{2} \quad (2.2.17)$$

$$\Delta r_{j+\frac{1}{2}} = \frac{\Delta r_{j+1} + \Delta r_j}{2} \quad (2.2.18)$$

and similar expressions apply to the liquid phase.

With those rules the differencing scheme for the fluid flow conservation equations is completed. To complete the set of algebraic equations we need only to specify the relationships for the exchange terms and the equations of state. These will be discussed in Chapter 3. We now turn our attention to the numerical solution of the set of algebraic equations, equations 2.2.1 through 2.2.4 and 2.2.6 through 2.2.9.

### 2.3 The Numerical Scheme

In the above difference equations all variables evaluated at the time level  $n$  were determined in the previous time level, thus in the present level  $n+1$  they are known quantities. The problem is to extract from that set of equations the variables at time level  $n+1$ . A quick look at those equations reveal they are non linear, complicate equations, and a numerical iterative technique is practically the only option for their solution.

The equations of state represent unique relationships of the densities and internal energies for a given pair of pressure and temperature. We will replace these densities and internal energies by the liquid and vapor temperatures as primary variables, thus reducing the number of unknown to eight, namely the void fraction, the pressure, the vapor and liquid temperatures and the four velocity components.

The technique used in the solution of algebraic equation is a multidimensional extension of the Newton iterative solution of algebraic equations. Let us first define a vector whose components are the unknowns of the problem. Then:

$$X = \left[ \alpha, P, T_v, T_l, U_{vz}, U_{vr}, U_{lz}, U_{lr} \right]^{n+1} \quad (2.3.1)$$

and the equations 2.2.1 through 2.2.4 and 2.2.6 through 2.2.9 can be written in abbreviated form as:

$$F_p(X) = 0, \quad p = 1, \dots, 8 \quad (2.3.2)$$

Now suppose that at a certain iteration  $k$  we have come up with an approximate solution of 2.3.2  $X^k$ . Since this is not the exact solution, the left hand side of 2.3.2  $F_p(X^k)$  is not necessarily equal to zero. Then, let us make a Taylor expansion of  $F(X)$  around the point  $X^k$ :

$$F_p(X^{k+1}) = F_p(X^k) + \sum_{q=1}^8 \left( \frac{\partial F_p}{\partial x_q} \right)_{X^k} (x_q^{k+1} - x_q^k) \quad (2.3.3)$$

$$P = 1, 8$$

If  $X^{k+1}$  is required to be the solution of equation 2.3.2 it follows:

$$\sum_{q=1}^8 \left( \frac{\partial F_p}{\partial x_q} \right)_{X^k} (x_q^{k+1} - x_q^k) = -F_p(X^k), \quad P = 1, \dots, 8 \quad (2.3.4)$$

With equation 2.3.4 the iterative procedure is defined. Note that this set of equations is now linear in the unknowns  $\delta x_q = x_q^{k+1} - x_q^k$

If equation 2.3.4 are written explicitly, it follows:

$$\left[ \frac{\rho_v^k}{\Delta t} - \frac{\partial S}{\partial \alpha} \right] \delta \alpha + \left[ \frac{\alpha^k}{\Delta t} \frac{\partial \rho_v}{\partial P} - \frac{\partial S}{\partial P} \right] \delta P_{ij} + \left[ \frac{\alpha^k}{\Delta t} \frac{\partial \rho_v}{\partial T_v} - \frac{\partial S}{\partial T_v} \right] \delta T_v -$$

$$\begin{aligned}
& - \frac{\partial S}{\partial T} \delta T_\ell + \frac{(\alpha \rho_v)_{i+\frac{1}{2},j}}{\Delta z_i} \delta U_{vz\ i+\frac{1}{2},j} - \frac{(\alpha \rho_v)_{i-\frac{1}{2},j}}{\Delta z_i} \delta U_{vz\ i-\frac{1}{2},j} + \\
& + \left( \frac{Ar/V \alpha \rho_v}{i+\frac{1}{2}} \right) \delta U_{vr\ i+\frac{1}{2},j} - \left( \frac{Ar/V \alpha \rho_v}{i-\frac{1}{2}} \right) \delta U_{vr\ i-\frac{1}{2},j} = - F_1^k
\end{aligned}$$

(2.3.5)

$$\begin{aligned}
& \left[ \frac{\rho_v^k e_v^k + P^k}{\Delta t} - \frac{\partial Q_v}{\partial \alpha} \right] \delta \alpha + \left[ \left( \alpha^k \rho_v^k \frac{\partial e_v}{\partial P} + \alpha^k e_v^k \frac{\partial \rho_v}{\partial P} \right) \frac{1}{\Delta t} - \frac{\partial Q_v}{\partial P} \right] \delta P_{ij} + \\
& + \left[ \frac{\alpha^k}{\Delta t} \left( \rho_v^k \frac{\partial e_v}{\partial T_v} + e_v^k \frac{\partial \rho_v}{\partial T_v} \right) - \frac{\partial Q_v}{\partial T_v} \right] \delta T_v - \frac{\partial Q_v}{\partial T} \delta T_\ell + \\
& + \left[ \frac{\alpha_{i+\frac{1}{2},j}}{\Delta z_i} \left( P_{ij} + (\rho_v e_v)_{i+\frac{1}{2},j} \right) \right] \delta U_{vz\ i+\frac{1}{2},j} - \left[ \frac{\alpha_{i-\frac{1}{2},j}}{\Delta z_i} \left( P_{ij} + \right. \right. \\
& \left. \left. + (\rho_v e_v)_{i-\frac{1}{2},j} \right) \right] \delta U_{vz\ i-\frac{1}{2},j} + \left[ \left( \frac{Ar/V \alpha}{i+\frac{1}{2}} \right) \left( P_{ij} + (\rho_v e_v)_{i+\frac{1}{2},j} \right) \right] \delta U_{vr\ i+\frac{1}{2},j} \\
& - \left[ \left( \frac{Ar/V \alpha}{i-\frac{1}{2}} \right) \left( P_{ij} + (\rho_v e_v)_{i-\frac{1}{2},j} \right) \right] \delta U_{vr\ i-\frac{1}{2},j} = - F_2^k
\end{aligned}$$

(2.3.6)

$$\begin{aligned}
& - \left[ \frac{\rho_\ell^k}{\Delta t} - \frac{\partial S}{\partial \alpha} \right] \delta \alpha + \left[ \frac{(1-\alpha^k)}{\Delta t} \frac{\partial \rho_\ell}{\partial P} + \frac{\partial S}{\partial P} \right] \delta P_{ij} + \frac{\partial S}{\partial T_v} \delta T_v + \\
& + \left[ \frac{(1-\alpha^k)}{\Delta t} \frac{\partial \rho_\ell}{\partial T_\ell} + \frac{\partial S}{\partial T_\ell} \right] \delta T_\ell + \left[ (1-\alpha) \rho_\ell \right]_{i+\frac{1}{2},j} \delta U_{\ell z \ i+\frac{1}{2},j} - \\
& - \left[ (1-\alpha) \rho_\ell \right]_{i-\frac{1}{2},j} \delta U_{\ell z \ i-\frac{1}{2},j} + \left[ Ar/V (1-\alpha) \rho_\ell \right]_{i,j+\frac{1}{2}} \delta U_{\ell r \ i,j+\frac{1}{2}} - \\
& - \left[ Ar/V (1-\alpha) \rho_\ell \right]_{i,j-\frac{1}{2}} \delta U_{\ell r \ i,j-\frac{1}{2}} = - F_3^k \tag{2.3.7}
\end{aligned}$$

$$\begin{aligned}
& - \left[ \frac{\rho_\ell^k e_\ell^k + P}{\Delta t} - \frac{\partial Q}{\partial \alpha} \right] \delta \alpha + \left[ \frac{(1-\alpha)}{\Delta t} \left( \rho_\ell^k \frac{\partial e_\ell^k}{\partial P} + e_\ell^k \frac{\partial \rho_\ell}{\partial P} \right) - \frac{\partial Q_\ell}{\partial P} \right] \delta P_{ij} - \\
& - \frac{\partial Q_\ell}{\partial T_v} \delta T_v + \left[ \frac{(1-\alpha^k)}{\Delta t} \left( \rho_\ell^k \frac{\partial e_\ell}{\partial T_\ell} + e_\ell^k \frac{\partial \rho_\ell}{\partial T_\ell} \right) - \frac{\partial Q_\ell}{\partial T_\ell} \right] \delta T_\ell + \\
& + \left[ \frac{(1-\alpha)_{i+\frac{1}{2},j}}{\Delta z_1} \left( P_{ij} + (\rho_\ell e_\ell)_{i+\frac{1}{2},j} \right) \right] \delta U_{\ell z \ i+\frac{1}{2},j} - \frac{(1-\alpha)_{i-\frac{1}{2},j}}{\Delta z_1} \left[ \left( P_{ij} + \right. \right. \\
& \left. \left. + (\rho_\ell e_\ell)_{i-\frac{1}{2},j} \right) \right] \delta U_{\ell z \ i-\frac{1}{2},j} + \left[ \frac{Ar/V(1-\alpha)}{i,j+\frac{1}{2}} \left( P_{ij} + (\rho_\ell e_\ell)_{i,j+\frac{1}{2}} \right) \right] \delta U_{\ell r \ i,j+\frac{1}{2}}
\end{aligned}$$

$$- \left[ \left( \frac{\Delta r}{V} (1-\alpha) \right)_{i,j-\frac{1}{2}} \left( P_{ij} + (\rho_\ell e_\ell)_{i,j-\frac{1}{2}} \right) \right] \delta U_{\ell r i,j-\frac{1}{2}} = - F_4^k \quad (2.3.8)$$

$$\begin{aligned} & \left[ \frac{(\alpha \rho_v)_{i+\frac{1}{2},j}}{\Delta t} + \frac{\partial M_v}{\partial U_{vz}} \right] \delta U_{vz i,+\frac{1}{2},j} + \frac{\partial M_v}{\partial U_{\ell z}} \delta U_{\ell z i+\frac{1}{2},j} + \\ & + \frac{\alpha_{i+\frac{1}{2},j}}{\Delta z_{i+\frac{1}{2}}} (\delta P_{i+1,j} - \delta P_{ij}) = - F_5^k \end{aligned} \quad (2.3.9)$$

$$\begin{aligned} & \left[ \frac{[(1-\alpha)\rho_\ell]_{i+\frac{1}{2},j}}{\Delta t} + \frac{\partial M_\ell}{\partial U_{\ell z}} \right] \delta U_{\ell z i+\frac{1}{2},j} + \frac{\partial M_\ell}{\partial U_{vz}} \delta U_{vz i+\frac{1}{2},j} + \\ & + \frac{(1-\alpha)_{i+\frac{1}{2},j}}{\Delta z_{i+\frac{1}{2}}} (\delta P_{i+1,j} - \delta P_{ij}) = - F_6^k \end{aligned} \quad (2.3.10)$$

$$\begin{aligned} & \left[ \frac{(\alpha \rho_v)_{i,j+\frac{1}{2}}}{\Delta t} + \frac{\partial M_v}{\partial U_{vr}} \right] \delta U_{vr i,j+\frac{1}{2}} + \frac{\partial M_v}{\partial U_{\ell r}} \delta U_{\ell r i,j+\frac{1}{2}} + \\ & + \frac{\alpha_{i,j+\frac{1}{2}}}{\Delta r_{j+\frac{1}{2}}} (\delta P_{i,j+1} - \delta P_{ij}) = F_7^k \end{aligned} \quad (2.3.11)$$

$$\begin{aligned} & \left[ \frac{(1-\alpha)\rho_\ell_{i,j+\frac{1}{2}}}{\Delta t} + \frac{\partial M_\ell}{\partial U_{\ell r}} \right] \delta U_{\ell r i,j+\frac{1}{2}} + \frac{\partial M_\ell}{\partial U_{vr}} \delta U_{vr i,j+\frac{1}{2}} + \\ & + \frac{(1-\alpha)_{i,j+\frac{1}{2}}}{\Delta r_{j+\frac{1}{2}}} (\delta P_{i,j+1} - \delta P_{ij}) = - F_8^k \end{aligned} \quad (2.3.12)$$

Note that the last four of the above equations depend only on pressures and the four velocity components. Grouping equations 2.3.9 and 2.3.10 in a pair and again 2.3.11 and 2.3.12 we can without difficulty isolate the velocity components in the left hand side:

$$\delta U_{vz\ i+\frac{1}{2},j} = W_{vz\ i+\frac{1}{2},j} (\delta P_{i+1,j} - \delta P_{ij}) + f_{uvz} \quad (2.3.13)$$

$$\delta U_{lz\ i+\frac{1}{2},j} = W_{lz\ i+\frac{1}{2},j} (\delta P_{i+1,j} - \delta P_{ij}) + f_{ulz} \quad (2.3.14)$$

$$\delta U_{vr\ i,j+\frac{1}{2}} = W_{vr\ i,j+\frac{1}{2}} (\delta P_{i,j+1} - \delta P_{ij}) + f_{uvr} \quad (2.3.15)$$

$$\delta U_{lr\ i,j+\frac{1}{2}} = W_{lr\ i,j+\frac{1}{2}} (\delta P_{i,j+1} - \delta P_{ij}) + f_{ulr} \quad (2.3.16)$$

where the coefficients W are given by:

$$W_{vz\ i+\frac{1}{2},j} = - \left[ \frac{\alpha}{\Delta z} \left( \frac{(1-\alpha)\rho_\ell}{\Delta t} + \frac{\partial M_\ell}{\partial U_{lz}} \right) + \frac{(1-\alpha)}{\Delta z} \cdot \frac{\partial M_v}{\partial U_{vz}} \right]_{i+\frac{1}{2},j} \times$$

$$\left[ \left( \frac{(1-\alpha)\rho_\ell}{\Delta t} + \frac{\partial M_\ell}{\partial U_{lz}} \right) \left( \frac{\alpha\rho_v}{\Delta t} + \frac{\partial M_v}{\partial U_v} \right) - \frac{\partial M_\ell}{\partial U_{lz}} \frac{\partial M_v}{\partial U_{vz}} \right]_{i+\frac{1}{2},j}^{-1} \quad (2.3.17)$$

and similar expressions for the other component velocities.

Now, with equation 2.3.13 through 2.3.16 we can eliminate all velocities in equations 2.3.5 through 2.3.8. Rearranging these equations, they can be written in the matrix form:

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \times \begin{bmatrix} \delta\alpha \\ \delta T_v \\ \delta T_\ell \\ \delta P \end{bmatrix}_{i,j} + \begin{bmatrix} b_{11} & b_{12} & b_{13} & b_{14} \\ b_{21} & b_{22} & b_{23} & b_{24} \\ b_{31} & b_{32} & b_{33} & b_{34} \\ b_{41} & b_{42} & b_{43} & b_{44} \end{bmatrix} \times \begin{bmatrix} \delta P_{i-1,j} \\ \delta P_{i,j-1} \\ \delta P_{i,j+1} \\ \delta P_{i+1,j} \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{bmatrix} \quad (2.3.18)$$

The expressions for the coefficients  $a$ 's and  $b$ 's are not given here for brevity. We will return to them and show representatives of them when we discuss the diagonal dominance of the pressure problem and the limiting case of only one phase present.

If we transform the matrix of the coefficients  $a$  in equation 2.3.18 into an upper triangular matrix, this equation becomes:

$$\begin{bmatrix} 1 & a'_{12} & a'_{13} & a'_{14} \\ 0 & 1 & a'_{23} & a'_{24} \\ 0 & 0 & 1 & a'_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} \delta\alpha \\ \delta T_v \\ \delta T_\ell \\ \delta P_{i,j} \end{bmatrix} + \begin{bmatrix} b'_{11} & \dots & b'_{14} \\ \dots & \dots & \dots \\ b'_{41} & \dots & b'_{44} \end{bmatrix} \times \begin{bmatrix} \delta P_{i-1,j} \\ \delta P_{i,j-1} \\ \delta P_{i,j+1} \\ \delta P_{i+1,j} \end{bmatrix} = \begin{bmatrix} f'_1 \\ f'_2 \\ f'_3 \\ f'_4 \end{bmatrix}$$

(2.3.19)

The last line of the above equation is an expression involving only pressures. Since this expression relates the pressure at a cell  $(i,j)$  to its four neighbors' pressure, this equation must be solved simultaneously for all mesh cells. The solution of this pressure problem is the subject of section 2.4.

It is important to point out that this solution technique reduces the inversion of a matrix with dimensions  $8N$  by  $8N$ , with  $N$  being the number of mesh cells, to the inversion of a matrix of dimensions  $N$  by  $N$  by performing for each mesh cell the inversion of two  $2$  by  $2$  matrices and one  $4$  by  $4$  matrix.

Before the closing of this section, it is appropriate to make three comments. The first one concerns the limiting case of single phase flow. The transformation of equation 2.3.18 into equation 2.3.19 requires that all diagonal elements of matrix of the coefficients  $a$  be non-zero. We will explore how those coefficients behave as the void fraction assumes the values  $\alpha = 0$  and  $\alpha = 1$ , and the mass exchange rate  $S = 0$ .

First let us consider the case  $\alpha = 0$ . If we look back into equation 2.3.5 we can see that in the first line of equation 2.3.18 all coefficients  $b_{1q}$ , as well as all  $a_{1q}$ , with the exception of  $a_{11}$  have a factor  $\alpha$  on them. Therefore, except for  $a_{11}$ , all those coefficients are zero. If we look into equation 2.2.1 we see that the right hand side of 2.3.5 is also zero, and the first line of 2.3.18 corresponds to the equation:

$$\frac{\rho_v^k}{\Delta t} \delta\alpha = 0$$

Now, consider equation 2.3.6. It is seen that without the presence of vapor this equation is trivial, and all coefficients  $a_{2q}$  and  $b_{2q}$  in second line of 2.3.18 are zero. But in this case, a trivial equation would cause us a problem, since an element of the diagonal of the matrix of coefficients  $\underline{a}$  in 2.3.18 would be zero, thus invalidating the triangularization of this matrix. To avoid this problem, we impose that the interphase heat exchange term be in the form:

$$Q_{lv} = h (T_\ell^{n+1} - T_n^{n+1})$$

with the coefficient  $h$  being non-zero even if one of the phases is not present. In this way, equation 2.3.6 reduces to:

$$h \partial T_v - h \partial T_\ell = h (T_v^k - T_\ell^k)$$

which implies the model will force the vapor temperature to be equal to the liquid temperature when we have one of the phases absent.

If we repeat this analysis for the vapor single phase flow, it is easy to see that we will reach the same conclusions. Therefore

we can be confident that the matrix of coefficients  $a$  in 2.3.18 does not have a diagonal element equal to zero and the triangularization of this matrix is always possible.

One last question in this subject concerns the inversion of the submatrices of the velocity components. If we look into equation 2.3.17 we will see that the absence of one phase would lead to a division by zero. We again avoid this problem by imposing the interphase momentum exchange term to be in the form:

$$M_{lv} = K (U_l^{n+1} - U_v^{n+1})$$

again with the coefficient  $K$  being non-zero even if one of the phases is not present. As for the energy equation, this will force the vapor velocity to be equal to the liquid velocity when one of the phases is not present.

The second question we would like to discuss concerns the diagonal dominance of the pressure problem. The solution of this problem requires that the diagonal element of the pressure problem matrix be greater or equal to the sum of the absolute value of the elements in the line corresponding to that diagonal element. In terms of the coefficients of equation 2.3.19 this translate to:

$$|b'_{41}| + |b'_{42}| + |b'_{43}| + |b'_{44}| \leq 1$$

An exact proof that this condition is satisfied would require a prohibitive amount of algebraic work. So instead of trying to follow this line, we will present only a partial view, which can bring

some understanding to this problem. Then, let us consider the elements of the first two lines of the matrix of the coefficients  $\underline{b}$  in equation 2.3.18. We evaluate these coefficients with the help of equations 2.3.5, 2.3.6, 2.3.13 and 2.3.15. We get the expressions:

$$b_{11} = \left( \frac{\alpha \rho_v}{\Delta z} W_{vz} \right)_{i-\frac{1}{2}, j}$$

$$b_{12} = \left( \frac{Ar}{V} \alpha \rho_v W_{vr} \right)_{i, j-\frac{1}{2}}$$

$$b_{13} = \left( \frac{Ar}{V} \alpha \rho_v W_{vr} \right)_{i, j+\frac{1}{2}}$$

$$b_{14} = \left( \frac{\alpha \rho_v}{\Delta z} W_{vz} \right)_{i+\frac{1}{2}, j}$$

$$b_{21} = \left[ \frac{\alpha}{\Delta z} (\rho_v e_v + P) W_{vz} \right]_{i-\frac{1}{2}, j}$$

$$b_{22} = \left[ \frac{Ar}{V} \alpha (\rho_v e_v + P) W_{vr} \right]_{i, j+\frac{1}{2}}$$

$$b_{23} = \left[ \frac{Ar}{V} \alpha (\rho_v e_v + P) W_{vr} \right]_{i, j+\frac{1}{2}}$$

$$b_{24} = \left[ \frac{\alpha}{\Delta z} (\rho_v e_v + P) W_{vz} \right]_{i+\frac{1}{2}, j}$$

From the expression for the value of the coefficients  $W$ , equation 2.3.17 we can see that all these coefficients  $\underline{b}$  are negative.

Now consider the coefficients of the central pressure, coefficients  $a_{14}$  and  $a_{24}$  in equation 2.3.18. From the same equations we used before we get:

$$a_{14} = \frac{\alpha}{\Delta t} \frac{\partial \rho_v}{\partial P} - \frac{\partial S}{\partial P} - (b_{11} + b_{12} + b_{13} + b_{14})$$

$$a_{24} = \frac{\alpha}{\Delta t} \rho_v \frac{\partial e_v}{\partial P} + e_v \frac{\partial \rho_v}{\partial P} - \frac{\partial Q}{\partial P} - (b_{21} + b_{22} + b_{23} + b_{24})$$

Let us examine in detail each of these coefficients. The way equation 2.2.1 was written the mass exchange rate  $S$  is positive when we have evaporation. It is easy to see that an increment in pressure will produce a decrease in the rate of evaporation, so the term  $\partial S/\partial P$  is negative. The other term making  $a_{14}$  is the vapor compressibility, which is a positive quantity. Therefore we conclude:

$$a_{14} > |b_{11}| + |b_{12}| + |b_{13}| + |b_{14}|$$

Consider next the coefficient  $a_{24}$ . The first term in  $a_{24}$  is  $\partial e_v/\partial P$ , which is a very small quantity. Indeed the equation of state used in our model puts a zero in this derivative. The other term  $\partial \rho_v/\partial P$  we have already investigated and seen it is a positive quantity. Finally we have the term  $\partial Q_v/\partial P$ . The heat transfers equations used in the model have only one term in the heat exchange rate dependent on the pressure, representing the heat transferred due to evaporation or condensation. In this way  $\partial Q_v/\partial P$  has the same sign as  $\partial S/\partial P$ , which we saw

before is a negative quantity. We thus conclude again:

$$a_{24} > |b_{21}| + |b_{22}| + |b_{23}| + |b_{24}|$$

We omit here a similar analysis of the coefficients appearing in the liquid equations. The general form of them is the same, on following a similar reasoning we would reach the same conclusions as we did for the vapor equations.

Now, since the pressure problem equation (the last line of equation 2.3.19) was obtained as a linear combination of equations whose coefficient of the central pressure exceeds the sum of the absolute value of the coefficients of the neighboring pressures, this pressure problem equation also has this same property, which shows us the pressure problem matrix is diagonal dominant.

Finally there is the question of the boundary conditions. We start with the radial direction. At the fuel assembly centerline there is simply the zero radial flow condition at  $r = 0$ . This is accomplished by just putting a zero in the terms  $U_{r1, \frac{1}{2}}$  appearing in the divergent differences. At the other radial boundary, corresponding to the fuel assembly hexcan there is also a zero flow boundary condition, which is translated in the model by setting the radial velocities at that boundary equal to zero. Note that since these velocities are identically zero, there is no need to evaluate the momentum equations at these nodes  $J+\frac{1}{2}$ , thus there will be only  $J-1$  radial momentum equations at each level  $i$ . Besides the flow conditions at this boundary,

there is also a thermal boundary condition, taking into account the heat transferred between the fluid and the structure, represented in the code by the hex can model. This model will be fully analysed in Chapter 3.

For the axial direction more complicated conditions appear. To explain this refer to figure 2.4. It can be seen in that figure that two fictitious cells were added to the actual fuel assembly. In these cells the conditions determining a particular problem must be specified. Thus the user of the model needs to specify as a function of time, an outlet pressure in cells  $i = I + 1$  and inlet pressure, vapor and liquid temperatures and the void fraction in cells  $i = 0$ . For the momentum equations in cells  $i = \frac{1}{2}$  and  $i = I + \frac{1}{2}$  the following conditions are imposed

$$U_{-\frac{1}{2},j} = U_{\frac{1}{2},j}$$

$$U_{I+\frac{3}{2},j} = U_{I+\frac{1}{2},j}$$

Finally, to completely determine the particular problem to be studied, the user also needs to specify the fuel pin heat generation rate as a function of time.

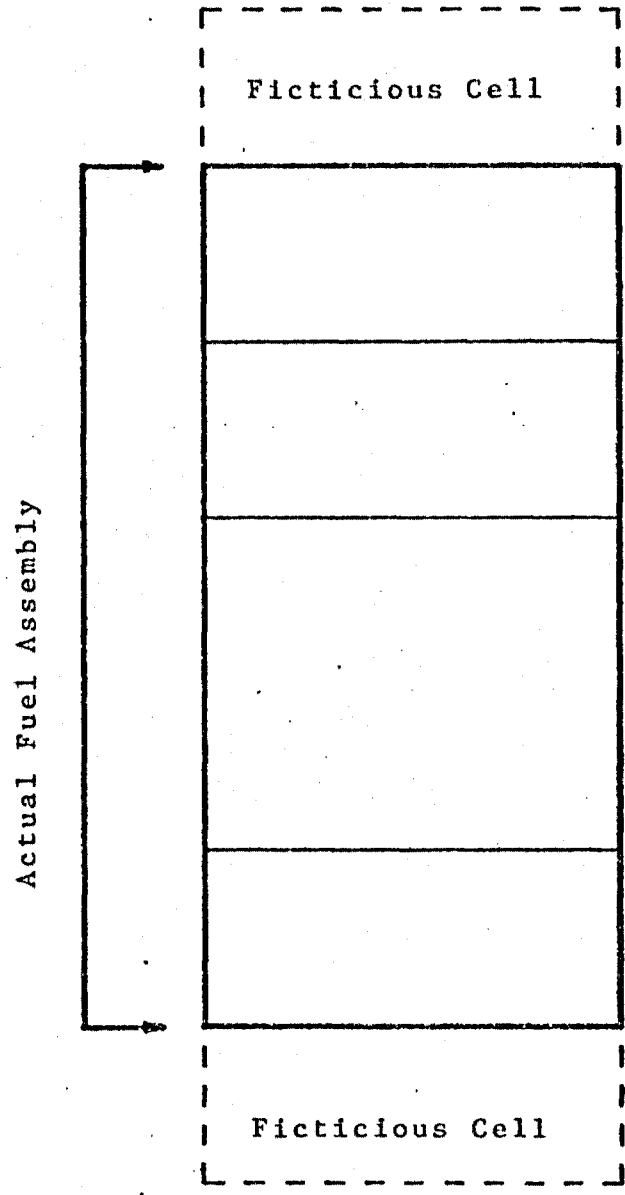


Figure 2.4 The Fictitious Cells

## 2.4 The Pressure Problem

So far, we have collapsed the eight conservation equations, the equation of state and the equations governing the exchange terms into a single equation (i.e. one for each mesh cell), involving the pressure in the cell itself and its neighbours. Because of this coupling between cells, those equations must be solved simultaneously. Since this matrix inversion rests inside an iterative process, which is to be repeated for each time step, it is clear that the overall efficiency of the model is strongly dependent on the way this pressure problem solution is done.

The approach to the problem was to take advantage of two particular characteristics of the case at hand. The first one is the fact that most of the elements of the matrix are zeros, the non-zeros being only the elements on five diagonals. The second one has to do with the fact that LMFBR fuel assemblies have one of its dimensions, the axial one, much larger than the other. This has a surprisingly strong effect on the time required for the matrix inversion, as explained in the following paragraphs.

The large number of zeros in the matrix was used to our advantage by adopting an iterative solution known as block-tri-diagonal, which is an extension of the Gauss-Siedel iterative technique (see Ref. ).

Recalling the pressure equation, for each mesh cell we have:

$$A_{ij} P_{ij-1} + B_{ij} P_{i-1j} + C_{ij} P_{ij} + D_{ij} P_{i+1j} + E_{ij} P_{ij+1} = R_{ij} \quad (2.4.1)$$

To perform the  $k$ th iteration in the cells at level  $i$ , we pass to the right-hand side of the equation the terms containing the pressure at the bottom and top of cell  $(i,j)$ .

$$A_{ij} P_{ij-1}^k + C_{ij} P_{ij}^k + E_{ij} P_{ij+1}^k = R_{ij} - B_{ij} P_{i-1j}^k - D_{ij} P_{i+1j}^{k-1} \quad (2.4.2)$$

Note that the term in  $P_{i-1j}$  takes the value obtained at iteration  $k$ . This is a known quantity since it was obtained in the previous step of the calculation, when this procedure was applied for the cells at level  $i-1$ .

With this manipulation, we ended with only three unknowns in the equation, and we now can use the tridiagonal matrix inversion technique (Ref. 42 ) which gives the exact solution of equation 2.4.2 for all values of  $j$ , with a very few operations.

This procedure is repeated for all values of the subscript  $i$ , and the pass over all cells is repeated again until the desired convergence is obtained.

The second characteristic which was taken into consideration influences the number of passes required to attain convergence.

An iterative solution sets arbitrary initial values for the unknowns, and by recalculating these unknowns with the appropriate set of equations aims to reduce the error contained in the previous value of the unknowns. The smaller the error carried from one pass to the other, the fewer the number of passes necessary to meet the required convergence criterion.

In the technique used in the model, the new value obtained for the pressure will have an error because in the right-hand side of equation 2.4.2 the values of the pressure are not the exact solution of the problem, but for each level  $i$ , the values of the pressure will have the correct relationship between themselves, since the tri-diagonal technique will give the exact solution for a given right-hand side. If we could make our scheme in such a way as to minimize the influence of the error carried into the right-hand side of equation 2.4.2, we would have the iterations converging quickly. The difference in dimensions for the axial and radial directions provides this way.

When a fuel assembly is divided into mesh cells, the radial dimension of these mesh cells will be a few pitches in length, or for usual LMFBR fuel assemblies, this dimension will be of the order of one centimeter. On the other hand, typically a fuel assembly is a few meters in length, and in order to keep the number of cells at a minimum, to shorten the time required for the calculations, we expect the axial dimension of a mesh cell to be of the order of tens of centimeter.

In this situation, the pressure at radially neighbouring cells must have a very close value, or in other words, a small increment in the pressure in one cell would be propagated to its radial neighbours almost in full. On the other hand, for the axial direction this propagation of error would not be so strong, since the larger distance between cells would act in the sense of attenuating the propagation.

We will try next to express the previous statement in mathematical terms. To avoid the formidable algebraic complication of

working with the full set of two fluid equations, we will use a simplified model, keeping only the parts relevant to this analysis.

We will consider only the mass and momentum equations for a single phase. We also put all explicit terms, which are not relevant to this problem into a generic term  $R^n$ . Then the conservation equations become:

$$\frac{\partial}{\partial t} \rho + \nabla \cdot \rho \vec{U} = 0 \quad (2.4.3)$$

$$\rho \frac{\partial \vec{U}}{\partial t} + \vec{U} \nabla \vec{U} + \frac{1}{\rho} \nabla P = -k \vec{U} \quad (2.4.4)$$

and the equation of state:

$$\frac{\partial \rho}{\partial P} = \frac{1}{c^2} \quad (2.4.5)$$

with  $c$  being the sonic speed.

Applying the differentiating scheme to these equations we get:

$$\frac{P_{ij}^{n+1} - P_{ij}^n}{c^2 \Delta t} + \frac{(\rho U_z^{n+1})_{i+\frac{1}{2}j} - (\rho U_z^{n+1})_{i-\frac{1}{2}j}}{\Delta z} + \frac{(\rho U_r^{n+1})_{ij+\frac{1}{2}} - (\rho U_r^{n+1})_{ij-\frac{1}{2}}}{\Delta r} = 0 \quad (2.4.6)$$

$$\frac{(U_z^{n+1} - U_z^n)_{i+\frac{1}{2}j}}{\Delta t} + \frac{1}{\rho_{i+\frac{1}{2}j}} \frac{P_{i+1j}^{n+1} - P_{ij}^{n+1}}{\Delta z} + k_z U_{zi+\frac{1}{2}j}^{n+1} = R_z^n \quad (2.4.7)$$

$$\frac{(U_r^{n+1} - U_r^n)_{ij+\frac{1}{2}}}{\Delta t} + \frac{1}{\rho_{ij+\frac{1}{2}}} \frac{P_{i+1j}^{n+1} - P_{ij}^{n+1}}{\Delta r} + k_r U_{rij+\frac{1}{2}}^{n+1} = R_r^n \quad (2.4.8)$$

We isolate  $U_z^{n+1}$  and  $U_r^{n+1}$  in equations 2.4.7 and 2.4.8:

$$(U_z^{n+1})_{i+\frac{1}{2}j} = \frac{1}{\rho_{ij+\frac{1}{2}}} \frac{\Delta t}{\Delta z} \frac{1}{1+k_z \Delta t} (P_{ij}^{n+1} - P_{i+1j}^{n+1}) + R_z^n \quad (2.4.9)$$

$$(U_r^{n+1})_{ij+\frac{1}{2}} = \frac{1}{\rho_{ij+\frac{1}{2}}} \frac{\Delta t}{\Delta z} \frac{1}{1+k_r \Delta t} (P_{ij}^{n+1} - P_{ij+1}^{n+1}) + R_r^n \quad (2.4.10)$$

It is possible now to eliminate the velocities in equation 2.4.6 to get an expression involving the pressure alone. If this equation is put in the form of equation 2.4.1 we then have the expression for the coefficients of the pressure problem matrix:

$$A_{ij} = -\left(\frac{\Delta t}{\Delta r}\right)^2 \frac{1}{1+k_r \Delta t} \quad (2.4.11a)$$

$$B_{ij} = -\left(\frac{\Delta t}{\Delta z}\right)^2 \frac{1}{1+k_z \Delta t} \quad (2.4.11b)$$

$$C_{ij} = -A_{ij} - B_{ij} - D_{ij} - E_{ij} + \frac{1}{c} \quad (2.4.11c)$$

$$D_{ij} = -\left(\frac{\Delta t}{\Delta z}\right)^2 \frac{1}{1+k_z \Delta t} \quad (2.4.11d)$$

$$E_{ij} = -\left(\frac{\Delta t}{\Delta r}\right)^2 \frac{1}{1+k_r \Delta t} \quad (2.4.11e)$$

The first point to be considered in these equations is the coefficient  $C_{ij}$  in equation 2.4.11c: Note that  $C_{ij}$  exceeds the sum of

the absolute values of the other coefficients by the factor  $1/c^2$ . In the numerical analysis language this means that the matrix of the coefficients is diagonal dominant, and it guarantees that the numerical inversion of this matrix will converge. Later on, when discussing the equations of state, we will insist that the equation for the density of both phases reflect some sort of compressibility, or in other words, that the derivative of the density with respect to the pressure be always a real positive number. Looking at equation 2.4.11c it can be seen that this requirement guarantees the diagonal dominance of the pressure problem matrix.

We now compare the coefficients  $A_{ij}$  and  $B_{ij}$  (which are in all similar to the pair of coefficients  $D_{ij}$  and  $E_{ij}$ ): As it has been established before,  $\Delta z$  is ten or more times larger than  $\Delta r$ , which means  $B_{ij}$  will be one hundred or more times smaller than  $A_{ij}$ . If we go back to equation 2.4.2 it can be seen that in the proposed scheme the errors contained in the pressure terms in the right-hand side will be multiplied by a coefficient which is very small compared to the coefficients in the left-hand side; therefore, the influence of these errors will be minimized, and the convergence of the scheme will be drastically improved.

In the comparison we have just made, the friction terms  $\frac{1}{1+k\Delta t}$  were neglected. This was done first because their influences are small, being the product  $k\Delta t$  not a large number compared to one.

Second, their influence is in the direction to enhance the disparity between the coefficients  $A_{ij}$  and  $B_{ij}$ . Clearly in all situations of practical interest the axial velocity will be two or three orders of magnitude larger than the radial velocity, which means the axial friction factor  $k_z$  will be larger than its radial counterpart.

Finally, to illustrate this point we ran a case with mesh cells whose dimensions were  $\Delta z = 30$  cm and  $\Delta r = 1$  cm, with the proposed scheme and with one which did the same procedure but exchanged the  $z$  axis by the  $r$  axis. In the first case we attained a convergence criterion of  $10^{-6}$  in less than 10 iterations. While with the second scheme the same convergence criterion could not be attained in ten thousand iterations.

## 2.5 Stability Analysis of the Numerical Method.

This chapter would not be complete without a study on the stability of the numerical method, and in the following paragraphs we will attempt to fulfill this requirement. We want to emphasize at this point that the following analysis is not rigorous in the mathematical sense, nor is it a definitive proof of the two fluid model stability. Because the tools of numerical analysis known to date were developed for systems of linear equations, they cannot be applied to the non-linear thermohydraulic equations without a few assumptions and simplifications, made to fit into the limitations of our tools. Even with this "local-linear" treatment of the system of equations, sometimes the algebraic complication of the study imposed a few approximations in order that we could have an intelligible conclusion. Nonetheless, this analysis gives a picture, if not rigorous, at least sufficiently clear for the understanding of the stability problems of the two-fluid model.

We will be following in this study a line developed by Stewart/ 51 / in which the stabilizing effects of the exchange terms are identified.

The first simplification made in this analysis was to reduce the full set of eight equations which make the two-dimensional, two-fluid problem to a system of only four equations, by taking the momentum equations in only one direction and neglecting the energy equations. Physically this situation corresponds to a one-dimensional, isothermal flow.

As we shall see later, we will be solving in this study, determinants and algebraic equations whose order is equal to the number of equations in our model. It is easy to understand that the algebraic difficulty of working with eighth order determinants and equations would be large enough to make it nearly impossible to visualize any kind of conclusion.

We will not be losing the desired degree of generalization with these simplifications, since the momentum equations are exactly the same for both directions, and the energy equations are differentiated in all similar to the mass equations. Therefore, all the characteristics of the eight-equation model will be represented in this analysis and the simplified system of equations will be, from the numerical point of view, analogous to the full two-fluid model.

We then write down the fluid-dynamic equations as:

$$\frac{\partial}{\partial t} \alpha \rho_v + \frac{\partial}{\partial z} \alpha \rho_v U_v = S \quad (2.5.1)$$

$$\frac{\partial}{\partial t} (1-\alpha) \rho_\ell + \frac{\partial}{\partial z} (1-\alpha) \rho_\ell U_\ell = -S \quad (2.5.2)$$

$$\alpha \rho_v \left[ \frac{\partial U_v}{\partial t} + U_v \frac{\partial U_v}{\partial z} \right] + \alpha \frac{\partial P}{\partial z} = k(U_\ell - U_v) \quad (2.5.3)$$

$$(1-\alpha) \rho_\ell \left[ \frac{\partial U_\ell}{\partial t} + U_\ell \frac{\partial U_\ell}{\partial z} \right] + (1-\alpha) \frac{\partial P}{\partial z} = k(U_v - U_\ell) \quad (2.5.4)$$

and the equations of state:

$$\frac{\partial \rho_v}{\partial P} = \frac{1}{c_v^2} \quad (2.5.5)$$

$$\frac{\partial \rho_l}{\partial P} = \frac{1}{c_l^2} \quad (2.5.6)$$

In the canonical form the above equations would appear as:

$$A \frac{\partial X}{\partial t} + B \frac{\partial X}{\partial z} = f(X) \quad (2.5.7)$$

with

$$X = [\alpha, P, U_v, U_l]^T \quad (2.5.8)$$

$$A = \begin{bmatrix} \rho_v & \alpha/c_v^2 & 0 & 0 \\ -\rho_l & (1-\alpha)/c_l^2 & 0 & 0 \\ 0 & 0 & \alpha\rho_v & 0 \\ 0 & 0 & 0 & (1-\alpha)\rho_l \end{bmatrix} \quad (2.5.9)$$

$$B = \begin{bmatrix} \rho_v U_v & \alpha U_v/c_v^2 & \alpha\rho_v & 0 \\ -\rho_l U_l & (1-\alpha)U_l/c_l^2 & 0 & (1-\alpha)\rho_l \\ 0 & \alpha & \alpha_v U_v & 0 \\ 0 & (1-\alpha) & 0 & (1-\alpha)\rho_l U_l \end{bmatrix} \quad (2.5.10)$$

With this formalism the characteristic roots of the system can be found, which are solutions of the equation:

$$\det[B - \lambda A] = 0 \quad (2.5.11)$$

The reduction of this characteristic determinant results in the algebraic equation:

$$\alpha \rho_l (U_l - \lambda)^2 + (1 - \alpha) \rho_v (U_v - \lambda)^2 - \left[ \frac{\alpha \rho_l}{c_v^2} + \frac{(1 - \alpha) \rho_v}{c_l^2} \right] (U_v - \lambda)^2 (U_l - \lambda)^2 = 0 \quad (2.5.12)$$

Since we are interested only in the qualitative aspect of the roots of this equation, rather than its precise value, we will make some approximations, in order to get a solution of 2.5.12 which are representative of the true value. We note that for the cases of practical interest the liquid density is much higher than the vapor density. Then it is reasonable to neglect the terms in  $\rho_v$ , and two real roots are obtained, which are approximately:

$$\lambda \approx U_v \pm c_v \quad (2.5.13)$$

On the other hand, with this model we intend to study only sub-sonic flow, hence both  $U_v$  and  $U_l$  are much smaller than the sonic velocities. Then, if the terms in  $1/c_v^2$  and  $1/c_l^2$  are neglected, the two other roots become:

$$\lambda = \frac{U_l + \epsilon^2 U_v}{1 + \epsilon^2} \pm \frac{i\epsilon(U_v - U_l)}{1 + \epsilon^2} \quad (2.5.14)$$

with

$$\epsilon^2 = \frac{(1-\alpha)\rho_v}{\alpha\rho_l} \quad (2.5.15)$$

It can be seen that whenever the phase velocities are different, the system will have two complex characteristic roots. This means the system of equations is not hyperbolic and consequently not well posed as an initial value problem. Nonetheless, with this conclusion it can only be said that the two-fluid problem failed to meet a sufficient condition, but it cannot be concluded that the problem is necessarily unstable. The previous analysis did not take into consideration the important stabilizing effect of the interphase exchange terms, and as we shall see later on, these terms are responsible for the stability of the two-fluid models.

To verify this effect, we will proceed with the Von Neumann analysis of the numerical scheme. The difference equations corresponding to equations 2.5.1 through 2.5.6 are:

$$\frac{\alpha_j^{n+1} \rho_j^{n+1} - \alpha_j^n \rho_j^n}{\Delta t} + \frac{\alpha_j^n \rho_j^n U_j^{n+1} - \alpha_{j-1}^n \rho_{j-1}^n U_{j-1}^{n+1}}{\Delta z} = S \quad (2.5.16)$$

$$\frac{(1-\alpha_j^{n+1})\rho_{\ell j}^{n+1} - (1-\alpha_j^n)\rho_{\ell j}^n}{\Delta t} + \frac{(1-\alpha_j^n)\rho_{\ell j}^n U_{\ell j+\frac{1}{2}}^{n+1} - \alpha_{j-1}^n \rho_{vj-1}^n U_{\ell j-\frac{1}{2}}^{n+1}}{\Delta z} = -S \quad (2.5.17)$$

$$\alpha_j^n \rho_{vj}^n \left[ \frac{U_{vj+\frac{1}{2}}^{n+1} - U_{vj+\frac{1}{2}}^n}{\Delta t} + U_{vj+\frac{1}{2}}^n \frac{(U_{vj+\frac{1}{2}}^n - U_{vj-\frac{1}{2}}^n)}{\Delta z} \right] + \alpha_j^n \frac{(P_{j+1}^{n+1} - P_j^{n+1})}{\Delta z} = k_{j+\frac{1}{2}} (U_{\ell j+\frac{1}{2}}^{n+1} - U_{vj+\frac{1}{2}}^{n+1}) \quad (2.5.18)$$

$$(1-\alpha_j^n)\rho_{\ell j}^n \left[ \frac{U_{\ell j+\frac{1}{2}}^{n+1} - U_{\ell j+\frac{1}{2}}^n}{\Delta t} + U_{\ell j+\frac{1}{2}}^n \frac{(U_{\ell j+\frac{1}{2}}^n - U_{\ell j-\frac{1}{2}}^n)}{\Delta z} \right] + (1-\alpha_j^n) \frac{(P_{j+1}^{n+1} - P_j^{n+1})}{\Delta z} = k_{j+\frac{1}{2}} (U_{\ell j+\frac{1}{2}}^{n+1} - U_{\ell j+\frac{1}{2}}^{n+1}) \quad (2.5.19)$$

The convective terms in the mass and momentum equations involve donor cell differencing, so the above equations are written for both  $U_v$  and  $U_\ell$  positive. To apply the Von Neumann method these equations must first be linearized. We thus expand the differences in terms of differences of the four basic variables individually, and treat the coefficient of these differences as constant. For simplicity we will neglect the liquid compressibility, so that we can substitute the difference terms in pressure by terms involving the vapor density alone and treat this variable as a basic one. If we recall the Von Neumann method, the error of any variable at a given time and location is expressed as:

$$\varepsilon_{xj+s}^{n+r} = \varepsilon_{xj}^n \xi^r e^{is\theta}$$

where

$\theta = \pi/m$  is the wave number.

Applying this formalism to equations 2.5.16 through 2.5.19 it follows:

$$\begin{aligned} \frac{\alpha}{\Delta t}(\xi-1)\varepsilon_{\rho v j}^n + \frac{\rho_v}{\Delta t}(\xi-1)\varepsilon_{\alpha j}^n + \frac{\alpha\rho_v}{\Delta z}(1-e^{-i\theta})\xi\varepsilon_{U v j+\frac{1}{2}}^n + \frac{\alpha U_v}{\Delta z}(1-e^{-i\theta})\varepsilon_{\rho v j}^n + \\ + \frac{\rho_v U_v}{\Delta z}(1-e^{-i\theta})\varepsilon_{\alpha j}^n = 0 \end{aligned} \quad (2.5.20)$$

$$-\frac{\rho_\ell}{\Delta t}(\xi-1)\varepsilon_{\alpha j}^n + \frac{(1-\alpha)\rho_\ell}{\Delta z}(1-e^{-i\theta})\xi\varepsilon_{U \ell j+\frac{1}{2}}^n - \frac{\rho_\ell U_\ell}{\Delta z}(1-e^{-i\theta})\varepsilon_{\alpha j}^n = 0 \quad (2.5.21)$$

$$\alpha\rho_v \left[ \frac{(\xi-1)}{\Delta t}\varepsilon_{U v j+\frac{1}{2}}^n + \frac{U_v}{\Delta z}(1-e^{-i\theta})\varepsilon_{U v j+\frac{1}{2}}^n \right] + \frac{\alpha C_v^2}{\Delta z}(e^{i\theta}-1)\xi\varepsilon_{\rho v j}^n = k\xi(\varepsilon_{U \ell j+\frac{1}{2}}^n - \varepsilon_{U v j+\frac{1}{2}}^n) \quad (2.5.22)$$

$$\begin{aligned} (1-\alpha)\rho_\ell \left[ \frac{(\xi-1)}{\Delta t}\varepsilon_{U \ell j+\frac{1}{2}}^n + \frac{U_\ell}{\Delta z}(1-e^{-i\theta})\varepsilon_{U \ell j+\frac{1}{2}}^n \right] + \frac{(1-\alpha)C_v^2}{\Delta z}(e^{i\theta}-1)\xi\varepsilon_{\rho v j}^n = \\ = k\xi(\varepsilon_{U v j+\frac{1}{2}}^n - \varepsilon_{U \ell j+\frac{1}{2}}^n) \end{aligned} \quad (2.5.23)$$

Rearranging these equations and putting them into matrix form it follows:

$$\mathbf{A} \mathbf{x} = \mathbf{0}$$

with

$$\mathbf{E} = [\varepsilon_{\rho v}, \varepsilon_{\alpha}, \varepsilon_{U v}, \varepsilon_{U \ell}]_{U, j}^T$$

and

$$A = \begin{bmatrix} \alpha(\xi-1+\tilde{U}_v) & \rho_v(\xi-1+\tilde{U}_v) & \xi \frac{\Delta t}{\Delta z} \alpha \rho_v i\tilde{\theta} & 0 \\ 0 & -\rho_\ell(\xi-1+\tilde{U}_\ell) & 0 & \xi \frac{\Delta t}{\Delta z} (1-\alpha) \rho_\ell i\tilde{\theta} \\ \xi C \frac{2\Delta z}{m\Delta t} i\tilde{\theta} & 0 & \rho_v(\xi-1+\tilde{U}_v) + \xi \rho_v \kappa & -\xi \rho_v \kappa \\ \xi C \frac{2\Delta z}{m\Delta t} i\tilde{\theta} & 0 & -\xi \rho_v \kappa & \rho_\ell(\xi-1+\tilde{U}_\ell) + \xi \rho_v \kappa \end{bmatrix}$$

Where we have abbreviated

$$\tilde{U}_v = U \frac{\Delta t}{v\Delta z} (1 - e^{-i\theta})$$

$$\tilde{U}_\ell = U_\ell \frac{\Delta t}{\ell\Delta z} (1 - e^{-i\theta})$$

$$\tilde{\theta} = 2 \sin \theta/2$$

$$C_m = \frac{C_v \Delta t}{\Delta z} \cdot 2 \sin \theta/2$$

$$\kappa = k \Delta t / \rho_\ell$$

$$\varepsilon^2 = \frac{(1-\alpha)\rho_v}{\alpha\rho_\ell}$$

In order for the errors in the basic variables not to grow geometrically, the absolute value of the eigenvalues  $\xi$  of the amplification matrix  $A$  must be all less than one. To find these eigenvalues we solve the equation  $\det[A] = 0$ . After reducing this determinant we end up with the algebraic equation

$$\begin{aligned} & \xi^2 C_m^2 [(\xi-1+\tilde{U}_v+2\xi\kappa)(\xi-1+\tilde{U}_v)\varepsilon^2 + (\xi-1+\tilde{U}_\ell)(\xi-1+U_\ell+2\xi\kappa) + \\ & + (\xi-1+\tilde{U}_v)(\xi-1+\tilde{U}_\ell) [(\xi-1+\tilde{U}_v+\xi\kappa)(\xi-1+U_\ell+\xi\kappa\rho_v/\rho_\ell) - (\xi\kappa)^2 \rho_v/\rho_\ell] = 0 \end{aligned} \quad (2.5.24)$$

The next step in the analysis would be to find the roots of this characteristic equation and see if their values would be less than one. But the expressions for the exact solution of the quartic equation are so complicated that it would be almost impossible to draw any conclusion from them. Instead we prefer to make some approximations which would give reasonably good values for the roots we are searching, but with the advantage of simple expressions which can give a clear visualization of them.

Since we want to emphasize the importance of the interphase exchange terms, we will first evaluate the characteristic roots of 2.5.24 with the momentum exchange coefficient  $k$  set to zero, and afterwards compare the results of this analysis with those obtained with a positive real non-zero value of  $k$ .

With  $k$  set to zero, equation 2.5.24 reduces to

$$\xi^2 C_m^2 [(\xi-1+\bar{U}_v)^2 \epsilon^2 + (\xi-1+\bar{U}_l)^2] + (\xi-1+\bar{U}_v)^2 (\xi-1+\bar{U}_l)^2 = 0 \quad (2.5.25)$$

First consider the high frequency behavior. As has been said before, the model uses the time step size  $\Delta t$  equal to the convective limit:  $\Delta t = \min (\Delta z/U_v, \Delta z/U_l)$

Also notice that the phase velocities are small compared to the vapor sonic velocity. Thus  $C_v \Delta t/\Delta z \gg 1$  and for small  $m$ ,

$C_m^2 \gg 1$ . It follows that equation 2.5.25 will have two roots of magnitude approximately  $\xi \approx \pm 1/C_m$ , which are smaller than one.

The other two roots approximately satisfy:

or

$$\xi \approx \frac{1 - \tilde{U}_\ell (1 \pm i\epsilon \tilde{U}_v / \tilde{U}_\ell)}{1 \pm i\epsilon} \quad (2.5.26)$$

In the complex plane this is represented by a circle of radius  $\tilde{U}_\ell \Delta t / \Delta z$ , touching the point one, tilted by an angle  $\pm \arctan(\epsilon \tilde{U}_v / \tilde{U}_\ell)$  and back through an angle  $\pm \arctan \epsilon$ . Clearly, with small  $m$ , points on this circle will not be outside the unit circle if the limit is satisfied:

$$\frac{\tilde{U}_\ell \Delta t}{\Delta z} \leq 1 \quad (2.5.27)$$

and

$$\frac{\tilde{U}_v \Delta t}{\Delta z} \leq 1 \quad (2.5.28)$$

We then conclude that even without momentum exchange the high frequency modes will not grow geometrically if the convective limit is observed.

Now let us turn to the low frequency modes. As  $m \rightarrow \infty$ ,  $C_m \rightarrow 0$ ) and in the limit the roots of 2.5.25 will be:

$$\xi = \pm 1 - \tilde{U}_v \quad (2.5.29)$$

and

$$\xi = \pm 1 - \tilde{U}_\ell \quad (2.5.30)$$

Then, let us say that for  $m$  large but finite the roots of 2.5.25 are:

$$\xi = 1 - \tilde{U}_\ell + \delta \quad (2.5.31)$$

We can evaluate this perturbation  $\delta$  by substituting 2.5.31 into 2.5.25, and neglecting the terms of order higher than  $\delta^2$ . The resulting quadratic equation will be:

$$\left[1 + \varepsilon^2 - \frac{(\tilde{U}_v - \tilde{U}_\ell)^2}{C_m^2}\right] \delta^2 - 2\varepsilon^2(1 - \tilde{U}_\ell)(\tilde{U}_v - \tilde{U}_\ell)\delta - \varepsilon^2(1 - \tilde{U}_\ell)^2(\tilde{U}_v - \tilde{U}_\ell)^2 = 0 \quad (2.5.32)$$

and the roots of this equation are:

$$\delta = (1 - \tilde{U}_\ell)(\tilde{U}_v - \tilde{U}_\ell) \left[ \frac{\varepsilon^2 \pm \varepsilon \sqrt{1 - (\tilde{U}_v - \tilde{U}_\ell)^2 / C_m^2}}{1 + \varepsilon^2 - (\tilde{U}_v - \tilde{U}_\ell)^2 / C_m^2} \right] \quad (2.5.33)$$

and again using the fact that  $(\tilde{U}_v - \tilde{U}_\ell) / C_v \ll 1$  we can write the expression for the characteristic root  $\xi$  as:

$$\xi \approx (1 - \tilde{U}_\ell) \left[ 1 - \frac{(\tilde{U}_v - \tilde{U}_\ell) \varepsilon (\varepsilon \pm 1)}{1 + \varepsilon^2} \right] \quad (2.5.34)$$

Since  $|1 - \tilde{U}_\ell|$  and  $|1 + (\tilde{U}_v - \tilde{U}_\ell)|$  are of the same order of magnitude for some value of  $\epsilon$  one root  $\xi$  may lie outside the unit circle. Therefore, without the momentum exchange term  $k$ , the low frequency modes will grow geometrically and the method would be unstable. Nonetheless, with very few spacial mesh cells, i.e., with small  $m$  the model may have a well behaved solution even without the momentum exchange term.

We now return to equation 2.5.24 to verify the effect of the momentum exchange term. For the high frequency modes, the same considerations are made as in the previous analysis with  $k$  equal to zero and it is clear that the roots will be of the same form, only multiplied by a factor which is approximately  $1/(1+\kappa)$ . Since in that analysis we concluded that the characteristic roots were less than one in magnitude, we can extend with confidence this result to the present case and conclude that for small values of  $m$  the model will present a well behaved solution.

To study the low frequency behavior again consider the limiting case as  $m \rightarrow \infty$  and then introduce a perturbation of order  $1/m$ . Then, for  $m \rightarrow \infty$  equation 2.5.24 becomes:

$$(\xi - 1 + \tilde{U}_v)(\xi - 1 + \tilde{U}_\ell) \left[ (\xi - 1 + \tilde{U}_v + \xi\kappa)(\xi - 1 + \tilde{U}_\ell + \xi\kappa \rho_v/\rho_\ell) - \xi^2 \kappa^2 \rho_v/\rho_\ell \right] = 0$$

(2.5.35)

and the roots of this equation are:

$$\xi = 1 - \tilde{U}_v \quad (2.5.36a)$$

$$\xi = 1 - \tilde{U}_\ell \quad (2.5.36b)$$

$$\xi = \sqrt{(1 - \tilde{U}_v)(1 - \tilde{U}_\ell)} \quad (2.5.36c)$$

$$\xi = \sqrt{\frac{(1 - \tilde{U}_v)(1 - \tilde{U}_\ell)}{1 + \kappa \rho_v / \rho_\ell}} \quad (2.5.36d)$$

Recall that when the difference equations 2.5.16 - 2.5.19 were formed a donor cell scheme was used, which guarantees the reduced velocities  $\tilde{U}_v$  and  $\tilde{U}_\ell$  are always positive, so all four characteristic roots in 2.5.36 are always real positive and strictly less than one.

As done before we will investigate the effect of a perturbation  $\delta$  in those roots, which stands for a large but finite value of  $m$ . It is clear from the expressions of equations 2.5.36 that if we analyze the effect of the perturbation in one of the first two values of  $\xi$ , the conclusion obtained in this way will stand for all the other three roots.

We then substitute  $\xi = 1 - \tilde{U}_v + \delta$  into equation 2.5.24 and keep only the first order terms in  $\delta$ . This will give a first order equation, and the single root of this equation gives an expression for  $\xi$  as:

$$\xi = (1 - \tilde{U}_v) \frac{\kappa + C_m^2}{\kappa + 2C_m^2} \quad (2.5.37)$$

which is strictly less than unity. To get this result we have assumed:

$$\rho_\ell |\tilde{u}_\ell - \tilde{u}_v| \ll \rho_v \kappa \quad (2.5.38)$$

This condition establishes a minimum value for the momentum exchange coefficient, in order to avoid exponentially growing modes. Stewart / 51 / showed that the condition in 2.5.38 implies that the wave length  $m\Delta z$  will not have a growing mode if it is larger than a certain multiple of the radius of an individual bubble or droplet.

To summarize, in this section we have seen that although the two-fluid formulation have at least two complex characteristic roots, this does not imply that a well behaved solution cannot be achieved. With the Von Neumann stability analysis we have shown that the numerical scheme used in our model, with a donor cell differencing will have non-growing high frequency modes for any value of the momentum exchange coefficient, and for the low frequency modes a well behaved solution requires a minimum value for  $k$ , expressed in 2.5.38.

### III. THE CONSTITUTIVE EQUATIONS AND FUNCTIONS OF STATE

#### 3.1 The Sodium Functions of State and Transport Properties.

The basic source for the sodium properties is a compilation by Golden and Tokar / 46 /, dated 1966. This source has been used extensively since then in sodium technology with great success. Although a recent compilation by the Argonne National Laboratory / 15 / has come to our knowledge, but not yet published, we decided to stay with that of Golden and Tokar on the basis of its wide use and acceptance. A comparison between the new compilation and the one used by us showed a wider range of validity in terms of temperatures and pressure in favor of the new one, but no significant disagreement between them.

A few modifications were made in the original expressions to satisfy program requirements, and all properties were converted to S I units. To help a quick reference to these properties we list them in table 3.1, with the correspondence to usual units.

##### 3.1.1 Saturation Temperature

From the several correlations for the saturation temperature listed in / 46 /, the one which showed the best agreement in the most important range of temperatures 870 - 1100°C (1600 - 2000°F) is the one from Makansi et al, which is valid in the range 620 - 1150°C.

Table 3.1

Units Used in this Work and the Correspondent Usual Ones

<u>Property</u>	<u>SI Units</u>	<u>Equal to</u>
Temperature	°K	°C + 273.15
Pressure	Pa	$14.05 \times 10^{-5}$ lbf/in <sup>2</sup>
Density	kg/m <sup>3</sup>	0.06243 lbm/ft <sup>3</sup>
Internal Energy	J/kg	$4.2992 \times 10^{-4}$ BTU/lbm
Viscosity	kg/m-sec	0.672 lbm/ft sec
Thermal Conductivity	W/m °K	0.5778 BTU/hr ft °F
Specific Heat	J/kg °K	$2.3884 \times 10^{-4}$ BTU/lbm °F
Surface Tension	N/m	—

The expression is:

$$T_{\text{sat}}(P) = \frac{a}{b - \ln P}$$

with

$$a = 1.2020 \times 10^4$$

$$b = 21.9358$$

valid for

$$4.8 \times 10^3 < P < 6.6 \times 10^5$$

### 3.1.2 Vapor Density

For the vapor density the expression which gives the density at saturation conditions was used and a perfect gas behaviour in the superheated zone was assumed:

$$\rho_v(P, T) = (rv_0 + rv_1 P + rv_2 P^2) \frac{T_{\text{sat}}(P)}{T}$$

with

$$rv_0 = 1.605 \times 10^{-2}$$

$$rv_1 = 2.510 \times 10^{-6}$$

$$rv_2 = 3.230 \times 10^{-13}$$

valid for

$$3.4 \times 10^4 < P < 2.3 \times 10^6$$

### 3.1.3 Liquid Density

From all correlations we have reviewed for the liquid density none showed a pressure dependence. This can be explained because the compressibility effect for the liquid phase is very small, usually smaller than the accuracy of the expressions themselves. Therefore it is reasonable, if one is interested only in the absolute value of that property, to neglect the liquid compressibility. But as seen in chapter 2, the model requires not only the value of the properties but also their derivatives with respect to pressure and temperature. It is clear from the physical point of view that however small, a liquid compressibility exists (otherwise the sonic speed would be infinity).

The estimate of liquid compressibility does not have to be very accurate, since as said before its effect is smaller than the accuracy of the equation of state. Therefore, a simple expression will satisfy the program requirements. With this idea in mind, the approximation was used:

$$\left( \frac{\partial \rho}{\partial P} \right)_{\text{constant temperature}} = \left( \frac{\partial \rho}{\partial P} \right)_{\text{constant entropy}} = \frac{1}{C^2}$$

where C is the speed of sound.

A constant sonic speed was taken, equal to 2,100 m/sec, which corresponds to a temperature of approximately 900°C, and the expression for the liquid density becomes:

$$\rho_l(P,T) = r\ell_0 + r\ell_1 T + r\ell_2 T^2 + r\ell_3 T^3 + r\ell_4 P$$

with

$$r\ell_0 = 1.0116 \times 10^3$$

$$r\ell_1 = - 0.2205$$

$$r\ell_2 = - 1.9224 \times 10^{-5}$$

$$r\ell_3 = 5.6377 \times 10^{-9}$$

$$r\ell_4 = 2.26 \times 10^{-7}$$

which is valid in the range

$$100 < T < 1370^\circ\text{C}$$

### 3.1.4 Internal Energies

The source of sodium properties gives only the expressions for the enthalpies. Therefore the internal energies were derived as

$$e = h - P/\rho$$

For the liquid enthalpy the following expression has been

used:

$$h_l(T) = h\ell_0 + h\ell_1 T + h\ell_2 T^2 + h\ell_3 T^3$$

with:

$$h\ell_0 = - 6.7507 \times 10^4$$

$$h\ell_1 = 1.6301 \times 10^3$$

$$h\ell_2 = - 0.41672$$

$$h\ell_3 = 1.5427 \times 10^{-4}$$

valid in the range

$$100 < T < 1500^{\circ}\text{C}$$

The vapor enthalpy is derived from the liquid expression. Again a perfect gas behavior is assumed for the vapor phase, in which the enthalpy of the super heated vapor is equal to that of saturated vapor at the same temperature. It follows:

$$h_v(T) = h_l(T) + hv_0 + hv_1 T$$

with

$$hv_0 = 5.089 \times 10^6$$

$$hv_1 = -1.043 \times 10^3$$

valid for  $600^{\circ} < T < 1200^{\circ}\text{C}$

### 3.1.5 Transport Properties

Following a list of the transport properties used in the model, again from reference / 46 / is presented

#### Liquid Thermal Conductivity

$$K_l(T) = Cl_0 + Cl_1 T + Cl_2 T^2$$

with

$$Cl_0 = 1.0969 \times 10^2$$

$$Cl_1 = -6.4494 \times 10^{-2}$$

$$Cl_2 = 1.1727 \times 10^{-5}$$

valid for

$$100 < T < 1370^{\circ}\text{C}$$

Vapor Thermal Conductivity

$$K_V(T) = Cv_0 + Cv_1 T + Cv_2 T^2$$

with

$$Cv_0 = - 3.2349 \times 10^{-2}$$

$$Cv_1 = 1.5167 \times 10^{-4}$$

$$Cv_2 = - 5.4376 \times 10^{-8}$$

for

$$700 < T < 5000^\circ\text{C}$$

Liquid Viscosity

$$\eta(T) = \exp[v\ell_0 + \frac{v\ell_1}{T} + v\ell_2 \ln T]$$

with

$$v\ell_0 = - 5.732$$

$$v\ell_1 = 508.7$$

$$v\ell_2 = - 0.4925$$

for

$$100 < T < 1370^\circ\text{C}$$

Vapor Viscosity

$$\eta_V(T) = vv_0 + vv_1 T$$

with

$$vv_0 = 1.261 \times 10^{-5}$$

$$vv_1 = 6.085 \times 10^{-9}$$

for

$$700 < T < 5000^\circ\text{C}$$

Liquid Specific Heat

$$C_{p^l}(T) = C_{p^l_0} + C_{p^l_1} T + C_{p^l_2} T^2$$

with

$$C_{p^l_0} = 1.6301 \times 10^3$$

$$C_{p^l_1} = -0.83344$$

$$C_{p^l_2} = 4.6281 \times 10^{-4}$$

for

$$100 < T < 1500^\circ\text{C}$$

Vapor Specific Heat

$$C_{p^v}(T) = C_{p^v_0} + C_{p^v_1} T + C_{p^v_2} T^2$$

with

$$C_{p^v_0} = 0.5871 \times 10^3$$

$$C_{p^v_1} = -0.83344$$

$$C_{p^v_2} = 4.6281 \times 10^{-4}$$

for

$$600 < T < 1200^\circ\text{C}$$

Surface Tension

$$\sigma(T) = st_0 + st_1 T$$

with

$$st_0 = 0.18$$

$$st_1 = -1.0 \times 10^{-4}$$

in the range

$$100 < T < 1370^\circ\text{C}$$

Finally we observed that the vapor Prandtl number showed a very smooth variation with temperature. Thus in order to save computation time a quadratic expression for the Prandtl number was fitted

$$\text{Pr}_v(T) = \text{pv}_0 + \text{pv}_1 (T - \text{pv}_2)^2$$

with

$$\text{pv}_0 = 0.7596$$

$$\text{pv}_1 = 0.810 \times 10^{-6}$$

$$\text{pv}_2 = 844.4$$

where the range of validity for this expression is taken as the smallest of the ranges of the properties composing this dimensionless number:

$$600 < T < 1200^\circ\text{C}$$

### 3.2 Mass Exchange Rate

It has been stated in Chapter 2 that the interphase exchange terms play a key role in the stability of the Two-Fluid Model. Of all exchange terms, the mass exchange rate is the most critical one to the code stability. Because of the large difference in densities between the liquid and vapor phases for the usual range of pressures encountered in sodium technology, a small amount of mass transferred between phases corresponds to a very large volume change, and consequently large pressure and velocity variations.

In particular for this model, where the solution of the fluid dynamic equations is reduced to a pressure problem, these large pressure variations must be handled with extreme care. To insure the code stability, a choice is to be made of an adequate model for the mass exchange rate and its most strongly varying terms are to be implicitly treated.

In general, the mass exchange rate  $S$  will be a function of the void fraction, pressure, temperatures and velocities, evaluated both at the old and new time levels. If the solution technique of chapter 2 is recalled, the derivatives of  $S$  with respect to the properties at the new time value are required, therefore the mass exchange rate is to be a continuous, differentiable function in these variables.

The mass exchange model used in the code is derived from the principles of the kinetic theory, in which the net mass flux  $j$  crossing an imaginary plane between phases is given by:

$$j = \sqrt{\frac{M}{2\pi R}} \cdot \frac{P_v}{\sqrt{T_v}} - \frac{P_l}{\sqrt{T_l}} \quad (3.2.1)$$

where

$J$  = mass flux (mass per unit time per unit area)

$M$  = molecular weight

$R$  = universal gas constant

$P$  and  $T$  = absolute pressure and temperature for both phases.

For small differences in pressure and temperature, the above expression can be reduced to:

$$j = \sqrt{\frac{M}{2\pi R}} \frac{P}{\sqrt{T_s}} \left[ \frac{\Delta P}{P} - \frac{\Delta T}{2T_s} \right] \quad (3.2.2)$$

the Clayperon equation

$$\left. \frac{dP}{dT} \right)_{\text{sat}} = \frac{h_{fg}}{T v_{fg}} \quad (3.2.3)$$

is used to eliminate  $\Delta P$  in equation 3.2.2 leading to:

$$j = \sqrt{\frac{R}{2\pi M}} \rho v \left( \frac{h_{fg}}{P v_{fg}} - \frac{1}{2} \right) \frac{\Delta T}{\sqrt{T_s}} \quad (3.2.4)$$

where

$h_{fg}$  = difference in enthalpy between phases

$v_{fg}$  = difference in specific volume between phases

and where the simplification was made:

$$v = \frac{P}{RT/M}$$

For the particular case of sodium, a few more simplifications in equation 3.2.4 can be made. First note that  $\rho_l \gg \rho_v$ , thus:

$$v_{fg} = \frac{1}{\rho_v} - \frac{1}{\rho_l} \approx \frac{1}{\rho_v}$$

second, for the actual values of  $h_{fg}$ ,  $P$  and  $v_{fg}$  it follows

$$\frac{h_{fg}}{P v_{fg}} \gg \frac{1}{2}$$

Therefore equation 3.2.4 becomes:

$$j = \sqrt{\frac{R}{2\pi M}} \frac{\rho_v^2 h_{fg}}{P} \frac{\Delta T}{\sqrt{T_s}} \quad (3.2.5)$$

The above equation was obtained with the assumptions of ideal conditions embodied in the kinetic theory. Although this model's predictions are in good agreement with experimental data for evaporation, large discrepancies appear when condensation is considered. Silver and Simpson /41/ suggested a correction factor, which modifies equation 3.2.5 for condensation:

$$j_c = \frac{2\sigma}{2-\sigma} \sqrt{\frac{R}{2\pi M}} \frac{\rho_v^2 h_{fg}}{P} \frac{\Delta T}{\sqrt{T_s}} \quad (3.2.6)$$

Figure 3.1 reproduced from reference 41 shows the value of  $\sigma$  as a function of pressure. From this figure, it can be seen that for the range of pressures expected to be encountered in LMFBR safety analysis, the value of  $\sigma$  is relatively small, thus the simplification can be made:

$$\frac{2\sigma}{2-\sigma} \approx \sigma$$

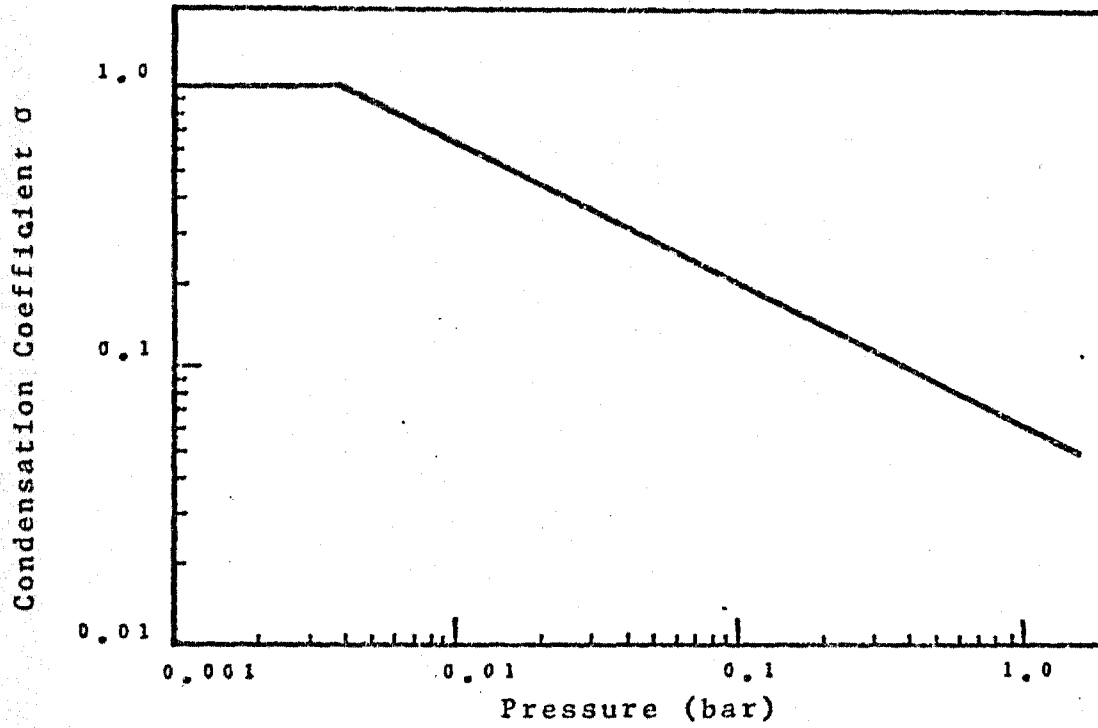


Figure 3.1 Condensation Coefficient as a Function of Pressure  
(From Reference 41)

Considering also the small variation of  $\sigma$  with the pressure, and the uncertainties involved in obtaining this coefficient, a reasonable approximation is to take a constant value for  $\sigma$ . Thus, for the pressure equal to one atmosphere the value of  $\sigma$  is:

$$\sigma = 0.005$$

The next factor to be evaluated in the mass exchange rate is the specific area between phases. Wilson /11/ proposed a model which takes into account three flow regimes - bubbly, annular flow and dry out. For the bubbly regime, with void fraction less than 0.6, he assumes the bubbles forming in the middle of each subchannel, packed on top of each other. (Figure 3.2) With this assumption, the expression for the specific area becomes:

$$\frac{A}{V} = \frac{4}{D} \sqrt{\frac{3\pi\alpha}{2/3 (P/D)^2 - \pi}} \quad \alpha < 0.6 \quad (3.2.7)$$

where

D = fuel pin diameter

P/D = pitch to diameter ratio

Although this model predicts reasonable values for the specific area at high values of the quality, for small void fractions this model would postulate the existence of unreasonably small vapor bubbles, thus overestimating the specific area. To correct this we introduced a minimum value for the bubble radius, so that for

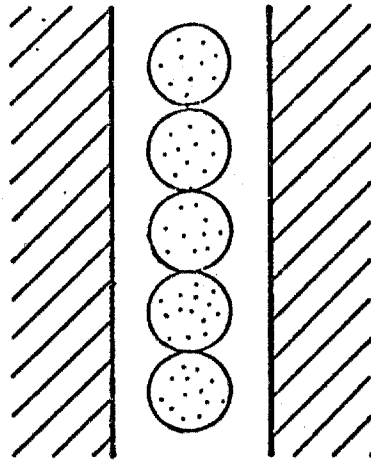


Figure 3.2 Bubbly Flow Representation

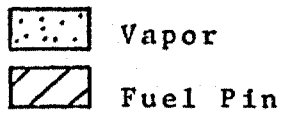
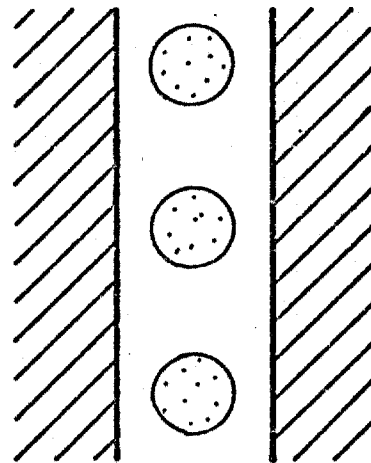


Figure 3.3 Low Void Fraction Bubbly Flow Representation

small void fractions the model would be pictured as in Figure 3.3.

The expression for the specific area becomes:

$$\frac{A}{V} = \frac{3\alpha}{r_m} \quad \alpha < \alpha_m \quad (3.2.8)$$

where

$$\alpha_m = \frac{8}{3} \left( \frac{r_m}{D} \right)^2 \frac{\pi}{\sqrt{3} (P/P)^2 - \pi/2}$$

where  $\alpha_m$  was chosen so that the two expressions of equations 3.2.7 and 3.2.8 be continuous at  $\alpha_m$ , and  $r_m$  is the minimum bubble radius, which was taken in our model equal to  $6 \times 10^{-4}$  m.

For the annular flow, all the liquid is assumed to be flowing in a circular annulus around the fuel rods, and the expression for the specific area becomes:

$$\frac{A}{V} = \frac{4}{D} \sqrt{\frac{2\sqrt{3} \pi (P/D)^2}{[2\sqrt{3} (P/D)^2 - \pi]^2}} - \frac{\pi\alpha}{2\sqrt{3} (P/D)^2 - \pi} \quad (3.2.9)$$

for  $0.6 < \alpha < 0.957$

Finally in the dryout regime a partial contact of the vapor with the fuel pin walls is assumed, and the expression for the area becomes:

$$\frac{A}{V} = \frac{4}{D} \sqrt{\left[ \frac{2 \sqrt{3} \pi (P/D)^2}{(2 \sqrt{3} (P/D)^2 - \pi)^2} - \frac{\pi \alpha}{2 \sqrt{3} (P/D)^2 - \pi} \right] \left[ \frac{1 - \alpha}{1 - .957} \right]} \quad (3.2.10)$$

for  $\alpha > 0.957$

where the dryout transition point were taken from the work by Autruffe /50/ analysing the KFK experiments /52/.

Note that the transition from bubbly to annular flow presents a discontinuity in the specific area, whose magnitude is a function of the pitch to diameter ratio. The transition at  $\alpha = 0.6$  was chosen to minimize this discontinuity for the usual pitch to diameter ratio of 1.25. Finally note that in the limiting case  $\alpha = 0$  or  $\alpha = 1$  the interphase area is obviously zero. This would prevent the initiation of boiling or condensation. To overcome this difficulty a "seed" void fraction is introduced to account for the initiation of phase transition. In this way  $\alpha$  is substituted in equations 3.2.8 and 3.2.10 by  $\hat{\alpha}$  which is defined as:

$$\hat{\alpha} = \begin{cases} \alpha & \text{if } \alpha > 10^{-4} \\ 10^{-4} & \text{if } \alpha \leq 10^{-4} \end{cases}$$

$$\hat{\alpha} = \begin{cases} \alpha & \text{if } \alpha < .9999 \\ .9999 & \text{if } \alpha \geq .9999 \end{cases}$$

Now the question of determining which terms are to be evaluated at the new or old time level can be addressed. The specific area must be evaluated at the old time level since the discontinuity in the transition from bubbly to annular flow makes it impossible to obtain the derivative of the mass exchange rate.

Both the enthalpy of vaporization  $h_{fg}$  and the vapor density does not show a marked dependence on the primary variables pressure and temperatures, therefore they can also be evaluated at the old time level.

On the other hand, the temperatures and pressure appearing in the expression of the mass flux have a very important dependence, thus they must be taken at the new time value.

Following is a summary of the equations used for the mass exchange rate:

$$S = S_e - S_c \quad (3.2.11)$$

$$S_e = A\alpha \sigma_e \sqrt{\frac{R}{2\pi M}} \left[ \frac{\rho_v^2 h_{fg}}{P} \right]^n \left[ \frac{(T_l - T_s)(1-\alpha)}{T_s} \right]^{n+1} \quad (3.2.12)$$

$$S_c = A(1-\alpha) \sigma_c \sqrt{\frac{R}{2\pi M}} \left[ \frac{\rho_v^2 h_{fg}}{P} \right]^n \left[ \frac{(T_s - T_v)\alpha}{T_s} \right]^{n+1} \quad (3.2.13)$$

where

$$\sigma_e = \begin{cases} 0 & \text{if } T_l < T_s \\ 1.0 & \text{if } T_l \geq T_s \end{cases} \quad (3.2.14)$$

$$\sigma_c = \begin{cases} 0 & \text{if } T_v > T_s \\ 0.005 & \text{if } T_v \leq T_s \end{cases} \quad (3.2.15)$$

$$A = \frac{3 \hat{\alpha}}{r_m} \quad \text{for } \alpha < \alpha_m \quad (3.2.16)$$

$$\alpha_m = \frac{8}{3} \left( \frac{r_m}{D} \right)^2 \frac{\pi}{\sqrt{3} (P/D)^2 - \pi/2} \quad (3.2.17)$$

$$A = \frac{4}{D} \sqrt{\frac{3 \pi \alpha}{2\sqrt{3} (P/D)^2 - \pi}} \quad \alpha_m < \alpha < 0.6 \quad (3.2.18)$$

$$A = \frac{4}{D} \sqrt{\frac{2\sqrt{3} \pi (P/D)^2}{2\sqrt{3} (P/D)^2 - \pi^2} - \frac{\pi \alpha}{2\sqrt{3} (P/D)^2 - \pi}} \quad (3.2.19)$$

$$0.6 < \alpha < 0.957$$

$$A = \frac{4}{D} \sqrt{\left[ \frac{2\sqrt{3} \pi (P/D)^2}{(2\sqrt{3} (P/D)^2 - \pi)^2} - \frac{\pi \alpha}{2\sqrt{3} (P/D)^2 - \pi} \right] \left[ \frac{1 - \hat{\alpha}}{1 - .957} \right]} \quad (3.2.20)$$

$$\alpha > 0.957$$

$$\hat{\alpha} = \begin{cases} 10^{-4} & \text{if } \alpha \leq 10^{-4} \\ \alpha & \text{if } 10^{-4} < \alpha < .9999 \\ .9999 & \text{if } \alpha \geq .9999 \end{cases} \quad (3.2.21)$$

$$r_m = 6 \times 10^{-4} \text{ m}$$

$$\frac{R}{M} = 361.30 \quad \text{J/kg } ^\circ\text{K}$$

### 3.3 Momentum Exchange

In this section we identify two kinds of momentum transfer in the fluids dynamic equations. One represents the interaction of the fluid with the fuel pins and fuel assembly structure, and the second one accounts for the momentum exchange between the phases themselves. Furthermore, because the fuel assembly geometry presents a very marked difference in the flow path for the axial and radial directions, we will have a different set of correlations for each direction.

Starting with the axial direction, a set of correlations developed by Autruffe/ 50 / analyzing the KFK experiments/ 52 / is used. The experiments were a series of steady state, single tube tests for several mass flow rates and qualities. Studying the pressure drop in the unheated zone (thus with no change in quality) the following correlations were proposed.

#### Liquid wall friction: axial direction

$$F_{lz} = \left[ \frac{0.18}{2D_H} Re_l^{-.2} \rho_l |U_{lz}| \right]^n U_{lz}^{n+1} \quad \alpha < \alpha_{dry} \quad (3.3.1)$$

$$F_{lz} = \left[ \frac{0.18}{2D_H} Re_l^{-.2} \rho_l |U_{lz}| \frac{(1-\alpha)}{(1-\alpha_{dry})} \right]^n U_{lz}^{n+1} \quad \alpha \geq \alpha_{dry} \quad (3.3.2)$$

with

$$Re_l = \frac{(1-\alpha)\rho_l |U_{lz}| D_H}{\eta_l} \quad (3.3.3)$$

$$D_H = 4 \times \frac{\text{free volume in tube bank}}{\text{exposed surface area of tubes}}$$

$$\alpha_{\text{dry}} = 0.957$$

Vapor wall friction: axial direction

$$F_{vz} = 0 \quad \alpha \leq \alpha_{\text{dry}}$$

$$F_{vz} = \left[ \frac{0.2}{2D_H} \text{Re}_v^{-0.2} \alpha \rho_v |U_{vz}| \right]^n U_{vz}^{n+1} \quad \alpha > \alpha_{\text{dry}} \quad (3.3.4)$$

with

$$\text{Re}_v = \frac{\alpha \rho_v |U_{vz}| D_H}{\eta_v} \quad (3.3.5)$$

Interphase momentum exchange: axial direction

$$M_z = K_z^n (U_{vz} - U_{lz})^{n+1} \quad (3.3.6)$$

with

$$K_z = \frac{4.31}{2D_H} \rho_v |U_{vz} - U_{lz}| [(1-\alpha)(1+75(1-\alpha))]^{.95} \quad (3.3.7)$$

Wilson/ 11 / introduced another term in the expression for the interphase momentum exchange, taking into account the momentum transport associated with the interphase mass exchange. In this formulation, the equation for the momentum exchange becomes:

$$M_z = (K_z + S)^n (U_{vz} - U_{lz})^{n+1} \quad (3.3.8)$$

where  $S$  is the mass exchange rate.

We also introduced in the above set of equations a term to represent a localized pressure drop, thus enabling the model to simulate fuel pin spacers or blockages. The expression, which adds up to the liquid wall friction is:

$$\Delta P_L = [K_L |U_{\ell z}|]^n U_{\ell z}^{n+1} \quad (3.3.9)$$

where  $K_L$  is an input parameter.

If for the axial direction momentum exchange we could find in the literature a number of sodium experiments, for the radial direction this abundance of data does not exist. But if we look into the dimensionless numbers involved in the momentum exchange models, we note the absence of the Prandtl number. Indeed, this number represents the energy transfer associated with momentum transport, and does not influence the pure momentum transfer we are interested here. Since of all dimensionless numbers involved in transport processes the Prandtl number is the only one which differentiates sodium from the other usually encountered fluids, we can expect to have good results if we use for our sodium momentum exchange a model developed for another fluid.

For the wall friction two correlations widely accepted in heat exchanges and boiler technology, were considered. One is by Kays and London/ 48 /and the other by Gunter and Shaw/ 49 /. Both correlations present approximately the same value for the friction factor, thus we made our choice in favor of the second one because its formulation is more conveniently adapted to our code. The correlations adopted are:

Liquid wall friction: radial direction

$$F_{lr} = \left[ \frac{f_{lr}}{2D_H} \rho \ell |U_{lr}^m| \right]^n (U_{lr}^m)^{n+1} \quad (3.3.10)$$

where

$$f_{lr} = \begin{cases} \frac{180}{Re_{lr}} & Re_{lr} \leq 202.5 \\ 1.92 Re_{lr}^{-.145} & Re_{lr} \geq 202.5 \end{cases} \quad (3.3.11)$$

$$Re_{lr} = \frac{\rho \ell |U_{lr}^m| D_H}{\eta_l} \quad (3.3.12)$$

and  $U_{lr}^m$  is the radial velocity at the point of maximum flow constriction between rods, and the hydraulic diameter  $D_H$  is the same as for the axial direction.

For the vapor wall friction and interphase momentum exchange we found very little in the literature. Therefore we proposed a formulation for these terms consistent with the one used for the other terms.

Vapor wall friction: radial direction

$$F_{vr} = 0 \quad \alpha \leq \alpha_{dry} \quad (3.3.13)$$

$$F_{vr} = \left[ \frac{f_{vr}}{2D_H} \rho v |U_{vr}^m| \right]^n (U_{vr}^m)^{n+1} \quad \alpha > \alpha_{dry}$$

with

$$f_{vr} = \begin{cases} \frac{180}{Re_{vr}} & Re_{vr} \leq 202.5 \\ 1.92 Re_{vr}^{-.145} & Re_{vr} > 202.5 \end{cases} \quad (3.3.14)$$

with

$$Re_{vr} = \frac{\rho v |U_{vr}^m| D_H}{\eta_v} \quad (3.3.15)$$

and here again,  $U_{vr}^m$  is the vapor radial velocity at the point of maximum constriction between the fuel pins.

Interphase momentum exchange: radial direction

$$M_r = K_r^n (U_{vr}^m - U_{lr}^m)^{n+1} \quad (3.3.16)$$

with

$$K_r = \frac{4.31}{2D_H} \rho v |U_{vr}^m - U_{lr}^m| [(1-\alpha)(1+75(1-\alpha))]^{.95} \quad (3.3.17)$$

To evaluate the velocities at the point of minimum transverse flow area we recall Chapter Two, where the primary radial velocities were defined as being the volume average velocities in the cell. One of the assumptions made in the derivations of that chapter was:

$$U_r(r)A_r(r) = \text{constant}$$

Thus the average velocity in the cell is:

$$\begin{aligned} \langle U_r \rangle &= \frac{1}{V} \int_V U_r(r) dV = \frac{1}{V} \int_{r_k}^{r_{k+1}} U_r(r) A_r(r) dr \\ \langle U_r \rangle &= \frac{U_r^m A_r^m}{V} (r_{k+1} - r_k) \end{aligned}$$

or

$$U_r^m = \frac{V}{A_r^m (r_{k+1} - r_k)} \langle U_r \rangle \quad (3.3.18)$$

### 3.4 Energy Exchange

As done for the momentum exchange, here again we divide the energy interactions into two parts, the energy exchange between phases and the heat exchange between fluid and fuel pins and structural materials. For the latter, we identify three subdivisions, the fuel pin heat conduction, the convective heat transfer between the fuel pin walls and the fluid, and finally the fuel assembly structure model.

#### 3.4.1 Fuel Pin Heat Conduction

A single rod in each volume (node) is selected to represent the fuel pin heat conduction, which is assumed to be thermally equivalent to any other rod in that cell. Axial heat conduction is neglected, so that the radial heat conduction equation is:

$$\rho C \frac{\partial T}{\partial t} - \frac{1}{r} \frac{\partial}{\partial r} (rK \frac{\partial T}{\partial r}) = q''' \quad (3.4.1)$$

For the time being all material properties are assumed to be known quantities and we proceed to analyze the solution of equation 3.4.1. Later in section 3.4.2 these material properties are discussed.

The fuel and the clad are now divided into mesh cells, the number of these cells being an input parameter. We only impose that all mesh spacings in the same region, whether fuel or clad, be of the same size, but mesh spacings may be different in different regions. One

cell is assumed for the gap. Fuel temperatures are located at the boundaries of mesh cells, represented by the subscript  $k$ . Fuel pin properties are evaluated in the center of mesh cells, and are represented with the subscript  $k+\frac{1}{2}$ . If we integrate equation 3.4.1 between the center of two adjacent cells we get:

$$\int_{r_{k-\frac{1}{2}}}^{r_{k+\frac{1}{2}}} \left[ r \rho C_p \frac{\partial T}{\partial t} - \frac{\partial}{\partial r} \left( r K \frac{\partial T}{\partial r} \right) \right] dr = \int_{r_{k-\frac{1}{2}}}^{r_{k+\frac{1}{2}}} q''' r dr \quad (3.4.2)$$

Using the approximation:

$$\langle \rho C_p \rangle_k = \frac{r_{k+\frac{1}{2}}^2 - r_k^2}{2} (\rho C_p)_{k+\frac{1}{2}} + \frac{r_k^2 - r_{k-\frac{1}{2}}^2}{2} (\rho C_p)_{k-\frac{1}{2}} \quad (3.4.3)$$

and

$$\int_{r_{k-\frac{1}{2}}}^{r_{k+\frac{1}{2}}} r \rho C_p \frac{\partial T}{\partial t} dr = \langle \rho C_p \rangle_k \frac{\partial T_k}{\partial t} \quad (3.4.4)$$

Also in equation 3.4.2 we have:

$$\begin{aligned} \int_{r_{k-\frac{1}{2}}}^{r_{k+\frac{1}{2}}} \left[ \frac{\partial}{\partial r} r K \frac{\partial T}{\partial r} \right] dr &= \left[ r K \frac{\partial T}{\partial r} \right]_{r_{k-\frac{1}{2}}}^{r_{k+\frac{1}{2}}} \\ &= (rK)_{k+\frac{1}{2}} \frac{T_{k+1} - T_k}{\Delta r_{k+\frac{1}{2}}} - (rK)_{k-\frac{1}{2}} \frac{T_k - T_{k-1}}{\Delta r_{k-\frac{1}{2}}} \end{aligned} \quad (3.4.5)$$

Finally, the right hand side of equation 3.4.2 becomes:

$$\int_{r_{k-\frac{1}{2}}}^{r_{k+\frac{1}{2}}} q''' r dr = \frac{r_{k+\frac{1}{2}}^2 - r_k^2}{2} q'''_{k+\frac{1}{2}} + \frac{r_k^2 - r_{k-\frac{1}{2}}^2}{2} q'''_{k-\frac{1}{2}} \quad (3.4.6)$$

and the difference equation corresponding to equation 3.4.2 becomes:

$$\begin{aligned} & \langle \rho C_p \rangle_k^n \left( \frac{T_k^{n+1} - T_k^n}{\Delta t} \right) - \left( \frac{rK}{\Delta r} \right)_{k+\frac{1}{2}}^n (T_{k+1}^{n+1} - T_k^{n+1}) + \left( \frac{rK}{\Delta r} \right)_{k-\frac{1}{2}}^n (T_k^{n+1} - T_{k-1}^{n+1}) \\ & = \left[ \frac{r_{k+\frac{1}{2}}^2 - r_k^2}{2} q_{k+\frac{1}{2}}''' + \frac{r_k^2 - r_{k-\frac{1}{2}}^2}{2} q_{k-\frac{1}{2}}''' \right]^n \end{aligned} \quad (3.4.7)$$

There are four locations where equation 3.4.7 must be modified to accommodate boundary conditions. For the center of the fuel pin, equation 3.4.1 is integrated from  $r = r_{\frac{1}{2}} = 0$  to  $r = r_{1\frac{1}{2}}$ , and the resulting equation is:

$$\langle \rho C_p \rangle_1 \left( \frac{T_1^{n+1} - T_1^n}{\Delta t} \right) - \left( \frac{rK}{\Delta r} \right)_{1\frac{1}{2}}^n (T_2^{n+1} - T_1^{n+1}) = \frac{r_{1\frac{1}{2}}^2}{2} q_{1\frac{1}{2}}''' \quad (3.4.8)$$

with

$$\langle \rho C_p \rangle_1 = \frac{r_{1\frac{1}{2}}^2}{2} (\rho C_p)_{1\frac{1}{2}} \quad (3.4.9)$$

For the clad outside surface we obtain the difference equation by integrating equation 3.4.1 from  $r = r_{N-\frac{1}{2}}$  to  $r = r_N =$  outside fuel pin radius, and introducing the clad surface heat flux  $q''$ . We obtain the equation:

$$\begin{aligned} & \langle \rho C_p \rangle_N \left( \frac{T_N^{n+1} - T_N^n}{\Delta t} \right) + \left( \frac{rK}{\Delta r} \right)_{N-\frac{1}{2}}^n (T_N^{n+1} - T_{N-1}^{n+1}) + q'' r_N \\ & = \frac{r_N^2 - r_{N-\frac{1}{2}}^2}{2} q_{N-\frac{1}{2}}''' \end{aligned} \quad (3.4.10)$$

with

$$\langle \rho C_p \rangle_N = \frac{r_N^2 - r_{N-1/2}^2}{2} (\rho C_p)_N^n \quad (3.4.11)$$

The general expression for the heat flux  $q''$  (later in section 3.5 the correlations for the heat flux will be discussed in detail) is:

$$q'' = h_\ell^n (T_w^{n+1} - T_\ell^{n+1}) + h_v^n (T_w^{n+1} - T_v^{n+1}) + h_{NB}^n (T_w^{n+1} - T_s^{n+1}) \quad (3.4.12)$$

where

$T_w = T_N$  = outside clad temperature

$T_\ell, T_v, T_s$  = liquid, vapor and saturation temperatures

$h_\ell, h_v, h_{NB}$  = heat transfer coefficients

Finally for the two equations involving gap properties the term  $\frac{k}{\Delta r}$  is replaced by  $h_{GAP}$ , the gap conductance.

Returning to equation 3.4.7 note that a fully implicit differentiating scheme was used in this equation. This difference equation can be shown to be unconditionally stable. In this way we ensure that a time step determined by the fluid equations stability does not cause any stability problem for the heat conduction problem.

Equation 3.4.7 couples the temperature at a cell  $k$  with its neighbors  $k+1$  and  $k-1$ , thus the temperature for all cells must be solved simultaneously. We incorporate an efficient technique to save computational time for this solution. This technique, proposed by Reed and Stewart / 21 / is a modification of the tridiagonal matrix inversion.

In matrix form, the set of equations 3.4.7 become:

$$\begin{bmatrix} a_{11} & a_{12} & 0 & \dots & 0 \\ a_{21} & a_{22} & a_{31} & 0 & \dots & 0 \\ 0 & a_{31} & a_{33} & a_{34} & \dots & 0 \\ \vdots & 0 & & & & \vdots \\ 0 & \vdots & & & & \vdots \\ 0 & 0 & & a_{N-1,N} & a_{NN} & \end{bmatrix} \times \begin{bmatrix} T_1^{n+1} \\ T_2^{n+1} \\ \vdots \\ \vdots \\ T_T^{n+1} \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ \vdots \\ f_N \end{bmatrix} \quad (3.4.13)$$

where the coefficients a's and f's depend only on the fuel geometry, the power density and material properties. All these quantities are evaluated at the old time step, therefore they do not change during the new time step iterations.

The usual tridiagonal solution for this equation replaces the matrix of coefficients a's (which we will call by the capital letter A) by a product:

$$A = C \times B$$

with

$$C = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 & 0 \\ c_{21} & 1 & 0 & \dots & 0 & 0 \\ 0 & c_{31} & 1 & 0 & \dots & \\ \vdots & 0 & & & & \\ \vdots & 0 & & & & \\ \vdots & 0 & & & & \\ 0 & 0 & \dots & c_{N,N-1} & & 1 \end{bmatrix}$$

and

$$B = \begin{bmatrix} b_{11} & b_{12} & 0 & 0 & \dots & 0 \\ 0 & b_{22} & b_{23} & 0 & \dots & 0 \\ \vdots & 0 & b_{33} & & & \\ \vdots & \vdots & \vdots & & & \\ \vdots & \vdots & \vdots & b_{N-1,N-1} & b_{N-1,N} & \\ 0 & 0 & 0 & 0 & & b_{NN} \end{bmatrix}$$

In order for this factorization to be true, we must require:

$$b_{11} = a_{11}$$

$$b_{12} = a_{12}$$

$$\vdots$$

$$c_{p,p-1} = a_{p,p-1} / b_{p-1,p-1}$$

$$b_{pp} = a_{pp} - c_{p,p-1} \cdot b_{p-1,p}$$

$$b_{p,p+1} = a_{p,p+1}$$

Now define a vector X such that:

$$F = C \times X$$

where F is the vector of coefficients f's in equation 3.4.13. This factorization requires:

$$x_1 = f_1$$

$$\vdots$$

$$x_p = f_p - c_{p,p-1} x_{p-1}$$

In this way, equation 3.4.13 becomes:

$$B \times T = X \tag{3.4.14}$$

where B is an upper triangular matrix, and once we have gotten the value of  $T_{N,N}^{n+1}$ , all other temperatures are easily obtained by backward substitution.

The important characteristic of all these operations is that they are performed only on explicit terms. Thus this procedure must be carried only once, at the beginning of the new time step.

The last line of equation 3.4.14 is the one used to determine the clad outside wall temperature. This is the only equation which involves implicit temperatures in the right hand side. If we recall equations 3.4.10 and 3.4.12, we can write this equation for the wall temperature, isolating the implicit terms:

$$b_{NN} T_{NN}^{n+1} = f_N^n + h_{\ell}^n T_{\ell}^{n+1} + h_{v}^n T_v^{n+1} + h_{NB}^n T_s^{n+1} \quad (3.4.15)$$

Then, after any Newton iteration  $k$  we use equation 3.4.15 to calculate the new wall temperature  $T_{NN}$ , without the need for calculating all the other temperatures, and only after the Newton iteration has converged we return to equation (3.4.14) to calculate the fuel temperatures.

### 3.4.2 Fuel Pin Material Properties

For the clad heat capacity and thermal conductivity the properties of stainless steel are incorporated in the code. From reference / 53 / the following expressions were selected:

$$(\rho C_p)_{\text{clad}} = a_0 + a_1 T + a_2 T^2 \quad (3.4.16)$$

$$K_{\text{clad}} = b_0 + b_1 T \quad (3.4.17)$$

with

$$a_0 = 4.28 \cdot 10^6$$

$$a_1 = 3.75 \cdot 10^2$$

$$a_2 = 7.45 \cdot 10^3$$

$$b_0 = 16.27$$

$$b_1 = 1.204 \cdot 10^{-2}$$

Two axially different zones are implemented in the code to represent the fuel itself. One with the properties of Plutonium-Uranium oxides to represent the active core region, and the second one to simulate the fission gas upper plenum.

From reference / 21 / the following expressions were selected to represent the fuel region:

$$(\rho C_p)_{\text{FUEL}} = (a_0 + a_1 T + a_2 T^2 + a_3 T^3) (1 + 0.045 \theta_{\text{pu}}) \cdot \theta_d \quad (3.4.18)$$

$$K_{\text{FUEL}} = (b_0 + b_1 T + b_2 T^2) (1 - (1 - \theta_d) \cdot X) \quad (3.4.19)$$

with

$$X = 2.74 - 5.8 \times 10^{-4} T$$

$\theta_{\text{pu}}$  = fraction of  $\text{PuO}_2$  in the mixed oxide fuel

$\theta_d$  = fraction of theoretical density

$$a_0 = 1.81 \times 10^6$$

$$a_1 = 3.72 \times 10^3$$

$$a_2 = -2.51$$

$$a_3 = 6.59 \times 10^{-4}$$

$$b_0 = 10.8$$

$$b_1 = -8.84 \times 10^{-3}$$

$$b_2 = 2.25 \times 10^{-6}$$

The fission gas plenum is simulated with a zero heat capacity.

The gap heat capacity is also assumed to be zero. For its conductance the following expression was incorporated, from reference / 5 /:

$$h_{\text{Gap}} = h_c + h_r \quad (3.4.20)$$

with

$$h_r = (T_f^2 + T_c^2)(T_f + T_c)1.70 \times 10^{-8} \quad (3.4.21)$$

and

$$h_c = \left[ \frac{dg + 1.32 \times 10^{-4}}{C_g} + 0.61 \times 10^{-4} \right]^{-1} + 1.8 \times 10^3 \quad (3.4.22)$$

with

$$C_g = 15. \times 2^{\text{dil}}$$

where

dg = gap thickness

dil = fraction of helium in gap composition.

$T_f$  and  $T_c$  = outside fuel pellet and inside clad temperature

### 3.4.3 Convective Heat Transfer Coefficient

It has been mentioned in the previous section the expression for the heat transfer between the fluid and the fuel pins as:

$$q'' = h_{\ell}^n (T_w^{n+1} - T_{\ell}^{n+1}) + h_v^n (T_w^{n+1} - T_v^{n+1}) + h_{NB}^n (T_w^{n+1} - T_s^{n+1})$$

This expression is an extension of the correlations proposed by Chen / 23 / for non-metallic coolants. Although this correlation

has not been verified by comparison with experimental data, we anticipate good agreement with experiments, based on the great success the assumptions of micro and macro-convective heat transfer mechanism has encountered for non-metallic coolants. Nevertheless, only an extensive experimental program could give a definitive confirmation of this model.

The conditions for validity of the correlation are stable, vertical, axial convective flow of saturated liquid, with wetted heat transfer surface. These conditions are in general encountered in convective boiling in annular or mist-annular flow. The model is based on the postulate that there are two mechanisms contributing to the total heat transfer, and these mechanisms interact with each other. The macro-convective mechanism is associated with overall flow heat transfer, and the micro-convective mechanism is associated with bubble growth in the annular liquid film.

The expression for the micro-convective heat transfer coefficient is:

$$h_{NB} = 0.00122 \frac{K_l^{.79} C_{pl}^{.45} \rho_l^{.49} \Delta p^{.75} S_f}{\sigma^{.5} \eta_l^{.29}} \left( \frac{\Delta T}{h_{lg} \rho_v} \right)^{.24} \quad (3.4.23)$$

where  $S_f$  the suppression factor defined as:

$$S_f = \left( \frac{\Delta T_e}{\Delta T} \right)^{.99}$$

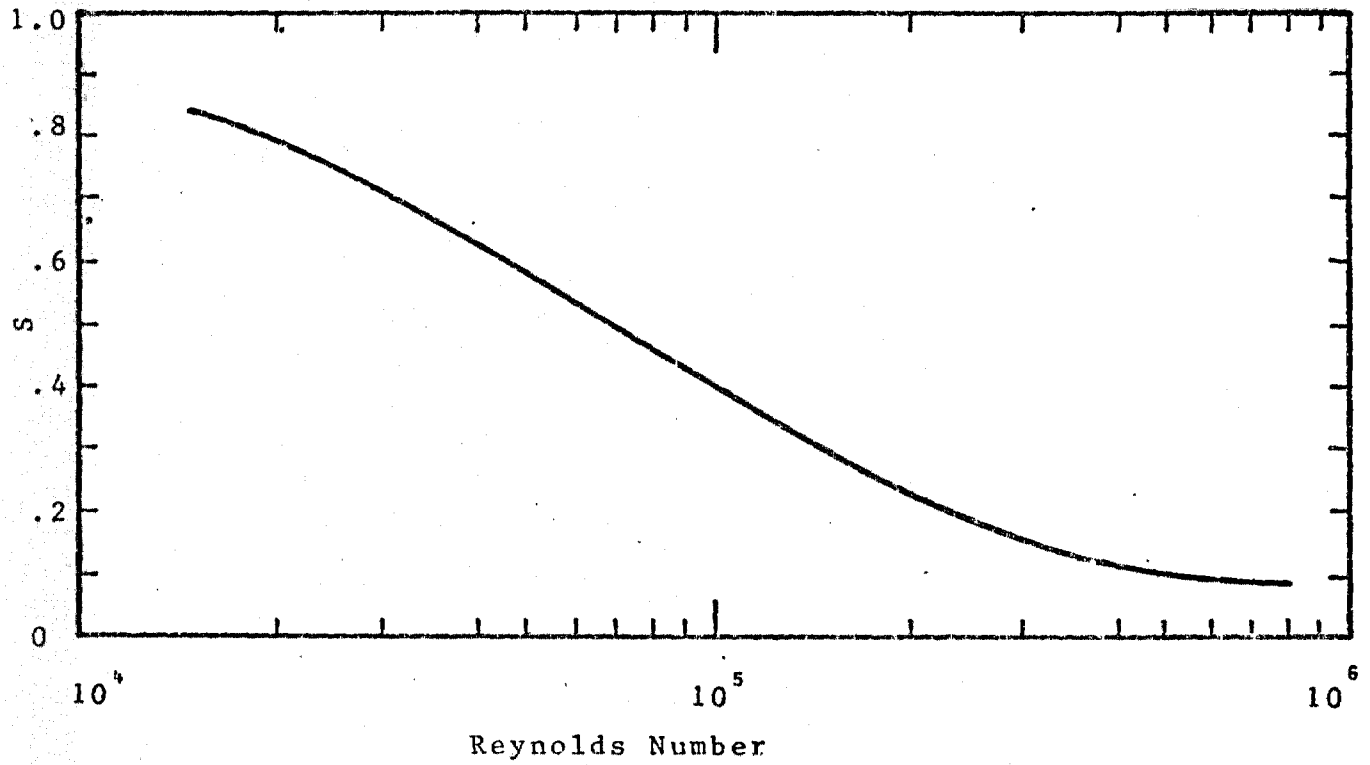


Figure 3.4 Supression Factor vs. Reynolds Number

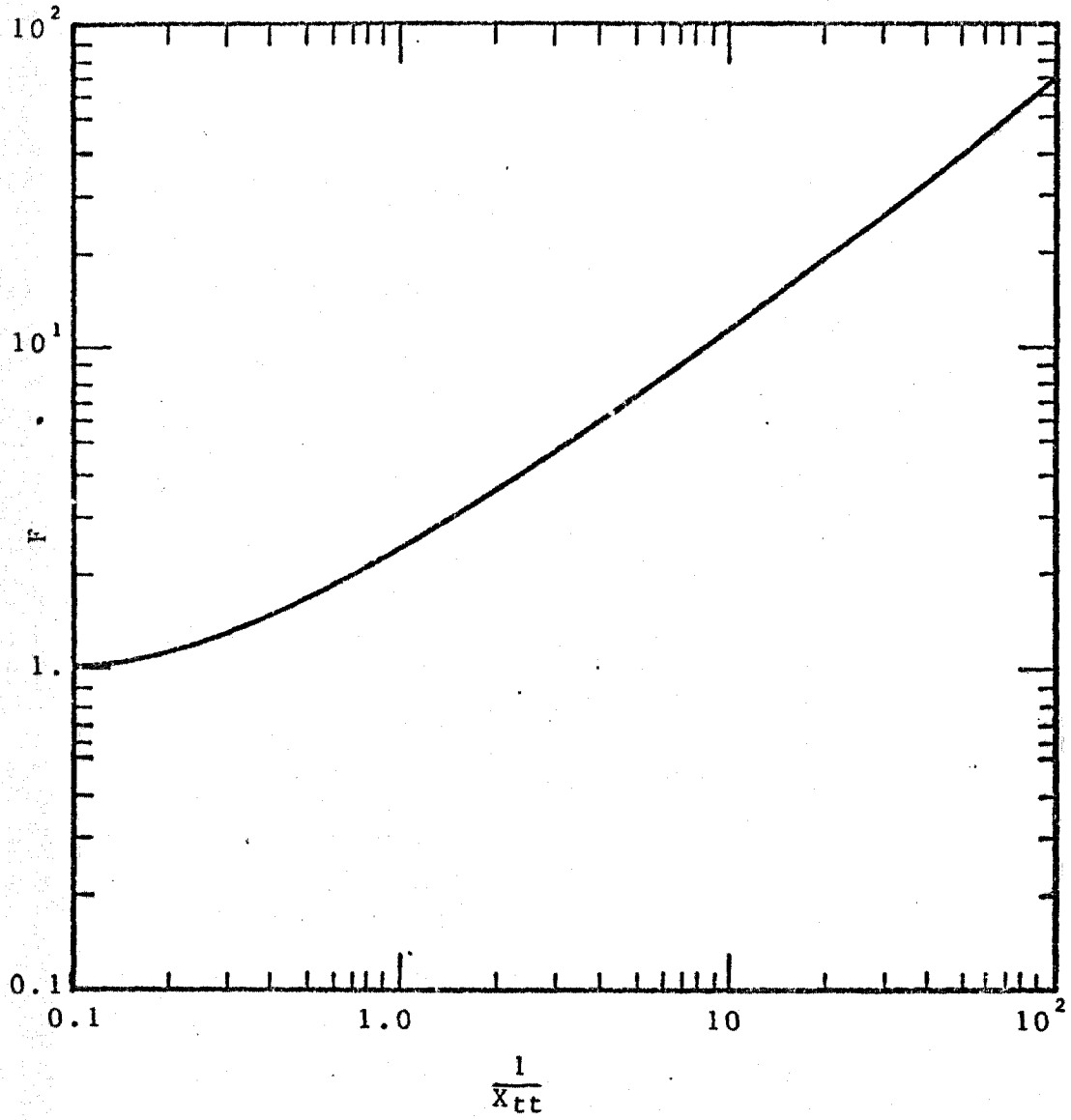


Figure 3.5 The Reynolds Number Factor

with

$\Delta T_e$  = effective superheat for bubble growth in annular liquid

$\Delta T$  = difference between wall temperature and saturation temperature

$\Delta p$  = difference between pressure at the wall and liquid temperature

Figure 3.4 shows the dependence of  $S$  on the Reynolds number.

From that figure, we extract the correlation for  $S_f$ :

$$S_f = \begin{cases} (1 + .12 \text{Re}_{TP}^{1.14})^{-1} & \text{Re}_{TP} \leq 32.5 \\ (1 + .42 \text{Re}_{TP}^{.78})^{-1} & 32.5 < \text{Re}_{TP} \leq 70 \\ 0.1 & \text{Re}_{TP} > 70 \end{cases} \quad (3.4.24)$$

and the two phase Reynolds number is defined as:

$$\text{Re}_{TP} = F^{1.25} \frac{(1-\alpha)\rho_\ell U_\ell D_H}{\eta_\ell} \quad (3.4.25)$$

where  $F$  is the Reynolds number factor, shown in Figure 3.5. The analytical expression from this figure is:

$$F = 2.35 \left( .213 + \frac{1}{X_{tt}} \right)^{.736} \quad X_{tt} < 10. \quad (3.4.26)$$

$$F = 1.0 \quad X_{tt} \geq 10.$$

and

$X_{tt}$  is the Martinelli parameter:

$$X_{tt} = \left( \frac{1-X}{X} \right)^9 \left( \frac{\rho_v}{\rho_\ell} \right)^5 \left( \frac{\eta_\ell}{\eta_v} \right)^1 \quad (3.4.27)$$

For the macroscopic heat transfer coefficient, Manahan / 11 / proposed a modified form of the Lyon-Martinelli equation:

$$h_{\ell} = F^{.375} h_{\ell sp} \quad (3.4.28)$$

where  $F$  is the same Reynolds number factor used for the microscopic heat transfer coefficient and  $h_{\ell sp}$  is the liquid single phase heat transfer coefficient. The CHAD correlation was used for this single phase heat transfer coefficient:

$$h_{\ell sp} = N_u \frac{k_{\ell}}{D_H}$$

with

$$N_u = \begin{cases} 4.5R & Pe \leq 150 \\ R Pe^{.3} & Pe > 150 \end{cases} \quad (3.4.29)$$

with

$$R = -16.15 + 24.96(P/d) - 8.55(P/d)^2 \quad (3.4.30)$$

and  $Pe$  is the Peclet number =  $RePr$

Finally, for the vapor single phase heat transfer coefficient the Dittus-Boelter correlation is used

$$h_v = 0.023 Re_v^{.8} Pr_v^{.4} \frac{k_v}{D_H} \quad (3.4.31)$$

#### 3.4.4 Fuel Assembly Structure Models

For the structural materials in fuel assembly two elements are considered: the wire wrap and the fuel assembly hex can.

The wire wrap is modeled by assuming it has the same temperature as the outside clad surface. In this way, a thin layer of stainless steel, corresponding to the wire wrap heat capacity is added to the heat capacity of the last cell of the clad in the fuel pin model.

Presently the model considers the fuel assembly hex can as an adiabatic boundary condition, modeling only the effect of its heat capacity in transients, although the model was designed to accommodate changes which would consider a heat sink outside the hex can.

The equation used to model the hex can heat capacity is:

$$\begin{aligned} (\rho C_p)_c \left( \frac{T_c^{n+1} - T_c^n}{\Delta t} \right) + h_l^n (T_c^{n+1} - T_l^{n+1}) + h_v^n (T_c^{n+1} - T_v^{n+1}) \\ + h_{NB}^n (T_c^{n+1} - T_s^{n+1}) = 0 \end{aligned} \quad (3.4.32)$$

where

$h_l$ ,  $h_v$ ,  $h_{NB}$  are the heat transfer coefficients discussed in the previous section;

$T_c$ ,  $T_l$ ,  $T_v$ ,  $T_s$  are the hex can, liquid, vapor and saturation temperatures.

#### 3.4.5 Interphase Heat Exchange

Of all models presented in this section, the interphase heat exchange is the least developed. Whereas other constitutive equations,

like those for the momentum exchange or fuel pin heat transfer, are applied to all models of two phase flow, the interphase heat exchange constitutive equation has its only application in the two-fluid model, which has been given attention only in recent years.

Thus, because of the lack of experimental data, we had to rely on a purely theoretical basis to produce a correlation for this exchange term.

Two mechanisms can be identified in which heat is transferred between phases. One represents the enthalpy transported by the mass exchange between phases, and the other accounts for the convective heat transfer. Then, we propose the following expression for this exchange term:

$$q_{lv} = S_e^{n+1} h_{vs}^{n+1} - S_c^{n+1} h_{ls}^{n+1} + HA(T_l^{n+1} - T_v^{n+1}) \quad (3.4.33)$$

where

$S_e$  = evaporation rate

$S_c$  = condensation rate

$h_{vs}$  = enthalpy for the saturated vapor

$h_{ls}$  = enthalpy for the saturated liquid

$H$  = overall heat transfer coefficient

$A$  = interfacial area

For the interfacial area, the same model developed for the mass exchange rate is used, and the expressions to evaluate this interfacial area can be found in equations 3.2.7 to 3.2.10.

In general, the overall heat transfer coefficient  $H$  can be written as:

$$H = \frac{Nu K_{\ell}}{D_H}$$

where

$K_{\ell}$  = liquid thermal conductivity

$D_H$  = hydraulic diameter

$Nu$  = Nusselt number

A great deal of uncertainty is embodied in the Nusselt number used in this model, which cannot be resolved without a consistent set of experimental data on the heat exchange between phases.

Therefore, we tentatively recommend the value  $Nu = 100$ .

## CHAPTER 4

Experimental Tests Simulation

The models and methods presented in the two previous chapters were assembled into a computer program named NATOF-2D.

In order to evaluate the model results, as well as to test the program capabilities, three tests were simulated with NATOF-2D.

The first experiment simulated was the SLSF P3A test which was used to evaluate the performance of the constitutive equations and to determine the sensibility of the code to these equations.

Next the W-1 experiment was simulated, a test which has been completed recently. Finally a steady-state experiment, the GR19 was analyzed.

4.1 The P3A Experiment

The Sodium Loop Safety Facility P3A Experiment was an in-pile test performed in the Engineering Test Reactor in the period July 16, 1977 to September 11, 1977. The experiment was made with a 37-pin bundle simulating an FFTR unprotected loss of flow accident. The test bundle

was irradiated for 26 full power days prior to the final experiment. The subassembly power was 1240 KW with a mass flow rate of 9.2 lbm/sec (4.173 Kg/sec).

Coolant boiling was detected at 8.8 seconds into the test. Inlet flow reversal occurred at 10 seconds, followed by inlet flow and temperature oscillations. Non-condensable gas passing through the bundle exit flow meter at 10.8 seconds was indicative of clad failure. Table 4.1 summarizes the design and steady-state operational data for the test.

Steady-state measurements made prior to the test indicated the existence of a discrepancy between the actual thermocouple readings and their expected values. A temperature gradient in the radial direction was observed which did not agree with the predicted values. This discrepancy was attributed to a non-uniform radial power distribution in the bundle, due to a non-uniform neutronic flux across the test bundle. Therefore a radial power distribution was assumed in the numerical simulation of the test. Table 4.2 shows the assumed radial power profile [39].

An inlet pressure decay was imposed to simulate the loss of flow transient. The expression used was:

$$P_{1u}(\text{bar}) = 1.7187 + 7.4380 \exp(-.21t)$$

Table 4.3 shows the timing of events for the NATOF-2D predictions, along with the experimental results and the values obtained with SOBOIL code [10].

Following a series of figures showing the results obtained with NATOF-2D is presented.

Figure 4.2 shows the inlet mass flow rate as a function of time. The flow oscillations observed in the test were also predicted by NATOF-2D. Figure 4.3 shows the curve for the mass flow rate obtained from the experimental data.

Figures 4.4 and 4.5 show the temperature evolution at the top of the heated zone for the central and the edge channels respectively. Here again the oscillations after the flow reversal encountered in the experiment are also observed.

Figures 4.6 and 4.7 show the axial temperature profile at the central channel and the radial temperature profile at the top of the heated zone for different times. In this last figure one can observe an increase in the radial temperature gradient up to the time 9.0 seconds. This is attributed to the effect of the duct wall heat capacity. After 9.0 seconds the boiling in the central channels creates a strong radial flow, with the effect of reducing again the radial temperature gradient.

Finally figure 4.8 shows the void fraction maps for the three radial channels.

From the numerical method point of view, the most encouraging result was the ability of the model to represent the transient beyond the point of flow reversal without numerical instability, a flow condition which has challenged the sodium two phase flow modeling for years.

Table 4.1: SLSF-P3A Test Bundle Data

	<u>Geometry</u>	(British Units)
Number of Pins	37	.1945 in
Fuel Pellet OD (m)	$4.94 \times 10^{-3}$	
Clad OD (m)	$5.842 \times 10^{-3}$	.230 in
Clad ID (m)	$5.080 \times 10^{-3}$	.200 in
Wire Wrap OD (m)		
inner pins	$1.422 \times 10^{-3}$	.056 in
outer pins	$7.11 \times 10^{-4}$	.028 in
Flat to Flat (m)	$4.501 \times 10^{-2}$	1.772 in
Duct Wall Thickness (m)	$3.048 \times 10^{-3}$	.12 in
Length of Fuel		36.0 in
Inlet to Bottom of Fuel (m)	$1.857 \times 10^{-1}$	7.31 in
Top of Fuel to End Cap (m)	5.334	210.0 in
Wire Wrap Lead (m)	$3.048 \times 10^{-1}$	12.0 in
Fill Gas	Helium, 1 atm at 20°C with xenon tag gas	
Fuel	Uranium-Plutonium mixed oxide, Pu 25% of total mass	

Table 4.1 continued

Thermo-Hydraulics

Inlet Temperature	(°C)	422	792°F
Outlet Temperature at Steady State	(°C)	658	1216°F
Bundle Power	(kw)	1240	
Test Bundle Flow	(Kg/sec)	4.173	9.20 lbm/sec
Pressure at Top of Heated Zone	(atm)	4.27	62.7 psia
Cover Gas Pressure	(atm)	.957	14.1 psia
Net Pump Head	(atm)	7.619	112 psi

Numerics of Simulation

Number of Axial Mesh Cells	10
Number of Radial Mesh Cells	3

Table 4.2

Assumed Non-Uniform Radial Power  
Distribution in P3A Test Bundle

<u>Pin Number*</u>	<u>Power Factor</u>
1	.90
2	.95
3	1.07
4	1.156

---

\*see figure 4.1 for Pin Number location

Table 4.3

Event Sequence Times (Seconds) of the P3A Experiment

	<u>NATOF-2D</u>	<u>SOBOIL</u>	<u>Experiment</u>
Boiling Inception	8.9	8.9	8.8
Inlet Flow Reversal	10.08	9.9	10.15

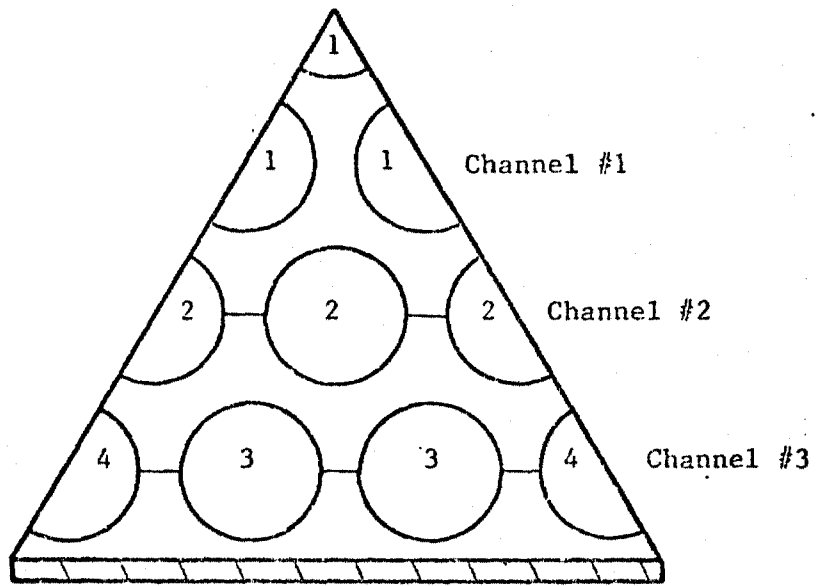


Figure 4.1

Pin Number Location

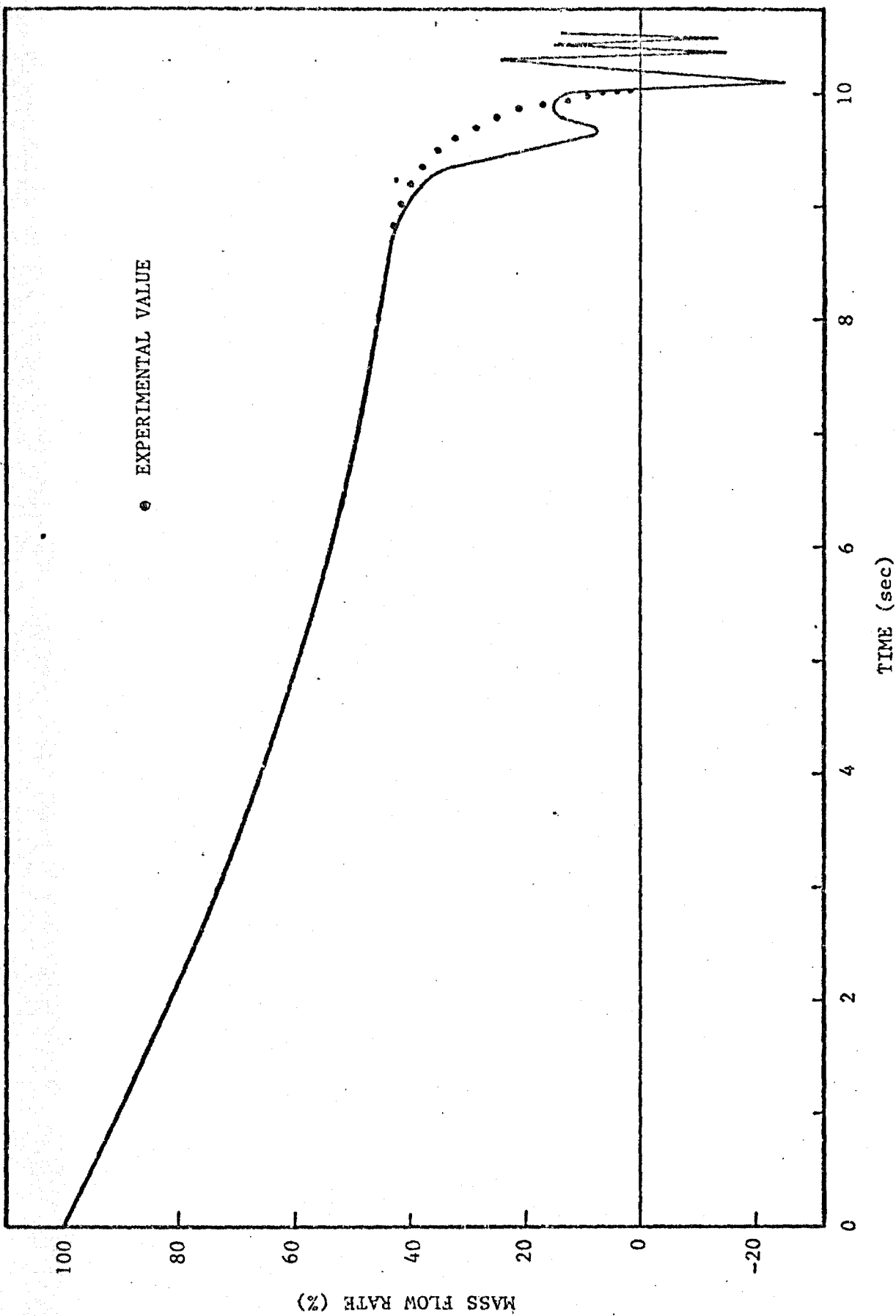


Figure 4.2: P3A: Mass Flow Rate Vs. Time

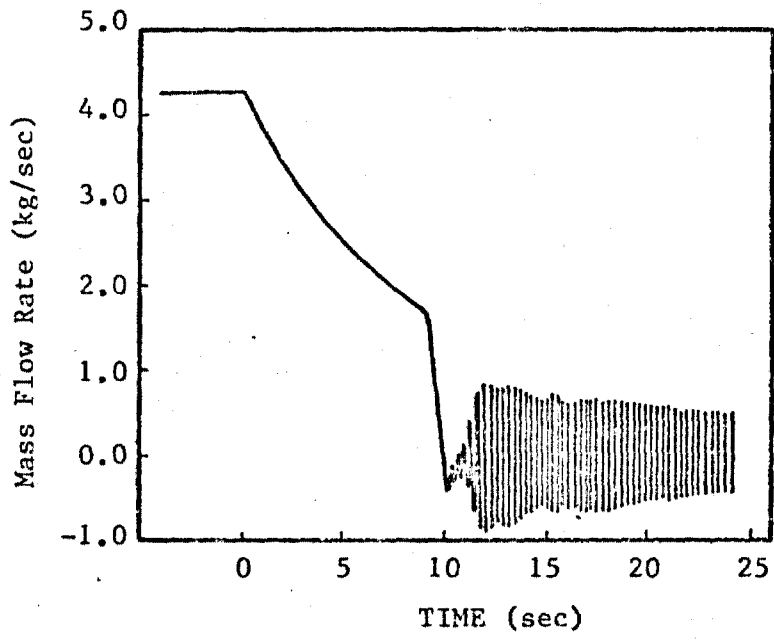


Figure 4.3

Experimental Inlet Mass Flow Rate

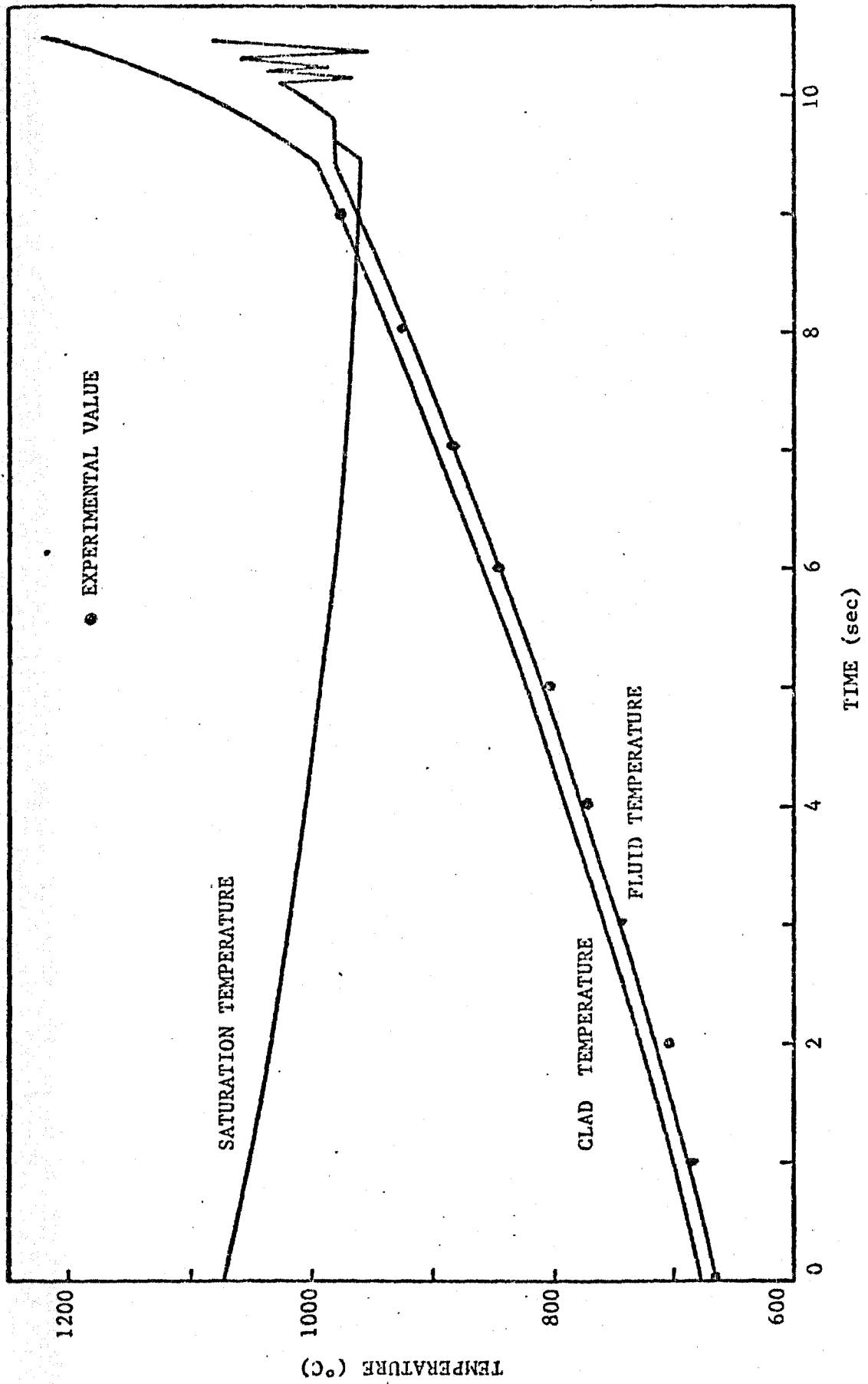


Figure 4.4: P3A - Temperature Vs. Time: Central Channel

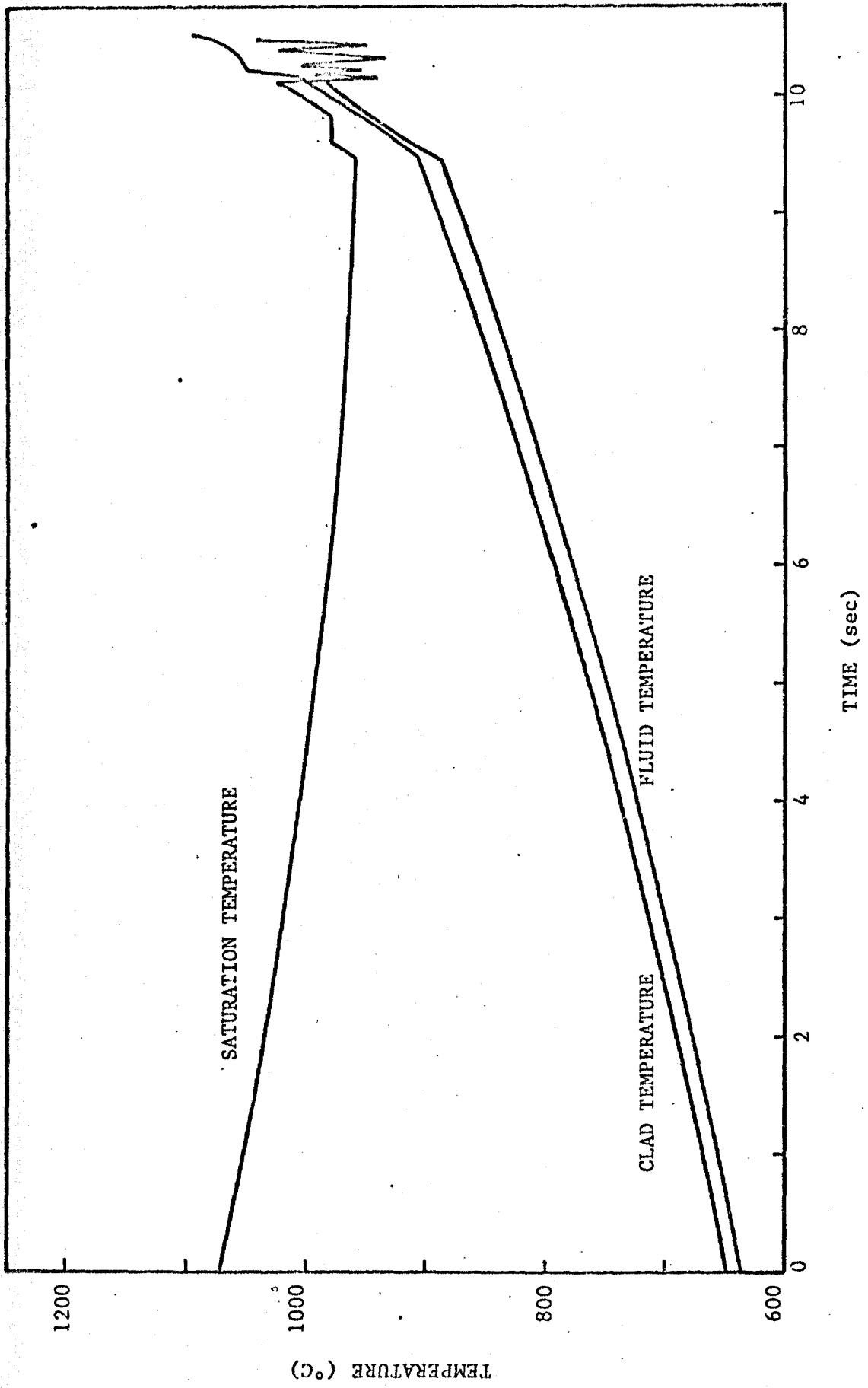


Figure 4.5: P3A — Temperature Vs. Time Edge Channel

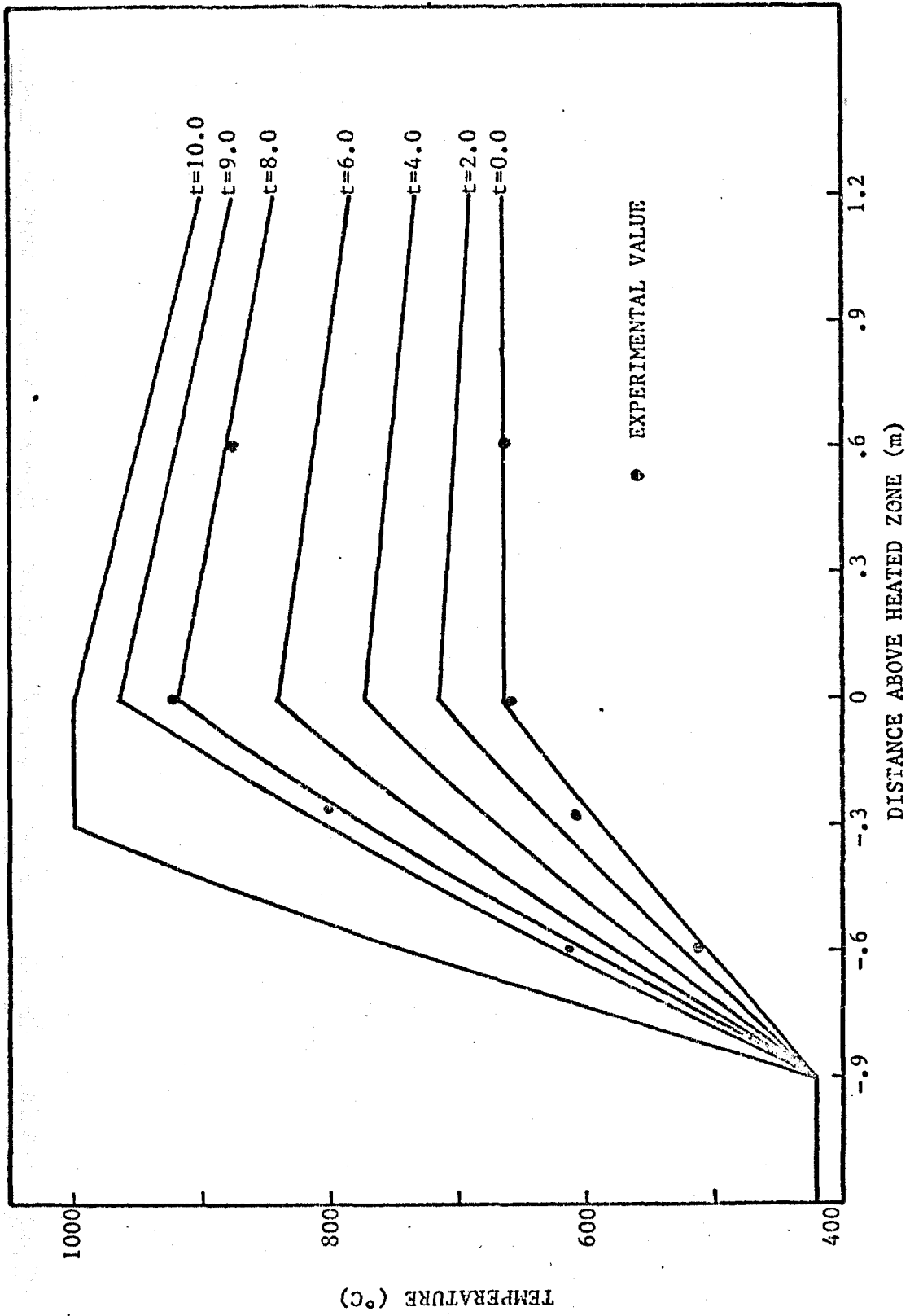


Figure 4.6: P3A: Axial Temperature Profile

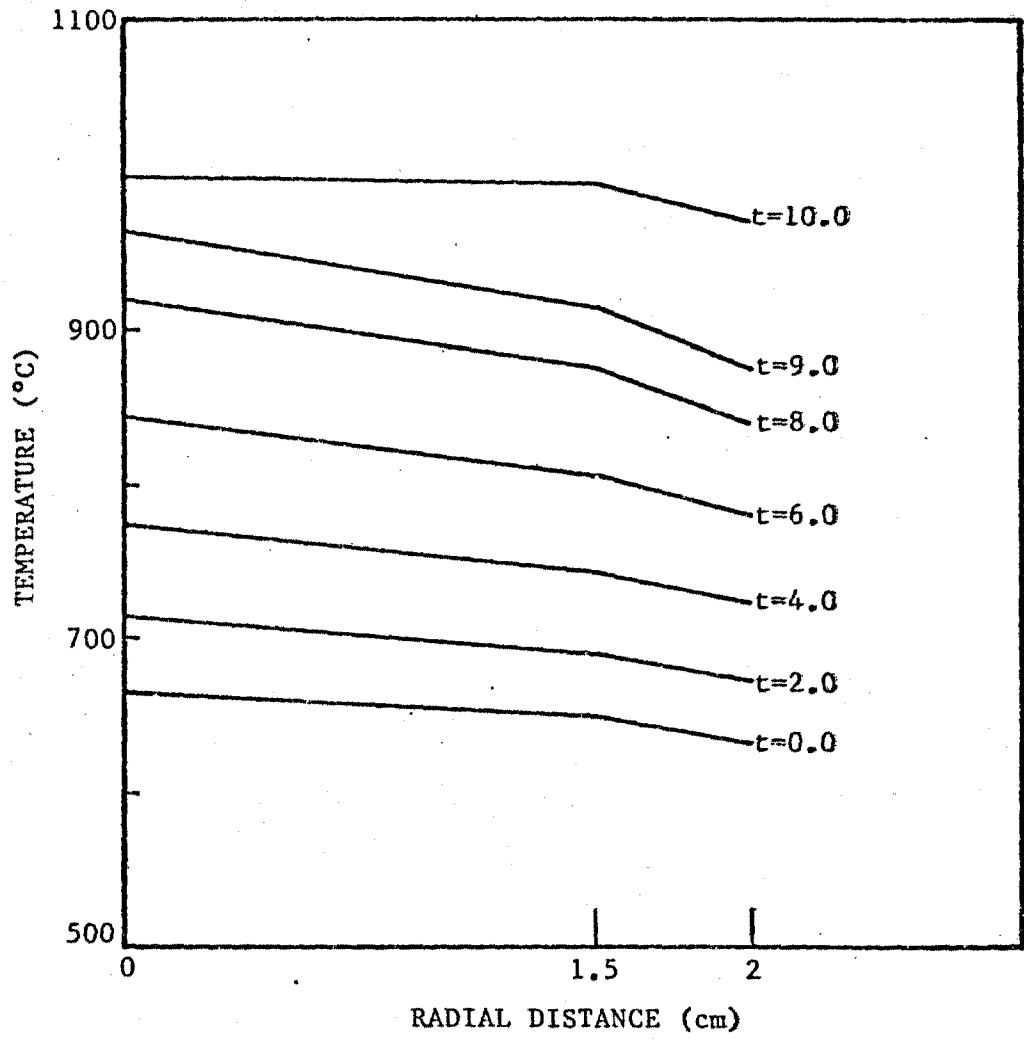


Figure 4.7

P3A: Radial Temperature Profile

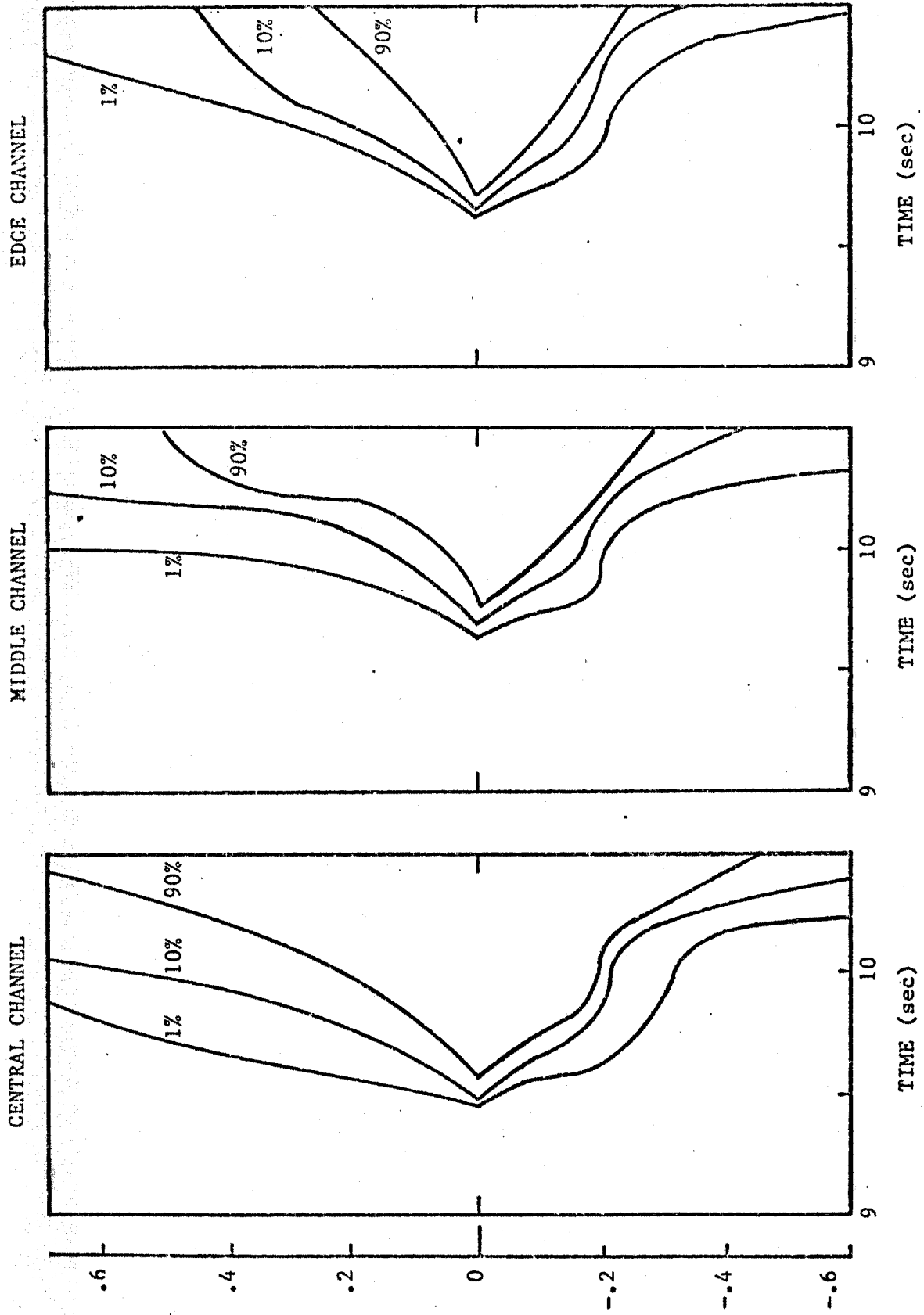


Figure 4.8: Void Fraction Maps for the P3A Experiments

#### 4.2 One Dimensional Analysis of the P3A Experiment

In order to determine the importance of the two dimensional characteristic of NATOF-2D, a comparison of the results presented in the previous section was made with a one dimensional analysis of the same test.

NATOF-2D was modified to allow a one dimensional representation of the fuel assembly, and the P3A test was reanalyzed under the same conditions.

Figure 4.9 shows the inlet mass flow rate as a function of time, figure 4.10 the temperature evolution at the top of the heated zone and figure 4.11 the axial temperature profile for this one dimensional analysis. Finally figure 4.12 shows the inlet mass flow rate for both one and two dimensional representations in the boiling period.

These figures show two interesting results. The onset of boiling occurred at 9.2 seconds for the one dimensional analysis, a delay of 0.3 seconds with respect to the two dimensional case. This can be explained by the fact that in the 1D case, both the central and the edge channels are represented by a single average temperature which is less than the maximum fluid temperature encountered in the central channel, and it takes longer for the average temperature to reach the saturation conditions.

The second result which differed from the two dimensional representation was the time of flow reversal, which occurred at 9.8 seconds, 0.28 seconds before the 2D result. This is explained by the fact that while voiding is taking place in the central channels, the edge channel which is relatively colder maintains a substantial liquid flow for a longer time, thus providing a path for an upwards liquid flow. This effect is lost with the one dimensional representation.

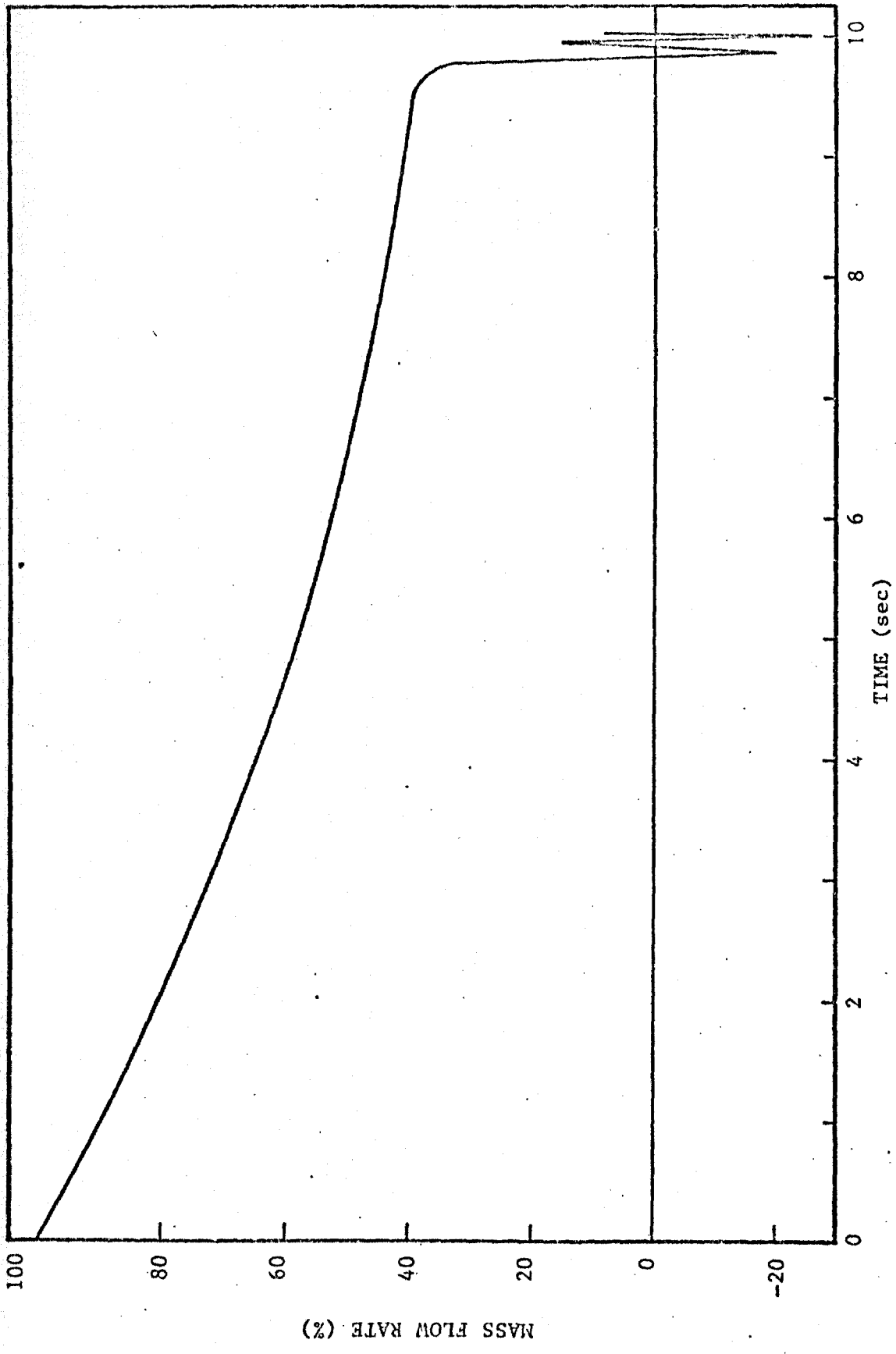


Figure 4.9: P3A — ID: Mass Flow Rate Vs. Time

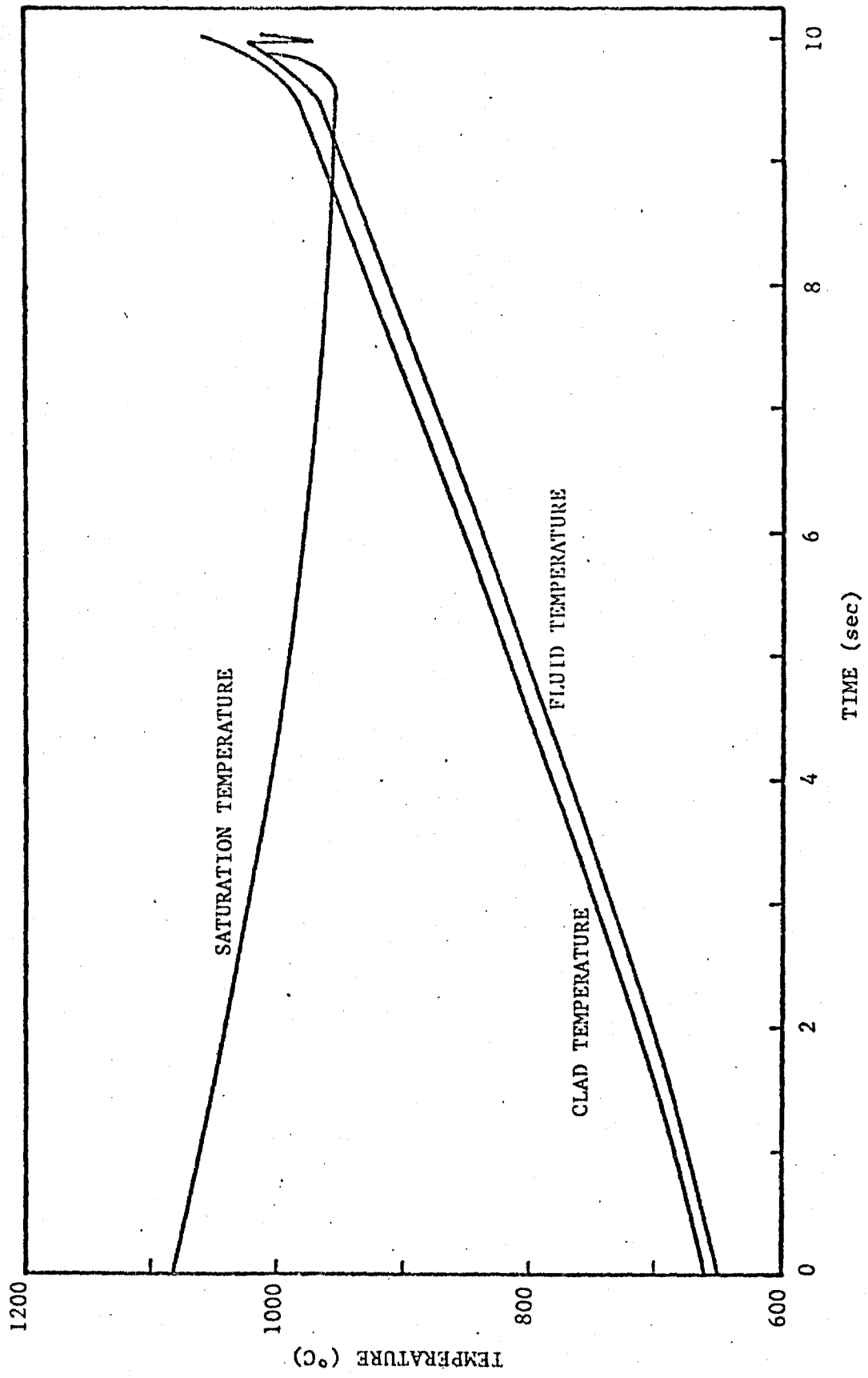


Figure 4.10: P3A — 1D: Temperature Vs. Time

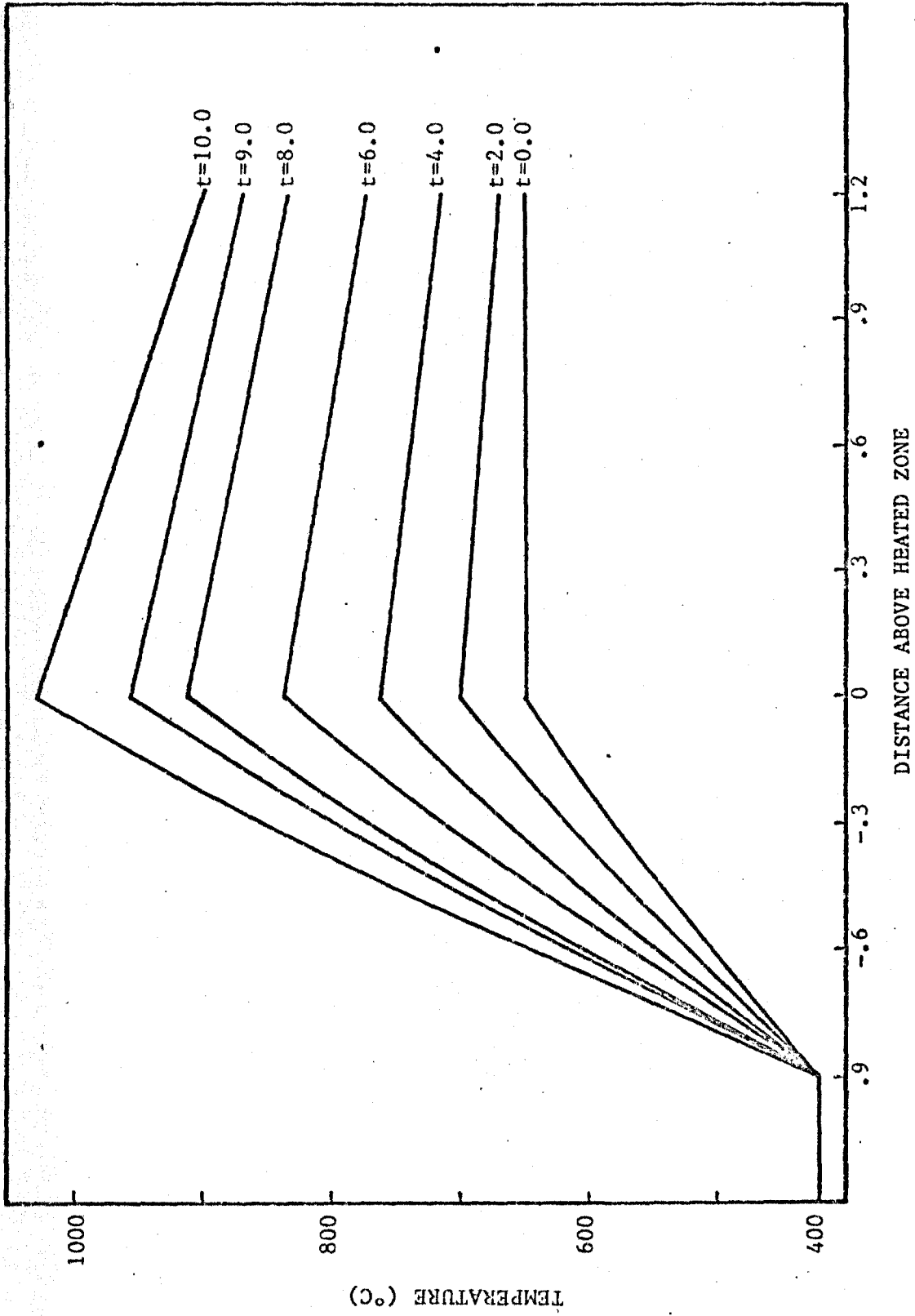


Figure 4.11: P3A-1D: Temperature Profiles

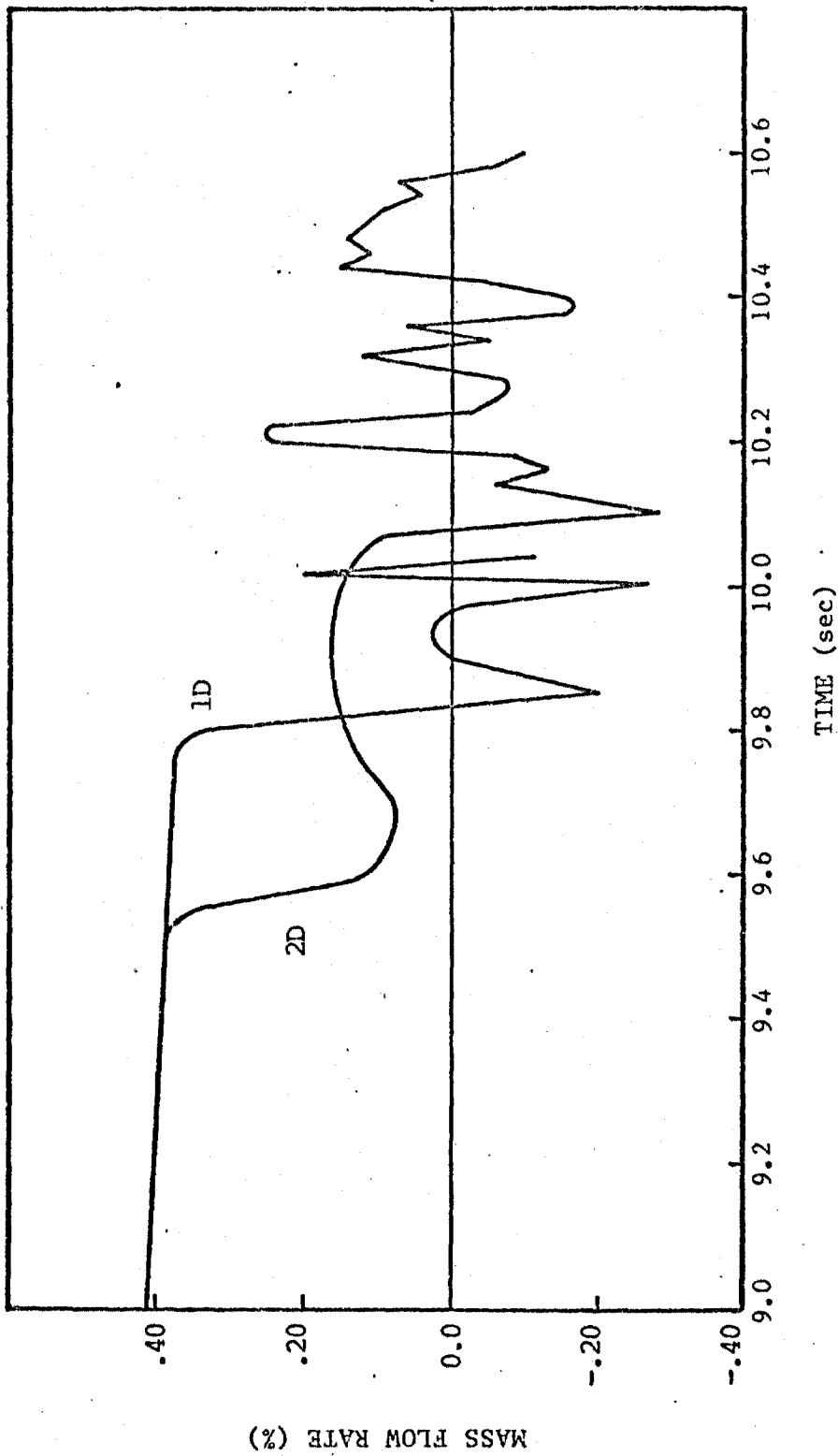


Figure 4.12: P3A: Comparison Between 1D and 2D: Mass Flow Rate

#### 4.2 W1 - SLSF Test

The W1 experiment is a test recently conducted under the direction of the Hanford Engineering Development Laboratory. Although the test has been completed, their results are not yet made public.

The test was divided in two parts. The first one aimed at determining the fuel pin heat released characteristics during a loss of pipe integrity accident. This part of the test does not involve boiling.

The second part of the test was directed to determine stable boiling and recovery limits as a function of fuel pin power. This part is the object of the numerical simulation presented in this section. Table 4.4 shows the relevant design data for the test [36], [38].

A series of flow transients were performed with several values of bundle power and flow decrease. Figure 4.12 is the graph of a typical Boiling Window Test flow transient. Table 4.5 shows the bundle power and percentage of full flow for each of the tests.

Following a series of figures present the results of the NATOF-2D simulation of the tests. For each case analyzed a figure shows the evolution in time of the saturation temperature, clad and fluid temperatures for the central channel and the fluid temperature for the edge channel. For sequences 6a, 7a', 7b', 3 and 4 the axial temperature profile for the central channel is also shown. Finally

for the cases where substantial voiding occurred, namely sequences 7a', 7b' and 4 a figure showing the void maps for the three channels is also presented.

In general the results obtained for the high power tests (14.4 kw/ft) seem to present values which agree with the predictions of the test plan (Table 4.5). On the other hand, for the lower power (and longer) tests, NATOF-2D predicted boiling conditions more severe than the expected in the test plan. One possible explanation for this discrepancy is an overestimating of the gap conductance by NATOF-2D, but of course an analysis of the results will be conclusive only when the test results are made available.

As an extension of the test, a 217-pin bundle simulation was performed under the same conditions of test sequence 7b'. The simulation was made with five radial mesh cells and the same geometric and fuel pin design parameters as the ones used for the W1 test. The results are presented in figures 4.31 through 4.33. Comparing these figures with the correspondent figures for the 19-pin test, figures 4.23 through 4.25, the following conclusions can be drawn:

- ¶ The onset of boiling occurred at approximately the same time.
- ¶ The flow reversal occurred earlier and the voiding of the subassembly were much sharper for the 217-pin bundle.

These results confirm what was expected, since the onset of boiling occurs in the central channel and is not influenced by the size of the subassembly. The second conclusion was also expected, since in a large fuel assembly, the edge channel which is submitted to a smaller heat flux and also has the hexcan wall as a heat sink, occupies a fraction of the total flow area which is much smaller than the correspondent edge channel for a 19-pin bundle.

TABLE 4.4

W1 Test Bundle DataGeometry

Number of Pins	19	
Fuel Pellet OD (m)	$4.94 \times 10^{-3}$	.1945 in
Clad OD (m)	$5.842 \times 10^{-3}$	.230 in
Clad ID (m)	$5.030 \times 10^{-3}$	.200 in
Wire Wrap OD (m)		
inner pins	$1.422 \times 10^{-3}$	.056 in
outer pins	$7.11 \times 10^{-4}$	.028 in
Flat to Flat (m)	$3.26 \times 10^{-2}$	1.283 in
Duct Wall Thickness (m)	$1.016 \times 10^{-4}$	.040 in
Length of Fuel (m)	.9144	36.0 in
Inlet to Bottom of Fuel (m)	.279	11 in
Top of Fuel to End of Pins (m)	1.27	50 in
Wire Wrap Lead (m)	.3048	12.0 in
Fill Gas	Helium-Neon (10%), 25 psia at 68°F	
Fuel	Uranium-Plutonium mixed oxide, Pu 25% of total mass.	

Table 4.4 continued

Thermo-Hydraulics

Inlet Temperature (°C)	388	732°F
Test Bundle Flow (kg/sec)	1.95	4.29 lbm/sec
Cover Gas Pressure (atm)	1.18	17 psia
Inlet Pressure (atm)	6.42	91.8 psia

Numerics of Simulation

Number of Axial Mesh Cells	12
Number of Radial Mesh Cells	3

TABLE 4.5

Boiling Window Matrix for the W1 Experiment

Fuel Bundle Power = 348 kw

Peak Pin Power = 7.5 kw/ft

	<u>Percentage of Full Flow</u>	<u><math>\Delta t_z</math></u>	<u>Test Sequence</u>
Approach to Boiling	29	5.0	1
Incipient Boiling			
Normal Procedure	24	5.0	2
Fallback Procedure A	24	7.0	2a
Fallback Procedure B	22	4.5	2b

Fuel Bundle Power = 532 kw

Peak Pin Power = 11.1 kw/ft

	<u>Percentage of Full Flow</u>	<u><math>\Delta t_z</math></u>	<u>Test Sequence</u>
Approach to Boiling	42	5.0	3
Incipient Boiling			
Normal Procedure	35	4.0	4
Fallback Procedure A	35	6.0	4a
Fallback Procedure B	33	3.0	4b

Fuel Bundle Power = 662 kw  
 Peak Pin Power = 14.4 kw/ft

	<u>Percentage of Full Flow</u>	<u><math>\Delta t_z</math></u>	<u>Test Sequence</u>
Approach to Boiling	53	5.0	5
Incipient Boiling			
Normal Procedure	45	3.0	6
Fallback Procedure A	45	5.0	6a
Fallback Procedure B	43	2.5	6b
Dryout or Fuel Pin Failure			
Normal Procedure A	42	2.0	7
Normal Procedure B	40	2.0	7'
Fallback Procedure A	42	3.0	7a
Fallback Procedure B	40	3.0	7a'
Fallback Procedure C	40	3.0	7b
Fallback Procedure D	38	3.0	7b'

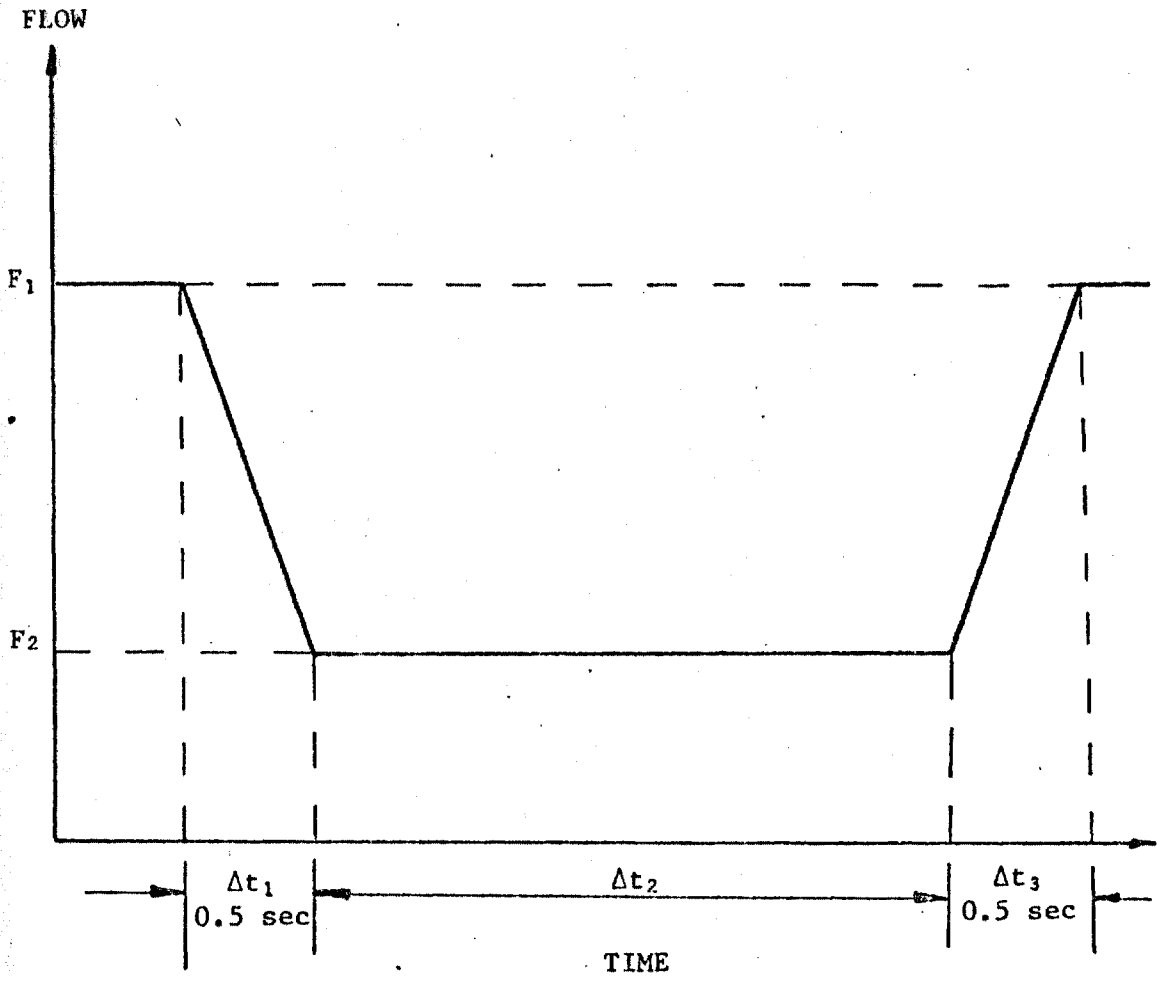


Figure 4.13

Typical Boiling Window Flow Decay for the WI Test

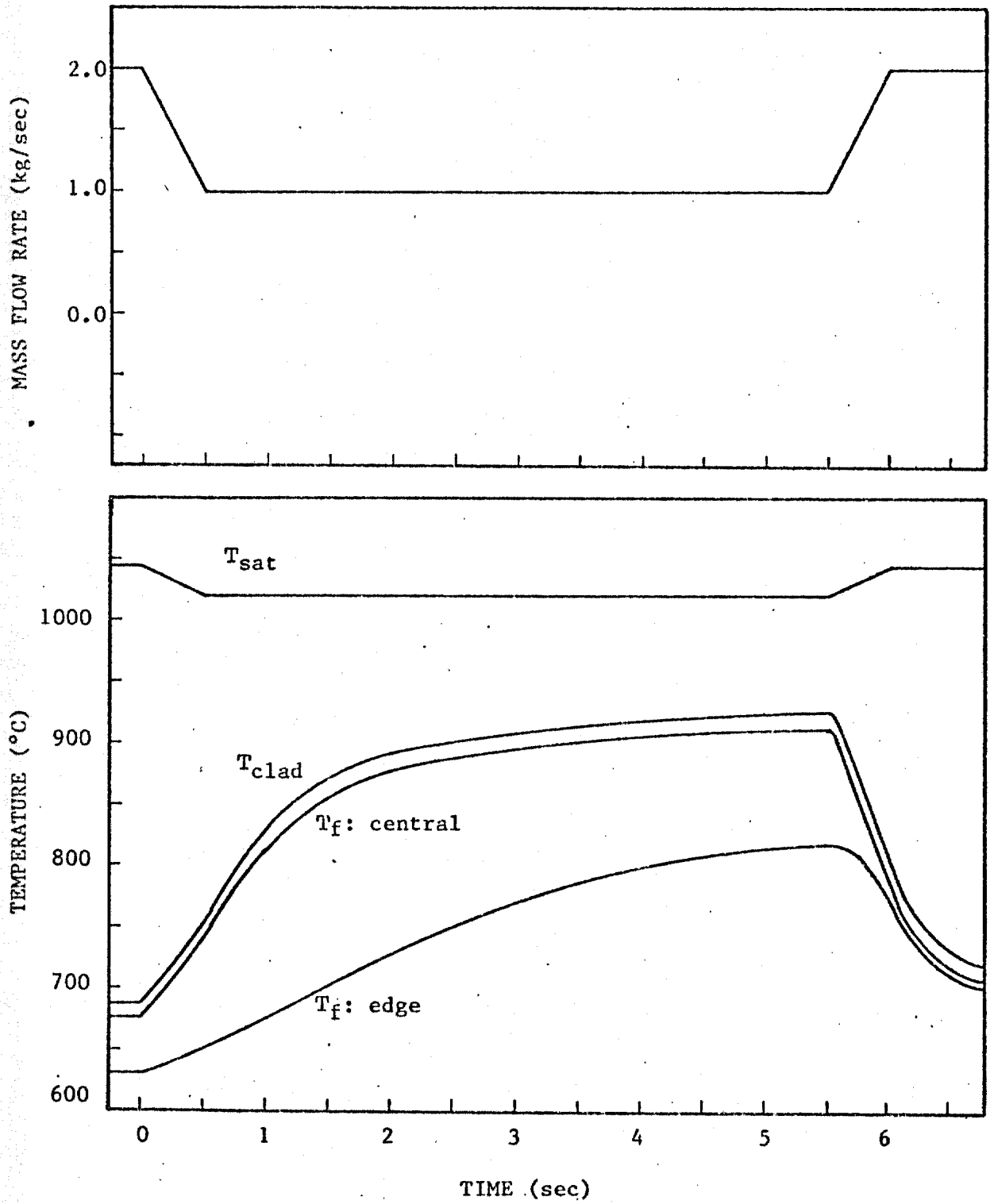


Figure 4.14: W1: Temperatures and Mass Flow Rate for the Sequence 5

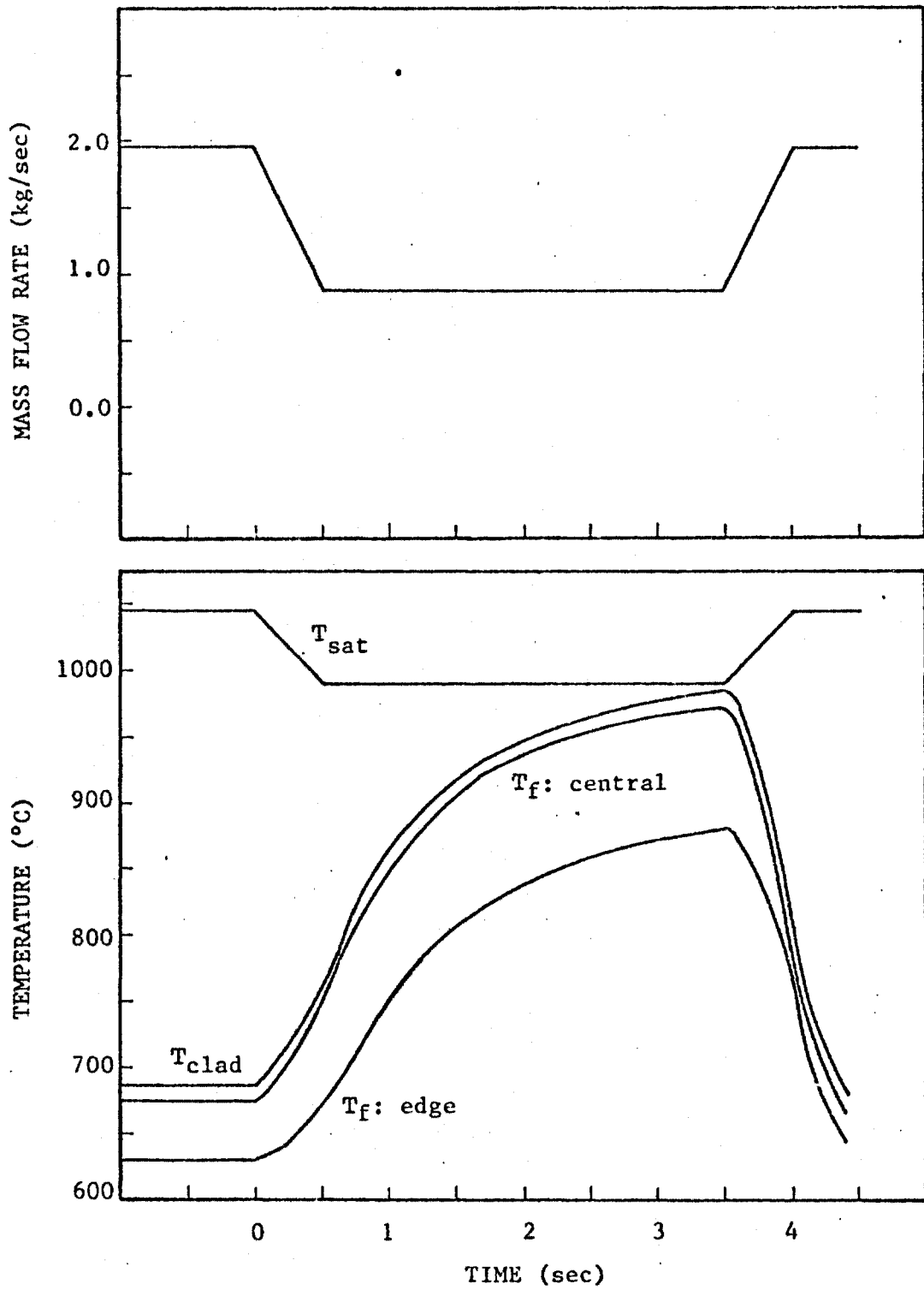


Figure 4.15: W1: Temperature and Mass Flow Rate for Sequence 6

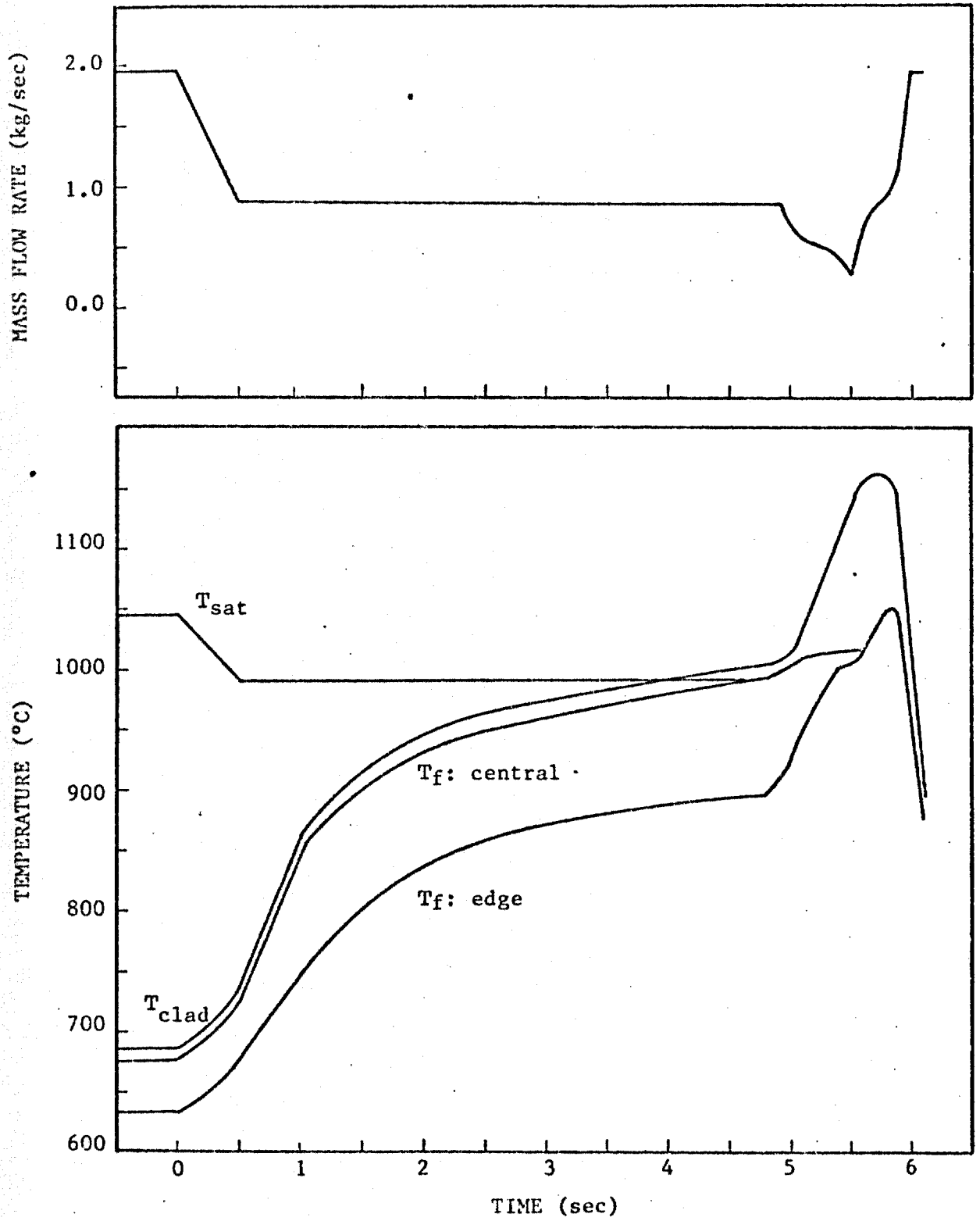


Figure 4.16: W1: Temperatures and Mass Flow Rate for the Sequence 6a

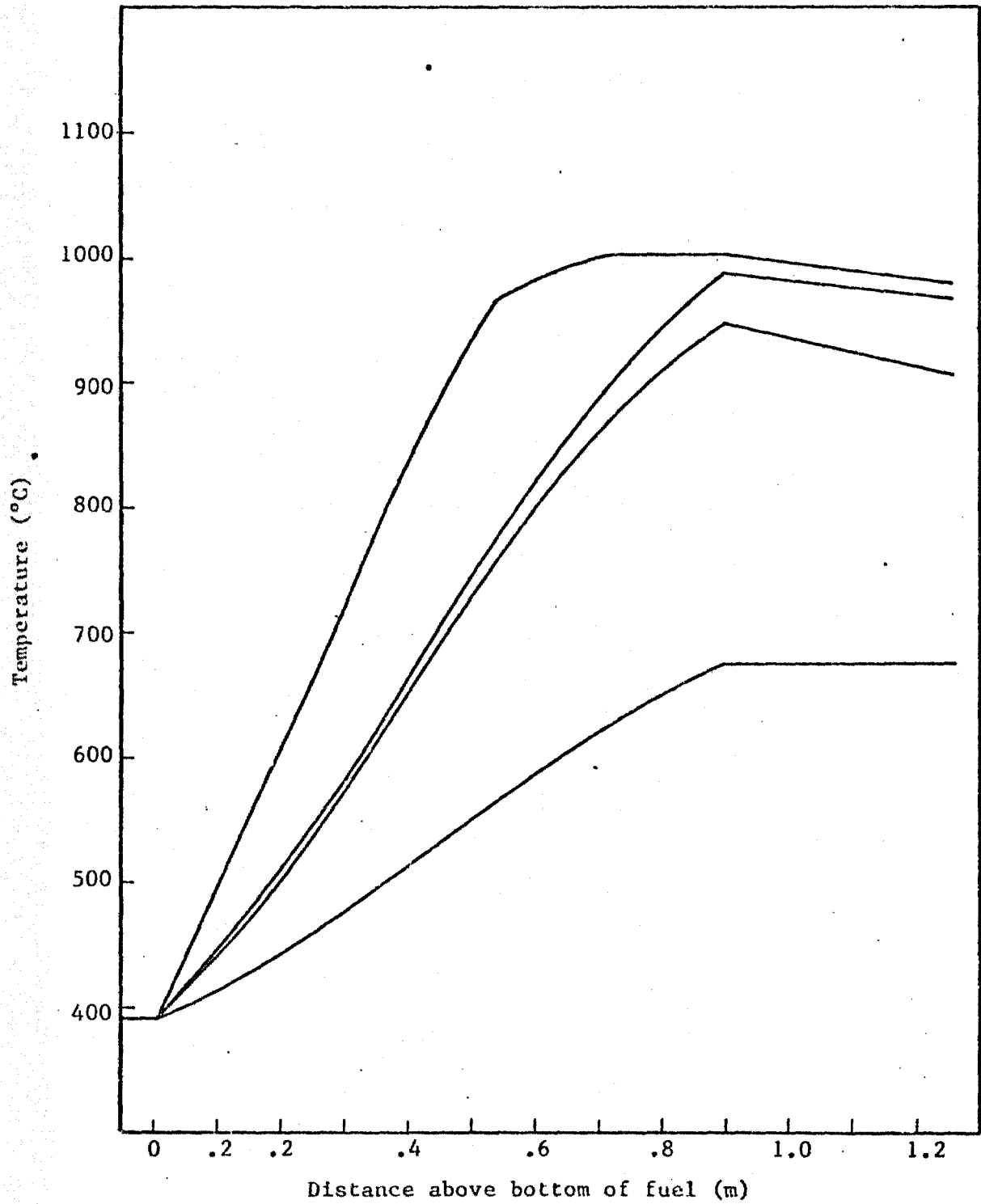


Figure 4.17: W1: Axial Temperature Profile for Sequence 6a

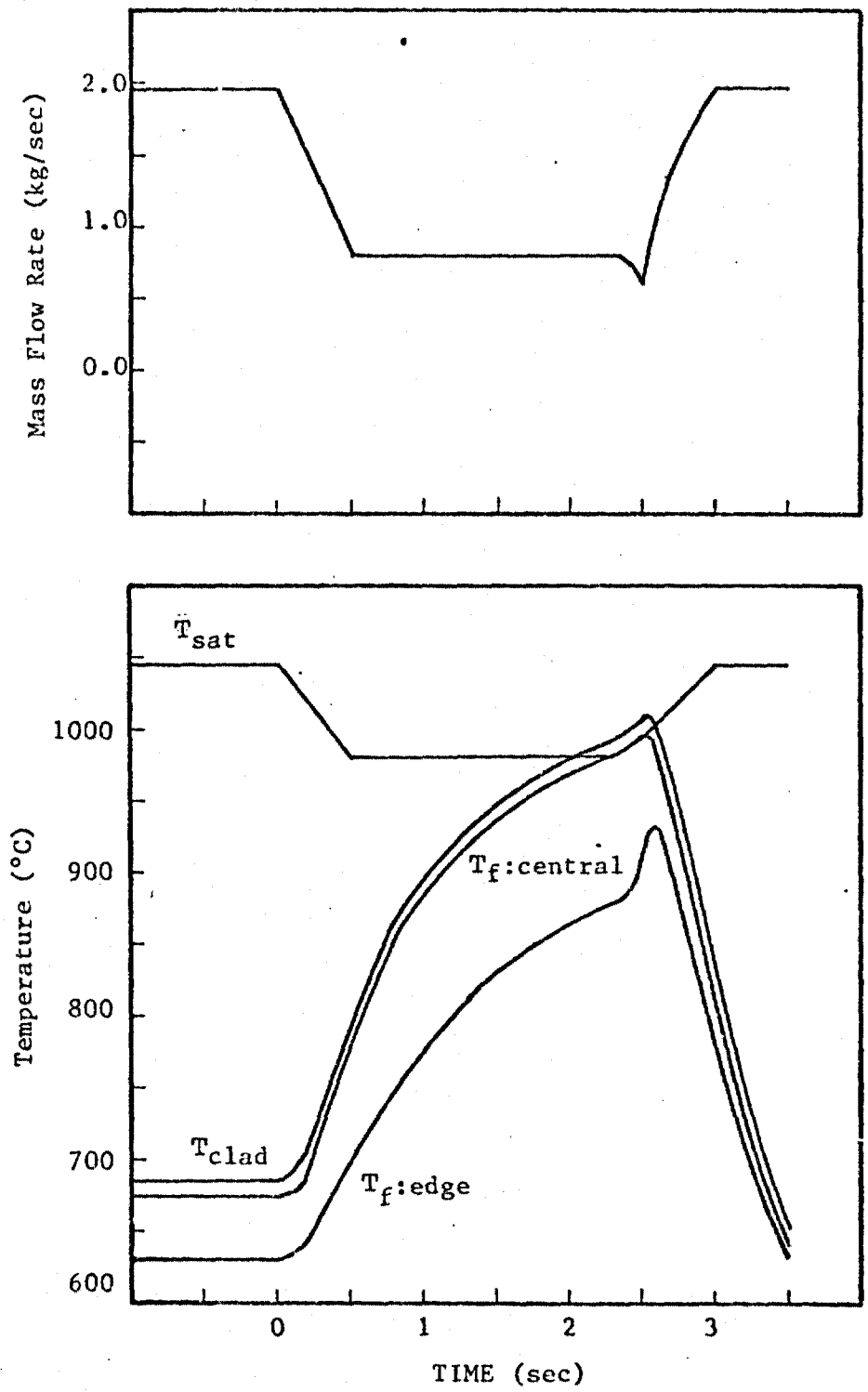


Figure 4.18: W1: Temperature and Mass Flow Rate for Sequence 7

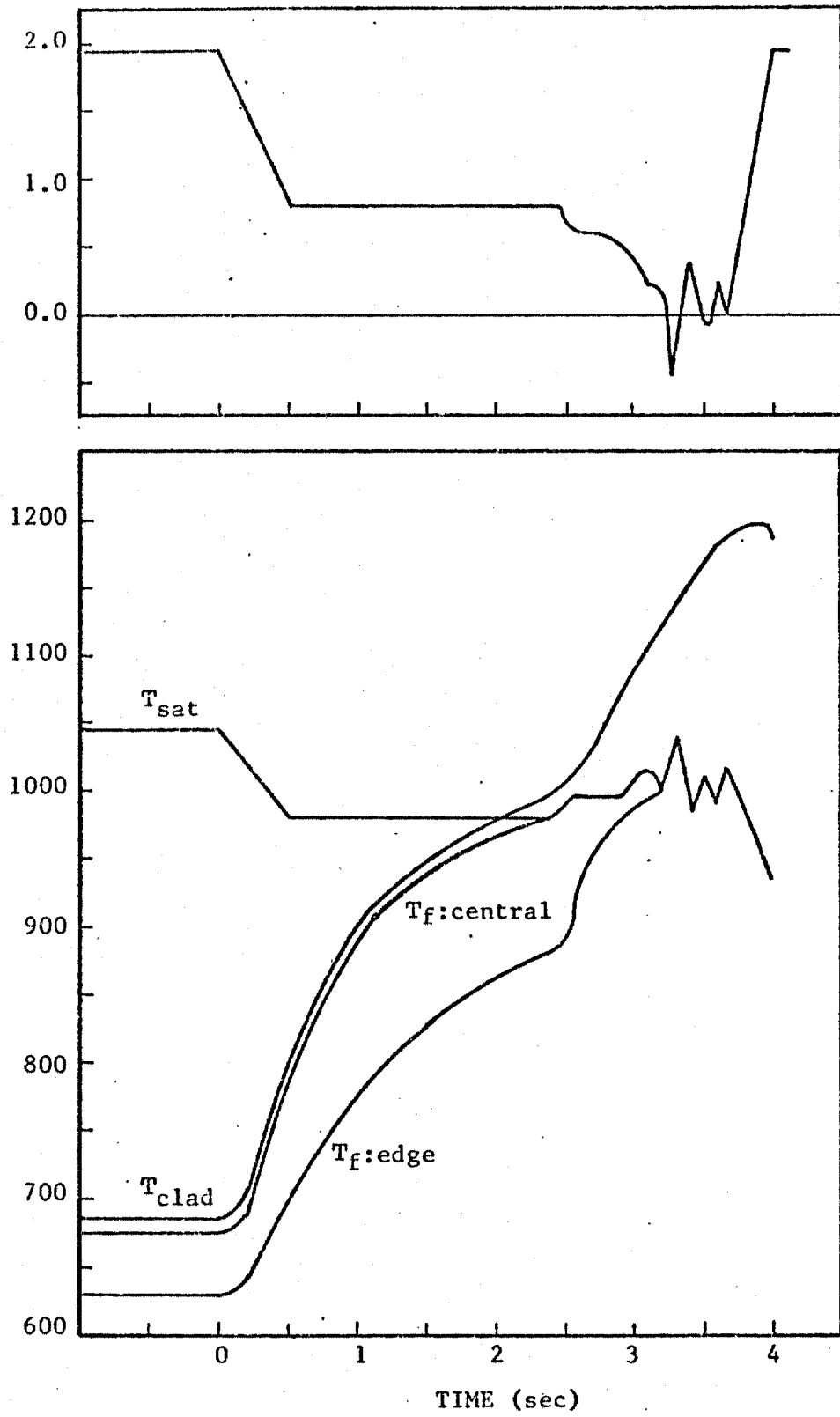


Figure 4.19: W1: Temperature and Mass Flow Rate for Sequence 7a

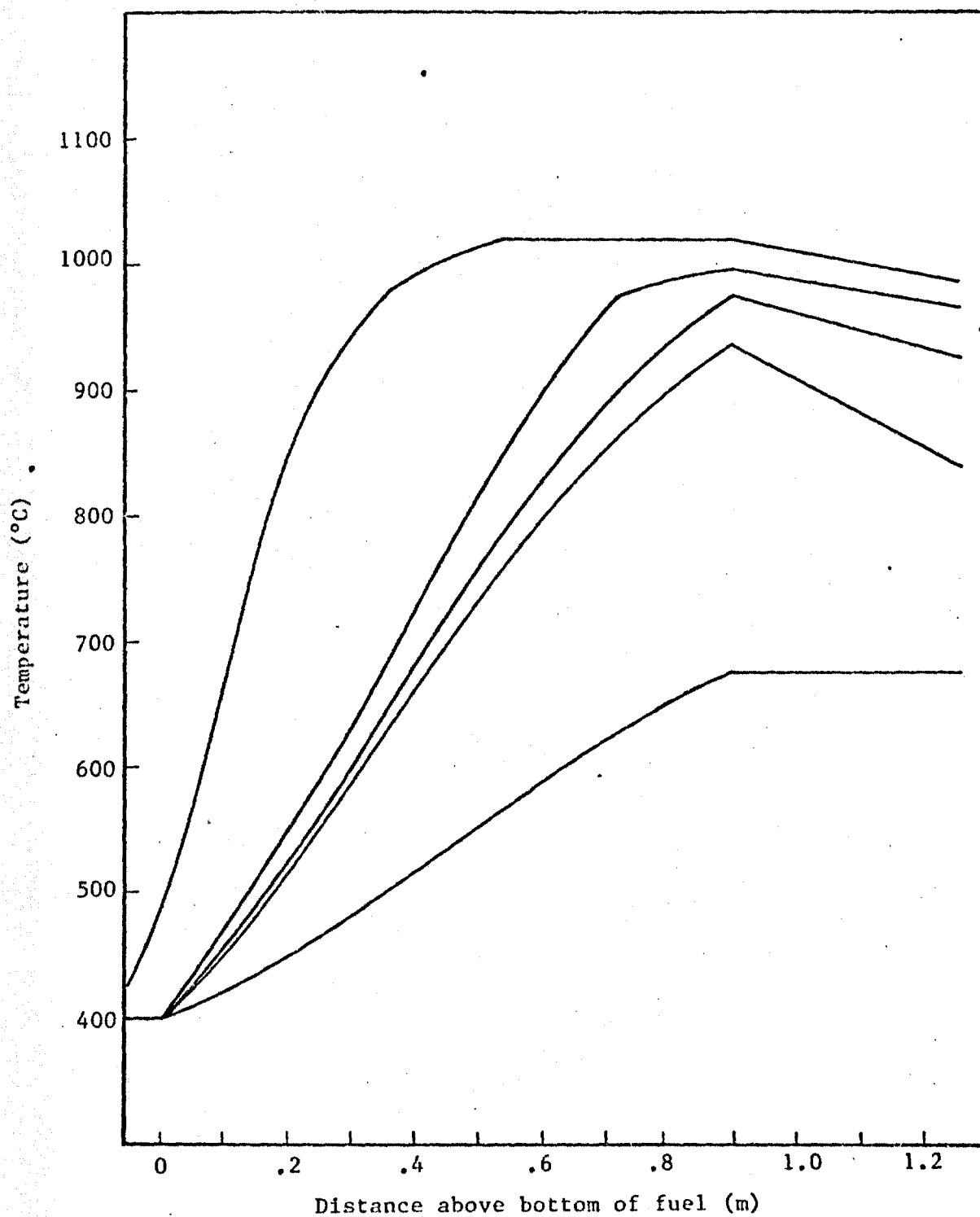


Figure 4.20: W1: Axial Temperature Profile for Sequence 7a

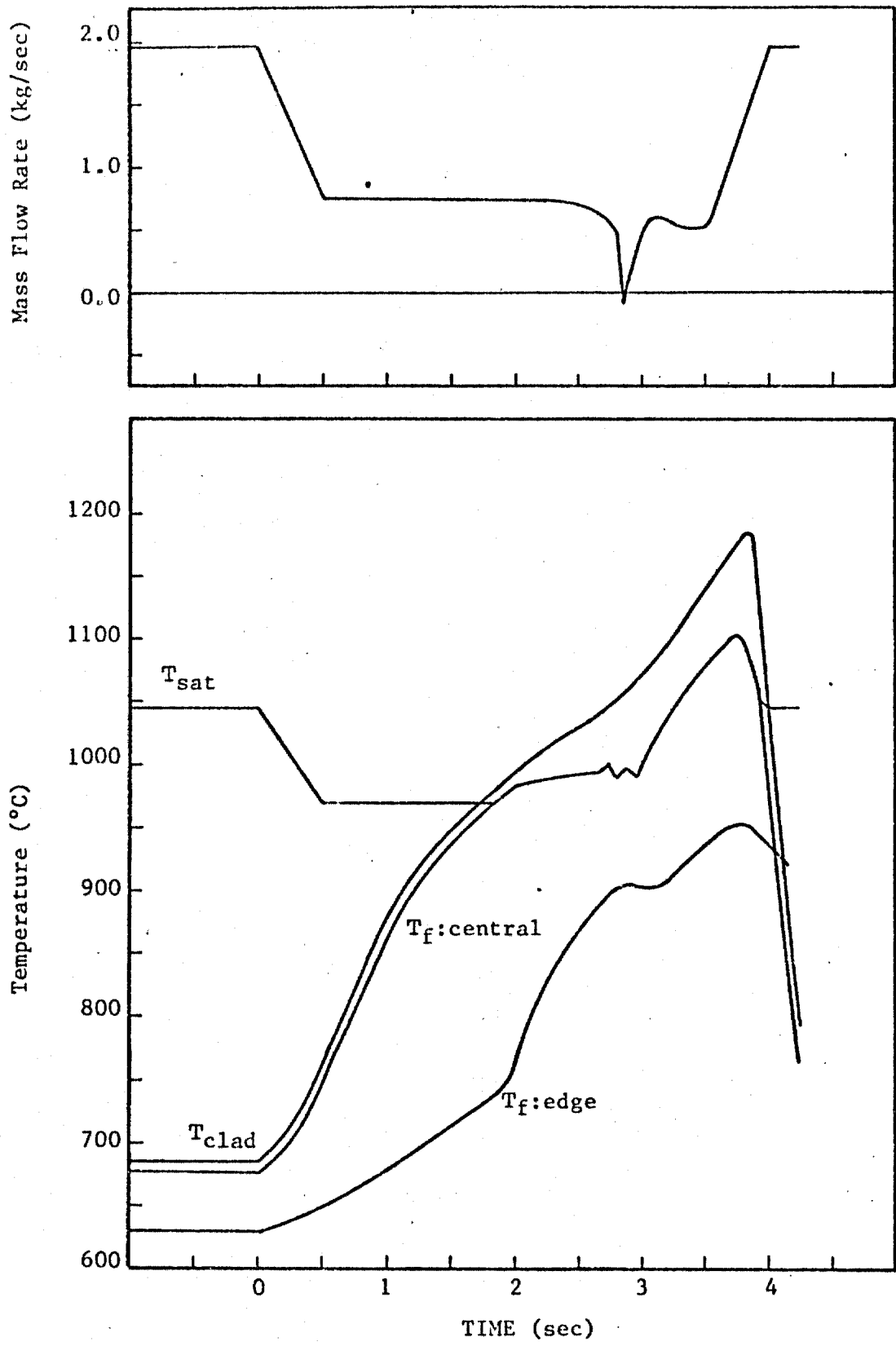


Figure 4.21: W1: Temperatures and Mass Flow Rate for Sequence 7a'

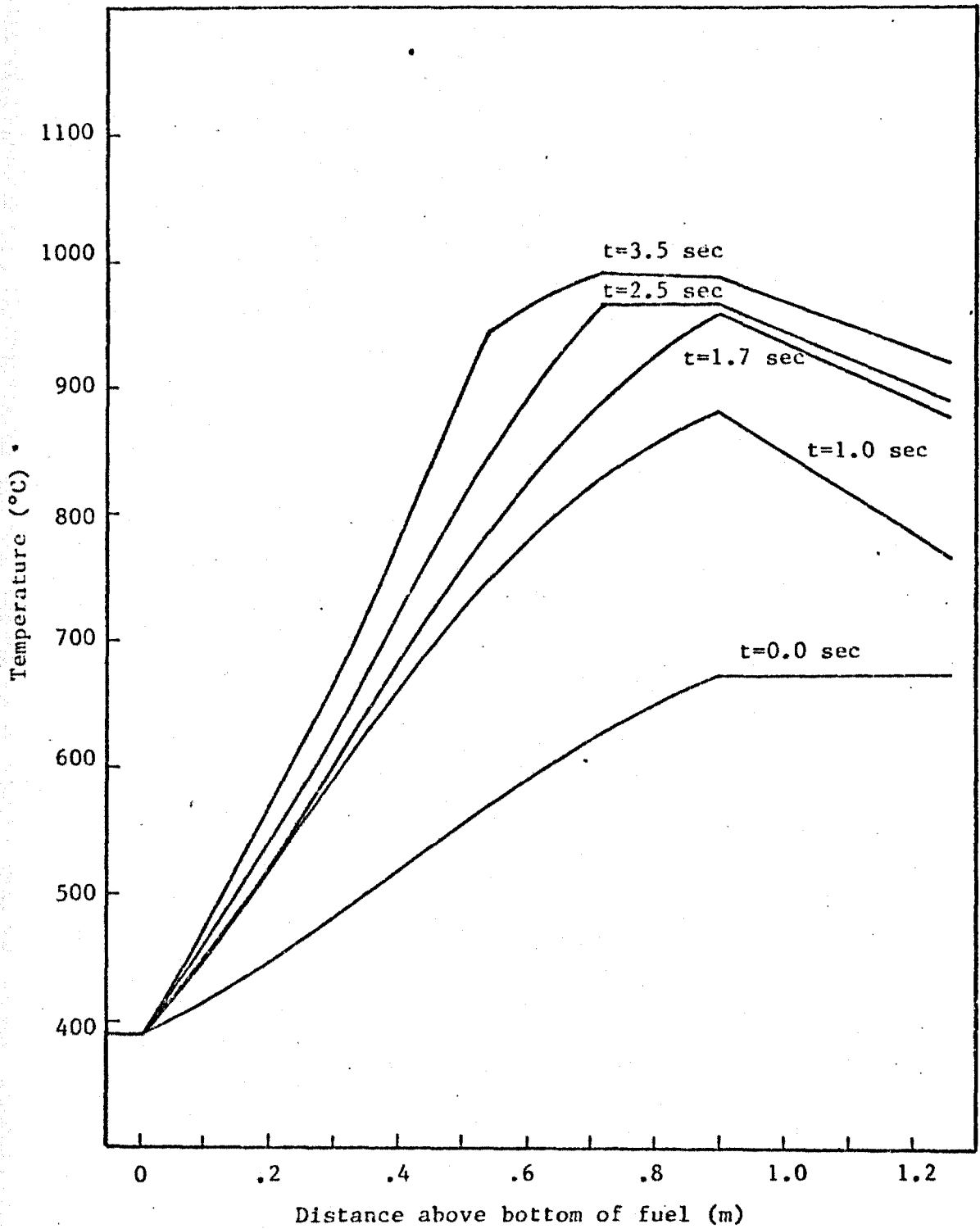


Figure 4.22: Axial Temperature Profile for Sequence 7a'

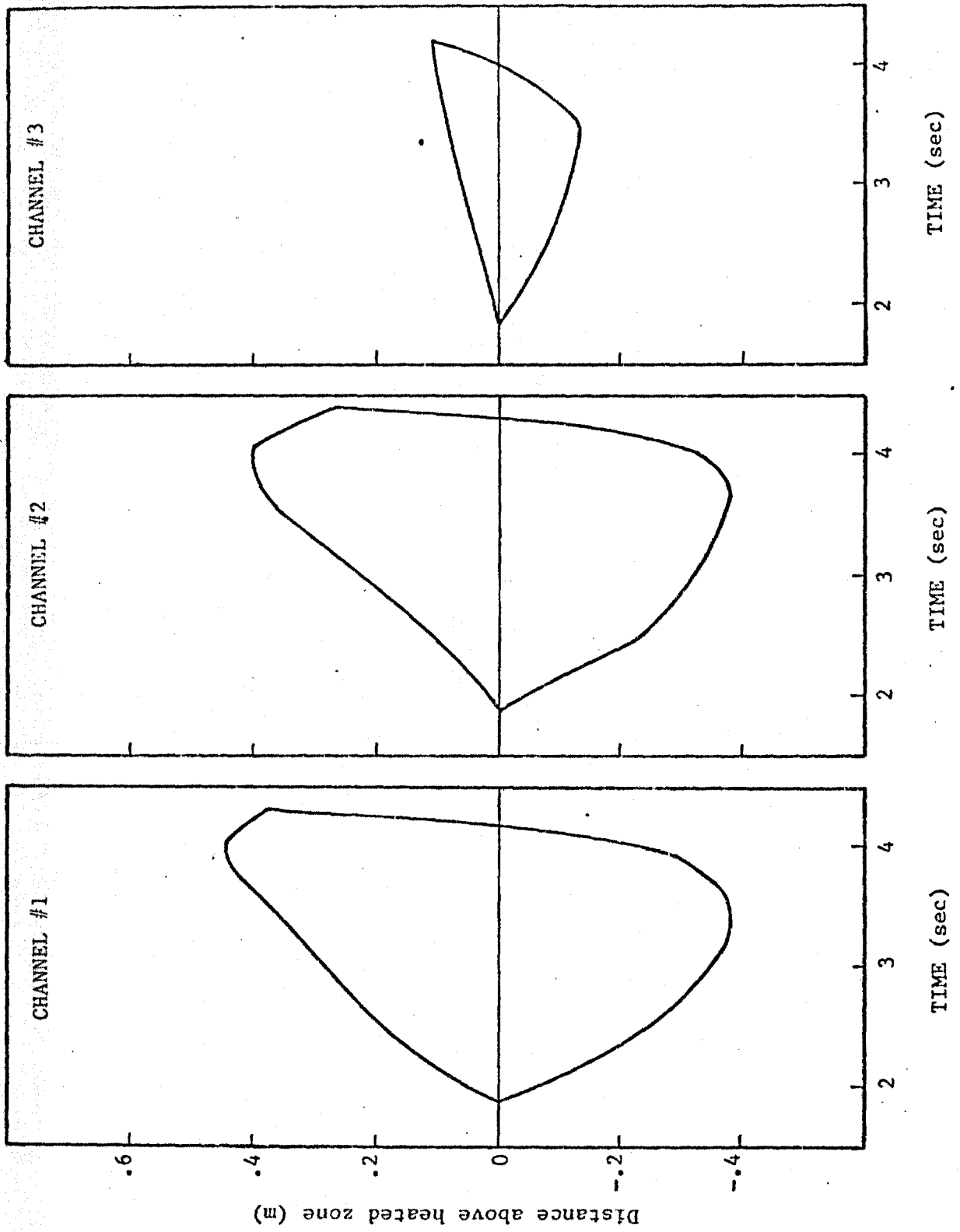


Figure 4.23: W1: Void Maps for Sequence 7a'

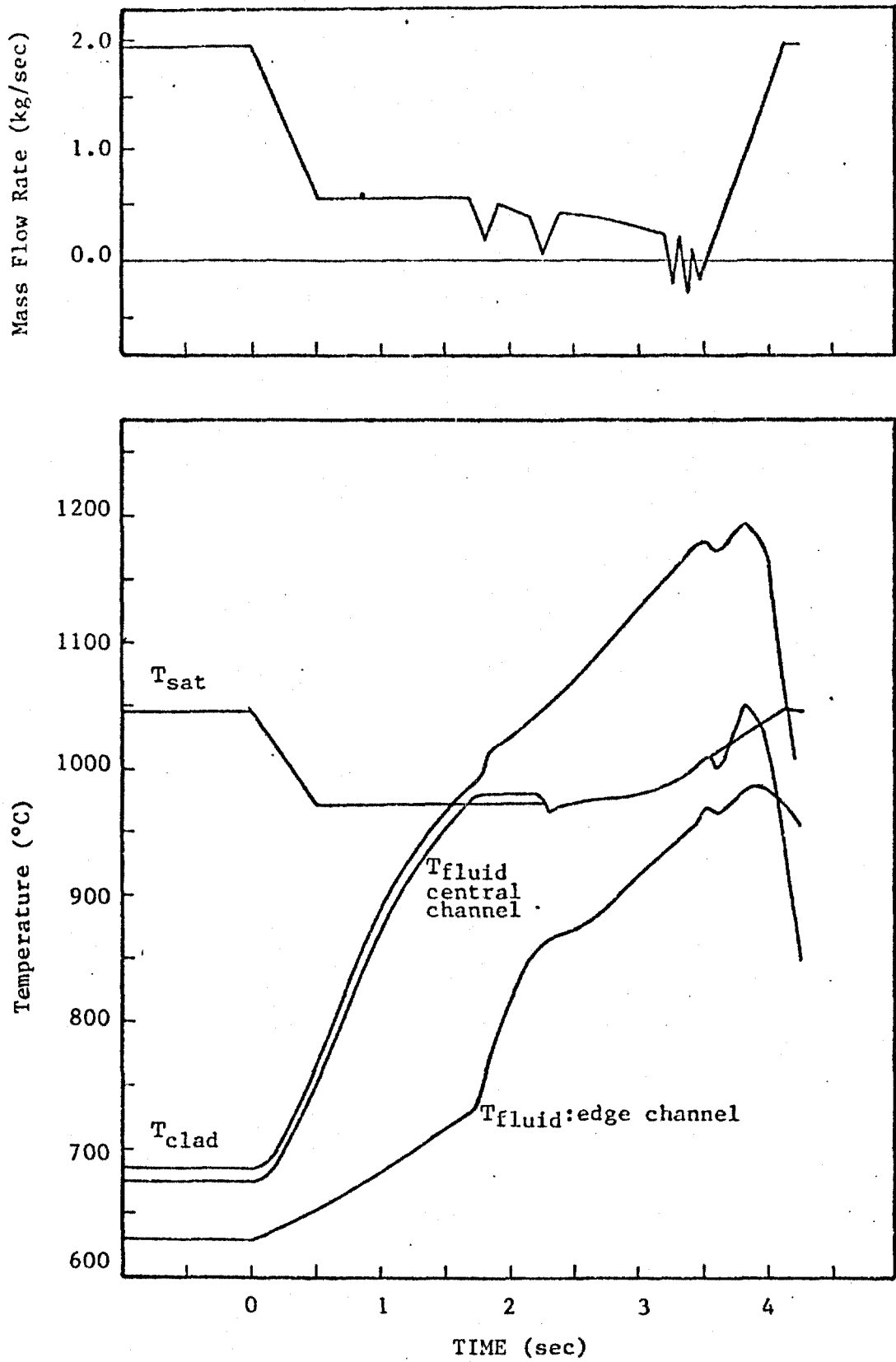


Figure 4.24: W1: Temperatures and Mass Flow Rate for Sequence 7b

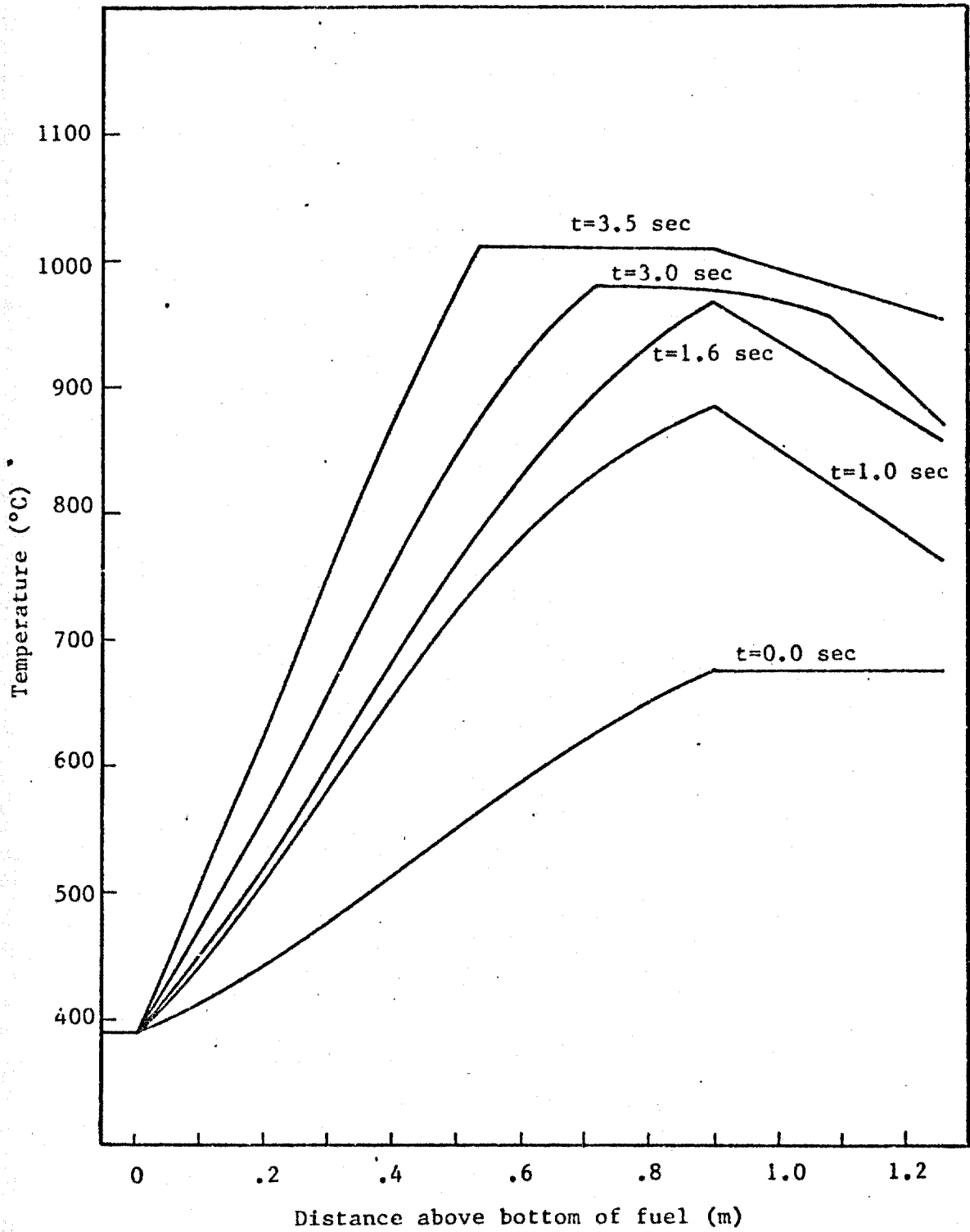


Figure 4.25: W1: Axial Temperature Profile for Sequence 7b'

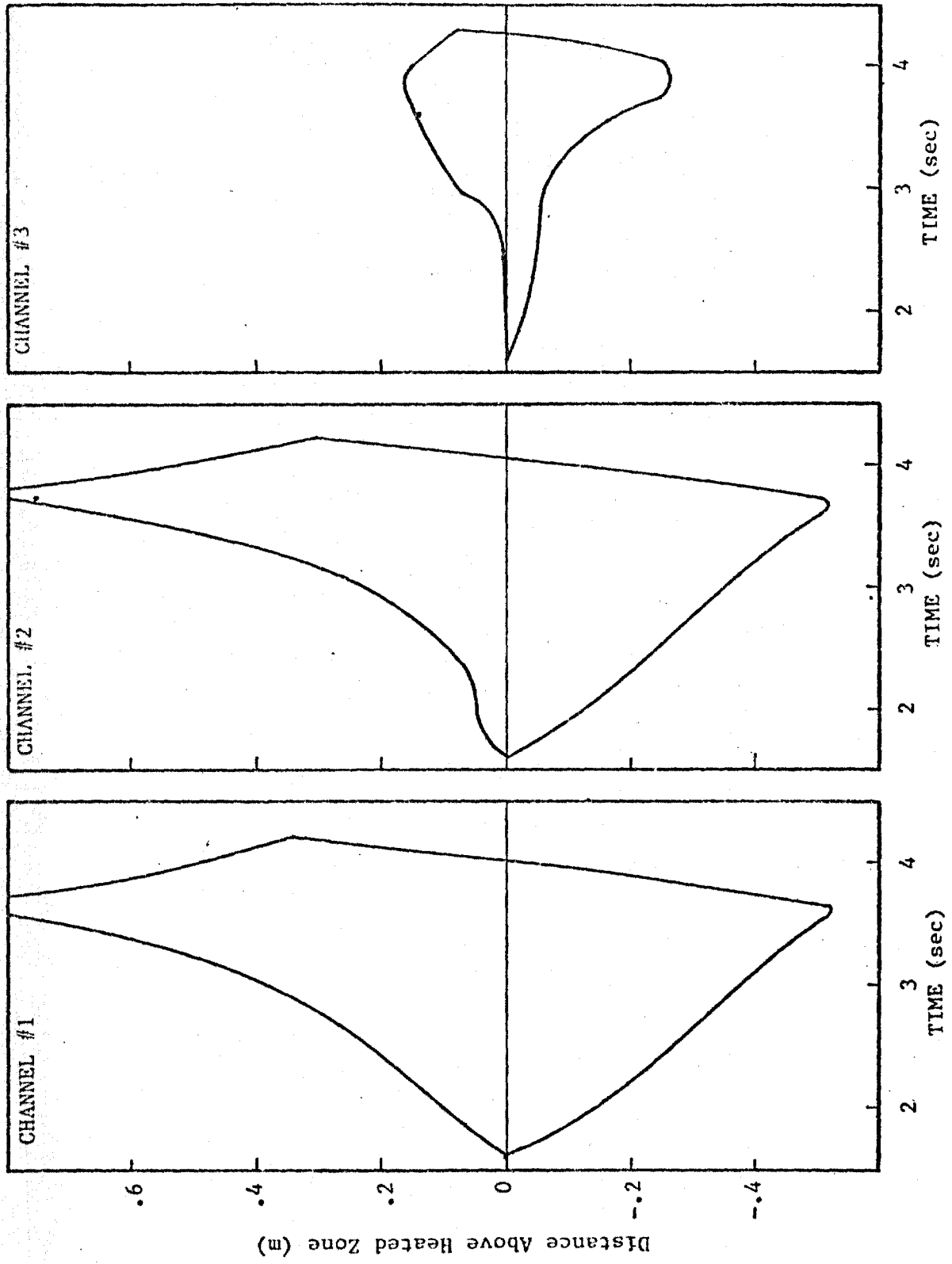


Figure 4.26: W1: Void Maps for Sequence 7b'

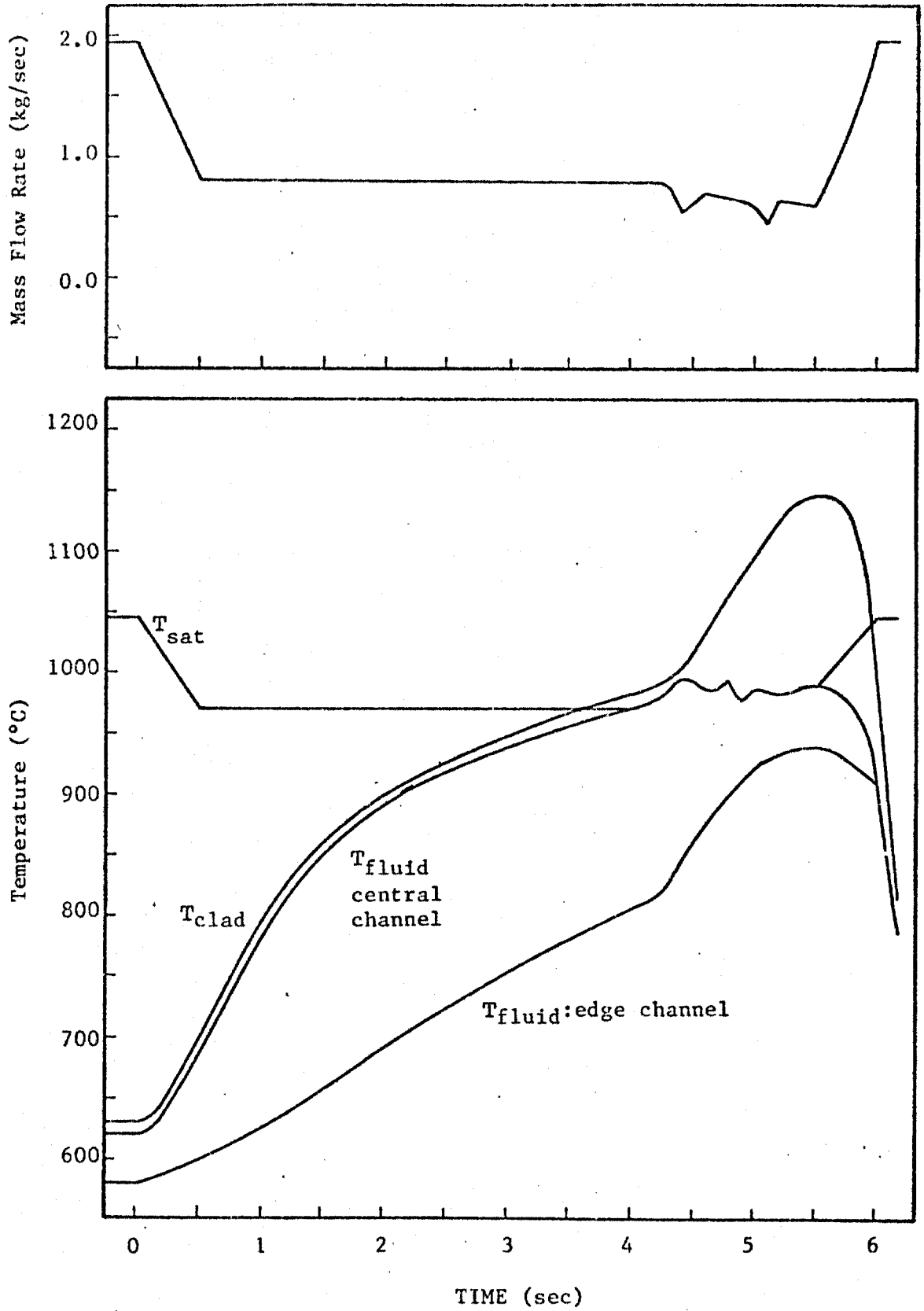


Figure 4.27: W1: Temperature and Mass Flow Rate for Sequence 3

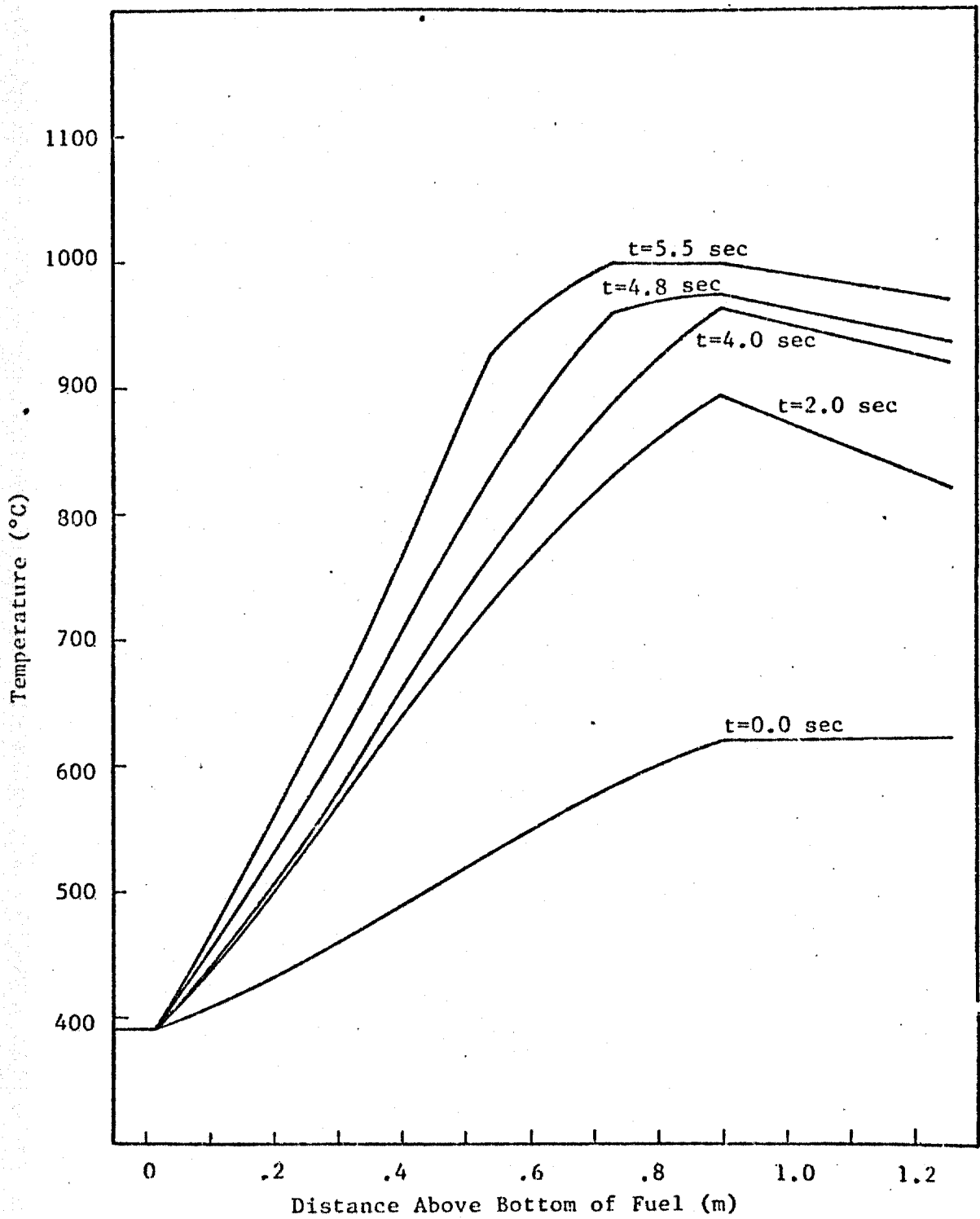


Figure 4.28: W1: Axial Temperature Profile for Sequence 3

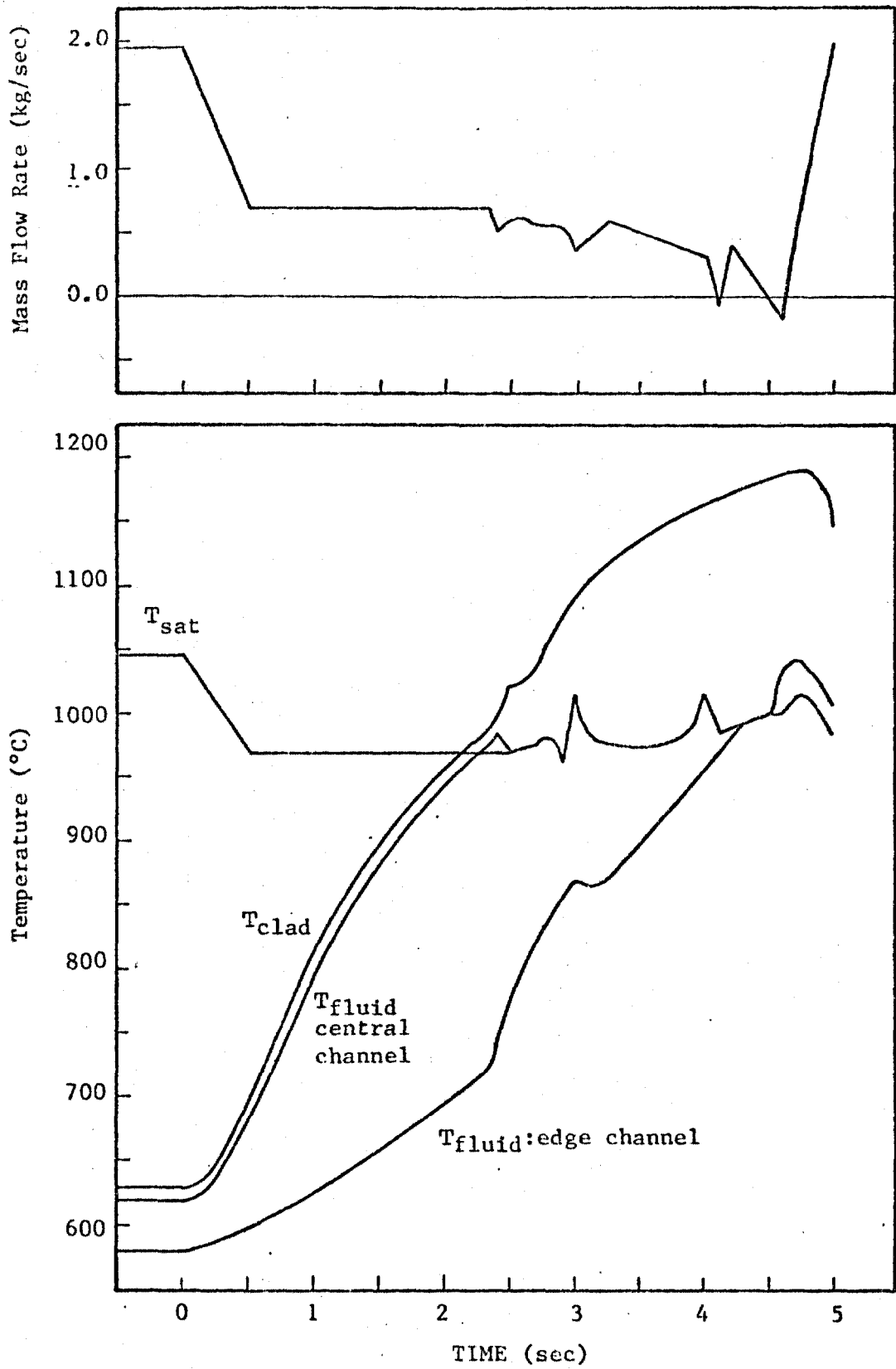


Figure 4.29: W1: Temperature and Mass Flow Rate for Sequence 4

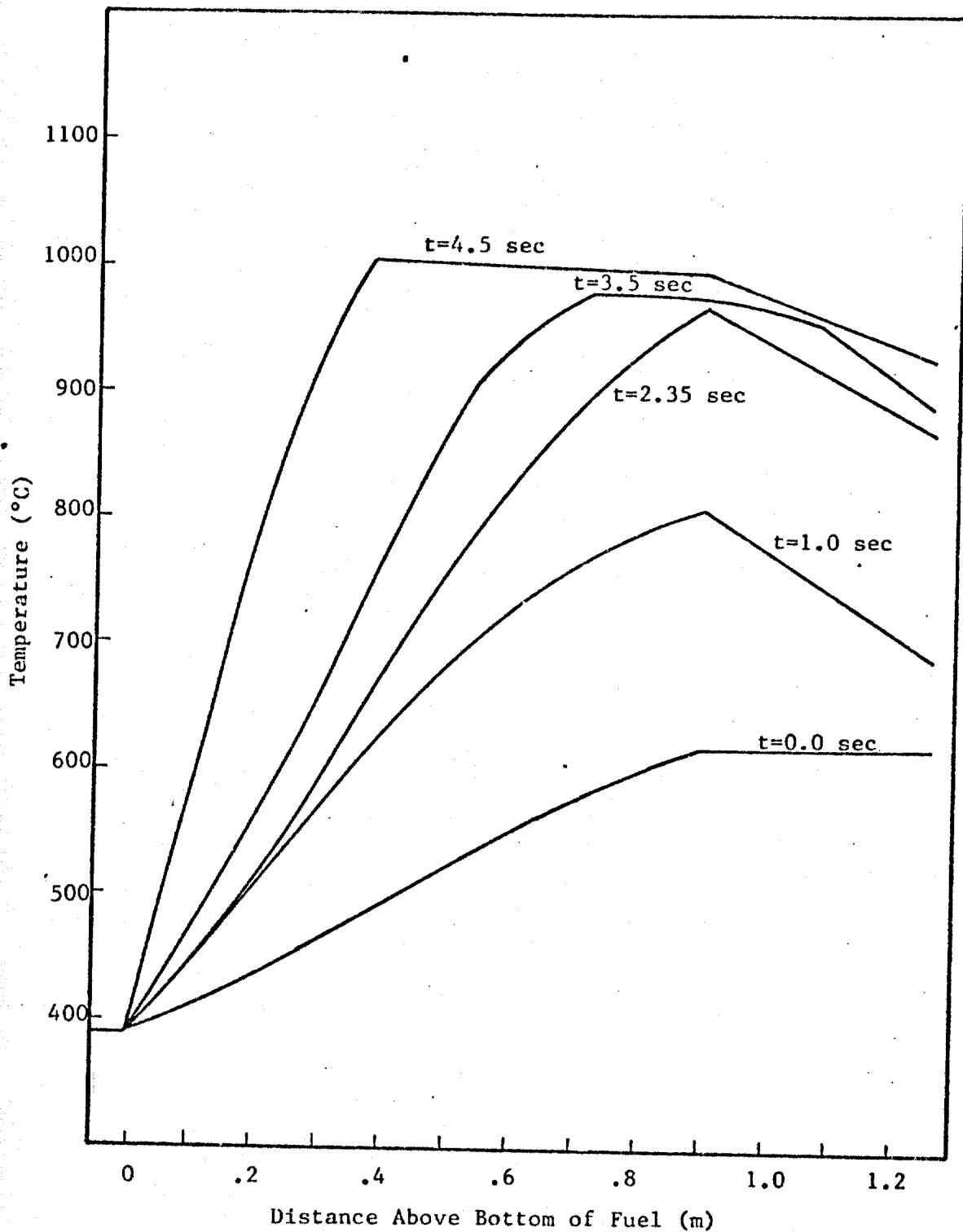


Figure 4.30: W1: Axial Temperature Profile for Sequence 4

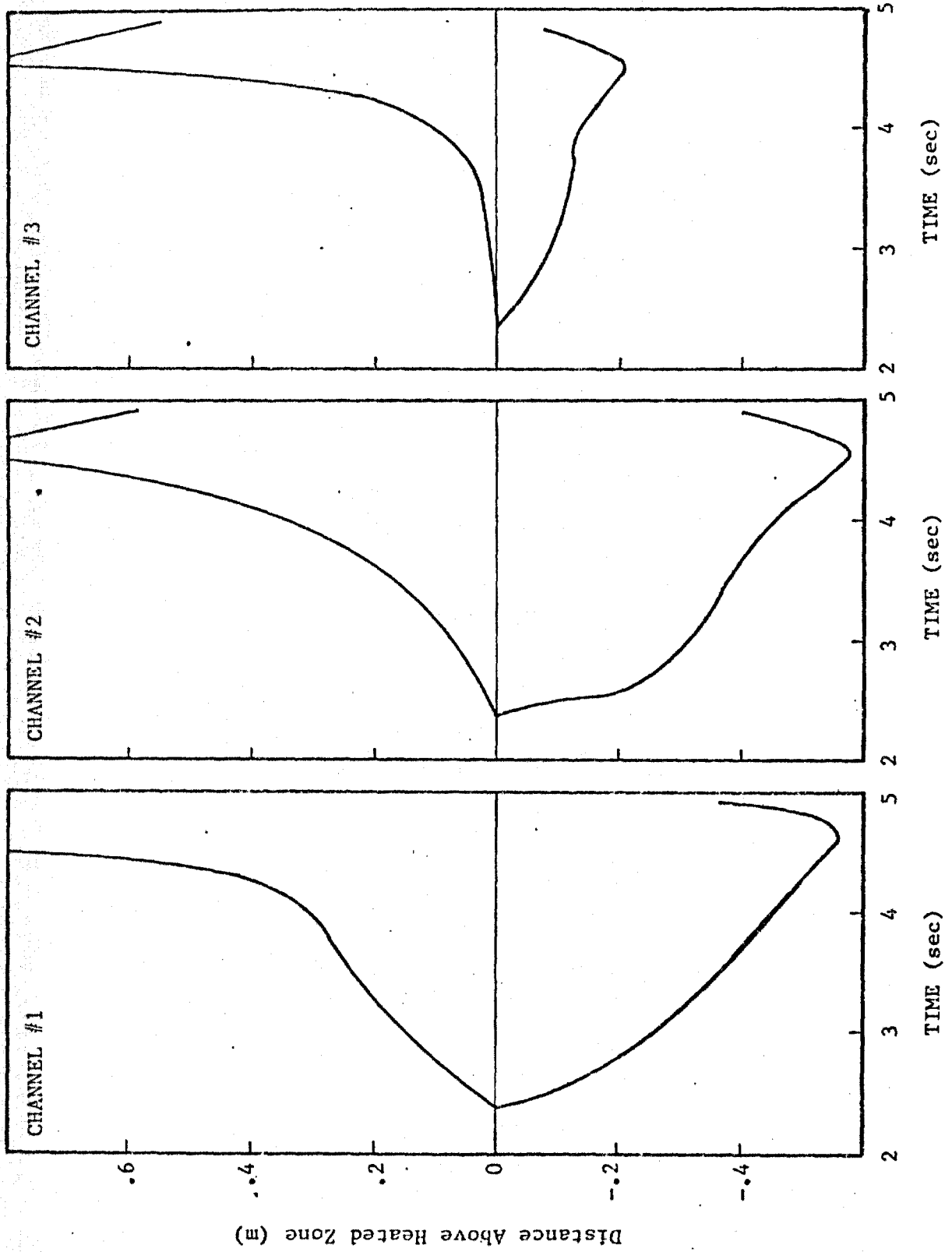


Figure 4.31: W1: Void Maps for Sequence 4

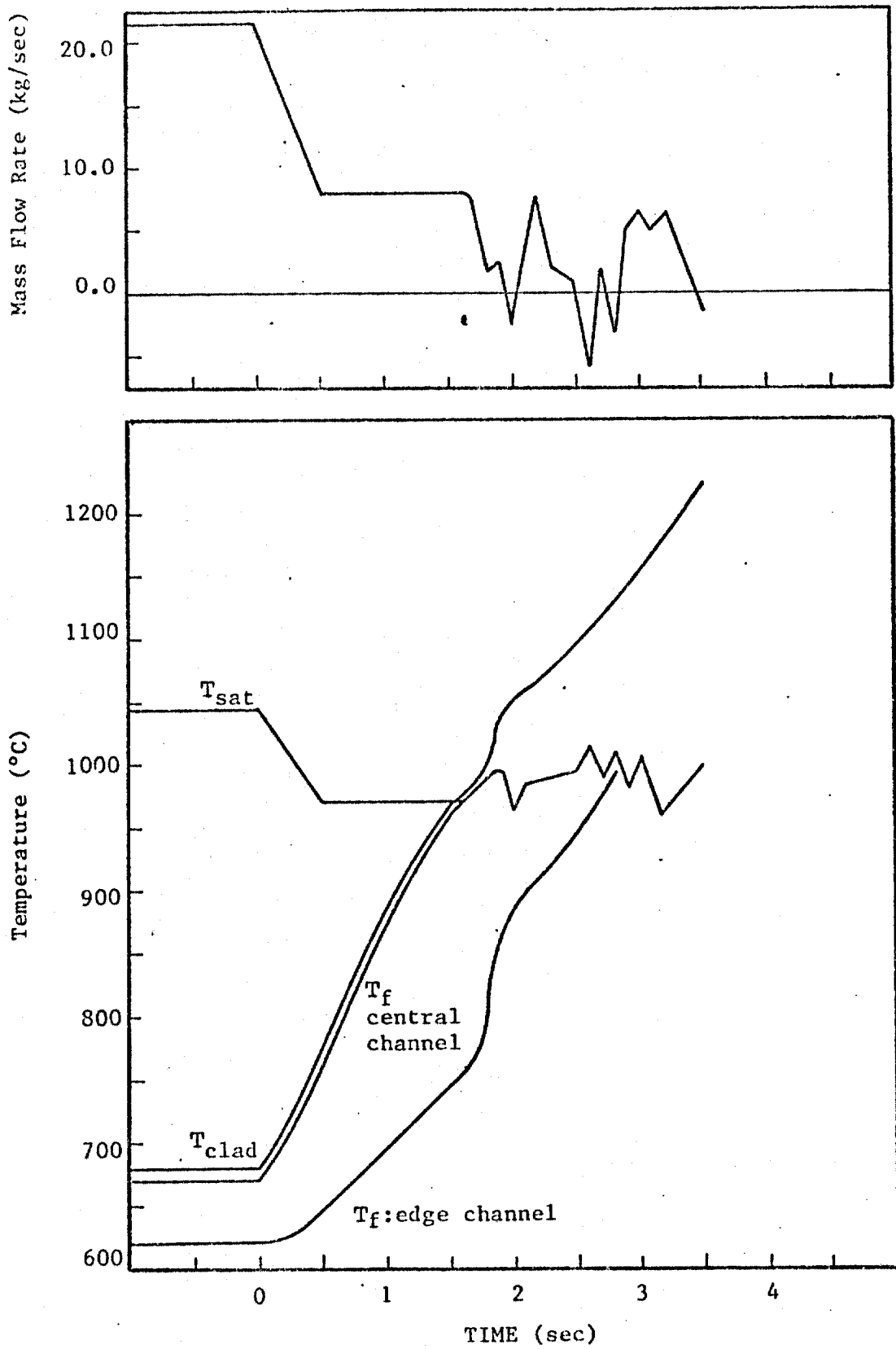


Figure 4.32: Temperature and Mass Flow Rate for 217-Pin Bundle Under Sequence 7b Conditions

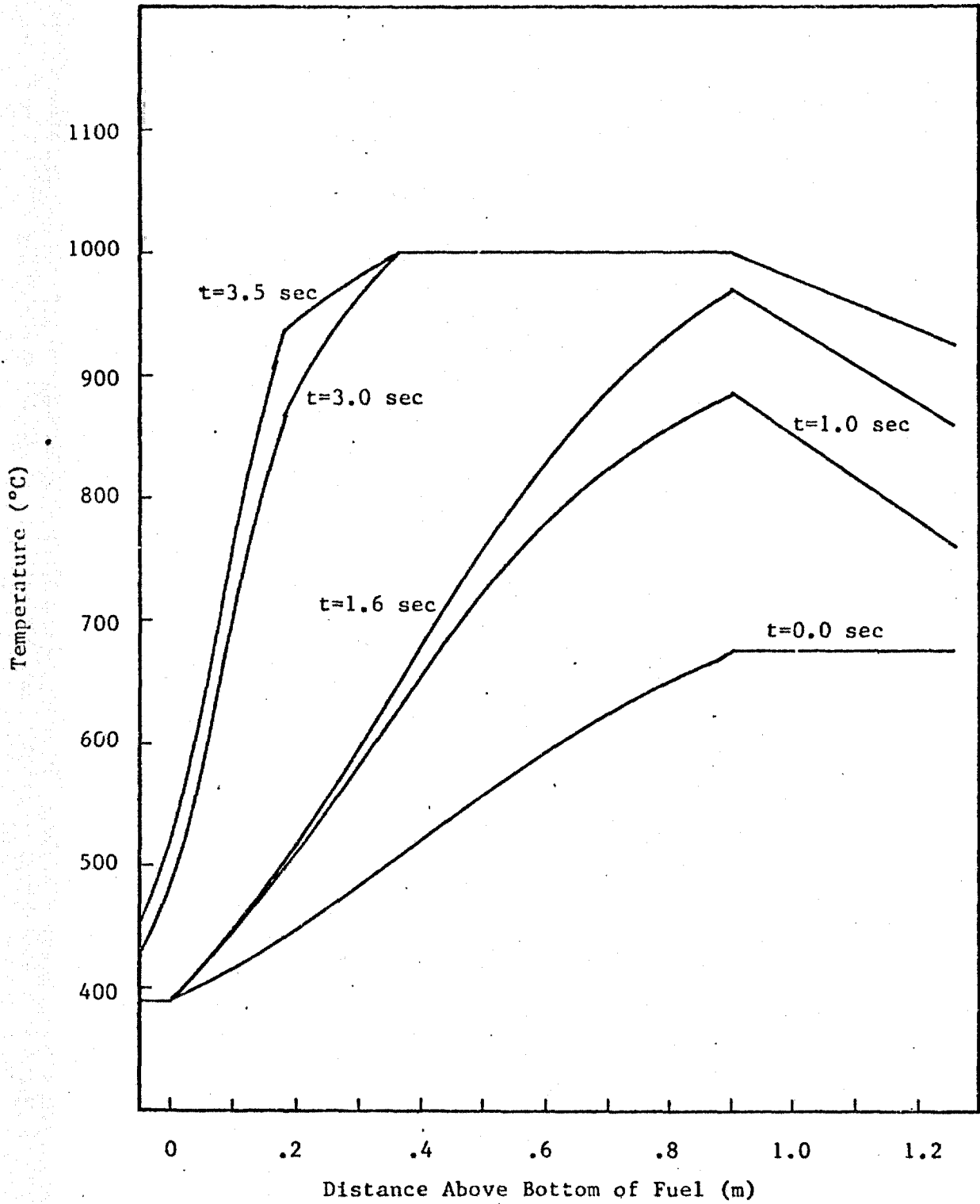


Figure 4.33: Axial Temperature Profile for 217-Pin Bundle Under Sequence 7b' Conditions

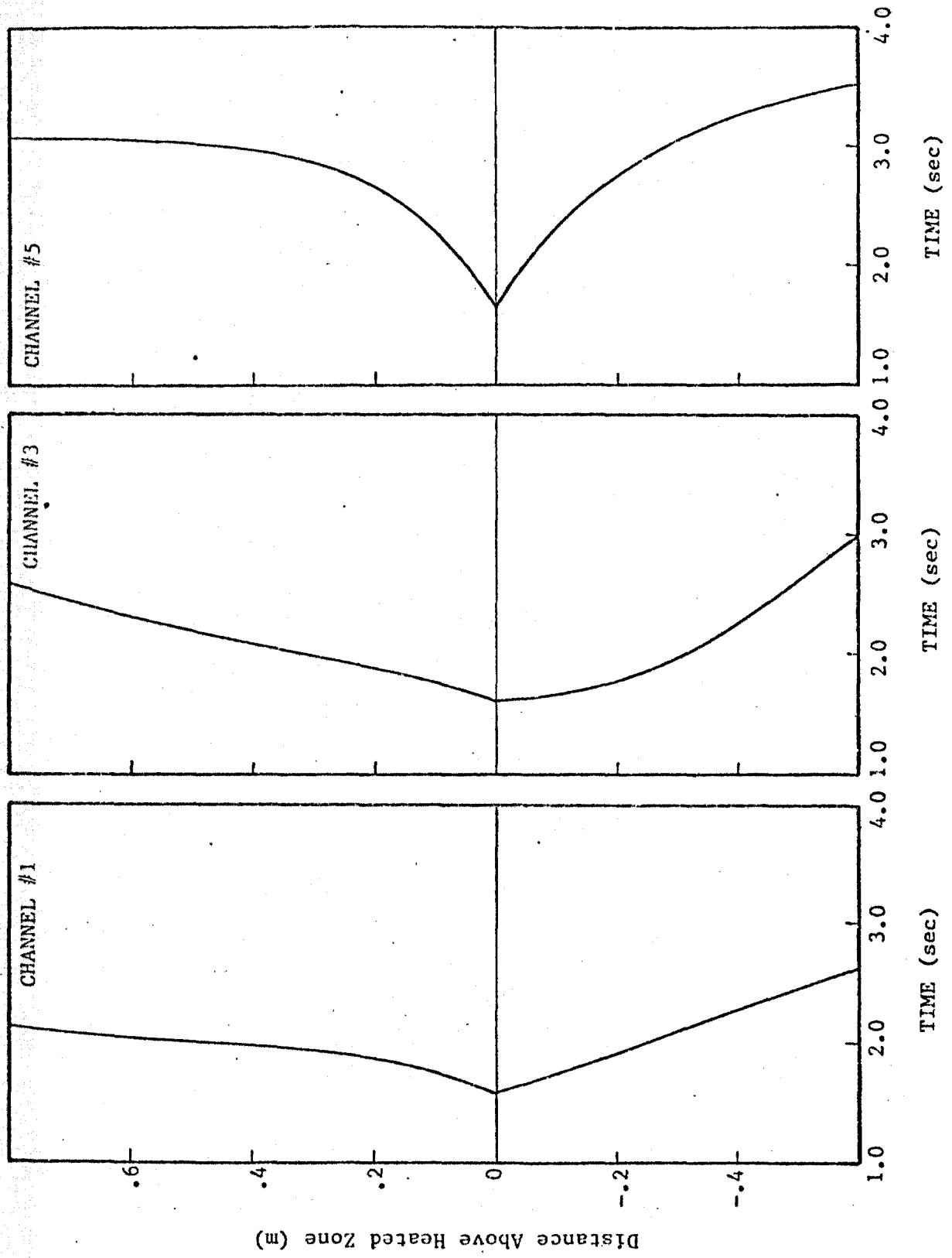


Figure 4.34: Void Maps for 217-Pin Bundle Under Sequence 7b' Conditions

### 4.3 The GR19 Experiment

As a final test of NATOF-2D the GR19 Experiment was simulated. This is a 19-pin, electrically heated, steady-state test performed on the CFNa loop, France. The test was analyzed with the BACCHUS code [35], and their results are presented here for comparison.

Table 4.6 presents the significant design data of the test. Table 4.7 shows the mass flow rate for the different tests performed, along with the measured maximum temperature and the NATOF-2D results.

Figures 4.32 through 4.34 show the axial temperature profile for these values of the mass flow rate. Figures 4.35 and 4.36 show the quality contours obtained for the values 0.265 and 0.260 kg/sec of the mass flow rate, along with the results of the BACCHUS code.

One interesting feature encountered in this simulation was a stable oscillation of the void fraction for the mass flow rate around the value .320 kg/sec, with the void fraction ranging from 10 to 50%, indicating the presence of a slug flow.

TABLE 4.6

Design Data for the GR19 Experiment

Number of Pins	19
Clad OD (m)	$8.65 \times 10^{-3}$
Heated Length (m)	0.6
Downstream Unheated Length (m)	0.494
Upstream Unheated Length (m)	0.12
Wire Wrap OD (m)	$1.28 \times 10^{-3}$
Flat to Flat (m)	$4.58 \times 10^{-2}$
Inlet Temperature (°C)	400
Saturation Temperature at the Top of Heated Zone (°C)	920
Power (kw)	170 (axially uniform)

TABLE 4.7

Mass Flow Rate  
And Temperatures for the GR19 Experiment

<u>FLOW</u> <u>(kg/sec)</u>	<u>T<sub>max</sub>(°C)</u> <u>(MEASURED)</u>	<u>T<sub>max</sub>(°C)</u> <u>(NATOF-2D)</u>
.606	693	694
.476	766	768
.405	825	827
.350	890	892
.329	918	920(Boiling)
.311	923	921
.293	926	921
.277	926	922
.265	926	925
.260	944	927

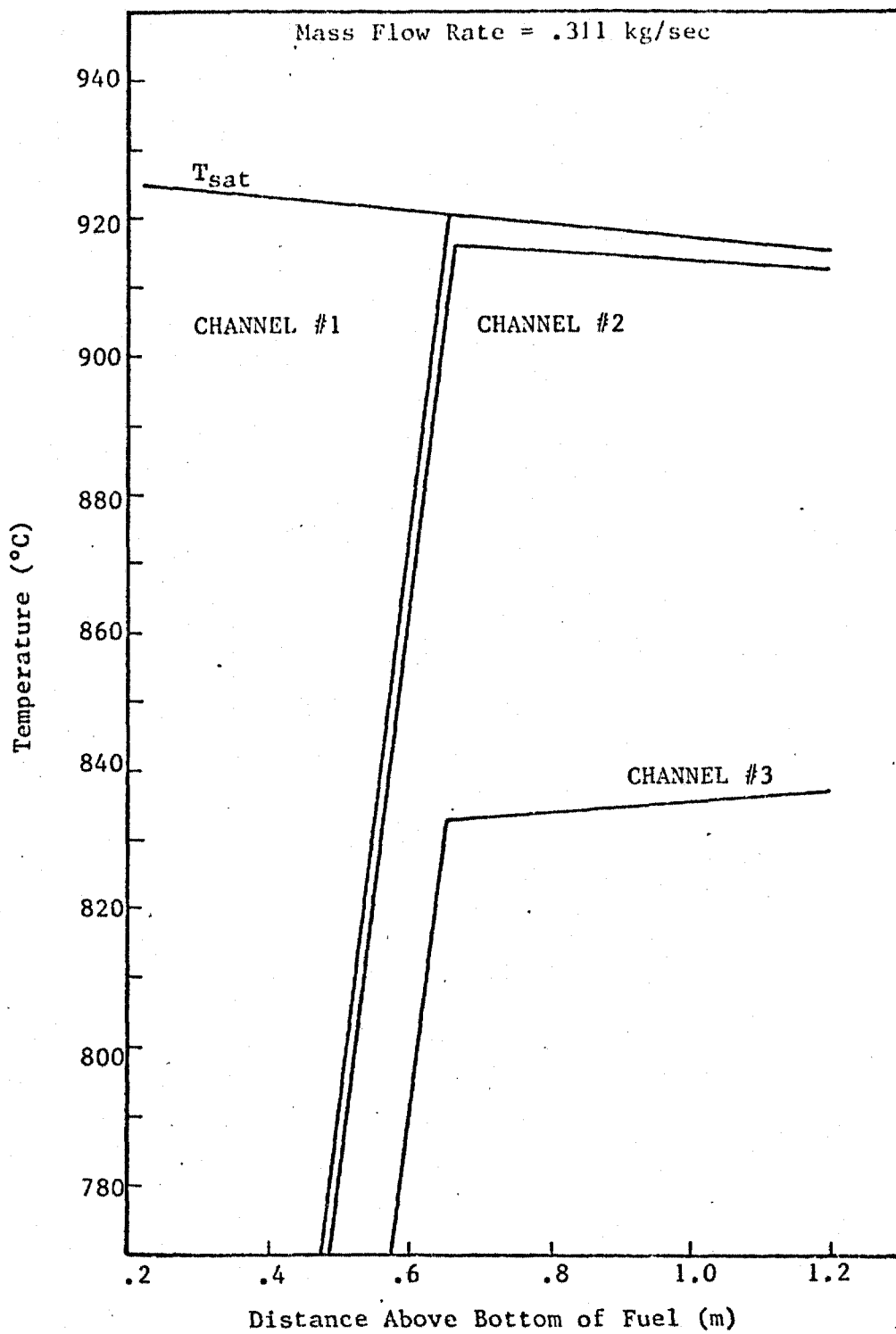


Figure 4.35: GR19: Temperature Profiles  
• For .311 kg/sec Mass Flow Rate

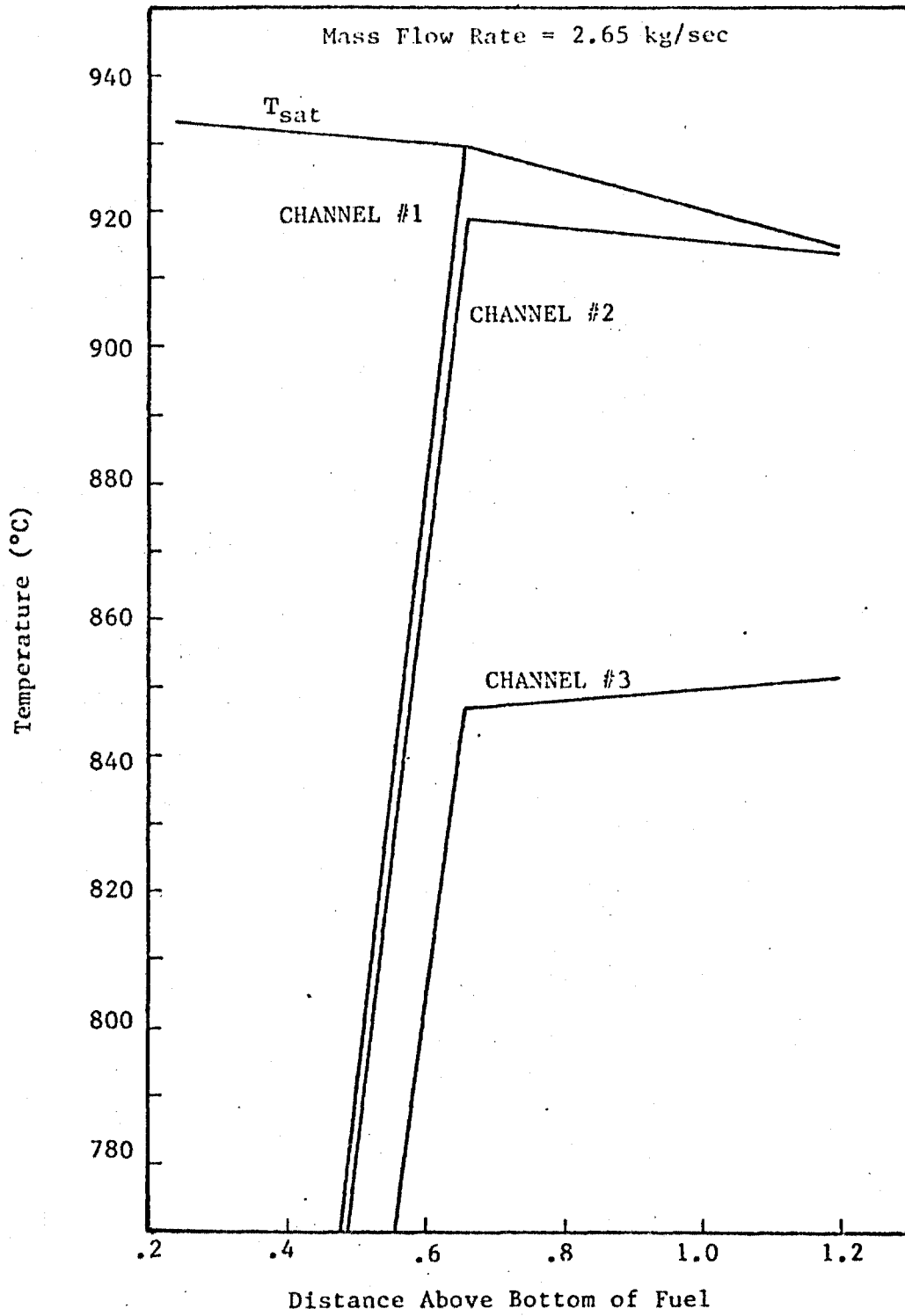


Figure 4.36: GR19: Temperature Profiles  
For .265 kg/sec Mass Flow Rate

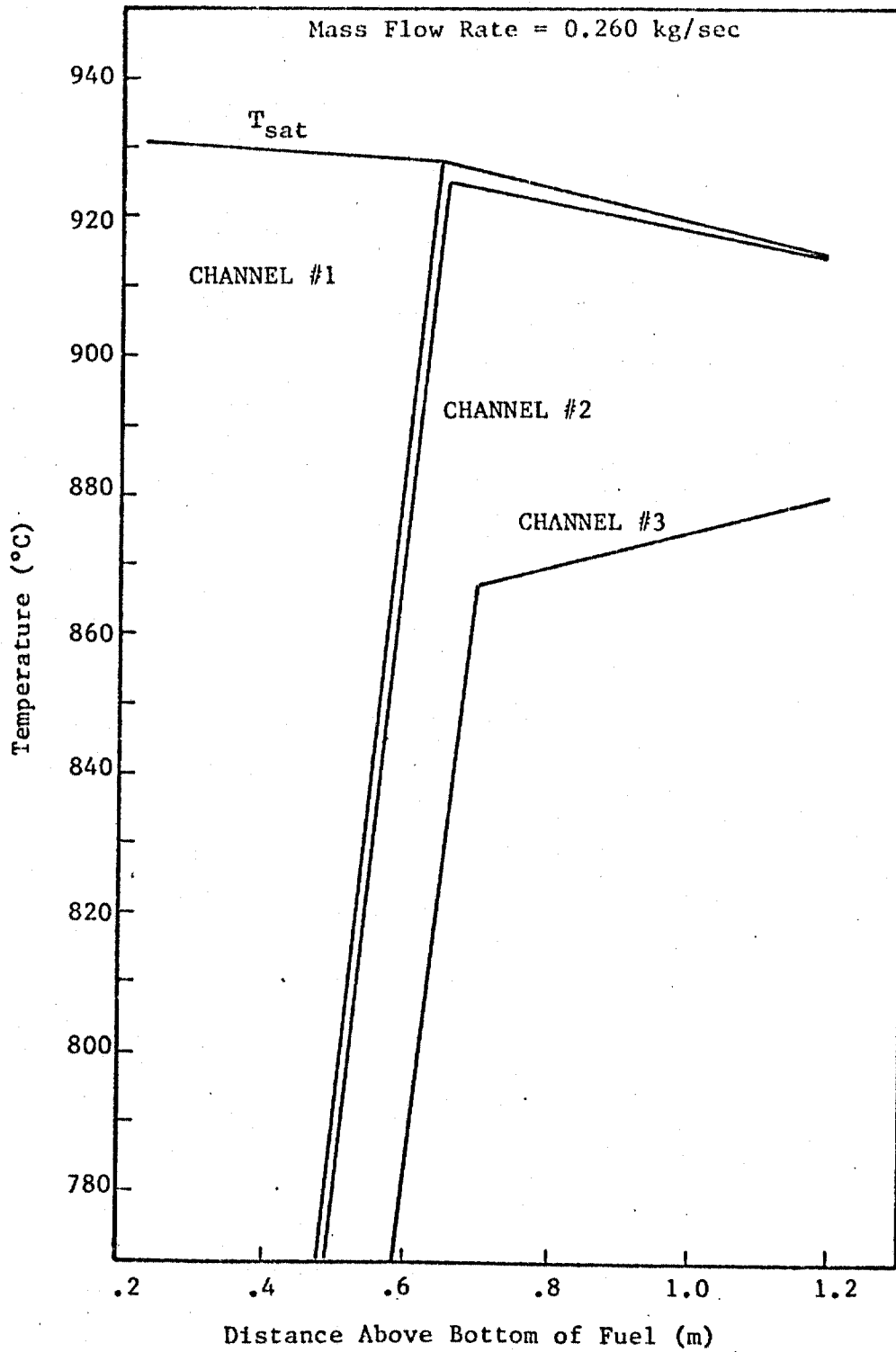


Figure 4.37: GR19: Temperature Profiles  
For .260 kg/sec Mass Flow Rate

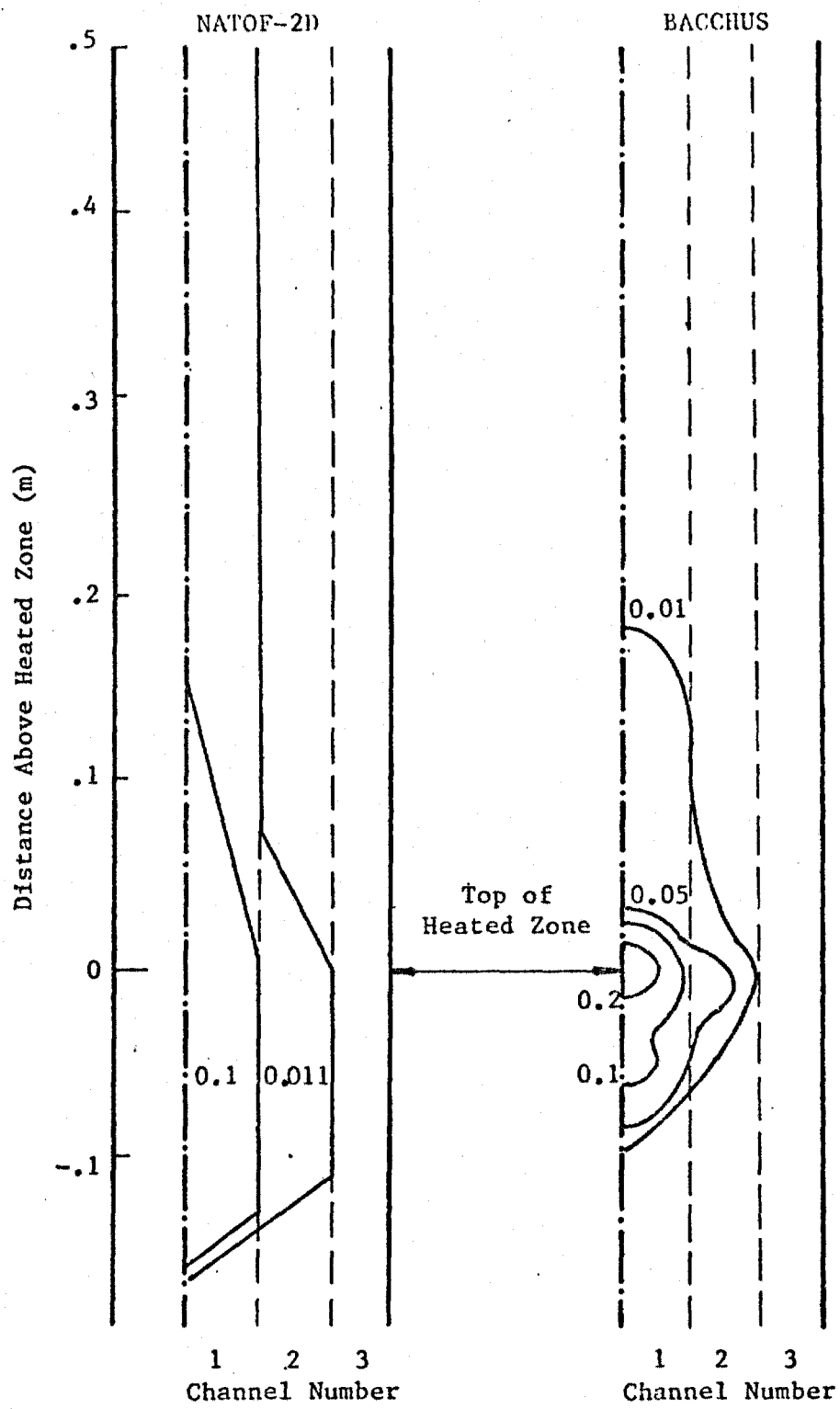


Figure 4.38

GR19 Quality Contours for 0.265 kg/sec Mass Flow Rate

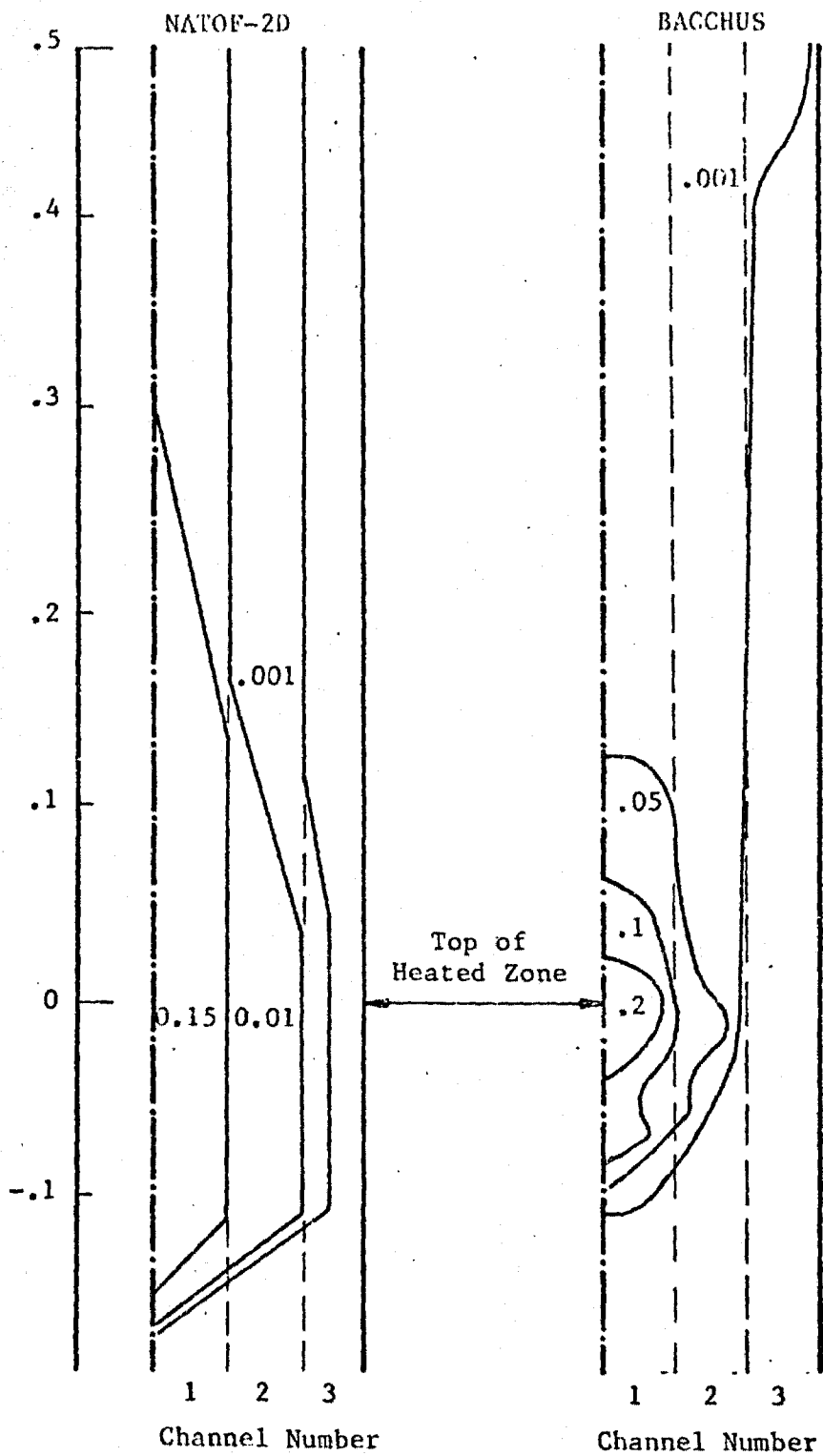


Figure 4.39

GR19: Quality Contours for 0.260 kg/sec Mass Flow Rate

## CHAPTER 5

CONCLUSIONS AND RECOMMENDATIONS5.1 Conclusion

A two dimensional computer code for the simulation of sodium boiling transients was developed using the two fluid model of conservation equations. A semi-implicit numerical differencing scheme, capable of handling the problems associated with the ill-posedness implied by the complex characteristic roots of the two fluid model was used, which took advantage of the dumping effect of the exchange terms. The stability of the method was demonstrated theoretically in Section 2.5 and also by the practical results obtained with the model, shown in Chapter 4. The stability of the model imposes an upper limit on the time step size, which is related to the mesh spacing the phase velocity by the expression

$$\Delta t < \max [\Delta z/\mu_2, \Delta r/\mu_r]$$

Of particular interest in the development of the model was the identification of the numerical problem used by the strong disparity between the axial and radial dimensions of fuel assemblies used in the current design of Liquid Metal Fast Breeder Reactors. A solution to this problem was found, which used the particular geometry of fuel assemblies to its advantage, reducing drastically the computation time.

Most of the constitutive equations incorporated in the model were

obtained through previous work. In general, adequate models were found for most equations, but for a few of them no satisfactory correlations could be produced. These models involve areas of the sodium technology not yet fully understood, and a substantial effort of development must be done in these areas. These models are identified and discussed in the recommendations of this work.

The models and methods of this work were incorporated into the computer program called NATOF-2D. With this program three series of experiments were simulated in order to demonstrate the model capabilities. The results of this simulation, which were presented in Chapter 4 showed good agreement with the experimental results obtained in the tests. One important capability demonstrated in these simulations was the ability of the model to represent the most severe boiling conditions, including flow reversal.

## 5.2 Recommendations

A word of caution must be said to the eventual users of NATOF-2D. The purpose of this work was to develop a numerical framework capable of solving the set of conservation equations of fluid flow under severe conditions of transient sodium boiling. In this way, most of the effort put into the work was dedicated to developing and organizing the numerical methods and models for solving this set of equations.

Of course the system of equations of fluid flow is not closed unless the constitutive relations describing the interaction of the fluids with the structural components and with themselves is provided, and a

set of constitutive equations were incorporated into NATOF-2D.

Some judgment was exercised in order to select constitutive equations representative of the sodium behavior, especially those characterizing the explosive volume change associated with sodium boiling at low pressure. This part of the code development was treated as complimentary to the numerical model construction. Therefore, the constitutive models may not be as realistic as the correct representation of sodium boiling in LMFBR fuel assemblies would require, and the overall results of simulations with NATOF-2D may be improved by the eventual improvement of some of the constitutive models incorporated in the code. Thus, this word of caution.

The relatively superficial treatment of the constitutive models is not incidental. Only recently did the interest in LMFBR safety reach the point where extensive investigation of sodium boiling became justified, and a substantial amount of research is yet to be done. Therefore, the present status of knowledge of the physical phenomena associated with sodium boiling does not lead immediately to significantly accurate models of the constitutive equations involved in sodium boiling. The task of developing these models is not a simple one, requiring a considerable effort in theoretical analysis and experimental work, well beyond the scope of a one person thesis.

But if NATOF-2D cannot claim to be a complete analytical model for sodium boiling simulation, because of the uncertainties contained in the constitutive models, it is an invaluable tool for the development of these models, where they can be implemented and tested against experimental results.

One of the most important benefits which NATOF-2D can provide to the development of sodium boiling is to identify, by the execution of sensitivity analysis, those constitutive models which affect most of the overall results, thus directing the research effort of sodium boiling to the directions which will lead to more fruitful results.

From the experience we had with NATOF-2D calculations, by far the most important model affecting the end results of sodium boiling simulation is the one for the interphase mass exchange rate (which unfortunately is the one that showed the widest disagreement between authors). Therefore, we recommend as a first step in the continuation of the work presented here that a substantial effort be made in developing a dependable model for the interphase mass exchange rate.

Of the same magnitude in importance is the two phase heat transfer coefficients. Here again the presently available models are few and incomplete. Thus a theoretical and experimental work in this area is recommended, in order to acquire a thorough understanding of the sodium boiling curve.

Another area which could be the object of future investigation is the one related to the interphase heat transfer. Although the direct effect of this exchange term on the overall results is not very marked, the relatively simple model incorporated in NATOF-2D could be replaced by a more refined one. The close relationship between this exchange term and the two previously mentioned would make this model a natural by product of the development of the above-mentioned ones.

## REFERENCES

1. Griffith, J. D., "Safety Considerations in Commercial Fast Breeder Reactor Plant Design," MIT summer course — Fast Breeder Reactor Safety, July 1977
2. Hinkle, W. D., LMFBR "Safety and Sodium Boiling, A State of the Art Report," draft, MIT, December 1977
3. Chawla, T. C., and Fauske, H. K., "On the Incoherence in Subassembly Voiding in FTR and its Possible Effects on the Loss of Flow Accident Sequence," Trans Am Nucl Soc 17, p285, November 1973
4. Rowe, D.S., COBRA III-C: "A Digital Computer Program for Steady-State and Transient Thermo-hydraulic Analysis for Rod Bundle Nuclear Fuel Elements," BNWL 1695, March 1973
5. Bohl, W.R., et al, "An Analysis of Transient Undercooling and Transient Overpower Accidents without Scram in the Clinch River Breeder Reactor," ANL/RAS 75-29, July 1975
6. Miao, C., and Theofanous, "A Numerical Simulation of the Two-Dimensional Boiling (Voiding) in LMFBR Fuel Subassemblies," Purdue University
7. Sha, W.T., et al, COMMIX1: "A Three Dimensional Transient Single-Phase Component Computer Program for Thermal Hydraulic Analysis," MIREG-CR 0415, ANL-77-96, September 1978
8. Grand, D., and Basque, G., "Two-Dimensional Calculation of Sodium Boiling in Sub-Assemblies," Service des Transferts Thermique, Centre d'Etude Nucleaires de Grenoble
9. Stewart, H.B., "Fractional Step Methods for Thermo-hydraulic Calculation," Brookhaven National Laboratory, March 1980
10. Shih, T.A., "The SOBOIL Program, A Transient, Multichannel Two Phase Flow Model for Analysis of Sodium Boiling in LMFBR Fuel Assemblies," Technical Note ST-TN-79008, March 1979
11. Hinkle, W.D., et al, "MIT Sodium Boiling Project FY 1979 Interim Report," draft, 1979

12. Shah, et al, "A Numerical Procedure for Calculating Steady/Unsteady Single-Phase/Two-Phase Three-Dimensional Fluid Flow with Heat Transfer," ANL-CT-79-31, June 1979
13. Ishii, M., "One-Dimensional Drift-Flux Model and Constitutive Equations for Relative Motion Between Phases in Various Two-Phase Flow Regimes," ANL 77-47, October 1977
14. Grolmes, M.A., and Henry, R.E., "Heat Transfer in Nuclear Power Reactors, Part II: Safety of Liquid Metal Cooled Fast Breeder Reactors," Argonne National Laboratory
15. Weber, M., et al, "Reactor Development Program," Progress Report ANL-RDP-78, December 1978.
16. Carter, J.C., et al, SASIA, "A Computer Code for the Analysis of Fast Reactor Power and Flow Transients," ANL7607, October 1970
17. Dunn, F.E., "The SAS3A LMFBR Accident Analysis Computer Code," ANL/RAS 75-17, 1975
18. The Separate Flow Model of Two Phase Flow, EPRI NP275, December 1976
19. Agrawal, A.K., et al, "Simulation of Transients in Liquid Metal Fast Breeder Reactor Systems," Nuclear Science and Engineering, Vol 64, 480-491, 1977
20. Boure, J.A., and Latrobe, A., "On Well-Posedness of Two Phase Flow Problems," 16th National Heat Transfer Conference, August 1976
21. Reed, Wm. H., and Stewart, H.B., "THERMIT, A Computer Program For Three-Dimensional Thermal-Hydraulic Analysis of Light Water Reactor Cores, MIT, 1978.
22. Rivard, W.C. and Torrey, M.D., "Numerical Calculation of Flashing from Long Pipers Using a Two Field Model," LA6104-MS, Los Alamos, November 1975
23. Chen, J.C., "A Proposed Mechanism and Method of Correlation for Convective Boiling Heat Transfer with Liquid Metals," BNL 7319, August 1973
24. Chao, B.T., Sha, W.T., and Soo, S.L., "On Inertial Coupling in Dynamic Equations of Components in a Mixture," Int J Multiphase Flow, Vol 4, pp219-223, 1978

25. Fabric, S., "Computer Codes in Water Reactor Safety: Problems in Modeling of Loss-of-Coolant Accident," I Mech E Conference, Manchester, 13-15 September 1977
26. Chawla, T.C., and Ishii, M., "Equations of Motion for Two-Phase Flow in a Pin Bundle of a Nuclear Reactor," J Heat Mass Transfer, Vol 21, pp1057-1068, 1978
27. Ramshaw, J.D., and Trapp, J.A., "Characteristics, Stability and Short Wavelength Phenomena in Two-Phase Flow Equations Systems," Nuclear Science and Engineering, Vol 66, pp93-102, 1978
28. Murray, S.E., and Smith, L.L., "Two Dimensional Sodium Voiding Analysis with SIMMER-I," LA-NUREG-6342-PR, June 1977
29. Lyczkowsky, R.W., and Solbrig, C.W., "Calculation of the Governing Equations for Seriated Unequal Velocity, Equal Temperature Two-Phase Flow, Nat Heat Transfer Conference 1977
30. Jones, O.C., Jr., and Pradip, S., "Non-Equilibrium Aspects of Water Reactor Safety," Brookhaven National Laboratory
31. Nigmatulin, R.I., "Equations of Hydromechanics and Compression Shock in Two Velocity and Two Temperature Continuum with Phase Transformation," Fluid Dynamics, Vol 2, No 5, 1967
32. Rohsenow, W.M., and Sukhatme, S.P., "Condensation," Massachusetts Institute of Technology
33. Brinkmann, K.J. and deVries, J.E., "Survey of Local Boiling Investigations in Sodium at ECN-Petten," Netherlands Energy Research Foundation ECN
34. Garrison, P.W., "Superheat Simulation Requirements for the Next Generation of LMFBR Codes," Oak Ridge National Laboratory, March 1979
35. Basque, G., Grand, D., and Menant, B., "Theoretical Analysis and Experimental Evidence of Three Types of Thermohydraulic Incoherence in Undisturbed Cluster Geometry," Karlsruhe, 1979
36. Baker, A.N., et al, "SLSF W-1 Experiment Test Predictions," GEFR 00047-9(L), December 1977
37. Knight, D.D., "SLSF W-1 LODI Experiments Preliminary Evaluation Data," ST-TN-80015, October 1979
38. Henderson, J.M., "Sodium Loop Safety Facility Test Plan HEDL W-1 SLSF Experiment," Hanford Engineering Development Laboratory, September 1978

39. Thompson, D.H., et al, "SLSF In-Reactor Experiment P3A," Interim Post Test Report, Argonne National Laboratory, November 1977
40. Kraft, T.E., et al, "Simulations of an Unprotected Loss-of-Flow Accident with a 37-Pin Bundle in the Sodium Loop Safety Facility," Argonne National Laboratory
41. Collier, J.G., Convective Boiling and Condensation, McGraw-Hill, United Kingdom, 1972
42. Clark, M., Jr., and Hansen, K.F., Numerical Methods of Reactor Analysis, Academic Press, New York, 1964
43. Richtmeyer, R.D., and Morton, K.W., Differential Methods for Initial Value Problems, Interscience, New York, 1967
44. Wallis, G.B., One Dimensional Two Phase Flow, McGraw-Hill, New York, 1969
45. Courant, R., and Hilbert, D., Methods of Mathematical Physics, Interscience, New York 1962
46. Golden, G.H., and Tokar, J.V., Thermophysical Properties of Sodium, ANL 7323, August 1967
47. Van Wylen, G.J., and Sonntag, R.E., Fundamentals of Classical Thermodynamics, John Wiley & Sons, New York, 1973
48. Kays, W., and London, A.L., Compact Heat Exchanges, McGraw-Hill, New York, 1964
49. Gunter, A.Y., and Shaw, W.A., A General Correlation of Friction Factors for Various Types of Surfaces in Crossflow, ASME Transactions, 67, pp643-660, 1945
50. Autruffe, M.A., Theoretical Study of Thermohydraulic Phenomena for LMFBR Accident Analysis, MIT thesis, September 1978
51. Stewart, H.B., "Stability of Two Phase Flow Calculations Using Two Fluid Model," Journal of Computational Physics, Vol 33, No 2, November 1979
52. Kaiser, A., and Pepler, W., "Sodium Boiling Experiments in an Annular Test Section under Flow Rundown Conditions," KFK 2389, March 1977
53. El Wakil, M.M., "Nuclear Heat Transport," International Textbook Company, 1971

54. Fink, J.K., and Leibowitz, L., "Thermophysical Properties of Sodium," ANL-CEN-RSD-79-1, May 1979
55. Gantmacher, F.R., "The Theory of Matrices," Chelsea Publishing Company, 1977
56. Varga, R.S., "Matrix Iterative Analysis," Prentice Hall, 1962
57. Rohsenow, W.M., and Choi, H., "Heat, Mass and Momentum Transfer," Prentice Hall, 1961
58. Poter, M.C., and Foss, J.F., "Fluid Mechanics," Ronald Press, 1975
59. "CRC Handbook of Chemistry and Physics," 58th Edition, CRC Press, 1978
60. Thompson, D.H., et al. "SLSF In-Reactor Experiment P3A - Interim Posttest Report", ANL/RAS 77-48, November 1977

APPENDIX A - NATOF-2D INPUT DATA MANUAL

In this section the user supplied information necessary to operate NATOF-2D is presented. Before showing the description of the input cards, it is useful to review the array structure of the code. Figure A.1 shows an example of a full assembly and the corresponding cell arrangement in a r-z plane. Quantities appearing in this figure are:

NI = number of mesh cells in the axial direction. It includes two fictitious half-cells in the top and bottom of fuel assembly.

NJ = number of mesh cells in the radial direction.

All dimensioned variables appear in the program with only one index, therefore a single number identifies each cell in full assembly. The cells are numbered from bottom to top and radially from center to hex can.

Figure A.2 shows a cross section of the fuel assembly indicating the numbering of the fuel pins. Fuel pin rows are numbered from center to hex can, and the boundary between cells is indicated by the row number where this boundary lies.

Figure A.3 shows schematically the cell arrangement for the fuel pin heat conduction. The quantities describing this cell arrangement are:

NCF = number of mesh cells in fuel.

NCLD = number of mesh cells in clad.

NI	2xNI		NJxNI
NI-1	2xNI-1		NJxNI-1
		etc.	
3	NI+3		
2	NI+2		
1	NI+1		

Figure A1. Cell Arrangement in the R-Z Plane

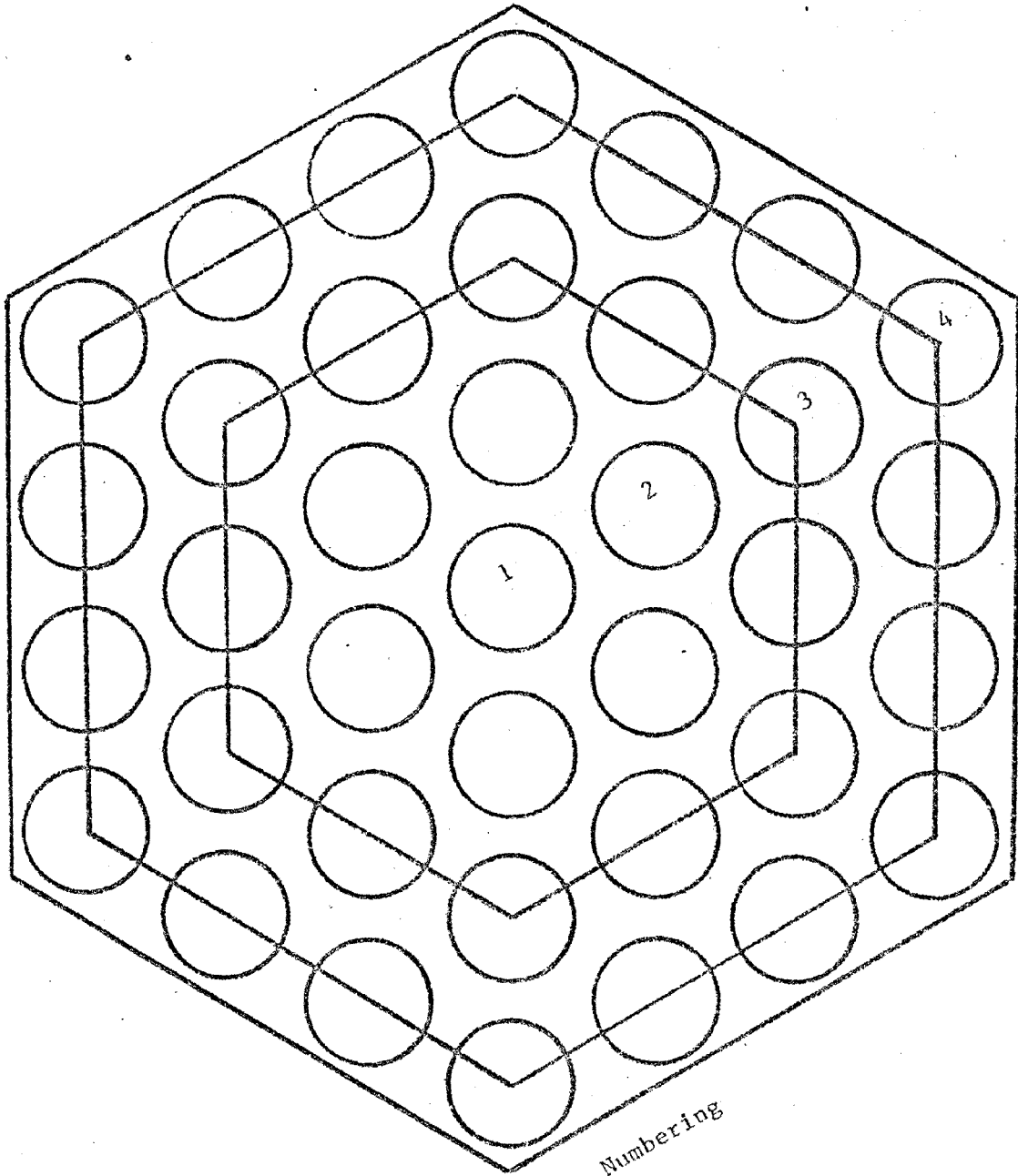


Figure A.2 Fuel Pin Numbering

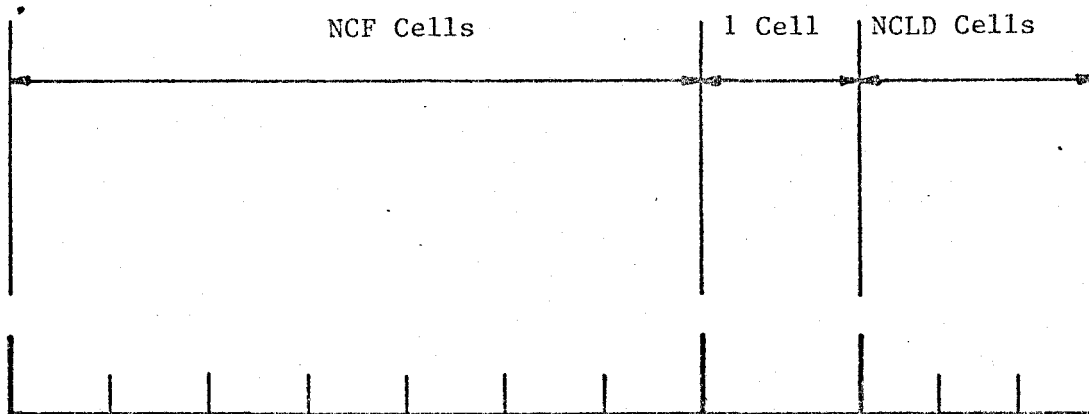


Figure A3. Cell Arrangement for Fuel Pin Heat Conduction.

A single cell is assumed by the code for the gap between fuel and clad.

Following is a presentation of the sequence of cards in the input data. Following the list of variables, in parenthesis, is the corresponding format for these variables.

1. General Description of the Problem

1st CARD: NI, NJ, NCF, NCLD (4I5)

NI = number of mesh cells in axial direction

NJ = number of mesh cells in radial direction

NCF = number of mesh cells in fuel

NCLD = number of mesh cells in clad

2nd CARD: NSET, TSET (I5, E15.4)

This card contains information which controls the printed output. the code will print NSET times the flow map, with a time interval TSET. This card can be repeated up to 49 times, so that the time interval between prints can be varied to reflect the desired degree of information at each time. Following these cards, a card containing only zeros in the position corresponding to NSET must be placed, to indicate the end of this subset.

3rd CARD: ITM, IGAUSS, DTMAX, EPS1, EPS2 (2I10, 3E15.9)

ITM = maximum number of iterations in the Newton iterative solution.

IGAUSS = maximum number of iterations in the pressure problem solution.

DTMAX = maximum value for the time step increment.

EPS1 = convergence criterion for the Newton iteration.

EPS2 = convergence criterion for the pressure problem.

EPS1 and EPS2 are criteria on the absolute value of the pressure. Their unit is N/m<sup>2</sup>.

## 2. Boundary Conditions

The next group of cards contains information governing the boundary conditions of the problem as a function of time. The simulation time is divided in up to 50 segments in which different functions can be prescribed for the boundary conditions. For a generic time segment L, the formulas used by the program for the boundary condition are:

$$X = (X_1(L) * DTIME + X_2(L)) * \exp(OMX(L) * DTIME) + X_3(L)$$

where:

$$DTIME = TIME - TB(L-1)$$

L = Index of current time segment

TB(L) = Time at the end of segment L

X<sub>1</sub>, X<sub>2</sub>, X<sub>3</sub>, OMX = Input parameters

and X stands for:

PNB = Pressure at the bottom of fuel assembly (N/m<sup>2</sup>)

PNT = Pressure at the top of fuel assembly (N/m<sup>2</sup>)

ALB = Void fraction at the inlet of fuel assembly.

TVB = Vapor temperature at inlet ( $^{\circ}\text{K}$ ).

TLB = Liquid temperature at inlet ( $^{\circ}\text{K}$ ).

HNW = Power density in fuel pins ( $\text{W}/\text{m}^3$ )

In order to save time, the code has an option to eliminate the exponential part in the formula to calculate the boundary condition. Thus, whenever the logical parameter LP is .TRUE., the boundary conditions are calculated as:

$$X = X_1(L) * \text{DTIME} + X_2(L)$$

1st CARD:	LP, TB	(L1, F15.5)
2nd CARD:	PNB1, PNB2, PNB3, OMP	(4E15.9)
3rd CARD:	PNT1, PNT2, PNT3, OMT	(4E15.9)
4th CARD:	ALB1, ALB2, ALB3, OMA	(4E15.9)
5th CARD:	TVB1, TVB2, TVB3, OMV	(4E15.9)
6th CARD:	TLB1, TLB2, TLB3, OML	(4E15.9)
7th CARD:	HNB1, HNB2, HNB3, OMH	(4E15.9)

This group of seven cards can be repeated for as much as the number of segments desired. To indicate the end of this subset, a card containing only a 'F' in the first position must be placed following the data.

### 3. Geometric Description of the Problem

1st CARD: NROW, PITCH; D, E (I5, 3E15.9)

NROW = Number of rows of fuel pins in fuel assembly.

PITCH = Distance between fuel pin centerlines (m).

D = Fuel pin diameter (m).

E = Minimum distance between fuel pin surface and hex can wall (m).

(see Figure A.2)

2nd CARD: N(J), J = 1, 20 (20I4)

N(J) is the row number where the boundary between cell J and cell J + 1 lies.

(see Figure A.2)

3rd CARD: LDATA, DZ(K) (L1, 5E15.9)

In this group of cards the axial mesh spacing DZ are written sequentially from 1 to NI, five per card. The logical parameter LDATA must have a .TRUE. value in each card where DZ is written. Following this group of cards, a card containing an 'F' in the first position must be placed to indicate the end of this set of data.

4th CARD: LDATA, CAN(K) (L1, 5E15.9)

The same arrangement of the previous group of cards.

CAN = Heat capacity of the hex can per unit area, for each axial mesh cell ( $J/m^2 \cdot K$ ). There must be one value for each axial mesh cell.

5th CARD: LDATA, SHAPE(K) (L1, 5E15.9)

The same arrangement as the previous group of cards.

SHAPE = Power density shape in fuel assembly. There must be one value of SHAPE for each mesh cell in fuel assembly.

6th CARD: LDATA, SPPD(K) (L1, 5E15.9)

The same arrangement as the previous group of cards.

SPPD = Spacer pressure drop. There must be one value of SPPD for each mesh cell in fuel assembly. The code will treat the spacer pressure drop as:

$$\Delta p = SPPD * \frac{\rho U^2}{2}$$

7th CARD: LDATA, PPP(K) (L1, 5E15.9)

The same arrangement as the previous group of cards.

PPP = Radial power profile inside fuel pin. There must be one value of PPP for each fuel pin mesh cell, including gap and clad (i.e., there is NCF + 1 + NCLD values).

The power density at each fuel pin mesh cell will be the product of the power density specified in the boundary conditions, multiplied by the value of SHAPE for the corresponding fuel assembly mesh cell, multiplied by the value of PPP for the corresponding fuel pin mesh cell.

8th CARD: AD, APU, DIL (3E15.9)

AD = Fraction of theoretical density of fuel.

APU = Fraction of plutonium in fuel.

DIL = Fraction of helium in gap composition.

\*9th CARD: LPLNM(I), I = 1, NI (39I2)

LPLNM is an integer which indicates the axial composition of fuel pin. LPLNM = 0 indicates gas composition (for upper plenum). LPLNM = 1 indicates mixed oxide U,PuO<sub>2</sub>. There must be one value of LPLNM for each axial node.

10th CARD: RADR, THC, THG (3E15.9)

RADR = Fuel pin outside radius (m).

THC = Clad thickness (m).

THG = Gap thickness (m).

#### 4. Initial Conditions

1st CARD: LSS, TINIT (L1, E15.9)

LSS is a logical parameter to indicate steady-state or transient problem.

LSS = .FALSE. indicates transient problem.

LSS = .TRUE. indicates steady-state problem.

In case LSS is .TRUE., the remaining initial condition input data resume to the next card:

2nd CARD: PIN, POUT, TIN, TAV (4E15.9)

PIN = Pressure at fuel assembly inlet ( $N/m^2$ )

POUT = Pressure at fuel assembly outlet ( $N/m^2$ )

TIN = Inlet liquid temperature ( $^{\circ}K$ )

TAV = An estimate of the average temperature in fuel assembly ( $^{\circ}K$ )

In case LSS = .FALSE., the next cards follow:

2nd CARD: KO, TV, TL, P, ALFA (I5, 4E15.9)

3rd CARD: KO, UVZ, ULZ, UVR, ULR (I5, 4E15.9)

KO is the cell number. It appears in both cards to put a check in the input data. Each pair of cards correspond to the same mesh cell. The group is to be repeated for as many as the number of mesh cells.

TV = Vapor temperature ( $^{\circ}K$ )

TL = Liquid temperature ( $^{\circ}K$ )

P = Pressure

ALFA = Void fraction

UVZ = Axial vapor velocity (m/sec)

UVR = Radial vapor velocity (m/sec)

ULR = Radial liquid velocity (m/sec)

4th CARD: LDATA, TR(K) (L1, 5E15.9)

The same arrangement as the group of cards for DZ.

TR = Fuel pin temperature ( $^{\circ}$ K).

This array must contain one value for each fuel pin mesh cell. The values of TR are ordered as:

TR(1) = Fuel centerline temperature at cell number 1.

TR(NCF + 1 + NCLD) = Surface clad temperature at cell number 1.

TR(NCF + 1 + NCLD + 1) = Fuel centerline temperature at cell number 2.

etc.

5th CARD: LDATA, TCAN(K) (L1, 5E15.9)

The same arrangement as the previous group of cards.

TCAN = Hex can initial temperature ( $^{\circ}$ K).

There must be one value of TCAN for each axial node.

## APPENDIX B

NATOF - 2D Programming Information

When NATOF-2D was programmed, it was recognized that the field of sodium boiling is presently the subject of a large effort of research, and therefore it can be expected that in the future this research will produce better correlations for the constitutive laws governing the sodium two-phase flow. In order to make changes in the program as easy as possible, NATOF-2D was programmed with its subroutines in a modular structure, particularly the parts of the program dealing with the constitutive laws.

In this way, the programmer working on modification of one particular subroutine does not have to worry about the rest of the program, provided the expressions introduced in that subroutine meet the requirements of consistency of the derivatives with respect to new time variables, which were discussed in chapter 2.

Following is a description of NATOF-2D subroutines, their functions and structure. The reader is referred to figure B1, which shows the structure of NATOF-2D.

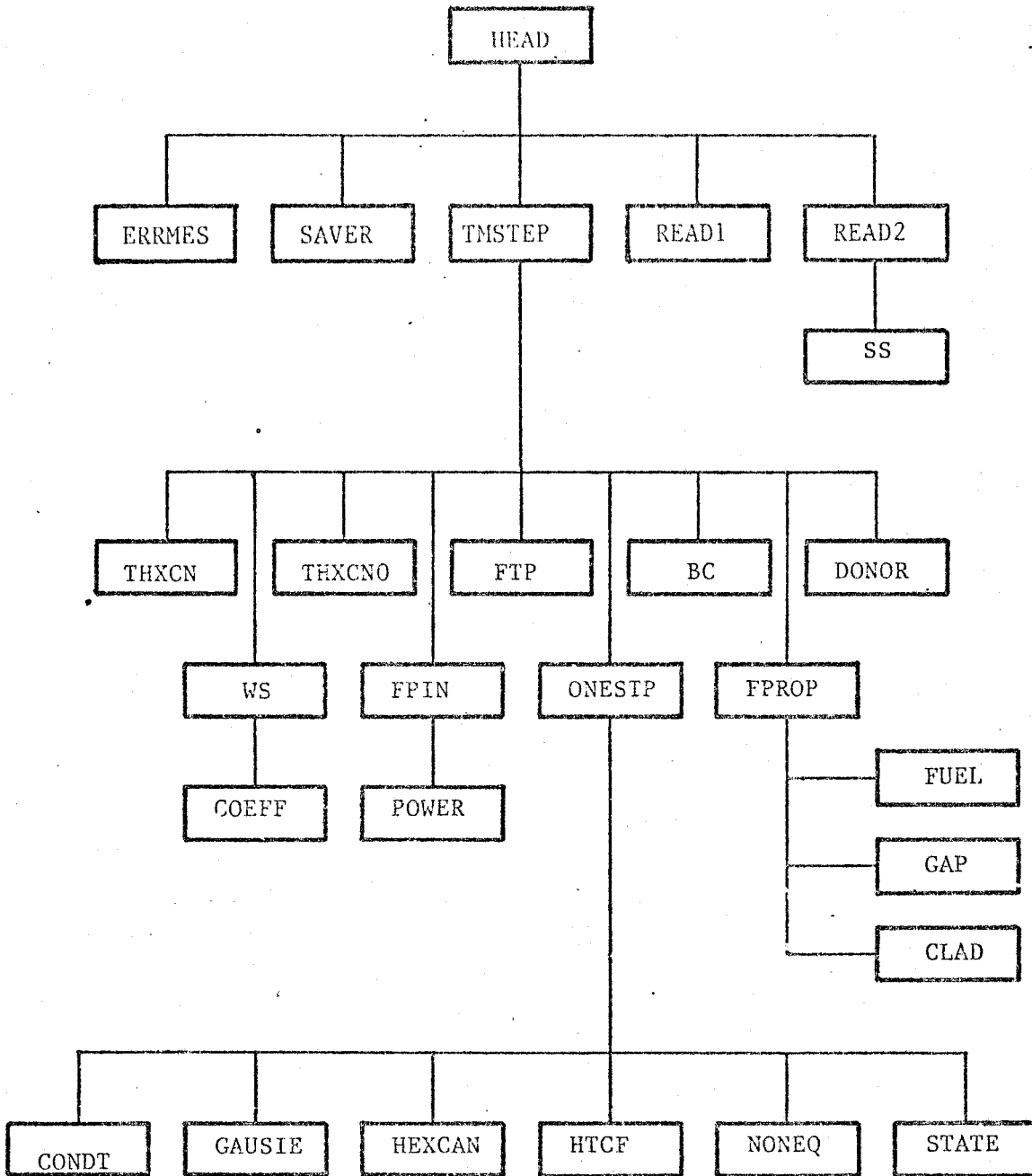


Figure B1. NATOF-2D Subroutine Structure

Main:

The main program's only function is to allocate memory storage space for the dimensioned arrays and transfer the control of the program to subroutine HEAD.

All arrays whose dimensions are a function of the number of mesh cells are placed within a single array ORBI. Individual arrays are located by pointers which determine the first element of each array. These pointers are grouped into the integer array M, and the correlation of the pointer to the variable is as following:

M(1)	=	P	=	New time, pressure, cell centered
M(2)	=	PO	=	Old time, pressure, cell centered
M(3)	=	TV	=	Vapor temperature, new time, cell centered
M(4)	=	TVO	=	Vapor temperature, old time, cell centered
M(5)	=	TL	=	Liquid temperature, new time, cell centered
M(6)	=	TLO	=	Liquid temperature, old time, cell centered
M(7)	=	ALFAN	=	Void fraction, new time, cell centered
M(8)	=	ALFAO	=	Void fraction, old time, cell centered
M(9)	=	ALFAZ	=	Void fraction, axial face centered
M(10)	=	ALFAR	=	Void fraction radial face centered
M(11)	=	RHOV	=	Vapor density, cell centered
M(12)	=	RHOL	=	Liquid density, cell centered

M(13) = RHOVZ = Vapor density, axial face centered  
M(14) = RHOLZ = Liquid density, axial face centered  
M(15) = RHOVR = Vapor density, radial face centered  
M(16) = RHOLR = Liquid density, radial face centered  
M(17) = EV = Vapor internal energy, cell centered  
M(18) = EL = Liquid internal energy, cell centered  
M(19) = EVZ = Vapor internal energy, axial face centered  
M(20) = ELZ = Liquid internal energy, axial face centered  
M(21) = EVR = Vapor internal energy, radial face centered  
M(22) = ELR = Liquid internal energy, radial face centered  
M(23) = UVZN = Axial vapor velocity, new time, axial face  
centered  
M(24) = ULZN = Axial liquid velocity, new time, axial face  
centered  
M(25) = UVRN = Radial vapor velocity, new time, radial face  
centered  
M(26) = ULRN = Radial liquid velocity, new time, radial face  
centered  
M(27) = UVZO = Axial vapor velocity, old time, axial face  
centered  
M(28) = ULZO = Axial liquid velocity, old time, axial face  
centered  
M(29) = UVRO = Radial vapor velocity, old time, radial face  
centered  
M(30) = ULRO = Radial liquid velocity, old time, radial face  
centered  
M(31) = UVRZ = Radial vapor velocity, axial face centered  
M(32) = ULRZ = Radial liquid velocity, axial face centered

M(33)	=	UVZR	=	Axial vapor velocity, radial face centered
M(34)	=	ULZR	=	Axial liquid velocity, radial race centered
M(35) to M(62)	=		=	Implicit terms for the conservation equations
M(63)	=	DH	=	Axial flow hydraulic diameter
M(64)	=	DHR	=	Radial flow hydraulic diameter
M(65)	=	DV	=	Fuel pin specific surface area
M(66)	=	QSI	=	Maximum-to-average radial velocity coefficient
M(67)	=	TS	=	Saturation temperature, new time
M(68)	=	TW	=	Fuel pin wall temperature, new time
M(69)	=	DTW	=	Increment in heat transfer for unit increment in TW
M(70)	=	HCONV	=	Vapor heat transfer coefficient
M(71)	=	HCONL	=	Macroscopic liquid heat transfer coefficient
M(72)	=	HNB	=	Microscopic liquid heat transfer coefficient
M(73) to M(79)	=		=	Coefficients for the pressure problem
M(80)	=	TR	=	Fuel pin temperature
M(81)	=	DTR	=	Auxiliary array for fuel pin heat conduction
M(82)	=	TWO	=	Fuel pin wall temperature, old time
M(83) to M(89)	=		=	Auxiliary arrays
M(90)	=	SPPD	=	Localized pressure drop coefficient
M(91)	=	TCAN	=	Hex can temperature

The storage space required by the array ORBI is given in double precision storage word by the formula:

$$[135 + 2(NCF + NCLD)]NI.NJ$$

- HEAD: — Defines the pointers of array ORBI  
— Controls the duration of the run  
— Controls the printouts
- READ 1: — Reads arrays' dimensions
- READ 2: — Reads all other information  
— Writes in FILE07 the input data for a restart  
— Calculate parameters which will remain constant throughout the problem
- SS: — Performs an initial guess for the steady-state problem
- TMSTEP: — Advances one time step  
— Controls convergence of the Newton iteration  
— Controls time step size. The time step is always kept below the convective limit. If an instability occurs during the run, such as non-convergence of the iterative procedures or a variable outside range of validity, TMSTEP reduces the time step size by a factor of ten and the run is resumed. If the difficulty is removed, the time step will be increased slowly towards the convective limit again. If after three time step reductions the instability still persists, an error message will be printed and the execution terminated.
- DONOR: — Transfers all centered quantities to face centered positions  
— Calculates explicit terms in momentum equation
- WS: — Calculates explicit terms for mass and energy equations

- ONESTP: — Performs one step of Newton iteration  
— Calculates new values of implicit variables  
— Checks variables against range of validity
- COEFF: — Calculates momentum exchange coefficients
- BC: — Calculates boundary conditions as a function of time
- HTCF: — Calculates heat transfer coefficients
- STATE: — Calculates sodium thermodynamic properties and its derivatives. The code stability imposes two requirements on the expressions for the sodium functions of state: the expressions for the densities must account for the pressure dependence which corresponds to a real, positive, finite sonic speed.  
The expressions for the property derivatives with respect to new time variables must be the analytic or numerical derivative of the expressions of the properties (but not approximated expressions).
- NONEQ: — Calculates the mass and energy exchange rates and its derivatives. The same requirement applied to the derivatives of the properties in STATE also applies here.
- CONDT: — Calculates the heat transfer between fluid and fuel pin and its derivatives. The requirement concerning the derivatives described above also applies here.
- HEXCAN: — Calculates the heat transfer between fluid and hexcan walls, and its derivatives. The requirement concerning the derivatives described above also applies here.

FPROP: — Finds the fuel pin transport properties

FUEL: — Transport properties of fuel

GAP: — Transport properties of gap

CLAD: — Transport properties of clad

FPIN: — Solves first part of heat conduction in fuel pin

FTP: — Solves second part of heat conduction in fuel pin

THXCN: — Solves the first part of hexcan heat conduction

THXCNO: — Solves the second part of hexcan heat conduction

POWER: — Calculates the power density as a function of time

GAUSIE: — Solves the pressure problem

ERRMES: — Prints error messages

SAVER: — Saves fluid flow variables at the end of run for  
eventual restart

Functions

- CONDL — Liquid thermal conductivity as function of temperature
- CONDV — Vapor thermal conductivity as function of temperature
- CPL — Liquid specific heat as function of temperature
- HFG — Enthalpy of vaporization as function of pressure
- PRL — Liquid Prandtl number as function of temperature
- PRV — Vapor Prandtl number as function of temperature
- SAT — Saturation temperature as function of pressure
- DTSDP — Pressure derivative of saturation temperature as  
function of pressure
- SURTEN — Surface tension as function of temperature
- VISCV — Vapor viscosity as function of temperature
- VISCL — Liquid viscosity as function of temperature

APPENDIX C

## NATOF - 2D I/O EXAMPLES

Fortran unit numbers for the data files are as follows:

- 5 is the standard input unit
- 6 is for the printed output
- 7 is the dump file to restart

After a successful run, the program creates in file 7 an input data set corresponding to an initial value problem starting at the time the last run was finished. This is particularly useful in generating a transient problem input data set, which requires a substantial amount of information for the initial conditions. In this way, a steady-state problem, which requires a relatively small amount of information, produces in file 7 the input data for the transient problem. The user must only change the cards which describe the boundary conditions, to represent the desired transient conditions, and the desired sequence of printouts.

Following is an example of the input data set for a steady state problem, a transient problem, and an example of the printed output. These examples were taken from the 217-pin simulation described in section 4.

STEADY-STATE INPUT DATA SET

EXAMPLE





241.

TRANSIENT INPUT DATA SET

EXAMPLE

12 5 4 2  
350.100000000D+00  
00.000000000D+00

8 10000.100000000D+000.100000000D-010.100000000D-03

T 0.50000

- .822080000D+060.675750000D+060.000000000D+000.000000000D+00  
- .180800000D+060.277600000D+060.000000000D+000.000000000D+00  
0.000000000D+000.000000000D+000.000000000D+000.000000000D+00  
0.000000000D+000.661140000D+030.000000000D+000.000000000D+00  
0.000000000D+000.661140000D+030.000000000D+000.000000000D+00  
0.000000000D+000.244508688D+100.000000000D+000.000000000D+00

T 3.5

0.000000000D+000.264710000D+06  
0.000000000D+000.187200000D+06  
0.000000000D+000.000000000D+00  
0.000000000D+000.661140000D+03  
0.000000000D+000.661140000D+03  
0.000000000D+000.244508688D+10

T 4.0

0.806340000D+060.272560000D+06  
0.176800000D+060.189200000D+06  
0.000000000D+000.000000000D+00  
0.000000000D+000.661140000D+03  
0.000000000D+000.661140000D+03  
- .4645665100D+100.244508688D+10

T 9.0

0.000000000D+000.675750000D+06  
0.000000000D+000.277600000D+06  
0.000000000D+000.000000000D+00  
0.000000000D+000.661140000D+03  
0.000000000D+000.661140000D+03  
0.000000000D+000.122254330D+09

T 0.00000

90.726440000D-020.584200000D-020.711200000D-03

5 7 8 9 0 0 0 0 0 0 0 0 0 0 0 0 0 0

T0.182900000D+000.182900000D+000.182900000D+000.182900000D+000.182900000D+00  
T0.182900000D+000.182900000D+000.182900000D+000.182900000D+000.182900000D+00  
T0.182900000D+000.182900000D+000.000000000D+000.000000000D+000.000000000D+00  
F0.000000000D+000.000000000D+000.000000000D+000.000000000D+000.000000000D+00  
T0.850000000D+040.850000000D+040.850000000D+040.850000000D+040.850000000D+04  
T0.850000000D+040.850000000D+040.000000000D+000.000000000D+000.000000000D+00  
T0.000000000D+000.000000000D+000.000000000D+000.000000000D+000.000000000D+00  
F0.000000000D+000.000000000D+000.000000000D+000.000000000D+000.000000000D+00  
T0.000000000D+000.000000000D+000.677403000D+000.838702000D+000.100000000D+01  
T0.838702000D+000.677403000D+000.000000000D+000.000000000D+000.000000000D+00



80.949882146D+030.949882146D+030.349355177D+060.000000000D+00  
80.698495222D+010.698495222D+01-.248258543D-03-.248258543D-03  
90.949882135D+030.949882135D+030.331413146D+060.000000000D+00  
90.698859789D+010.698859789D+01-.991732707D-04-.991732707D-04  
100.949882148D+030.949882148D+030.313474029D+060.000000000D+00  
100.699007538D+010.699007538D+01-.357622335D-04-.357622335D-04  
110.949882121D+030.949882121D+030.295536495D+060.000000000D+00  
110.699063111D+010.699063111D+01-.821326883D-05-.821326883D-05  
120.949882121D+030.949882121D+030.277600000D+060.000000000D+00  
120.699078638D+010.699078638D+010.000000000D+000.000000000D+00  
130.661140000D+030.661140000D+030.675750000D+060.000000000D+00  
130.645403526D+010.645403526D+010.000000000D+000.000000000D+00  
140.661139976D+030.661139976D+030.457944948D+060.000000000D+00  
140.645403526D+010.645403526D+010.134087592D-020.134087592D-02  
150.709005430D+030.709005430D+030.440286984D+060.000000000D+00  
150.644199773D+010.644199773D+010.928251848D-030.928251848D-03  
160.768786808D+030.768786808D+030.422313640D+060.000000000D+00  
160.651999493D+010.651999493D+010.548821310D-030.548821310D-03  
170.840563281D+030.840563281D+030.404309099D+060.000000000D+00  
170.662673574D+010.662673574D+010.427459315D-030.427459315D-03  
180.900987733D+030.900987733D+030.386077651D+060.000000000D+00  
180.676258771D+010.676258771D+010.215717403D-030.215717403D-03  
190.949814287D+030.949814287D+030.367828379D+060.000000000D+00  
190.688324996D+010.688324996D+010.945914321D-040.945914321D-04  
200.949814277D+030.949814277D+030.349357575D+060.000000000D+00  
200.698476007D+010.698476007D+01-.356041467D-03-.356041467D-03  
210.949814285D+030.949814285D+030.331413955D+060.000000000D+00  
210.698812060D+010.698812060D+01-.149952101D-03-.149952101D-03  
220.949814304D+030.949814304D+030.313474316D+060.000000000D+00  
220.698961889D+010.698961889D+01-.597175017D-04-.597175017D-04  
230.949814279D+030.949814279D+030.295536554D+060.000000000D+00  
230.699028057D+010.699028057D+01-.182037349D-04-.182037349D-04  
240.949814279D+030.949814279D+030.277600000D+060.000000000D+00  
240.699053848D+010.699053848D+010.000000000D+000.000000000D+00  
250.661140000D+030.661140000D+030.675750000D+060.000000000D+00  
250.645417845D+010.645417845D+010.000000000D+000.000000000D+00  
260.661139976D+030.661139976D+030.457935425D+060.000000000D+00  
260.645417845D+010.645417845D+010.155979087D-020.155979087D-02  
270.709007651D+030.709007651D+030.440281298D+060.000000000D+00  
270.644169688D+010.644169688D+010.108494207D-020.108494207D-02  
280.768792981D+030.768792981D+030.422310628D+060.000000000D+00  
280.651956761D+010.651956761D+010.638818786D-030.638818786D-03  
290.840572981D+030.840572981D+030.404306936D+060.000000000D+00  
290.662642196D+010.662642196D+010.504013753D-030.504013753D-03  
300.901000145D+030.901000145D+030.386076682D+060.000000000D+00

300. 676230243D+010. 676230243D+010. 256252601D-030. 256252601D-03  
 310. 949828329D+030. 949828329D+030. 367827986D+060. 000000000D+00  
 310. 688304299D+010. 688304299D+010. 127519496D-030. 127519496D-03  
 320. 949828094D+030. 949828094D+030. 349359311D+060. 000000000D+00  
 320. 698443939D+010. 698443939D+010. 419776595D-030. 419776595D-03  
 330. 949827962D+030. 949827962D+030. 331414574D+060. 000000000D+00  
 330. 698566424D+010. 698566424D+010. 177593108D-030. 177593108D-03  
 340. 949827916D+030. 949827916D+030. 313474560D+060. 000000000D+00  
 340. 698993622D+010. 698993622D+010. 718543316D-040. 718543316D-04  
 350. 949827871D+030. 949827871D+030. 295536625D+060. 000000000D+00  
 350. 699031921D+010. 699031921D+010. 227451686D-040. 227451686D-04  
 360. 949827871D+030. 949827871D+030. 277600000D+060. 000000000D+00  
 360. 699057870D+010. 699057870D+010. 000000000D+000. 000000000D+00  
 370. 661140000D+030. 661140000D+030. 675750000D+060. 000000000D+00  
 370. 645428949D+010. 645428949D+010. 000000000D+000. 000000000D+00  
 380. 661139976D+030. 661139976D+030. 457928641D+060. 000000000D+00  
 380. 645428949D+010. 645428949D+010. 179030824D-020. 179030824D-02  
 390. 709009385D+030. 709009385D+030. 440276865D+060. 000000000D+00  
 390. 644146447D+010. 644146447D+010. 124468169D-020. 124468169D-02  
 400. 768797831D+030. 768797831D+030. 422308292D+060. 000000000D+00  
 400. 651923359D+010. 651923359D+010. 726055293D-030. 726055293D-03  
 410. 840580594D+030. 840580594D+030. 404365235D+060. 000000000D+00  
 410. 662617911D+010. 662617911D+010. 579300377D-030. 579300377D-03  
 420. 901009888D+030. 901009888D+030. 386075914D+060. 000000000D+00  
 420. 676202049D+010. 676202049D+010. 294166726D-030. 294166726D-03  
 430. 949839319D+030. 949839319D+030. 367827626D+060. 000000000D+00  
 430. 688288602D+010. 688288602D+010. 177753970D-030. 177753970D-03  
 440. 949714824D+030. 949714824D+030. 349360678D+060. 000000000D+00  
 440. 698368498D+010. 698368498D+010. 461539051D-030. 461539051D-03  
 450. 94955968D+030. 94955968D+030. 331415063D+060. 000000000D+00  
 450. 698663932D+010. 698663932D+010. 207414256D-030. 207414256D-03  
 460. 949634152D+030. 949634152D+030. 313474756D+060. 000000000D+00  
 460. 698328621D+010. 698328621D+010. 922642250D-040. 922642250D-04  
 470. 949624131D+030. 949624131D+030. 295535668D+060. 000000000D+00  
 470. 698923257D+010. 698923257D+010. 371563794D-040. 371563794D-04  
 480. 949624131D+030. 949624131D+030. 277600000D+060. 000000000D+00  
 480. 698980314D+010. 698980314D+010. 000000000D+000. 000000000D+00  
 490. 661140000D+030. 661140000D+030. 675750000D+060. 000000000D+00  
 490. 541970490D+010. 541970490D+010. 000000000D+000. 000000000D+00  
 500. 661139976D+030. 661139976D+030. 457921379D+060. 000000000D+00  
 500. 541970490D+010. 541970490D+010. 000000000D+000. 000000000D+00  
 510. 699646325D+030. 699646325D+030. 440273500D+060. 000000000D+00  
 510. 548841579D+010. 548841579D+010. 000000000D+000. 000000000D+00  
 520. 747216772D+030. 747216772D+030. 422306195D+060. 000000000D+00  
 520. 559525484D+010. 559525484D+010. 000000000D+000. 000000000D+00

530.803998842D+030.803998842D+030.404303690D+060.000000000D+00  
530.569825826D+010.569885826D+010.000000000D+000.000000000D+00  
540.251601173D+030.851601173D+030.326075215D+060.000000000D+00  
540.581498449D+010.581498449D+010.000000000D+000.000000000D+00  
550.890031705D+030.890031705D+030.367827221D+060.000000000D+00  
550.590800788D+010.590800788D+010.000000000D+000.000000000D+00  
560.890031690D+030.890031690D+030.349361862D+060.000000000D+00  
560.598279362D+010.598279362D+010.000000000D+000.000000000D+00  
570.890031694D+030.890031694D+030.331415520D+060.000000000D+00  
570.596519113D+010.596519113D+010.000000000D+000.000000000D+00  
580.890031699D+030.890031699D+030.313474959D+060.000000000D+00  
580.595729676D+010.595729676D+010.000000000D+000.000000000D+00  
590.890031688D+030.890031688D+030.295536764D+060.000000000D+00  
590.595380175D+010.595380175D+010.000000000D+000.000000000D+00  
600.890031688D+030.890031688D+030.277600000D+060.000000000D+00  
600.595241215D+010.595241215D+010.000000000D+000.000000000D+00  
T0.661139976D+030.661139976D+030.661139976D+030.661139976D+030.661139976D+03  
T0.661139976D+030.661139976D+030.661139976D+030.661139976D+030.177692822D+040.169945358D+04  
T0.148493467D+040.119305137D+040.887163285D+030.762291343D+030.739650519D+03  
T0.718545367D+030.229407525D+040.220226250D+040.190408012D+040.143965213D+04  
T0.989632560D+030.835144320D+030.807124783D+030.780994205D+030.270225402D+04  
T0.262496466D+040.235072460D+040.175835324D+040.110423196D+040.920248747D+03  
T0.886840533D+030.855684557D+030.249915538D+040.242058256D+040.215147977D+04  
T0.165030075D+040.112246557D+040.968224687D+030.940205151D+030.914074572D+03  
T0.223810093D+040.216198092D+040.192056129D+040.153494856D+040.112893154D+04  
T0.100439023D+040.981759401D+030.960654249D+030.949882146D+030.949882146D+03  
T0.949882146D+030.949882146D+030.949882146D+030.949882146D+030.949882146D+03  
T0.949882146D+030.949882135D+030.949882135D+030.949882135D+030.949882135D+03  
T0.949882135D+030.949882135D+030.949882135D+030.949882135D+030.949882148D+03  
T0.949882148D+030.949882148D+030.949882148D+030.949882148D+030.949882148D+03  
T0.949882148D+030.949882148D+030.949882121D+030.949882121D+030.949882121D+03  
T0.949882121D+030.949882121D+030.949882121D+030.949882121D+030.949882121D+03  
T0.661139976D+030.661139976D+030.661139976D+030.661139976D+030.661139976D+03  
T0.661139976D+030.661139976D+030.661139976D+030.177689940D+040.169942569D+04  
T0.148491047D+040.119303284D+040.887149675D+030.762267718D+030.739636893D+03  
T0.718531742D+030.229401948D+040.220220365D+040.190401737D+040.143960338D+04  
T0.989601252D+030.835112961D+030.807093425D+030.780962946D+030.270219726D+04  
T0.262490331D+040.235064620D+040.175826399D+040.110418122D+040.920197885D+03  
T0.886789671D+030.855633695D+030.249907233D+040.242049368D+040.215137401D+04  
T0.165020046D+040.112240299D+040.968161977D+030.940142441D+030.914011862D+03  
T0.223799226D+040.216186803D+040.192044307D+040.153484908D+040.112886318D+04  
T0.100432175D+040.981690925D+030.960585773D+030.949814277D+030.949814277D+03  
T0.949814277D+030.949814277D+030.949814277D+030.949814277D+030.949814277D+03  
T0.949814277D+030.949814285D+030.949814285D+030.949814285D+030.949814285D+03  
T0.949814285D+030.949814285D+030.949814285D+030.949814285D+030.949814304D+03

T0.949814304D+030.949814304D+030.949814304D+030.949814304D+030.949814304D+03  
T0.949814304D+030.949814304D+030.949814279D+030.949814279D+030.949814279D+03  
T0.949814279D+030.949814279D+030.949814279D+030.949814279D+030.949814279D+03  
T0.661139976D+030.661139976D+030.661139976D+030.661139976D+030.661139976D+03  
T0.661139976D+030.661139976D+030.661139976D+030.177690446D+040.169943059D+04  
T0.148491472D+040.119303609D+040.887152066D+030.762270112D+030.739639289D+03  
T0.718534136D+030.229403090D+040.220221571D+040.190403022D+040.143961337D+04  
T0.989607666D+030.835119386D+030.807099849D+030.780969270D+030.270220841D+04  
T0.262491536D+040.235066159D+040.175828151D+040.110419118D+040.920207874D+03  
T0.885799659D+030.855643684D+030.249908909D+040.242051161D+040.215139534D+04  
T0.165022069D+040.112241561D+040.968174625D+030.940155089D+030.914024510D+03  
T0.223601488D+040.216189153D+040.192046768D+040.153486977D+040.112887741D+04  
T0.100433600D+040.981705178D+030.960600027D+030.949828094D+030.949828094D+03  
T0.949828094D+030.949828094D+030.949828094D+030.949828094D+030.949828094D+03  
T0.949828094D+030.949827962D+030.949827962D+030.949827962D+030.949827962D+03  
T0.949827962D+030.949827962D+030.949827962D+030.949827962D+030.949827962D+03  
T0.949827916D+030.949827916D+030.949827916D+030.949827916D+030.949827916D+03  
T0.949827916D+030.949827916D+030.949827916D+030.949827916D+030.949827916D+03  
T0.949827871D+030.949827871D+030.949827871D+030.949827871D+030.949827871D+03  
T0.661139976D+030.661139976D+030.661139976D+030.661139976D+030.661139976D+03  
T0.661139976D+030.661139976D+030.661139976D+030.177690842D+040.169943441D+04  
T0.148491804D+040.119303864D+040.887153934D+030.762271982D+030.739641157D+03  
T0.718536005D+030.229403988D+040.220222518D+040.190404032D+040.143962121D+04  
T0.989612704D+030.835124431D+030.807104895D+030.780974316D+030.270221716D+04  
T0.262492482D+040.235067368D+040.175829527D+040.110419900D+040.920215712D+03  
T0.886607497D+030.855651522D+030.249910223D+040.242052567D+040.215141208D+04  
T0.165023657D+040.112242552D+040.968184552D+030.940165015D+030.914034436D+03  
T0.223603277D+040.216191012D+040.192048714D+040.153488615D+040.112888866D+04  
T0.100434727D+040.981716450D+030.960611298D+030.949714824D+030.949714824D+03  
T0.949714824D+030.949714824D+030.949714824D+030.949714824D+030.949714824D+03  
T0.949714824D+030.949658968D+030.949658968D+030.949658968D+030.949658968D+03  
T0.949658968D+030.949658968D+030.949658968D+030.949658968D+030.949658968D+03  
T0.949634152D+030.949634152D+030.949634152D+030.949634152D+030.949634152D+03  
T0.949634152D+030.949634152D+030.949624131D+030.949624131D+030.949624131D+03  
T0.949624131D+030.949624131D+030.949624131D+030.949624131D+030.949624131D+03  
T0.661139976D+030.661139976D+030.661139976D+030.661139976D+030.661139976D+03  
T0.661139976D+030.661139976D+030.661139976D+030.175496958D+040.167823793D+04  
T0.146656179D+040.117897741D+040.876804866D+030.751911989D+030.729281164D+03  
T0.708176013D+030.225221059D+040.215821163D+040.185774891D+040.140409633D+04  
T0.966699134D+030.812174698D+030.784155162D+030.758024593D+030.265753179D+04  
T0.257657755D+040.228898081D+040.169079942D+040.106588979D+040.881819019D+03  
T0.848410304D+030.817254829D+030.242750750D+040.234398530D+040.206164460D+04  
T0.156873596D+040.107147044D+040.917126727D+030.889107191D+030.862976612D+03  
T0.213527169D+040.205587076D+040.181242410D+040.144655784D+040.106777019D+04  
T0.943126193D+030.920495368D+030.899390217D+030.890031690D+030.890031690D+03

T0.890031690D+030.890031690D+030.890031690D+030.890031690D+030.890031690D+03  
T0.890031690D+030.890031694D+030.890031694D+030.890031694D+030.890031694D+03  
T0.890031694D+030.890031694D+030.890031694D+030.890031694D+030.890031699D+03  
T0.890031699D+030.890031699D+030.890031699D+030.890031699D+030.890031699D+03  
T0.890031699D+030.890031699D+030.890031688D+030.890031688D+030.890031688D+03  
T0.890031688D+030.890031688D+030.890031688D+030.890031688D+030.890031688D+03  
T0.890031688D+030.890031688D+030.890031688D+030.890031688D+030.890031688D+03

F

T0.661139976D+030.661139976D+030.699646325D+030.747216772D+030.803998842D+03  
T0.851601173D+030.890031705D+030.890031690D+030.890031694D+030.890031699D+03  
T0.890031688D+030.890031688D+030.890031690D+030.890031694D+030.890031699D+03

F

PRINTED OUTPUT EXAMPLE

flow map at time = 1.9007 sec.

number of time steps = 192  
 number of iterations = 770  
 time step size = 0.1454D-02 sec.

inlet mass flow rate = 0.268297D+01 kg/sec      inlet enthalpy flow = 0.234219D+07 watt  
 outlet mass flow rate = 0.914200D+01 kg/sec      outlet enthalpy flow = 0.131079D+08 watt  
 total heat transfered = 0.272580D+07 watt

channel number 1										
iz	p (bar)	void	tv	tl ----- (degree celsius)	tsat	twall	uvz (m/sec)	ulz (m/sec)	uvr (m/sec)	ulr (m/sec)
12	1.8720	0.010783	851.598	851.598	1005.330	849.091	3.43381	3.37281	0.00000	0.00000
11	1.9192	0.010793	851.598	851.598	957.036	849.091	3.80309	3.92908	0.01375	0.01375
10	1.9053	0.120591	877.980	877.979	956.123	876.620	10.80335	8.29518	-0.04234	-0.04234
9	2.0836	0.778999	897.775	897.773	967.469	895.300	24.55758	8.75392	-0.80572	-0.79020
8	2.2962	0.983274	980.388	980.388	980.036	925.166	34.95135	5.32153	0.97598	0.93946
7	2.3966	0.939780	987.087	987.087	985.655	1070.378	3.46065	0.50722	0.40586	0.40327
6	2.4354	0.010680	989.274	989.274	987.774	997.784	0.85252	0.83565	0.06171	0.06170
5	2.4638	0.000000	886.795	886.795	989.308	898.459	0.89817	0.89817	0.00957	0.00957
4	2.4990	0.000000	717.602	717.602	991.196	726.899	0.86429	0.86429	0.00200	0.00200
3	2.5366	0.000000	559.583	559.583	993.184	567.597	0.81454	0.81454	0.00031	0.00031
2	2.5758	0.000000	388.000	388.000	995.231	388.000	0.81165	0.81165	-0.00020	-0.00020
1	2.6471	0.000000	388.000	388.000	1138.947	388.000	0.81165	0.81165	0.00000	0.00000

channel number 2										
iz	p (bar)	void	tv	tl ----- (degree celsius)	tsat	twall	uvz (m/sec)	ulz (m/sec)	uvr (m/sec)	ulr (m/sec)
12	1.8720	0.013694	850.618	850.618	1005.330	848.367	3.37639	3.30393	0.00000	0.00000
11	1.9180	0.013694	850.618	850.618	956.956	848.367	3.67583	3.82623	0.00991	0.00990
10	1.9052	0.161033	875.160	875.158	956.112	874.532	12.36248	8.03308	-0.17664	-0.17638
9	2.1451	0.990050	963.998	963.999	971.209	888.852	41.74311	6.83830	1.28744	1.21714
8	2.2468	0.942923	974.842	974.842	977.205	924.136	28.65856	6.20831	0.17670	0.16972
7	2.3659	0.941946	985.341	985.341	983.957	1056.445	7.71611	0.77429	0.91280	0.90757
6	2.4288	0.000272	985.940	985.940	987.414	994.873	0.98653	0.98567	0.05649	0.05649
5	2.4630	0.000000	885.666	885.666	989.268	897.504	0.93446	0.93446	0.00790	0.00790
4	2.4989	0.000000	717.325	717.325	991.187	726.660	0.87301	0.87301	0.00144	0.00144
3	2.5366	0.000000	559.447	559.447	993.183	567.473	0.81773	0.81773	0.00015	0.00015
2	2.5758	0.000000	388.000	388.000	995.233	388.000	0.81125	0.81125	-0.00050	-0.00050

1 2.6471 0.000000 388.000 388.000 1138.947 388.000 0.81125 0.81125 0.00000 0.00000

channel number 3

iz	p (bar)	void	tv	tl	tsat	twall	uvz (m/sec)	ulz (m/sec)	uvr (m/sec)	ulr (m/sec)
------(degree celsius)-----										
12	1.8720	0.008960	850.006	850.006	1005.330	847.885	3.31343	3.26321	0.00000	0.00000
11	1.9174	0.008960	850.006	850.006	956.916	847.885	3.71564	3.73243	0.01519	0.01519
10	1.9146	0.103637	871.035	871.034	956.734	871.663	9.99655	7.42375	-0.16207	-0.16200
9	2.1012	0.825045	889.776	889.775	968.551	886.600	22.08457	7.60331	0.54027	0.53450
8	2.2408	0.983656	977.425	977.426	976.855	924.390	34.67865	5.82414	0.72068	0.70740
7	2.3347	0.937245	983.685	983.685	982.215	1047.062	4.36154	1.16028	1.11469	1.10838
6	2.4263	0.000089	994.672	984.672	987.277	993.807	1.05234	1.05201	0.04685	0.04685
5	2.4627	0.000000	895.371	885.371	989.251	987.267	0.94564	0.94564	0.00587	0.00587
4	2.4988	0.000000	717.242	717.242	991.183	726.593	0.87624	0.87624	0.00104	0.00104
3	2.5366	0.000000	559.346	559.346	993.182	567.384	0.82035	0.82035	0.00008	0.00008
2	2.5758	0.000000	388.000	388.000	995.234	388.000	0.81083	0.81083	-0.00073	-0.00073
1	2.6471	0.000000	388.000	388.000	1138.947	388.000	0.81083	0.81083	0.00000	0.00000

channel number 4

iz	p (bar)	void	tv	tl	tsat	twall	uvz (m/sec)	ulz (m/sec)	uvr (m/sec)	ulr (m/sec)
------(degree celsius)-----										
12	1.8720	0.002959	843.195	843.195	1005.330	842.154	3.07388	3.05353	0.00000	0.00000
11	1.9169	0.002959	843.195	843.195	956.883	842.154	3.16124	3.15590	0.01478	0.01478
10	1.9199	0.021862	850.868	850.868	957.078	854.891	7.19584	6.58415	-0.05237	-0.05237
9	2.0887	0.164075	850.842	850.840	967.783	848.647	11.37246	8.19370	-0.03159	-0.03150
8	2.2338	0.987239	976.981	976.981	976.449	918.343	32.79295	5.62849	1.45981	1.40642
7	2.3097	0.931968	981.770	981.770	980.301	1048.771	4.75434	1.59974	1.00605	1.00252
6	2.4250	0.000007	983.910	983.910	987.207	993.180	1.09089	1.09087	0.02652	0.02652
5	2.4626	0.000000	885.240	885.240	989.243	897.164	0.95099	0.95099	0.00315	0.00315
4	2.4988	0.000000	717.213	717.213	991.182	726.570	0.87760	0.87760	0.00058	0.00058
3	2.5366	0.000000	559.321	559.321	993.182	567.360	0.82085	0.82085	-0.00001	-0.00001
2	2.5759	0.000000	388.000	388.000	995.236	388.000	0.81020	0.81020	-0.00098	-0.00098
1	2.6471	0.000000	388.000	388.000	1138.947	388.000	0.81020	0.81020	0.00000	0.00000

channel number 5

iz	p (bar)	void	tv	tl	tsat	twall	tcan	uvz (m/sec)	ulz (m/sec)
------(degree celsius)-----									
12	1.8720	0.000327	713.043	713.043	1005.330	709.701	713.043	2.29313	2.29167
11	1.9166	0.000327	713.043	713.043	956.861	709.701	713.043	1.75293	1.75221

251

10	1.9211	0.002928	741.112	741.112	957.157	735.525	741.112	4.04090	3.87783
9	2.0910	0.029919	803.879	803.879	967.923	795.314	803.879	8.19941	7.51813
8	2.2099	0.328978	856.139	856.136	975.053	846.926	856.136	6.59102	5.76649
7	2.2942	0.215748	862.126	862.123	979.923	867.617	841.491	2.56718	1.85150
6	2.4244	0.000000	783.632	783.632	987.179	793.138	767.650	0.86239	0.86239
5	2.4625	0.000000	710.987	710.987	989.240	724.489	700.308	0.71430	0.71430
4	2.4988	0.000000	603.494	603.494	991.161	614.385	595.840	0.66470	0.66470
3	2.5366	0.000000	499.709	499.709	993.182	508.341	494.512	0.63937	0.63937
2	2.5759	0.000000	388.000	388.000	995.238	388.000	388.000	0.67691	0.67691
1	2.6471	0.000000	388.000	388.000	1138.947	388.000	388.000	0.67691	0.67691

INSTITUTO DE RECOURSES EMPRESARIALES Y SERVICIOS  
 I.P.E.N.

flow map at time = 2.0013 sec.

number of time steps = 265  
 number of iterations = 1138  
 time step size = 0.3378D-02 sec.

inlet mass flow rate = -.271222D+01 kg/sec  
 outlet mass flow rate = 0.754283D+01 kg/sec  
 total heat transferred = -.562540D+06 watt

inlet enthalpy flow = -.248236D+07 watt  
 outlet enthalpy flow = 0.109099D+08 watt

channel number 1										
iz	p (bar)	void	tv	tl ----- (degree celsius)	tsat	twall	uvz (m/sec)	ulz (m/sec)	uvr (m/sec)	ulr (m/sec)
12	1.8720	0.303258	861.964	861.963	1005.330	860.980	0.51501	3.07887	0.00000	0.00000
11	1.5388	0.303258	861.964	861.963	929.841	860.980	15.50110	8.88634	-0.14508	-0.14462
10	1.8744	0.994372	945.344	945.344	954.065	884.862	27.85684	3.51727	1.07253	1.05714
9	1.8967	0.970861	956.143	956.144	955.553	885.346	22.69684	3.38400	0.32808	0.31151
8	1.9247	0.973085	958.142	958.142	957.394	918.024	16.03301	2.50611	0.27615	0.26388
7	1.9403	0.971530	959.403	959.403	958.410	1102.302	6.95722	1.05196	0.33247	0.31702
6	1.9438	0.977325	959.525	959.525	958.641	1055.607	1.96098	-2.71013	0.37385	0.37203
5	1.9863	0.276728	961.461	961.461	961.380	965.323	-0.53806	-1.36656	0.12127	0.12115
4	2.1552	0.003745	806.046	806.046	971.812	803.191	-0.89807	-0.94945	0.00318	0.00318
3	2.3318	0.000193	637.440	637.440	982.050	634.469	-0.88206	-0.88490	0.00252	0.00252
2	2.5118	0.000005	424.457	424.457	991.872	417.601	-0.84238	-0.84243	-0.00068	-0.00068
1	2.6471	0.000000	388.000	388.000	1138.947	417.601	-0.84238	-0.84243	0.00000	0.00000

channel number 2										
iz	p (bar)	void	tv	tl ----- (degree celsius)	tsat	twall	uvz (m/sec)	ulz (m/sec)	uvr (m/sec)	ulr (m/sec)
12	1.8720	0.267247	859.178	859.177	1005.330	859.009	2.07274	3.82092	0.00000	0.00000
11	1.5528	0.267247	859.178	859.177	930.932	859.009	15.03800	9.07103	-0.25991	-0.25926
10	1.8534	0.993174	944.524	944.524	952.657	883.012	27.19286	3.42531	1.60127	1.58285
9	1.8781	0.961646	954.645	954.645	954.316	878.283	21.72918	3.35950	0.43395	0.41368
8	1.9107	0.969096	957.194	957.194	956.473	923.430	13.99242	2.38723	0.42054	0.39789
7	1.9230	0.973095	958.282	958.282	957.287	1093.160	6.96456	1.07655	0.56712	0.54060
6	1.9263	0.980335	958.405	958.405	957.497	1049.437	6.78303	-2.64147	0.77521	0.77175
5	1.9771	0.190750	960.883	960.883	960.797	964.921	-0.35509	-1.16214	-0.06259	-0.06248
4	2.1552	0.003013	803.565	803.565	971.809	801.283	-0.84814	-0.89399	0.00015	0.00015
3	2.3316	0.000176	636.337	636.337	982.037	633.663	-0.86055	-0.86327	0.00200	0.00200
2	2.5118	0.000005	424.035	424.035	991.876	417.296	-0.84150	-0.84154	-0.00216	-0.00216

1 2.6471 0.000000 388.000 388.000 1138.947 417.296 -0.84150 -0.84154 0.00000 0.00000

channel number 3

iz	p (bar)	void	tv -----	tl (degree celsius)	tsat -----	twall -----	uvz (m/sec)	ulz (m/sec)	uvr (m/sec)	ulr (m/sec)
12	1.8720	0.194987	847.012	847.011	1005.330	849.999	1.93011	3.94404	0.00000	0.00000
11	1.5650	0.194987	847.012	847.011	931.873	849.999	13.82167	8.91571	-0.29528	-0.29497
10	1.8382	0.992186	937.570	937.571	951.627	873.235	31.75979	3.99577	2.07276	1.95726
9	1.8649	0.962474	953.507	953.507	953.433	875.425	19.54338	3.28950	0.14674	0.14020
8	1.8976	0.955072	955.987	955.987	955.609	929.787	10.60926	2.26209	0.42229	0.40226
7	1.9086	0.972566	957.333	957.333	956.340	1083.821	2.22228	0.93556	0.63869	0.60924
6	1.9089	0.981059	957.346	957.346	956.360	1046.766	1.84264	-2.38606	0.95529	0.95168
5	1.9822	0.235475	961.237	961.237	961.115	965.307	-0.35914	-1.06574	0.02240	0.02241
4	2.1553	0.002785	802.564	802.564	971.816	800.549	-0.81983	-0.86247	-0.00133	-0.00133
3	2.3315	0.000164	635.795	635.795	982.032	633.262	-0.84754	-0.84997	0.00136	0.00136
2	2.5120	0.000004	423.787	423.787	991.883	417.115	-0.84167	-0.84170	-0.00321	-0.00321
1	2.6471	0.000000	388.000	388.000	1138.947	417.115	-0.84167	-0.84170	0.00000	0.00000

channel number 4

iz	p (bar)	void	tv -----	tl (degree celsius)	tsat -----	twall -----	uvz (m/sec)	ulz (m/sec)	uvr (m/sec)	ulr (m/sec)
12	1.8720	0.057287	826.273	826.273	1005.330	830.661	2.56785	3.32214	0.00000	0.00000
11	1.5759	0.057287	826.273	826.273	932.711	830.661	9.94701	8.60545	-0.14279	-0.14277
10	1.7882	0.464318	845.903	845.899	948.201	841.528	9.54023	5.16638	0.03636	0.03632
9	1.8619	0.952944	935.220	935.219	953.233	860.605	11.26089	3.51069	0.24239	0.23809
8	1.8882	0.875778	927.424	927.424	954.991	926.111	7.09710	2.66816	0.16310	0.16036
7	1.8981	0.973212	956.609	956.609	955.648	1074.430	-2.40832	0.77884	0.64798	0.61870
6	1.8962	0.983523	956.620	956.620	955.518	1045.606	-0.37324	-2.25866	1.18517	1.17979
5	1.9827	0.091832	948.708	948.708	961.145	957.131	-0.58589	-0.98588	-0.01629	-0.01628
4	2.1554	0.002370	801.594	801.594	971.822	799.919	-0.80536	-0.84113	-0.00068	-0.00068
3	2.3314	0.000131	635.521	635.521	982.029	633.080	-0.84035	-0.84221	0.00045	0.00045
2	2.5121	0.000003	423.420	423.420	991.890	416.859	-0.84098	-0.84100	-0.00438	-0.00438
1	2.6471	0.000000	388.000	388.000	1138.947	416.859	-0.84098	-0.84100	0.00000	0.00000

channel number 5

iz	p (bar)	void	tv -----	tl (degree celsius)	tsat -----	twall -----	tcan -----	uvz (m/sec)	ulz (m/sec)
12	1.8720	0.003825	763.699	763.699	1005.330	753.066	763.699	1.16509	1.25295
11	1.5793	0.003825	763.699	763.699	932.972	753.066	763.699	6.99345	6.85619

254

10	1.7879	0.020460	823.341	823.341	948.180	816.611	823.341	6.65369	6.37594
9	1.8577	0.105617	859.689	859.688	952.946	853.058	859.688	5.87165	5.69386
8	1.8855	0.093630	877.018	877.017	954.806	872.255	877.017	4.81453	4.76605
7	1.8880	0.308404	892.240	892.239	954.977	899.516	879.397	1.42155	1.66511
6	1.8766	0.318665	847.592	847.590	954.213	856.880	815.256	-1.69625	-2.11925
5	1.9834	0.007083	750.548	750.548	961.195	760.944	727.796	-0.84139	-0.88960
4	2.1554	0.000247	634.591	634.590	971.823	643.339	618.204	-0.82661	-0.82885
3	2.3314	0.000007	526.750	526.750	982.029	533.441	513.420	-0.81987	-0.81992
2	2.5122	0.000000	400.197	400.197	991.897	399.065	394.352	-0.61606	-0.61606
1	2.6471	0.000000	388.000	388.000	1138.947	399.065	394.352	-0.61606	-0.61606

APPENDIX D

NATOF - 2D PROGRAM LISTING

COMPILATION LISTING OF NATOF (>user\_dir\_dir>BOIL>Granziera>NATOF.fortran)

Compiled by: Multics New Fortran Compiler, Release 6

Compiled on: 04/29/80 1304.9 edt Tue

Options: table card relocatable map

Main Program

```
1 C      MAIN PROGRAM
2 C
3 C      THE MAIN PROGRAM HAS THE ONLY FUNCTION OF ALOCATING
4 C      POSITIONS IN THE MEMORY FOR THE VARIABLES.
5 C
6 C      THE COMAND 'DIMENSION ORBI(XXXX)' ALOCATES MEMORY FOR
7 C      ALL THE VARIABLES.FOR EACH PROBLEM,THE USER SHOULD
8 C      SUPLY ITS DIMENSION,WHICH VALUE IS CALCULATED AS :
9 C
10 C     XXXX = (131 + 2*(NCF + NCLD))*NI*NJ
11 C
12 C     WITH :
13 C
14 C     NI = NUMBER OF AXIAL MESH POINTS
15 C     NJ = NUMBER OF RADIAL MESH POINTS
16 C
17 C
18 C     IMPLICIT REAL*8 (A-H,O-Z)
19 C
20 C     DIMENSION ORBI(12000)
21 C     NORBI = 12000
22 C     DO 10 II = 1,NORBI
23 C 10 ORBI(II) = 0.00
24 C
25 C     CALL HEAD(ORBI,NORBI)
26 C     STOP
27 C     END
```

Block Data

```
28 BLOCK DATA  
29 COMMON /NUMBER/ ZERO,ONE,BIG,SMALL  
30 REAL*8 ZERO/0.00/,ONE/1.00/,BIG/1.0+07/,SMALL/1.0-08/  
31 END
```

Subroutine head

```

32 SUBROUTINE HEAD(ORBI,NORBI)
33 IMPLICIT REAL*8 (A-H,O-Z)
34 COMMON /BCX/ ULO
35 COMMON /NUMBER/ ZERO,ONE,BIG,SMALL
36 COMMON /ERROR/ IERR
37 COMMON /RHEA/ TSET(40),TSHSET(40),DTMAX,DTM1
38 COMMON /REA/ NN,NP,NB,NW,NTR,NPIN,NPM1,NSET(40),NSHSET(40)
39 COMMON /DIM/ DZ(40),DZ1(40),DRO(40),DR1(40),DR2(40),DR3(40),
40 * DR4(40),NI,NJ,NIM1,NIM2,NJM1,NNI,NNJ,NNJJ
41 COMMON /CNTRL/ EPS1,EPS2,RES,IT1,IT2,IT3,ITM1,ITM2,IGAUSS
42 COMMON /TEMPO/ TIME,DT,DTO,DTLS,NDT
43 COMMON /PNTR1/ K(100),M(100)
44 DIMENSION ORBI(NORBI)
45 C
46 C THE MATRIX M CONTAINS POINTERS TO THE MATRIX ORBI
47 C WHICH CORRESPOND TO THE FIRST ELEMENT OF THE VARIABLE
48 C DIMENSIONED ARAYS IN THE FOLLOWING EQUIVALENCE :
49 C
50 C M( 1) = P M( 2) = PO
51 C M( 3) = TV M( 4) = TVO
52 C M( 5) = TL M( 6) = TLO
53 C M( 7) = ALFAN M( 8) = ALFAD
54 C M( 9) = ALFAZ M(10) = ALFAR
55 C M(11) = RHOV M(12) = RHOL
56 C M(13) = RHOVZ M(14) = RHOLZ
57 C M(15) = RHOVR M(16) = RHOVRZ
58 C M(17) = HV M(18) = HL
59 C M(19) = HVZ M(20) = HLZ
60 C M(21) = HVR M(22) = HLR
61 C M(23) = UVZN M(24) = ULZN
62 C M(25) = UVRN M(26) = ULRN
63 C M(27) = UVZO M(28) = ULZO
64 C M(29) = UVRO M(30) = ULRO
65 C M(31) = UVRZ M(32) = ULRZ
66 C M(33) = UVZR M(34) = ULZR
67 C M(35) = FUVZN M(36) = FULZN
68 C M(37) = FUVRN M(38) = FULRN
69 C M(39) = W(K( 1)) M(40) = W(K( 3))
70 C M(41) = W(K( 3)) M(42) = W(K( 4))
71 C M(43) = W(K( 5)) M(44) = W(K( 6))
72 C M(45) = W(K( 7)) M(46) = W(K( 8))
73 C M(47) = W(K( 9)) M(48) = W(K(10))
74 C M(49) = W(K(11)) M(50) = W(K(12))

```

```

75 C M(51) = W(K(13))
76 C M(53) = W(K(15))
77 C M(55) = W(K(17))
78 C M(57) = W(K(19))
79 C M(59) = W(K(21))
80 C M(61) = W(K(23))
81 C M(63) = DH
82 C M(65) = DV
83 C M(67) = TS
84 C M(69) = DTW
85 C M(71) = HCONL
86 C M(73) = DPN
87 C M(75) = A2
88 C M(77) = A4
89 C M(79) = B
90 C M(81) = DTR
91 C M(83) = BETA
92 C M(85) = QPP C
93 C M(87) = ALZD C
94 C M(89) = ALRD
95 C M(91) = TCAN
96 C
97 C
98 C
99 C
100 DO 1001 L = 2,79
101 M(L) = M(L-1) + NN
102 M(80) = M(79) + NB
103 M(81) = M(80) + NTR
104 M(82) = M(81) + NTR
105 M(83) = M(82) + NN
106 M(84) = M(83) + NN
107 M(85) = M(84) + NN
108 M(86) = M(85) + NN
109 M(87) = M(86) + NN
110 M(88) = M(87) + NN
111 M(89) = M(88) + NN
112 M(90) = M(89) + NN
113 M(91) = M(90) + NN
114 NCAN = 4*NI
115 M(92) = M(91) + NCAN
116 C
117 C
118 C
119 C

```

```

M(52) = W(K(14))
M(54) = W(K(16))
M(56) = W(K(18))
M(58) = W(K(20))
M(60) = W(K(22))
M(62) = W(K(24))
M(64) = DHR
M(66) = OSI
M(68) = TW
M(70) = HCONV
M(72) = HNB
M(74) = A1
M(76) = A3
M(78) = YP
M(80) = TR
M(82) = TWOLD
M(84) = GAMMA
M(86) = AVZD
M(88) = AVR
M(90) = SPPD
M(92) =

```

TIME = ZERO

```

120     CALL READ2(ORBI(M(1)),ORBI(M(3)),ORBI(M(5)),ORBI(M(7)),
121     *          ORBI(M(23)),ORBI(M(24)),ORBI(M(25)),ORBI(M(26)),
122     *          ORBI(M(63)),ORBI(M(65)),ORBI(M(66)),ORBI(M(80)),
123     *          ORBI(M(81)),TINIT,ORBI(M(68)),ORBI(M(90)),ORBI(M(91)),
124     *          NP,NTR,NPIN,NPM1,NN,NCAN)
125     DO 104 KO = 1,NN
126     KL = KO - 1
127     DO 103 L = 1,67
128     103 K(L) = M(L) + KL
129 C
130     ORBI(K(67)) = SAT(ORBI(K(1)))
131     ORBI(K(2)) = ORBI(K(1))
132     ORBI(K(4)) = ORBI(K(3))
133     ORBI(K(6)) = ORBI(K(5))
134     ORBI(K(8)) = ORBI(K(7))
135     ORBI(K(30)) = ORBI(K(26))
136     ORBI(K(27)) = ORBI(K(23))
137     ORBI(K(28)) = ORBI(K(24))
138     ORBI(K(29)) = ORBI(K(25))
139     104 CONTINUE
140 C
141 C
142 C
143 C
144     NPRI = 0
145     TPRI = TINIT + TSET(1)
146     TSHPRI = TINIT + TSHSET(1)
147     LSH = 1
148     L = 1
149     NTS = 0
150     NIT = 0
151     TIME = TINIT
152     LPRI = 0
153     LSHPRI = 0
154     1 CONTINUE
155     NDT = 0
156     IT3 = 0
157     CALL TMSTEP(ORBI,NORBI,
158     *          NN,NP,NB,NW,NTR,NPIN,NPM1,NCAN)
159 C
160     NIT = NIT + IT3
161     NTS = NTS + 1
162 C
163     IF(IERR.NE.0) GO TO 7
164     IF(TIME.LT.TPRI) GO TO 1

```

```

165 C      GO TO 8
166      2 CONTINUE
167 C
168 C
169 C
170      LPRI = LPRI + 1
171      IF(LPRI-NSET(L))3,4,4
172      3 TPRI = TPRI + TSET(L)
173      GO TO 1
174      4 L = L + 1
175      NPRI = NPRI + LPRI
176      IF(NSET(L))6,6,5
177      5 LPRI = 0
178      TPRI = TPRI + TSET(L)
179      GO TO 1
180      6 CALL SAVER(ORBI(M(1)),ORBI(M(3)),ORBI(M(5)),ORBI(M(7)),
181      ORBI(M(23)),ORBI(M(24)),ORBI(M(25)),ORBI(M(26))),
182      ORBI(M(80)),ORBI(M(91)),TIME,NTR,NN,NCAN,NI)
183      *
184      *
185      7 CALL ERRMES(TIME)
186 C      8 CONTINUE
187
188      QT = ZERO
189      FMI = ZERO
190      FME = ZERO
191      FHI = ZERO
192      FHE = ZERO
193      KP = 0
194      DO 9 J = 1,NU
195      KI = (J-1)*NI + 1
196      KE = J*NI
197 C      IF(ORBI(M(24))+KI-1).LT.ZERO) KI = KI + 1
198      TV = ORBI(M(3))+KI-1
199      TL = ORBI(M(5))+KI-1
200      PP = ORBI(M(1))+KI-1
201      UV = ORBI(M(23))+KI-1
202      UL = ORBI(M(24))+KI-1
203      AA = ORBI(M(7))+KI-1
204      RV = ORBI(M(11))+KI-1
205      RL = ORBI(M(12))+KI-1
206      EV = ORBI(M(17))+KI-1
207      EL = ORBI(M(18))+KI-1
208 C
209 C
210 C

```

```

210      FMVJ = AA*RV*UV*DR4(J)
211      FMLJ = (ONE - AA)*RL*UL*DR4(J)
212      HV = EV + PP/RV
213      HL = EL + PP/RL
214 C
215      FMI = FMI + FMVJ + FMLJ
216      FHI = FHI + FMVJ*HV + FMLJ*HL
217 C
218      TV = ORBI(M(3)+KE-1)
219      TL = ORBI(M(5)+KE-1)
220      PP = ORBI(M(1)+KE-1)
221      AA = ORBI(M(7)+KE-2)
222      UV = ORBI(M(23)+KE-1)
223      UL = ORBI(M(24)+KE-1)
224      RV = ORBI(M(11)+KE-1)
225      RL = ORBI(M(12)+KE-1)
226      EV = ORBI(M(17)+KE-1)
227      EL = ORBI(M(18)+KE-1)
228 C
229 C
230      FMVJ = AA*RV*UV*DR4(J)
231      FMLJ = (ONE - AA)*RL*UL*DR4(J)
232      HV = EV + PP/RV
233      HL = EL + PP/RL
234 C
235      FME = FME + FMVJ + FMLJ
236      FHE = FHE + FMVJ*HV + FMLJ*HL
237 C
238      DO 9 I = 1,NIM2
239      KP = KP + 1
240      KO = (J-1)*NI + 1
241      QT = QT + ORBI(M(85)+KP-1)*ORBI(M(65)+KO)*DZ(I+1)*DR4(J)
242 9 CONTINUE
243      WRITE(6,200) TIME
244      WRITE(6,201) NYS,NIT,DT
245      WRITE(6,202) FMI,FHI,FME,FHE,QT
246      DO 10 J = 1,NCM1
247 C
248      WRITE(6,203) J
249      WRITE(6,204)
250 C
251      DO 10 I = 1,NI
252      KI = NI - I + 1
253      KO = (J-1)*NI + KI
254      KP = KO - 2*J + 1

```

```

255      IF(KI.EQ.1) KP = KP + 1
256      IF(KI.EQ.NI)KP = KP - 1
257 C
258      PP = ORBI(M(1)+KO-1)/1.D+05
259      TVP = ORBI(M(3)+KO-1) - 273.14
260      TLP = ORBI(M(5)+KO-1) - 273.14
261      TSP = ORBI(M(67)+KO-1) - 273.14
262      TWP = ORBI(M(68)+KP-1) - 273.14
263      AP = ORBI(M(7)+KO-1)
264      UVZ = ORBI(M(23)+KO-1)
265      ULZ = ORBI(M(24)+KO-1)
266      UVR = ORBI(M(25)+KO-1)
267      ULR = ORBI(M(26)+KO-1)
268 C
269      WRITE(6,205) KI,PP,AP,TVP,TLP,TSP,TWP,
270      *      UVZ,ULZ,UVR,ULR
271 10 CONTINUE
272 C
273      J = NJ
274 C
275      WRITE(6,203) J
276      WRITE(6,206)
277 C
278      DO 11 I = 1,NI
279      KI = NI - I + 1
280      KO = (J-1)*NI + KI
281      KP = KO - 2*J + 1
282      IF(KI.EQ.1) KP = KP + 1
283      IF(KI.EQ.NI)KP = KP - 1
284 C
285      PP = ORBI(M(1)+KO-1)/1.D+05
286      TVP = ORBI(M(3)+KO-1) - 273.14
287      TLP = ORBI(M(5)+KO-1) - 273.14
288      TSP = ORBI(M(67)+KO-1) - 273.14
289      TWP = ORBI(M(68)+KP-1) - 273.14
290      TCP = ORBI(M(91)+KI-1) - 273.14
291      AP = ORBI(M(7)+KO-1)
292      UVZ = ORBI(M(23)+KO-1)
293      ULZ = ORBI(M(24)+KO-1)
294 C
295      WRITE(6,207) KI,PP,AP,TVP,TLP,TSP,TWP,TCP,
296      *      UVZ,ULZ
297 11 CONTINUE
298 C
299      IF(IERR.NE.0) GO TO 6

```

```

300      GO TO 2
301 206 FORMAT(1X,'IZ',5X,'P',10X,'VOID',7X,'TV',8X,'TL',7X,'TSAT',
302 *      5X,'TWALL',7X,'TCAN',6X,'UVZ',9X,'ULZ'/
303 *      6X,'(BAR)',19X,'------(DEGREE CELSIUS)',
304 *      '-----',2(5X,'(M/SEC)')/)
305 207 FORMAT(1X,I2,2X,F9.4,2X,F8.6,5(2X,F8.3),2(2X,F10.5))
306 200 FORMAT(1H1,10X,'FLOW MAP AT TIME = ',F10.4,' SEC.'/)
307 201 FORMAT(1X,'NUMBER OF TIME STEPS = ',I10/
308 *      1X,'NUMBER OF ITERATIONS = ',I10/
309 *      1X,'TIME STEP SIZE = 'D10.4,' SEC.'/)
310 202 FORMAT(1X,'INLET MASS FLOW RATE = ',D12.6,' KG/SEC',
311 *      6X,'INLET ENTHALPY FLOW = ',D12.6,' WATT'/
312 *      1X,'OUTLET MASS FLOW RATE = ',D12.6,' KG/SEC',
313 *      6X,'OUTLET ENTHALPY FLOW = ',D12.6,' WATT'/
314 *      1X,'TOTAL HEAT TRANSFERED = ',D12.6,' WATT'/)
315 203 FORMAT(1H0,40X,'CHANNEL NUMBER ',I5/)
316 204 FORMAT(1X,'IZ',5X,'P',10X,'VOID',7X,'TV',8X,'TL',7X,'TSAT',
317 *      5X,'TWALL',7X,'UVZ',9X,'ULZ',9X,'UVR',9X,'ULR'/
318 *      6X,'(BAR)',19X,'------(DEGREE CELSIUS)-----',
319 *      4(5X,'(M/SEC)')/)
320 205 FORMAT(1X,I2,2X,F9.4,2X,F8.6,4(2X,F8.3),4(2X,F10.5))
321 C
322      END

```

Subroutine read1

```

323 SUBROUTINE READ1
324 IMPLICIT REAL*8 (A-H,O-Z)
325 COMMON /RHEA/ TSET(40),TSHSET(40),DTMAX,DTM1
326 COMMON /REA/ NN,NP,NB,NW,NTR,NPIN,NPM1,NSET(40),NSHSET(40)
327 COMMON /DIM/ DZ(40),DZ1(40),DRO(40),DR1(40),DR2(40),DR3(40),
328 * DR4(40),NI,NJ,NIM1,NIM2,NJM1,NNI,NNJ,NNJJ
329 COMMON /GRVTY/ GZ,GR
330 COMMON /CNTRL/ EPS1,EPS2,RES,IT1,IT2,IT3,ITM1,ITM2,IGAUSS
331 COMMON /GAUSS/ NZ,NR,NZM1
332 COMMON /TEMPO/ TIME,DT,DTO,DTLS,NOT
333 COMMON /ICONST/ NCF,NCC,NG
334 C
335 READ(5,118) NI,NJ,NCF,NCLD
336 WRITE(7,118)NI,NJ,NCF,NCLD
337 NN = NI*NJ
338 NIM1 = NI - 1
339 NIM2 = NI - 2
340 NJM1 = NJ - 1
341 NP = (NI - 2)*NJ
342 NB = 21*NN
343 NW = 24*NN
344 NNJ = NN - NI
345 NNJJ = NNJ - NI
346 NNI = NN - NJ
347 NR = NJ
348 NZ = NI - 2
349 NZM1 = NZ - 1
350 NG = NCF + 1
351 NCC = NG + 1
352 NPIN = NCC + NCLD
353 NPM1 = NPIN - 1
354 NTR = NPIN*NP
355 C
356 C
357 L = 1
358 1 CONTINUE
359 READ(5,121) NSET(L),TSET(L)
360 WRITE(7,121)NSET(L),TSET(L)
361 L = L + 1
362 IF(L.GT.50) GO TO 2
363 IF(NSET(L-1)) 2,2,1
364 2 CONTINUE
365 C

```

```
366      GZ = 9.80665
367      GR = 0.00
368 C
369 C
370      READ(5,119) ITM1,IGAUSS,DTMAX,EPS1,EPS2
371      WRITE(7,119)ITM1,IGAUSS,DTMAX,EPS1,EPS2
372      DT = DTMAX
373 C
374      ITM2 = ITM1
375 118  FORMAT(4I5)
376 119  FORMAT(2I10,3D15.9)
377 120  FORMAT(2D15.9)
378 121  FORMAT(I5,D15.9)
379      RETURN
380      END
```

. Subroutine read2

```

381 SUBROUTINE READ2(P,TV,TL,ALFA,UVZ,ULZ,UVR,ULR,DH,DV,
382 *           QSI,TR,DTR,TINIT,TW,SPPD,TCAN,
383 *           NP,NTR,NPIN,NPM1,NN,NCAN)
384 IMPLICIT REAL*8 (A-H,O-Z)
385 LOGICAL LP,LDATA,LSS
386 COMMON /NUMBER/ ZERO,ONE,BIG,SMALL
387 COMMON /BCOND/ TB(51),PNB1(51),PNB2(51),PNB3(51),OMP(51),
388 *           PNT1(51),PNT2(51),PNT3(51),OMT(51),ALB1(51),
389 *           ALB2(51),ALB3(51),OMA(51),TVB1(51),TVB2(51),
390 *           TVB3(51),OMV(51),TLB1(51),TLB2(51),TLB3(51),
391 *           DML(51),HNW1(51),HNW2(51),HNW3(51),OMH(51),
392 *           LMAX,LP(51)
393 COMMON /PSHAPE/ SHAPE(100)
394 COMMON /DIM/ DZ(40),DZ1(40),DRO(40),DR1(40),DR2(40),DR3(40),
395 *           DR4(40),NI,NJ,NIM1,NIM2,NJM1,NNI,NNJ,NNJJ
396 COMMON /PINO/ RODR(20),VP(20),VM(20),RADR,PPP(20)
397 COMMON /GCONST/ DIL,RADFU,RADCL
398 COMMON /CCONST/ CA0,CA1,CA2,CA3,CB0,CB1,CB2,CB3
399 COMMON /FCONST/ FA0,FA1,FA2,FA3,FB0,FB1,FB2,AD,APU,LPLNM(40)
400 COMMON /ICONST/ NCF,NCC,NG
401 COMMON /PD/ D4,PDD2
402 COMMON /POVERD/ R
403 COMMON /HXCN/ ACDV
404 COMMON /STST/ TAFP,LSS
405 DIMENSION P(NN),TV(NN),TL(NN),ALFA(NN),UVZ(NN),ULZ(NN),
406 *           UVR(NN),ULR(NN),DH(NN),DV(NN),QSI(NN),TR(NTR),
407 *           DTR(NTR),TW(NP),SPPD(NN),TCAN(NCAN)
408 DIMENSION RAD(20),XIN(5),N(20)
409 C
410 FA0 = 1.81D+06
411 FA1 = 3.72D+03
412 FA2 = -2.51D0
413 FA3 = 6.59D-04
414 FB0 = 10.8D0
415 FB1 = -8.84D-03
416 FB2 = 2.25D-06
417 C
418 CA0 = 4.28D+06
419 CA1 = 3.75D+02
420 CA2 = -7.45D-03
421 CA3 = ZERO
422 CB0 = 16.27
423 CB1 = ZERO

```

```

424      CB2 = ZERO
425      CB3 = ZERO
426 C
427 C
428 C
429      TB(1) = ZERO
430      L = 2
431 2 CONTINUE
432      READ(5,1001) LP(L),TB(L)
433      WRITE(7,1001)LP(L),TB(L)
434      IF(TB(L).LE.TB(L-1)) GO TO 3
435      READ(5,1002) PNB1(L),PNB2(L),PNB3(L),OMP(L)
436      READ(5,1002) PNT1(L),PNT2(L),PNT3(L),OMT(L)
437      READ(5,1002) ALB1(L),ALB2(L),ALB3(L),OMA(L)
438      READ(5,1002) TVB1(L),TVB2(L),TVB3(L),OMV(L)
439      READ(5,1002) TLB1(L),TLB2(L),TLB3(L),OML(L)
440      READ(5,1002) HNW1(L),HNW2(L),HNW3(L),OMH(L)
441 C
442      WRITE(7,1002)PNB1(L),PNB2(L),PNB3(L),OMP(L)
443      WRITE(7,1002)PNT1(L),PNT2(L),PNT3(L),OMT(L)
444      WRITE(7,1002)ALB1(L),ALB2(L),ALB3(L),OMA(L)
445      WRITE(7,1002)TVB1(L),TVB2(L),TVB3(L),OMV(L)
446      WRITE(7,1002)TLB1(L),TLB2(L),TLB3(L),OML(L)
447      WRITE(7,1002)HNW1(L),HNW2(L),HNW3(L),OMH(L)
448 C
449      L = L + 1
450      IF(L.GT.51) GO TO 3
451      GO TO 2
452 3 CONTINUE
453      LMAX = L
454      DO 4 KO = 1,NN
455      QSI(KO) = (4.*D/(PITCH - D))*2
456 4 CONTINUE
457 C
458      READ(5,1003) NROW,PITCH,D,E
459      WRITE(7,1003)NROW,PITCH,D,E
460 C
461      POVD = PITCH/D
462      POD2 = POVD*POVD
463      D4 = 4./D
464      R = -16.15 + 24.96*POVD - 8.55*POVD*POVD
465 C
466      READ(5,1004) (N(J),J=1,19)
467      WRITE(7,1004)(N(J),J=1,19)
468      KRES = 0

```

```

469     5 CONTINUE
470     READ(5,1005) LDATA,(XIN(K),K=1,5)
471     WRITE(7,1005) LDATA,(XIN(K),K=1,5)
472     IF(.NOT.LDATA) GO TO 205
473     DO 105 I = 1,5
474     KO = KRES + I
475     IF(KO.GT.NI) GO TO 5
476     DZ(KO) = XIN(I)
477 105 CONTINUE
478     KRES = KRES + 5
479     GO TO 5
480 205 CONTINUE
481 C
482     KRES = 3*NI
483 305 CONTINUE
484     READ(5,1005) LDATA,(XIN(K),K=1,5)
485     WRITE(7,1005) LDATA,(XIN(K),K=1,5)
486     IF(.NOT.LDATA) GO TO 505
487     DO 405 I = 1,5
488     KO = KRES + I
489     IF(KO.GT.NCAN) GO TO 305
490     TCAN(KO) = XIN(I)
491 405 CONTINUE
492     KRES = KRES + 5
493     GO TO 305
494 505 CONTINUE
495     KRES = 0
496     6 CONTINUE
497     READ(5,1005) LDATA,(XIN(K),K=1,5)
498     WRITE(7,1005) LDATA,(XIN(K),K=1,5)
499     IF(.NOT.LDATA) GO TO 206
500     DO 106 I = 1,5
501     KO = KRES + I
502     IF(KO.GT.NN) GO TO 6
503     SHAPE(KO) = XIN(I)
504 106 CONTINUE
505     KRES = KRES + 5
506     GO TO 6
507 206 CONTINUE
508     KRES = 0
509 306 CONTINUE
510     READ(5,1005) LDATA,(XIN(K),K=1,5)
511     WRITE(7,1005) LDATA,(XIN(K),K=1,5)
512     IF(.NOT.LDATA) GO TO 506
513     DO 406 I = 1,5

```

```

514      KO = KRES + I
515      IF(KO.GT.NN) GO TO 306
516      SPPD(KO) = XIN(I)
517 406 CONTINUE
518      KRES = KRES + 5
519      GO TO 306
520 505 CONTINUE
521 C
522      DZ1(1) = DZ(1)
523      DO 7 I = 2,NI
524      DZ1(I) = (DZ(I) + DZ(I-1))/2.D0
525 7 CONTINUE
526 C
527      A1 = DSQRT(3.D0)/2.D0
528      A2 = 3.1415927/4.D0
529      W = PITCH - D
530 C
531      X = (PITCH*PITCH*A1 - (D*D + W*W)*A2)/A2/D
532      XI = 4.D0/X
533 C
534      DO 8 J = 1,NUM1
535      DO 8 I = 1,NI
536      KO = (J-1)*NI + I
537      DH(KO) = X
538      DV(KO) = XI
539 8 CONTINUE
540 C
541      DO 9 J = 2,NUM1
542 C
543      N41 = N(J) - 1
544      N42 = N(J-1) - 1
545      DN4 = N41*N41 - N42*N42
546      DR4(J) = DN4*X*A2*D*3.D0
547 C
548      NX = N(J) - N(J-1)
549      NX1 = 2*N41
550      NX2 = (2*N42 + NX)*NX
551      DNX1 = NX1
552      DR1(J) = DNX1/NX2/PITCH/A1
553      DR2(J) = 2.D0*N42/NX2/PITCH/A1
554      DRO(J) = PITCH*A1*NX
555 9 CONTINUE
556 C
557      DN4 = (N(1) - 1)*(N(1) - 1)
558      DR4(1) = DN4*X*A2*D*3.D0

```

```

559 C      DR1(1) = 2.DO/PITCH/A1/(N(1)-1)
560      DR2(1) = 0.DO
561      DRO(1) = PITCH*A1*(N(1)-1)
562
563 C      B1 = (N(NJM1) + NROW - 2)
564      B2 = (NROW - N(NJM1))
565      B3 = (NROW - 1)
566
567 C      XX = B1*B2/2.DO + B3/2.DO + 1.DO/6.DO
568      PT = B3*PITCH + (D/2.DO + E)/A1 + A2*D*XX*4.DO
569      AC = (B1*PITCH + (D/2.DO + E)/A1)*(B2*PITCH*A1 + D/2.DO + E)*
570      *      0.500 - A2*(D*D + E*E)*XX
571      Y = 4.DO*AC/PT
572      PP = A2*D*XX*4.DO
573      YY = PP/AC
574      ARM = (ONE - A2/A1*(D*D + W*W))/(PITCH*PITCH)*
575      *      (N(NJM1) - 1)*PITCH
576
577 C      DR1(NJ) = ZERO
578      DR2(NJ) = ARM/AC
579      DRO(NJ) = B2*PITCH + D/2.DO + E
580      DR4(NJ) = AC*6.DO
581      ACOV = (B3*PITCH + (D/2.DO + E)/A1)/AC
582
583 C      DO 10 I = 1,NI
584      KO = NJM1*NI + I
585      DH(KO) = Y
586      DV(KO) = YY
587
588      10 CONTINUE
589
589 C      DR3(NJ) = DRO(NJ)
590      DO 11 J = 1,NJM1
591      DR3(J) = (DRO(J) + DRO(J+1))/2.DO
592
593      11 CONTINUE
594      KRES = 0
595
595      12 CONTINUE
596      READ(5,1005) LDATA,(XIN(K),K=1,5)
597      WRITE(7,1005) LDATA,(XIN(K),K=1,5)
598      IF(.NOT.LDATA) GO TO 212
599      DO 112 I = 1,5
600      KO = KRES + I
601      IF(KO.GT.NPIN) GO TO 12
602      PPP(KO) = XIN(I)
603      112 CONTINUE

```

```

604      KRES = KRES + 5
605      GO TO 12
606 212 CONTINUE
607 C
608      READ(5,1006) AD,APU,DIL
609      READ(5,1007) (LPLNM(K),K = 1,39)
610      READ(5,1008) RADR,THC,THG
611 C
612      WRITE(7,1006)AD,APU,DIL
613      WRITE(7,1007)(LPLNM(K),K = 1,39)
614      WRITE(7,1008)RADR,THC,THG
615 C
616      RADFU = RADR - THG - THC
617      RADCL = RADFU + THG
618      NCLD = NPIN - NCC
619      DRF = RADFU/NCF
620      DRC = THC/NCLD
621      TAFP = RADFU*RADFU/D
622 C
623      RAD(1) = ZERO
624      DO 14 K = 1,NCF
625      RAD(K+1) = RAD(K) + DRF
626 14 CONTINUE
627      RAD(NG+1) = RAD(NG) + THG
628      DO 15 K = NCC,NPM1
629      RAD(K+1) = RAD(K) + DRC
630 15 CONTINUE
631      DO 16 K = 1,NPM1
632      IF(K.EQ.NG) RODR(K) = (RAD(K+1) + RAD(K))/2.00
633      IF(K.NE.NG) RODR(K) = (RAD(K+1)+RAD(K))/(RAD(K+1)-RAD(K))/2.00
634 16 CONTINUE
635 C
636      VM(1) = ZERO
637      VP(1) = DRF*DRF/8.00
638      RM = (RADR + RAD(NPM1))/2.00
639      VM(NPIN) = (RADR*RADR + W*W/4.00 - RM*RM)/2.00
640      VP(NPIN) = ZERO
641      DO 17 K = 2,NPM1
642      RP = (RAD(K+1) + RAD(K))/2.00
643      RM = (RAD(K) + RAD(K-1))/2.00
644      VP(K) = (RP*RP - RAD(K)*RAD(K))/2.00
645      VM(K) = (RAD(K)*RAD(K) - RM*RM)/2.00
646 17 CONTINUE
647 C
648      READ(5,1009) LSS,TINIT

```

```

649     TB(1) = ZERO
650     IF (LSS) GO TO 19
651     DO 1 KO = 1,NN
652     READ(5,1000) KCHECK,TV(KO),TL(KO),P(KO),ALFA(KO)
653     READ(5,1000) KCHECK,UVZ(KO),ULZ(KO),UVR(KO),ULR(KO)
654     IF(KCHECK.EQ.KO)GO TO 1
655     IERR = 4
656     RETURN
657     1 CONTINUE
658     KRES = 0
659     13 CONTINUE
660     READ(5,1005) LDATA,(XIN(K),K=1,5)
661     IF(.NOT.LDATA) GO TO 213
662     DO 113 I = 1,5
663     KO = KRES + I
664     IF(KO.GT.NTR) GO TO 13
665     TR(KO) = XIN(I)
666     113 CONTINUE
667     KRES = KRES + 5
668     GO TO 13
669     213 CONTINUE
670 C
671     KRES = 0
672     313 CONTINUE
673     READ(5,1005) LDATA,(XIN(K),K=1,5)
674     IF(.NOT.LDATA) GO TO 513
675     DO 413 I = 1,5
676     KO = KRES + I
677     K3 = KO + 2*NI
678     IF(KO.GT.NI) GO TO 313
679     TCAN(KO) = XIN(I)
680     TCAN(K3) = XIN(I)
681     413 CONTINUE
682     KRES = KRES + 5
683     GO TO 313
684     513 CONTINUE
685 C
686     DO 18 I = 1,NIM2
687     DO 18 J = 1,NJ
688     KP = (J-1)*NIM2 + I
689     KT = KP*NPIN
690     YW(KP) = YR(KT)
691     18 CONTINUE
692     RETURN
693     19 CONTINUE

```

```
694 READ(5,1010) PIN,POUT,TIN,TAV
695 QPP = HNW2(2)*RADFU*RADFU/RADR/2.DO
696 CALL SS (PIN,POUT,TIN,TAV,QPP,P,TV,TL,UVZ,ULZ,UVR,ULR,ALFA,
697 * TW,TR,DTR,OH,DV,NN,NP,NTR,NPIN,NPM1)
698 RETURN
699 C
700 1000 FORMAT(I5,4D15.9)
701 1001 FORMAT(L1,F15.5)
702 1002 FORMAT(4D15.9)
703 1003 FORMAT(I5,3D15.9)
704 1004 FORMAT(19I4)
705 1005 FORMAT(L1,5D15.9)
706 1006 FORMAT(3D15.9)
707 1007 FORMAT(39I2)
708 1008 FORMAT(3D15.9)
709 1009 FORMAT(L1,D15.9)
710 1010 FORMAT(4D15.9)
711 END
```

Subroutine ss

```

712 SUBROUTINE SS(PIN,POUT,TIN,TAV,Q,P,TV,TL,UVZ,ULZ,UVR,ULR,ALFA,
713 * TW,TR,DTR,DH,DV,NN,NP,NTR,NPIN,NPM1)
714 IMPLICIT REAL*8 (A-H,O-Z)
715 COMMON /NUMBER/ ZERO,ONE,BIG,SMALL
716 COMMON /DIM/ DZ(40),DZ1(40),DRO(40),DR1(40),DR2(40),DR3(40),
717 * DR4(40),NI,NJ,NIM1,NIM2,NJM1,NNI,NNJ,NNJJ
718 COMMON /PSHAPE/ SHAPE(100)
719 COMMON /GRVTY/ GZ,GR
720 DIMENSION PROP(3,4)
721 DIMENSION P(NN),TV(NN),TL(NN),UVZ(NN),ULZ(NN),UVR(NN),ULR(NN),
722 * ALFA(NN),TW(NP),TR(NTR),DTR(NTR),DH(NN),DV(NN)
723 C
724 C SUBROUTINE SS PUTS AN INITIAL GUESS IN THE VARIABLES
725 C TV,TL,P,UVZ,ULZ,UVR,ULR,ALFA AND TR, IN ORDER TO
726 C ACCELERATE THE CONVERGENCE TO THE STEADY STATE PROBLEM.
727 C
728 H = ZERO
729 DO 1 I = 2,NI
730 H = H + DZ1(I)
731 1 CONTINUE
732 DP = (PIN - POUT)/H
733 C
734 CALL STATE (TAV,TAV,PIN,PROP,0)
735 RHO = PROP(1,2)
736 DPG = DP - RHO*GZ
737 C
738 A = (RHO*DH(2)/VISCL(TAV))**.2*DH(2)/RHO/.100
739 X = ONE/1.800
740 V = (A*DPG)**X
741 C
742 DO 2 J = 1,NJM1
743 DO 2 I = 1,NI
744 KO = (J-1)*NI + I
745 ULZ(KO) = V
746 UVZ(KO) = V
747 ULR(KO) = ZERO
748 UVR(KO) = ZERO
749 ALFA(KO) = ZERO
750 2 CONTINUE
751 C
752 A = (RHO*DH(NNJ+2)/VISCL(TAV))**.2*DH(NNJ+2)/RHO/.100
753 V = (A*DPG)**X
754 C

```

```

755      DO 3 I = 1,NI
756      KO = NNU + I
757      ULZ(KO) = V
758      UVZ(KO) = V
759      ULR(KO) = ZERO
760      UVR(KO) = ZERO
761      ALFA(KO) = ZERO
762      3 CONTINUE
763 C
764      TL(1) = TIN
765      TV(1) = TIN
766      P(1) = PIN
767      DO 4 J = 1,NU
768      KO = J*NI - NIM1
769      TL(KO) = TIN
770      TV(KO) = TIN
771      P(KO) = PIN
772 C
773      DO 4 I = 2,NI
774      KO = (J-1)*NI + I
775      P(KO) = P(KO-1) - DP*DZ1(I)
776      UXX = ULZ(KO)
777      IF(UXX.EQ.ZERO) UXX = ONE
778      TL(KO) = TL(KO-1) + Q*SHAPE(KO)*DV(KO)*DZ1(I)/RHO/UXX/
779      /      CPL(TL(KO-1))
780      TV(KO) = TL(KO)
781      4 CONTINUE
782 C
783      DT = .100
784 C
785      DO 7 J = 1,NU
786      DO 7 I = 2,NIM1
787      KO = (J-1)*NI + I
788      KP = KO + 1 - J*2
789      KT = (KP-1)*NPIN + 1
790      KR = KP*NPIN
791 C
792      TW(KP) = TL(KO)
793      TS = SAT(P(KO))
794      CALL HTCFC (P(KO),TV(KO),TL(KO),ALFA(KO),PROP(1,1),
795      *      PROP(1,2),PROP(1,3),PROP(1,4),DH(KO),TS,TW(KP),
796      *      HCONV,HCONL,HNB,UVZ(KO),ULZ(KO))
797 C
798      DO 5 K = 1,NPIN
799      KTR = (KP-1)*NPIN + K

```

```

800     TR(KTR) = TW(KP)
801     5 CONTINUE
802     6 CONTINUE
803     TTR = TR(KT)
804     CALL FPROP(TR(KT),NPIN,NPM1,I)
805     CALL FPIN (TV(KO),TL(KO),TS,TW(KP),DTW,HCONV,HCONL,HNB,
806     *          TR(KT),DTR(KT),DT,NPIN,NPM1,KO)
807 C
808     TR(KR) = TW(KP)
809     DO 16 KK = 1,NPM1
810     KS = KR - KK
811     TR(KS) = TR(KS) - DTR(KS)*TR(KS+1)
812 16 CONTINUE
813     TTR = DABS(TTR - TR(KT))/DT
814     IF(TTR.GT.ONE) GO TO 6
815     7 CONTINUE
816     RETURN
817     END

```

Subroutine tmstep

```

818 SUBROUTINE TMSTEP(O,NO,
819 * NN, NP, NB, NW, NTR, NPIN, NPM1, NCAN)
820 IMPLICIT REAL*8 (A-H,O-Z)
821 COMMON /ERROR/ IERR
822 COMMON /NUMBER/ ZERO, ONE, BIG, SMALL
823 COMMON /RHEA/ TSET(40), TSHSET(40), DTMAX, DTM1
824 COMMON /CNTRL/ EPS1, EPS2, RES, IT1, IT2, IT3, ITM1, ITM2, IGAUSS
825 COMMON /DIM/ DZ(40), DZ1(40), DRO(40), DR1(40), DR2(40), DR3(40),
826 * DR4(40), NI, NJ, NIM1, NIM2, NJM1, NNI, NNJ, NNJJ
827 COMMON /TEMPO/ TIME, DT, DTO, DTLS, NDT
828 COMMON /PNTR1/ K(100), M(100)
829 C
830 C
831 DIMENSION O(NO)
832 C
833 C
834 C
835 DTLS = DT
836 TMS = ZERO
837 IERR = 0
838 DO 100 J = 1, NJ
839 DO 100 I = 2, NI
840 KO = (J-1)*NI + I - 1
841 K23 = KO + M(23)
842 K24 = KO + M(24)
843 K25 = KO + M(25)
844 JO = J
845 C
846 TSVZ = DABS(O(K23)/DZ1(I))
847 TSLZ = DABS(O(K24)/DZ1(I))
848 TSVR = DABS(O(K25)/DR3(J))
849 C
850 TMS = DMAX1(TSVZ, TSLZ, TSVR, TMS)
851 100 CONTINUE
852 C
853 IF(TMS) 101, 101, 102
854 101 DT = DTMAX
855 GO TO 103
856 102 DT = 0.9500/TMS
857 DT = DMIN1(DTMAX, DT, 2.0*DTLS)
858 103 CONTINUE
859 IT2 = 0
860 NDT = 0

```

```

861     TIME = TIME+DT
862 C
863     DO 104 KO = 1,NN
864     KL = KO - 1
865     K68 = M(68) + KL
866     K82 = M(82) + KL
867     O(K82) = O(K68)
868 104 CONTINUE
869 C
870     CALL DONOR(O(M( 1)),O(M( 2)),O(M( 3)),O(M( 4)),O(M( 5)),
871 *           O(M( 6)),O(M( 7)),O(M( 8)),O(M( 9)),O(M(10)),
872 *           O(M(11)),O(M(12)),O(M(13)),O(M(14)),O(M(15)),
873 *           O(M(16)),O(M(17)),O(M(18)),O(M(19)),O(M(20)),
874 *           O(M(21)),O(M(22)),O(M(23)),O(M(24)),O(M(25)),
875 *           O(M(26)),O(M(27)),O(M(28)),O(M(29)),O(M(30)),
876 *           O(M(31)),O(M(32)),O(M(33)),O(M(34)),O(M(41)),
877 *           O(M(42)),O(M(47)),O(M(48)),O(M(52)),O(M(53)),
878 *           O(M(58)),O(M(59)),O(M(73)),O(M(86)),O(M(87)),
879 *           O(M(88)),O(M(89)),NN,NP)
880 C
881 1 CONTINUE
882     CALL BC(O(M( 1)),O(M( 3)),O(M( 5)),O(M( 7)),TIME,
883 *           O(M(24)),NN,NI,NIM1)
884     CALL WS(O(M( 2)),O(M( 4)),O(M( 6)),O(M( 8)),O(M( 9)),
885 *           O(M(10)),O(M(11)),O(M(12)),O(M(13)),O(M(14)),
886 *           O(M(15)),O(M(16)),O(M(17)),O(M(18)),O(M(27)),
887 *           O(M(28)),O(M(29)),O(M(30)),O(M(39)),O(M(40)),
888 *           O(M(43)),O(M(44)),O(M(45)),O(M(46)),O(M(47)),
889 *           O(M(48)),O(M(49)),O(M(50)),O(M(51)),
890 *           O(M(54)),O(M(55)),O(M(56)),O(M(57)),O(M(58)),
891 *           O(M(59)),O(M(60)),O(M(61)),O(M(62)),O(M(63)),
892 *           O(M(65)),O(M(66)),O(M(90)),NN)
893 C
894     DO 1001 I = 2,NIM1
895     DO 1001 J = 1,NJ
896     KO = (J-1)*NI + I - 1
897     KP = KO + 1 -J*2
898     KT = KP*NPIN .
899     K01 = M( 1) + KO
900     K03 = M( 3) + KO
901     K05 = M( 5) + KO
902     K07 = M( 7) + KO
903     K11 = M(11) + KO
904     K12 = M(12) + KO
905     K17 = M(17) + KO

```

```

906      K18 = M(18) + KQ
907      K23 = M(23) + KQ
908      K24 = M(24) + KQ
909      K63 = M(63) + KQ
910      K67 = M(67) + KQ
911      K68 = M(68) + KP
912      K69 = M(69) + KP
913      K70 = M(70) + KP
914      K71 = M(71) + KP
915      K72 = M(72) + KP
916      K80 = M(80) + KT
917      K81 = M(81) + KT
918      KF = KQ + 1
919 C
920      UV = (O(K23) + O(K23 + 1))/2.00
921      UL = (O(K24) + O(K24 + 1))/2.00
922 C
923      CALL HTCFC(O(K01),O(K03),O(K05),O(K07),O(K11),O(K12),
924 *              O(K17),O(K18),O(K63),O(K67),O(K68),O(K70),
925 *              O(K71),O(K72),UV,UL)
926      CALL FPROP(O(K80),NPIN,NPM1,1)
927      CALL FPIN(O(K03),O(K05),O(K67),O(K68),O(K69),O(K70),
928 *             O(K71),O(K72),O(K80),O(K81),DT,NPIN,NPM1,KF)
929 1001 CONTINUE
930 C
931      CALL THXCN(O(M(3)),O(M(5)),O(M(70)),O(M(71)),O(M(91)),
932 *             DT,NN,NI,NJ,NCAN,NIM1,NIM2)
933 C
934      IF(IERR.NE.0) RETURN
935      IT2 = 0
936      2 CONTINUE
937      IT2 = IT2+1
938      CALL ONESTP(O(M( 1)),O(M( 2)),O(M( 3)),O(M( 5)),O(M( 7)),
939 *              O(M( 8)),O(M( 9)),O(M(10)),O(M(11)),O(M(12)),
940 *              O(M(17)),O(M(18)),O(M(19)),O(M(20)),O(M(21)),
941 *              O(M(22)),O(M(23)),O(M(24)),O(M(25)),O(M(26)),
942 *              O(M(35)),O(M(36)),O(M(37)),O(M(38)),O(M(39)),
943 *              O(M(65)),O(M(67)),O(M(68)),O(M(69)),O(M(70)),
944 *              O(M(71)),O(M(72)),O(M(73)),O(M(74)),O(M(75)),
945 *              O(M(76)),O(M(77)),O(M(78)),O(M(79)),O(M(83)),
946 *              O(M(84)),O(M(86)),O(M(87)),O(M(88)),O(M(89)),
947 *              O(M(91)),DT,NN,NB,NP,NW,NCAN)
948      IF(IERR.NE.0) GO TO 5
949      IF(RES.GT.EPS1) GO TO 4
950      IT3 = IT3 + IT2

```

```

951 CALL FTP(O(M( 3)),O(M( 5)),O(M(67)),O(M(68)),O(M(70)),
952 * O(M(71)),O(M(72)),O(M(80)),O(M(81)),O(M(85)),
953 * NI,NJ,NN,NP,NTR,NPM1,NIM2,NPIN)
954 CALL THXCNO(O(M(91)),NCAN,NI)
955 RETURN
956 4 IF(IT2.LT.ITM2) GO TO 2
957 5 CONTINUE
958 NDT = NDT+1
959 IT3 =IT3+IT2
960 IT2 = 0
961 TIME = TIME-DT
962 DTD = DT
963 DT = DT*0.1
964 IF(DT.LT.1.D-07) IERR = 21
965 TIME = TIME+DT
966 DO 6 KO = 1,NN
967 KL = KO - 1
968 K01 = M( 1) + KL
969 K02 = M( 2) + KL
970 K03 = M( 3) + KL
971 K04 = M( 4) + KL
972 K05 = M( 5) + KL
973 K06 = M( 6) + KL
974 K07 = M( 7) + KL
975 K08 = M( 8) + KL
976 K23 = M(23) + KL
977 K24 = M(24) + KL
978 K25 = M(25) + KL
979 K26 = M(26) + KL
980 K27 = M(27) + KL
981 K28 = M(28) + KL
982 K29 = M(29) + KL
983 K30 = M(30) + KL
984 K73 = M(73) + KL
985 K68 = M(68) + KL
986 K82 = M(82) + KL
987 C
988 O(K03) = O(K04)
989 O(K05) = O(K06)
990 O(K07) = O(K08)
991 O(K23) = O(K27)
992 O(K24) = O(K28)
993 O(K25) = O(K29)
994 O(K26) = O(K30)
995 O(K01) = O(K02)

```

```
996      O(K73) = ZERO
997      O(K68) = O(K82)
998      6 CONTINUE
999      CALL FTP(O(M( 3)),O(M( 5)),O(M(67)),O(M(68)),O(M(70)),
1000     *      O(M(71)),O(M(72)),O(M(80)),O(M(81)),O(M(85)),
1001     *      NI,NJ,NN,NP,NTR,NPM1,NIM2,NPIN)
1002     IF(IERR.GT.20) RETURN
1003     IF(NDT.GT.3) RETURN
1004     IERR = 0
1005     GO TO 1
1006     END
```

Subroutine donor

```

1007 SUBROUTINE DONOR(P,PO,TV,TVO,TL,TLO,ALFAN,ALFAO,ALFAZ,ALFAR,
1008 * RHOV,RHOL,RHOVZ,RHOLZ,RHOVR,RHOLR,
1009 * HV,HL,HVZ,HLZ,HVR,HLR,UVZN,
1010 * ULZN,UVRN,ULRN,UVZO,ULZO,UVRO,ULRO,
1011 * UVRZ,ULRZ,UVZR,ULZR,WZ1,WZ2,
1012 * WZ7,WZ8,WR1,WR2,WR7,WR8,DPN,AVZD,ALZD,
1013 * AVRD,ALRD,NN,NP)
1014 IMPLICIT REAL*8 (A-H,O-Z)
1015 COMMON /NUMBER/ ZERO,ONE,BIG,SMALL
1016 COMMON /GRVTY/ GZ,GR
1017 COMMON /DIM/ DZ(40),DZ1(40),DRO(40),DR1(40),DR2(40),DR3(40),
1018 * DR4(40),NI,NJ,NIM1,NIM2,NJM1,NNI,NNJ,NNJJ
1019 COMMON /TEMPO/ TIME,DT,DTO,DTLS,NDT
1020 DIMENSION P(NN),PO(NN),TV(NN),TVO(NN),TL(NN),TLO(NN),ALFAN(NN),
1021 * ALFAO(NN),ALFAZ(NN),ALFAR(NN),RHOV(NN),RHOL(NN),
1022 * RHOVZ(NN),RHOLZ(NN),RHOVR(NN),RHOLR(NN),
1023 * HV(NN),HL(NN),HVZ(NN),HLZ(NN),HVR(NN),HLR(NN),
1024 * UVZN(NN),ULZN(NN),UVRN(NN),ULRN(NN),UVZO(NN),
1025 * ULZO(NN),UVRO(NN),ULRO(NN),UVRZ(NN),ULRZ(NN),
1026 * UVZR(NN),ULZR(NN),WZ1(NN),WZ2(NN),WZ7(NN),WZ8(NN),
1027 * WR1(NN),WR2(NN),WR7(NN),WR8(NN),DPN(NN),
1028 * AVZD(NN),ALZD(NN),AVRD(NN),ALRD(NN)
1029 DIMENSION PROP(3,4),S(5,2)
1030 C
1031 IFLAG = 0
1032 DTR = DT/DTLS
1033 DO 101 KO = 1,NN
1034 CALL STATE (TV(KO),TL(KO),P(KO),PROP,IFLAG)
1035 C
1036 IF(ALFAN(KO).GT.1.D-08) GO TO 100
1037 ALFAN(KO) = ZERO
1038 TV(KO) = TL(KO)
1039 100 CONTINUE
1040 TVO(KO) = TV(KO)
1041 TLO(KO) = TL(KO)
1042 PO(KO) = P(KO)
1043 RHOV(KO) = PROP(1,1)
1044 RHOL(KO) = PROP(1,2)
1045 HV(KO) = PROP(1,3)
1046 HL(KO) = PROP(1,4)
1047 ALFAO(KO) = ALFAN(KO)
1048 C
1049 IF(DABS(UVRN(KO)).LT.1.D-10) UVRN(KO) = ZERO

```

```

1050      IF(DABS(ULRN(KO)).LT.1.D-10) ULRN(KO) = ZERO
1051      UVZO(KO) = UVZN(KO)
1052      UVRO(KO) = UVRN(KO)
1053      ULZO(KO) = ULZN(KO)
1054      ULRO(KO) = ULRN(KO)
1055      101 CONTINUE
1056      DO 1101 J = 1,NJ
1057      DO 1101 I = 2,NIM1
1058      KO = (J-1)*NI + I
1059      KP = (J-1)*NIM1 - J + I
1060      1101 DPN(KP) = P(KO) - PD(KO)
1061      DO 2101 I = 2,NI
1062      II = I - 1
1063      DO 2101 J = 1,NJM1
1064      JJ = J + 1
1065      KO = (J - 1)*NI + I
1066      KI = KO - 1
1067      KJ = KO + NI
1068      C
1069      DZM = DZ(I) + DZ(II)
1070      DRM = DRO(J) + DRO(JJ)
1071      C
1072      ALFAZ(KO) = (ALFAO(KO)*DZ(I) + ALFAO(KI)*DZ(II))/DZM
1073      RHOVZ(KO) = (RHOV(KO)*DZ(I) + RHOV(KI)*DZ(II))/DZM
1074      RHOLZ(KO) = (RHOL(KO)*DZ(I) + RHOL(KI)*DZ(II))/DZM
1075      C
1076      ALFAR(KO) = (ALFAO(KO)*DRO(J) + ALFAO(KJ)*DRO(JJ))/DRM
1077      RHOVR(KO) = (RHOV(KO)*DRO(J) + RHOV(KJ)*DRO(JJ))/DRM
1078      RHOLR(KO) = (RHOL(KO)*DRO(J) + RHOL(KO)*DRO(JJ))/DRM
1079      2101 CONTINUE
1080      C
1081      DO 3101 J = 1,NJ
1082      KO = (J - 1)*NI + 1
1083      ALFAZ(KO) = ALFAO(KO)
1084      RHOVZ(KO) = RHOV(KO)
1085      RHOLZ(KO) = RHOL(KO)
1086      3101 CONTINUE
1087      C
1088      DO 4101 I = 2,NI
1089      KO = NNJ + I
1090      II = I - 1
1091      KI = KO - 1
1092      DZM = DZ(I) + DZ(II)
1093      C
1094      ALFAZ(KO) = (ALFAO(KO)*DZ(I) + ALFAO(KI)*DZ(II))/DZM

```

```

1095      RHOVZ(KO) = (RHOV(KO)*DZ(I) + RHOV(KI)*DZ(II))/DZM
1096      RHOLZ(KO) = (RHOL(KO)*DZ(I) + RHOL(KI)*DZ(II))/DZM
1097 4101 CONTINUE
1098      DO 102 J = 2,NJM1
1099      DO 102 I = 2,NIM1
1100      KO = (J-1)*NI+I
1101 C
1102      UVRZ(KO) = (UVRO(KO)+UVRO(KO-1)+UVRO(KO-NI)+UVRO(KO-1-NI))/4.
1103      ULRZ(KO) = (ULRO(KO)+ULRO(KO-1)+ULRO(KO-NI)+ULRO(KO-1-NI))/4.
1104      UVZR(KO) = (UVZO(KO)+UVZO(KO+1)+UVZO(KO+NI)+UVZO(KO+1+NI))/4.
1105      ULZR(KO) = (ULZO(KO)+ULZO(KO+1)+ULZO(KO+NI)+ULZO(KO+1+NI))/4.
1106 C
1107      KD = 0
1108      IF(UVZO(KO).GE.ZERO) KD = -1
1109      KN = KO+KO
1110      IO = I +KD
1111 C
1112      HVZ(KO) = HV(KN)
1113      AVZD(KO) = ALFAO(KN)
1114      WZ1(KO) = ALFAO(KN)*RHOV(KN)
1115      WZ7(KO) = (UVZO(KN+1)-UVZO(KN))/DZ(IO)*UVZO(KO)
1116 C
1117      KD = 0
1118      IF(ULZO(KO).GE.ZERO) KD = -1
1119      IO = I+KD
1120      KN = KO+KO
1121      HLZ(KO) = HL(KN)
1122      ALZD(KO) = ONE - ALFAO(KN)
1123      WZ2(KO) = (ONE-ALFAO(KN))*RHOL(KN)
1124      WZ8(KO) = (ULZO(KN+1)-ULZO(KN))/DZ(IO)*ULZO(KO)
1125 C
1126 C
1127      KD = NI
1128      IF(UVRO(KO).GE.ZERO) KD = 0
1129      JO = J + KO/NI
1130      KN = KO + KD
1131 C
1132      HVR(KO) = HV(KN)
1133      AVRD(KO) = ALFAO(KN)
1134      WR1(KO) = ALFAO(KN)*RHOV(KN)
1135      WR7(KO) = (UVRO(KN)-UVRO(KN-NI))/DRO(JO)*UVRO(KO)
1136      IF(J.EQ.NJM1) WR7(KO) = -UVRO(KN-NI)*UVRO(KO)/DRO(JO)
1137 C
1138 C
1139      KD = NI

```

```

1140 IF(ULRO(KO).GE.ZERO) KD = 0
1141 JO = J + KD/NI
1142 KN = KO+KD
1143 C
1144 HLR(KO) = HL(KN)
1145 ALRD(KO) = ONE - ALFAO(KN)
1146 WR2(KO) = (ONE-ALFAO(KN))*RHOL(KN)
1147 WR8(KO) = (ULRO(KN)-ULRO(KN-NI))/DRD(JO)*ULRO(KO)
1148 IF(J.EQ.NUM1) WR8(KO) = -ULRO(KN-NI)*ULRO(KO)/DRD(JO)
1149 C
1150 C
1151 KD = NI
1152 IF(UVRZ(KO).GE.ZERO) KD = 0
1153 KN = KO + KD
1154 JO = J - 1 + KD/NI
1155 C
1156 WZ7(KO) = ((UVZO(KN)-UVZO(KN-NI))*UVRZ(KO)/DR3(JO) +
1157 + WZ7(KO) + GZ)*ALFAZ(KO)*RHOVZ(KO)
1158 C
1159 KD = NI
1160 IF(ULRZ(KO).GE.ZERO) KD = 0
1161 KN = KO + KD
1162 JO = J - 1 + KD/NI
1163 C
1164 WZ8(KO) = ((ULZO(KN)-ULZO(KN-NI))*ULRZ(KO)/DR3(JO) +
1165 + WZ8(KO) + GZ)*(ONE-ALFAZ(KO))*RHOLZ(KO)
1166 C
1167 KD = 0
1168 IF(UVZR(KO).GE.ZERO) KD = -1
1169 KN = KO + KD
1170 IO = I + KD + 1
1171 C
1172 WR7(KO) = ((UVRO(KN+1)-UVRO(KN))*UVZR(KO)/DZ1(IO) +
1173 + WR7(KO) + GR)*ALFAR(KO)*RHOVR(KO)
1174 C
1175 KD = 0
1176 IF(ULZR(KO).GE.ZERO) KD = -1
1177 KN = KO + KD
1178 IO = I + KD + 1
1179 C
1180 WR8(KO) = ((ULRO(KN+1)-ULRO(KN))*ULZR(KO)/DZ1(IO) +
1181 + WR8(KO) + GR)*(ONE-ALFAR(KO))*RHOLR(KO)
1182 + 102 CONTINUE
1183 C
1184 C TOP CELLS

```

```

1185 C
1186 DO 103 KO = NI,NN,NI
1187 KD = 0
1188 IF(UVZO(KO).GE.ZERO) KD = -1
1189 KN = KO+KD
1190 IO = NI+KD
1191 C
1192 HVZ(KO) = HV(KN)
1193 AVZD(KO) = ALFAO(KN)
1194 WZ1(KO) = ALFAO(KN)*RHOV(KN)
1195 WZ7(KO) = ((UVZO(KN+1)-UVZO(KN))/DZ(IO)*UVZO(KO)+GZ)*
1196 * ALFAZ(KO)*RHOVZ(KO)
1197 C
1198 C
1199 KD = 0
1200 IF(ULZO(KO).GE.ZERO) KD = -1
1201 KN = KO+KD
1202 IO = NI+KD
1203 C
1204 HLZ(KO) = HL(KN)
1205 ALZD(KO) = ONE - ALFAO(KN)
1206 WZ2(KO) = (ONE-ALFAO(KN))*RHOL(KN)
1207 WZ8(KO) = ((ULZO(KN+1)-ULZO(KN))/DZ(IO)*ULZO(KO)+GZ)*
1208 * (ONE-ALFAZ(KO))*RHOLZ(KO)
1209 C
1210 C
1211 103 CONTINUE
1212 C
1213 C THE CENTERLINE CELLS
1214 C
1215 DO 110 KO = 2,NIM1
1216 C
1217 UVRZ(KO) = (UVRO(KO)+UVRO(KO-1))/4.
1218 ULRZ(KO) = (ULRO(KO)+ULRO(KO-1))/4.
1219 UVZR(KO) = (UVZO(KO)+UVZO(KO+1)+UVZO(KO+NI)+UVZO(KO+1+NI))/4.
1220 ULZR(KO) = (ULZO(KO)+ULZO(KO+1)+ULZO(KO+NI)+ULZO(KO+1+NI))/4.
1221 C
1222 KD = 0
1223 IF(UVZO(KO).GE.ZERO) KD = -1
1224 KN = KO+KD
1225 IO = KO +KD
1226 C
1227 HVZ(KO) = HV(KN)
1228 AVZD(KO) = ALFAO(KN)
1229 WZ1(KO) = ALFAO(KN)*RHOV(KN)

```

```

1230      WZ7(KO) = (UVZO(KN+1)-UVZO(KN))/DZ(IO)*UVZO(KO)
1231 C
1232      KD = 0
1233      IF(ULZO(KO).GE.ZERO) KD = -1
1234      IO = KO + KD
1235      KN = KO+KD
1236      HLZ(KO) = HL(KN)
1237      ALZO(KO) = ONE - ALFAO(KN)
1238      WZ2(KO) = (ONE-ALFAO(KN))*RHOL(KN)
1239      WZ8(KO) = (ULZO(KN+1)-ULZO(KN))/DZ(IO)*ULZO(KO)
1240 C
1241 C
1242      KD = 0
1243      IF(UVZR(KO).GE.ZERO) KD = -1
1244      KN = KO + KD
1245      IO = KO + KD + 1
1246      WR7(KO) = (UVRO(KN+1)-UVRO(KN))*UVZR(KO)/DZ1(IO)
1247 C
1248      KD = 0
1249      IF(ULZR(KO).GE.ZERO) KD = -1
1250      KN = KO + KD
1251      IO = KO + KD + 1
1252      WR8(KO) = (ULRO(KN+1)-ULRO(KN))*ULZR(KO)/DZ1(IO)
1253 C
1254      IF(UVRO(KO))104,105,105
1255 104 KN = KO+NI
1256      JO = 2
1257 C
1258      HVR(KO) = HV(KN)
1259      AVR0(KO) = ALFAO(KN)
1260      WR1(KO) = ALFAO(KN)*RHOV(KN)
1261      WR7(KO) = ((UVRO(KN) - UVRO(KO))/DRO(JO)*UVRO(KO)+
1262 +          WR7(KO)+GR)*ALFAR(KO)*RHOVR(KO)
1263 C
1264 C
1265      GO TO 106
1266 105 HVR(KO) = HV(KO)
1267      AVR0(KO) = ALFAO(KO)
1268      WR1(KO) = ALFAO(KO)*RHOV(KO)
1269      WR7(KO) = (UVRO(KO)/DRO(1)*UVRO(KO)+WR7(KO)+GR)*
1270 *          ALFAR(KO)*RHOVR(KO)
1271 C
1272 C
1273 106 CONTINUE
1274      IF(ULRO(KO)) 107,108,108

```

```

1275 107 KN = KO+NI
1276      JO = 2
1277 C
1278      HLR(KO) = HL(KN)
1279      ALRD(KO) = ONE - ALFAO(KN)
1280      WR2(KO) = (ONE-ALFAO(KN))*RHOL(KN)
1281      WR8(KO) = ((ULRO(KN) - ULRO(KO))/DR0(JO)*ULRO(KO)+
1282 +          WR8(KO)+GR)*(ONE-ALFAR(KO))*RHOLR(KO)
1283 C
1284 C
1285      GO TO 109
1286 108 HLR(KO) = HL(KO)
1287      ALRD(KO) = ONE - ALFAO(KO)
1288      RHOLR(KO) = RHOL(KO)
1289      WR2(KO) = (ONE-ALFAO(KO))*RHOLR(KO)
1290      WR8(KO) = (ULRO(KO)/DR0(1)*ULRO(KO)+WR8(KO)+GR)*WR2(KO)
1291 C
1292 C
1293 109 CONTINUE
1294 C
1295      IF(UVRZ(KO)) 1108,2108,2108
1296 1108 WZ7(KO) = (WZ7(KO) + UVZO(KO+NI)*UVRZ(KO)/DR3(1) +
1297 +          GZ)*ALFAZ(KO)*RHOVZ(KO)
1298      GO TO 3108
1299 2108 WZ7(KO) = (WZ7(KO) + UVZO(KO)*UVRZ(KO)/DR3(1) +
1300 +          GZ)*ALFAZ(KO)*RHOVZ(KO)
1301 C
1302 3108 CONTINUE
1303      IF(ULRZ(KO)) 4108,5108,5108
1304 4108 WZ8(KO) = (WZ8(KO) + ULZO(KO+NI)*ULRZ(KO)/DR3(1) +
1305 +          GZ)*(ONE-ALFAZ(KO))*RHOLZ(KO)
1306      GO TO 6108
1307 5108 WZ8(KO) = (WZ8(KO) + ULZO(KO)*ULRZ(KO)/DR3(1) +
1308 +          GZ)*(ONE-ALFAZ(KO))*RHOLZ(KO)
1309 6108 CONTINUE
1310 C
1311 C
1312 110 CONTINUE
1313 C
1314 C          THE WALL CELLS
1315 C
1316      DO 111 I = 2,NIM1
1317      KO = NNJ+I
1318 C
1319      UVRZ(KO) = (UVRO(KO-NI)+UVRO(KO-1-NI))/4.

```

```

1320      ULRZ(KO) = (ULRO(KO-NI)+ULRO(KO-1-NI))/4.
1321 C
1322      KD = 0
1323      IF(UVZO(KO).GE.ZERO) KD = -1
1324      KN = KO+KD
1325      IO = I +KD
1326 C
1327      HVZ(KO) = HV(KN)
1328      AVZD(KO) = ALFAO(KN)
1329      WZ1(KO) = ALFAO(KN)*RHOV(KN)
1330      WZ7(KO) = (UVZO(KN+1)-UVZO(KN))/DZ(IO)+UVZO(KO)
1331 C
1332      KD = 0
1333      IF(ULZO(KO).GE.ZERO) KD = -1
1334      IO = I+KD
1335      KN = KO+KD
1336      HLZ(KO) = HL(KN)
1337      ALZD(KO) = ONE - ALFAO(KN)
1338      WZ2(KO) = (ONE-ALFAO(KN))*RHOL(KN)
1339      WZ8(KO) = (ULZO(KN+1)-ULZO(KN))/DZ(IO)*ULZO(KO)
1340 C
1341 C
1342      IF(UVRZ(KO)) 1110,2110,2110
1343 1110 WZ7(KO) = (WZ7(KO) - UVZO(KO)*UVRZ(KO)/DR3(3) +
1344      + GZ)*ALFAZ(KO)*RHOVZ(KO)
1345      GO TO 3110
1346 2110 WZ7(KO) = (WZ7(KO) + (UVZO(KO)-UVZO(KO-NI))*UVRZ(KO)/DR3(2) +
1347      + GZ)*ALFAZ(KO)*RHOVZ(KO)
1348 C
1349 3110 CONTINUE
1350      IF(ULRZ(KO)) 4110,5110,5110
1351 4110 WZ8(KO) = (WZ8(KO) - ULZO(KO)*ULRZ(KO)/DR3(3) +
1352      + GZ)*(ONE-ALFAZ(KO))*RHOLZ(KO)
1353      GO TO 6110
1354 5110 WZ8(KO) = (WZ8(KO) +(ULZO(KO)-ULZO(KO-NI))*ULRZ(KO)/DR3(3) +
1355      + GZ)*(ONE-ALFAZ(KO))*RHOLZ(KO)
1356 6110 CONTINUE
1357 C
1358 111 CONTINUE
1359      RETURN
1360      END

```

Subroutine ws

```

1361 SUBROUTINE WS(PO,TVO,TLO,ALFAO,ALFAZ,ALFAR,RHOV,
1362 *          RHOZ,RHOVZ,RHOLZ,RHOVR,RHOLR,HV,HL,
1363 *          UVZO,ULZO,UVRO,ULRO,
1364 *          WEV,WEL,WZ3,WZ4,WZ5,WZ6,WZ7,WZ8,WZ9,
1365 *          WZ10,WZ11,WR3,WR4,WR5,WR6,WR7,WR8,WR9,
1366 *          WR10,WR11,DH,DV,OSI,SPPD,NN)
1367 IMPLICIT REAL*8 (A-H,O-Z)
1368 COMMON /DIM/ DZ(40),DZ1(40),DRO(40),DR1(40),DR2(40),DR3(40),
1369 *          DR4(40),NI,NJ,NIM1,NIM2,NUM1,NNI,NNJ,NNJU
1370 COMMON /TEMPO/ TIME,DT,DT0,DTLS,NDT
1371 COMMON /NUMBER/ ZERO,ONE,BIG,SMALL
1372 DIMENSION PO(NN),TVO(NN),TLO(NN),ALFAO(NN),ALFAZ(NN),
1373 *          ALFAR(NN),RHOV(NN),RHOZ(NN),RHOVZ(NN),RHOLZ(NN),
1374 *          RHOVR(NN),RHOLR(NN),HV(NN),HL(NN),UVZO(NN),
1375 *          ULZO(NN),UVRO(NN),ULRO(NN),WEV(NN),WEL(NN),
1376 *          WZ3(NN),WZ4(NN),WZ5(NN),WZ6(NN),WZ7(NN),WZ8(NN),
1377 *          WZ9(NN),WZ10(NN),WZ11(NN),WR3(NN),WR4(NN),
1378 *          WR5(NN),WR6(NN),WR7(NN),WR8(NN),WR9(NN),WR10(NN),
1379 *          WR11(NN),DH(NN),DV(NN),OSI(NN),SPPD(NN)
1380 C
1381 C
1382 C
1383 C SUBROUTINE WS COMPLETE THE EVALUATION OF THE
1384 C EXPLICIT TERMS INVOLVED IN THE SOLUTION OF
1385 C THE PROBLEM STATED WITH SUBROUTINE DONOR.
1386 C HERE ARE SET THE TERMS CONTAINING THE TIME
1387 C INCREMENT DT.IT IS WRITTEN SEPARATELY FROM
1388 C SUSROUTINE DONOR.IN ORDER TO ALLOW A CHANGE
1389 C IN THE VALUE OF DT WHEN THE PROBLRM DOES NOT
1390 C CONVERGE WITH THE PREVIOUS DT.
1391 C (SEE NEXT COMENT IN THIS SUBROUTINE.)
1392 C
1393 C
1394 C
1395 DO 5 JO = 1,NI
1396 DO 5 IO = 2,NI
1397 KO = (JO-1)*NI+IO
1398 C
1399 WWZ1 = ALFAZ(KO)*RHOVZ(KO)
1400 WWZ2 = (ONE - ALFAZ(KO))*RHOLZ(KO)
1401 WWR1 = ALFAR(KO)*RHOVR(KO)
1402 WWR2 = (ONE - ALFAR(KO))*RHOLR(KO)
1403 C

```

```

1404 CALL COEFF(TVO(KO),TLO(KO),UVZO(KO),UVRO(KO),ULZO(KO),ULRO(KO),
1405 * ALFAZ(KO),ALFAR(KO),RHOVZ(KO),RHOVR(KO),
1406 * RHOLZ(KO),RHOLR(KO),DH(KO),DV(KO),QSI(KO),
1407 * SPPD(KO),WVZ1,WVZ2,WVR1,WVR2,
1408 * FVZ,FLZ,FVR,FLR,C1Z,C1R)
1409 C
1410 WEV(KO) = -(RHOV(KO)*HV(KO)+PO(KO))*ALFAO(KO)/DT
1411 WEL(KO) = -(RHOL(KO)*HL(KO)+PO(KO))*(ONE-ALFAO(KO))/DT
1412 C
1413 IF(NDT.NE.0) GO TO 1
1414 C
1415 C
1416 C SINCE THE PROGRAM ALLOWS A CHANGE IN THE VALUE
1417 C OF THE TIME INCREMENT DT,EVEN IF THE TIME STEP
1418 C IS NOT COMPLETED,WE PUT A CHECK HERE TO KNOW
1419 C IF SUCH A CHANGE DID OCCUR (IN THIS CASE NDT
1420 C WOULD BE DIFFERENT THAN ZERO) IN CASE THE TEST
1421 C BE TRUE,WE SUBTRACT THE TERMS WHICH HAVE THE
1422 C OLD DT AND ADD THEM BACK WITH THE NEW VALUE
1423 C OF DT.
1424 C
1425 WZ4(KO) = C1Z
1426 WZ6(KO) = C1Z
1427 WR4(KO) = C1R
1428 WR6(KO) = C1R
1429 C
1430 WZ3(KO) = WZ4(KO) + ALFAZ(KO)*RHOVZ(KO)/DT + FVZ
1431 WZ5(KO) = WZ6(KO) + (ONE-ALFAZ(KO))*RHOLZ(KO)/DT + FLZ
1432 WR3(KO) = WR4(KO) + ALFAR(KO)*RHOVR(KO)/DT + FVR
1433 WR5(KO) = WR6(KO) + (ONE-ALFAR(KO))*RHOLR(KO)/DT + FLR
1434 C
1435 C
1436 WZ7(KO) = WZ7(KO) - UVZO(KO)/DT*ALFAZ(KO)*RHOVZ(KO)
1437 WZ8(KO) = WZ8(KO) - ULZO(KO)/DT*(ONE-ALFAZ(KO))*RHOLZ(KO)
1438 WR7(KO) = WR7(KO) - UVRO(KO)/DT*ALFAR(KO)*RHOVR(KO)
1439 WR8(KO) = WR8(KO) - ULRO(KO)/DT*(ONE-ALFAR(KO))*RHOLR(KO)
1440 GO TO 2
1441 C
1442 1 DTC = ONE/DTC - ONE/DT
1443 C
1444 WZ7(KO) = UVZO(KO)*ALFAZ(KO)*RHOVZ(KO)*DTC + WZ7(KO)
1445 WZ8(KO) = ULZO(KO)*(ONE-ALFAZ(KO))*RHOLZ(KO)*DTC + WZ8(KO)
1446 WR7(KO) = UVRO(KO)*ALFAR(KO)*RHOVR(KO)*DTC + WR7(KO)
1447 WR8(KO) = ULRO(KO)*(ONE-ALFAR(KO))*RHOLR(KO)*DTC + WR8(KO)
1448 WZ3(KO) = WZ3(KO) - ALFAZ(KO)*RHOVZ(KO)*DTC

```

```

1404 CALL COEFF(TVO(KO),TLO(KO),UVZO(KO),UVRO(KO),ULZO(KO),ULRO(KO),
1405 * ALFAZ(KO),ALFAR(KO),RHOVZ(KO),RHOVR(KO),
1406 * RHOLZ(KO),RHOLR(KO),DH(KO),DV(KO),QSI(KO),
1407 * SPPD(KO),WWZ1,WWZ2,WWR1,WWR2,
1408 * FVZ,FLZ,FVR,FLR,C1Z,C1R)
1409 C
1410 WEV(KO) = -(RHOV(KO)*HV(KO)+PO(KO))*ALFAO(KO)/DT
1411 WEL(KO) = -(RHOL(KO)*HL(KO)+PO(KO))*(ONE-ALFAO(KO))/DT
1412 C
1413 IF(NDT.NE.0) GO TO 1
1414 C
1415 C
1416 C SINCE THE PROGRAM ALLOWS A CHANGE IN THE VALUE
1417 C OF THE TIME INCREMENT DT,EVEN IF THE TIME STEP
1418 C IS NOT COMPLETED,WE PUT A CHECK HERE TO KNOW
1419 C IF SUCH A CHANGE DID OCCUR (IN THIS CASE NDT
1420 C WOULD BE DIFFERENT THAN ZERO) IN CASE THE TEST
1421 C BE TRUE,WE SUBTRACT THE TERMS WHICH HAVE THE
1422 C OLD DT AND ADD THEM BACK WITH THE NEW VALUE
1423 C OF DT.
1424 C
1425 WZ4(KO) = C1Z
1426 WZ6(KO) = C1Z
1427 WR4(KO) = C1R
1428 WR6(KO) = C1R
1429 C
1430 WZ3(KO) = WZ4(KO) + ALFAZ(KO)*RHOVZ(KO)/DT + FVZ
1431 WZ5(KO) = WZ6(KO) + (ONE-ALFAZ(KO))*RHOLZ(KO)/DT + FLZ
1432 WR3(KO) = WR4(KO) + ALFAR(KO)*RHOVR(KO)/DT + FVR
1433 WR5(KO) = WR6(KO) + (ONE-ALFAR(KO))*RHOLR(KO)/DT + FLR
1434 C
1435 C
1436 WZ7(KO) = WZ7(KO) - UVZO(KO)/DT*ALFAZ(KO)*RHOVZ(KO)
1437 WZ8(KO) = WZ8(KO) - ULZO(KO)/DT*(ONE-ALFAZ(KO))*RHOLZ(KO)
1438 WR7(KO) = WR7(KO) - UVRO(KO)/DT*ALFAR(KO)*RHOVR(KO)
1439 WR8(KO) = WR8(KO) - ULRO(KO)/DT*(ONE-ALFAR(KO))*RHOLR(KO)
1440 GO TO 2
1441 C
1442 1 DTC = ONE/DTCO - ONE/DT
1443 C
1444 WZ7(KO) = UVZO(KO)*ALFAZ(KO)*RHOVZ(KO)*DTC + WZ7(KO)
1445 WZ8(KO) = ULZO(KO)*(ONE-ALFAZ(KO))*RHOLZ(KO)*DTC + WZ8(KO)
1446 WR7(KO) = UVRO(KO)*ALFAR(KO)*RHOVR(KO)*DTC + WR7(KO)
1447 WR8(KO) = ULRO(KO)*(ONE-ALFAR(KO))*RHOLR(KO)*DTC + WR8(KO)
1448 WZ3(KO) = WZ3(KO) - ALFAZ(KO)*RHOVZ(KO)*DTC

```

```

1449   WZ5(KO) = WZ5(KO) - (ONE-ALFAZ(KO))*RHOLZ(KO)*DTC
1450   WR3(KO) = WR3(KO) - ALFAR(KO)*RHOVR(KO)*DTC
1451   WR5(KO) = WR5(KO) - (ONE-ALFAR(KO))*RHOLR(KO)*DTC
1452 C
1453   2 IF(WZ3(KO).GT.SMALL) GO TO 3
1454 C
1455 C
1456 C           THIS TEST IS DONE TO CHECK THE PRESENCE OF
1457 C           VAPOR IN THE CELL AT THE PRESENT TIME STEP.
1458 C           IN CASE THERE IS NO VAPOR NOR EVAPORATION
1459 C           (WZ3 = ZERO),THE VAPOR MOMENTUM EQUATION
1460 C           BECOMES TRIVIAL AND THE LIQUID EQUATION
1461 C           STANDS ALONE.
1462 C
1463   WZ11(KO) = ZERO
1464   WZ9(KO) = ZERO
1465   WZ10(KO) = -(ONE-ALFAZ(KO))/DZ1(IO)/WZ5(KO)
1466   GO TO 5
1467 C
1468   3 IF(WZ5(KO).GT.SMALL) GO TO 4
1469 C
1470 C
1471 C           THIS TEST IS DONE TO CHECK THE PRESENCE OF
1472 C           LIQUID IN THE CELL AT THE PRESENT TIME STEP.
1473 C           IN CASE THERE IS NO LIQUID NOR CONDENSATION
1474 C           (WZ5 = ZERO),THE LIQUID MOMENTUM EQUATION
1475 C           BECOMES TRIVIAL AND THE VAPOR EQUATION
1476 C           STANDS ALONE.
1477 C
1478 C
1479   WZ11(KO) = ZERO
1480   WZ10(KO) = ZERO
1481   WZ9(KO) = -ALFAZ(KO)/DZ1(IO)/WZ3(KO)
1482   GO TO 5
1483 C
1484   4 WZ11(KO) = WZ3(KO)*WZ5(KO)-WZ4(KO)*WZ6(KO)
1485   WZ10(KO) = -(ALFAZ(KO)*WZ6(KO)+(ONE-ALFAZ(KO))*WZ3(KO))/
1486   /
1487   WZ9(KO) = -(ALFAZ(KO)*WZ5(KO)+(ONE-ALFAZ(KO))*WZ4(KO))/
1488   /
1489   DZ1(IO)/WZ11(KO)
1490   5 CONTINUE
1491 C
1492 C
1493 C           THE SAME TEST WHICH WAS DONE FOR THE

```

```

1494 C      Z-DIRECTION (SEE COMENTS ABOVE) IS
1495 C      DONE HERE FOR THE R-DIRECTION.NOTE
1496 C      THAT SINCE THE MOMENTUM EQUATIONS ARE
1497 C      EVALUATED AT DIFFERENT LOCATIONS FOR
1498 C      EACH DIRECTION,IT IS POSSIBLE THAT
1499 C      ONE PHASE IS ABSENT IN ONE DIRECTION
1500 C      EQUATIONS AND PRESENT IN THE OTHER
1501 C      DIRECTION EQUATIONS.
1502 C
1503 C
1504 C      DO 8 JO = 1,NJM1
1505 C      DO 8 IO = 2,NIM1
1506 C      KO = (JO-1)*NI + IO
1507 C
1508 C      IF(WR3(KO).GT.SMALL) GO TO 6
1509 C      WR11(KO) = ZERO
1510 C      WR9(KO) = ZERO
1511 C      WR10(KO) = -(ONE-ALFAR(KO))/DR3(JO)/WR5(KO)
1512 C      GO TO 8
1513 C
1514 C      6 IF(WR5(KO).GT.SMALL) GO TO 7
1515 C      WR11(KO) = ZERO
1516 C      WR10(KO) = ZERO
1517 C      WR9(KO) = -ALFAR(KO)/DR3(JO)/WR3(KO)
1518 C      GO TO 8
1519 C
1520 C      7 WR11(KO) = WR3(KO)*WR5(KO) - WR4(KO)*WR6(KO)
1521 C      WR10(KO) = -(ALFAR(KO)*WR6(KO)+(ONE-ALFAR(KO))*WR3(KO))/
1522 C      /
1523 C      DR3(JO)/WR11(KO)
1524 C      WR9(KO) = -(ALFAR(KO)*WR5(KO)+(ONE-ALFAR(KO))*WR4(KO))/
1525 C      /
1526 C      DR3(JO)/WR11(KO)
1527 C      8 CONTINUE
1528 C      RETURN
1529 C      END

```

Subroutine onestp

```

1528 SUBROUTINE ONESTP(PN,PO,TVN,TLN,ALFAN,ALFAO,ALFAZ,ALFAR,
1529 * RHOV,RHOL,HV,HL,HVZ,HLZ,HVR,HLR,
1530 * UVZN,ULZN,UVRN,ULRN,
1531 * FUVZN,FULZN,FUVRN,FULRN,W,DV,TS,
1532 * TW,DTW,HCONV,HCONL,HNB,DPN,A1,A2,A3,
1533 * A4,YP,B,BETA,GAMMA,AVZD,ALZD,AVRD,
1534 * ALRD,TCAN,DT,NN,NB,NP,NW,NCAN)
1535 C
1536 IMPLICIT REAL*8 (A-H,O-Z)
1537 C
1538 C
1539 COMMON /NUMBER/ ZERO,ONE,BIG,SMALL
1540 COMMON /ERROR/ IERR
1541 COMMON /DIM/ DZ(40),DZ1(40),DRO(40),DR1(40),DR2(40),DR3(40),
1542 * DR4(40),NI,NJ,NIM1,NIM2,NUM1,NNI,NNJ,NNJU
1543 C
1544 DIMENSION EPSLON(9),RES(9)
1545 DIMENSION PN(NN),PO(NN),TVN(NN),TLN(NN),ALFAN(NN),ALFAO(NN),
1546 * ALFAZ(NN),ALFAR(NN),RHOV(NN),RHOL(NN),HV(NN),HL(NN),
1547 * HVZ(NN),HLZ(NN),HVR(NN),HLR(NN),
1548 * UVZN(NN),ULZN(NN),UVRN(NN),ULRN(NN),
1549 * FUVZN(NN),FULZN(NN),FUVRN(NN),FULRN(NN),
1550 * W(NW),DV(NN),TS(NN),TW(NN),DTW(NN),
1551 * HCONV(NN),HCONL(NN),HNB(NN),DPN(NN),
1552 * A1(NN),A2(NN),A3(NN),A4(NN),YP(NN),B(NB),
1553 * BETA(NN),GAMMA(NN),AVZD(NN),ALZD(NN),AVRD(NN),
1554 * ALRD(NN),TCAN(NCAN)
1555 C
1556 DIMENSION A(65),F(9),PROP(3,4),S(5,2),Q(4,2),K(30),M(30)
1557 C
1558 IFLAG = 1
1559 C
1560 C THE MOMENTUM EQUATIONS (Z-DIRECTION) AT THE BOTTON
1561 C
1562 MM = NNJ + 2
1563 DO 4 KO = 2,MM,NI
1564 DO 1 L = 1,27
1565 1 K(L) = (L-1)*NN + KO
1566 C
1567 IF(W(K(5)).GT.SMALL) GO TO 2
1568 C
1569 C
1570 C ONLY LIQUID PRESENT IN THE CELL

```

```

1571 C
1572     V01 = (ONE-ALFAZ(KO))/DZ1(2)
1573     V05 = W(K(7))
1574 C
1575     FUVZN(KO) = ZERO
1576     FULZN(KO) = -(W(K(7))*ULZN(KO) + (PN(KO)-PN(KO-1))*V01 +
1577 +           W(K(10)))/V05
1578     W(K(11)) = ZERO
1579     W(K(12)) = -V01/V05
1580     GO TO 4
1581 C
1582     2 IF(W(K(7)).GT.SMALL) GO TO 3
1583 C
1584 C
1585 C           ONLY VAPOR PRESENT IN THE CELL
1586 C
1587     V02 = ALFAZ(KO)/DZ1(2)
1588     V03 = W(K(5))
1589 C
1590     FUVZN(KO) = -(W(K(5))*UVZN(KO) + (PN(KO)-PN(KO-1))*V02 +
1591 +           W(K(9)))/V03
1592     FULZN(KO) = ZERO
1593     W(K(11)) = -V02/V03
1594     W(K(12)) = ZERO
1595     GO TO 4
1596 C
1597 C           BOTH PHASES PRESENT IN THE CELL
1598 C
1599     3 CONTINUE
1600     V01 = (ONE-ALFAZ(KO))/DZ1(2)
1601     V02 = ALFAZ(KO)/DZ1(2)
1602     V03 = W(K(5))
1603     V04 = W(K(6))
1604     V05 = W(K(7))
1605     V06 = W(K(8))
1606     V07 = V04*V06 - V03*V05
1607 C
1608     F(5) = W(K(5))*UVZN(KO) - W(K(6))*ULZN(KO) + (PN(KO)-PN(KO-1))*
1609 +           V02 + W(K(9))
1610     F(6) = W(K(7))*ULZN(KO) - W(K(8))*UVZN(KO) + (PN(KO)-PN(KO-1))*
1611 +           V01 + W(K(10))
1612 C
1613     W(K(11))=(V05*V02+V04*V01)/V07
1614     W(K(12))=(V06*V02+V03*V01)/V07
1615     FUVZN(KO)=(F(5)*V05+F(6)*V04)/V07

```

```

1616      FULZN(KO)=(F(5)*V06+F(6)*V03)/V07
1617      4 CONTINUE
1618 C
1619 C      THE CENTRAL CELLS
1620 C
1621      A(4) = ZERO
1622      A(12)= ZERO
1623      A(20)= ZERO
1624      A(28)= ZERO
1625      DO 122 KO = 2,NIM1
1626      DO 5 L=1,27
1627      5 K(L) = (L-1)*NN+KO
1628      KM=KO+1
1629      KP = KO - 1
1630      CALL STATE (TVN(KO),TLN(KO),PN(KO),PROP,IFLAG)
1631      CALL NONEQ(ALFAO(KO),ALFAN(KO),TVN(KO),TLN(KO),PN(KO),
1632      *          RHOV(KO),RHOL(KO),TS(KO),S,IFLAG)
1633      CALL CONDT (TVN(KO),TLN(KO),PN(KO),ALFAO(KO),TS(KO),TW(KP),
1634      *          DTW(KP),HCONV(KP),HCONL(KP),HNB(KP),DV(KO),Q,KO)
1635      CALL IPHTC (HIF,ALFAN(KO))
1636 C
1637      V01=ALFAO(KO)/DT
1638      V02=(ONE-ALFAO(KO))/DT
1639      V03=ALFAN(KO)/DT
1640      V04=(ONE-ALFAN(KO))/DT
1641      V05=S(1,1)
1642      V06=S(2,1)
1643      V07=S(3,1)
1644      V08=S(4,1)
1645      V09=S(5,1)
1646      V10=W(K(3)+1)/DZ(KO)
1647      V11=W(K(3))/DZ(KO)
1648      V12=W(K(14))*DR1(1)
1649 C
1650      V14=W(K(4)+1)/DZ(KO)
1651      V15=W(K(4))/DZ(KO)
1652      V16=W(K(15))*DR1(1)
1653 C
1654      V18=HVZ(KM)*V10 + PO(KO)*AVZD(KM)/DZ(KO)
1655      V19=HVZ(KO)*V11 + PO(KO)/DZ(KO)*AVZD(KO)
1656      V20=HVR(KO)*V12 + PO(KO)*DR1(1)*AVRO(KO)
1657 C
1658      V22=HLZ(KM)*V14 + PO(KO)*ALZD(KM)/DZ(KO)
1659      V23=HLZ(KO)*V15 + PO(KO)*ALZD(KO)/DZ(KO)
1660      V24=HLR(KO)*V16 + PO(KO)*ALRD(KO)*DR1(1)

```

```

1661 C
1662 V26=(TVN(KO)-TLN(KO))*HIF
1663 V27=V03*PROP(1,1)
1664 V28=V04*PROP(1,2)
1665 V29 = V01*RHOV(KO)
1666 V30 = V02*RHOL(KO)
1667 V31 = HV(KO)*V29
1668 V32 = HL(KO)*V30
1669 C
1670 C THE RESIDUALS OF CONSERVATION EQUATIONS
1671 C
1672 F(1) = V27 -V29 +UVZN(KM)*V10 -UVZN(KO)*V11 + UVRN(KO)*
1673 * V12 -V05
1674 F(2) = PROP(1,3)*V27 - V31 +UVZN(KM)*V18 -
1675 * UVZN(KO)*V19 + UVRN(KO)*V20 - S(1,2) - Q(1,1) +
1676 + PQ(KO)*(V03 - V01)
1677 F(3) = V28 - V30 + ULZN(KM)*V14 - ULZN(KO)*V15 +
1678 + ULRN(KO)*V16 +V05
1679 F(4) = PROP(1,4)*V28 - V32 + ULZN(KM)*V22 -
1680 * ULZN(KO)*V23 + ULRN(KO)*V24 + S(1,2) - Q(1,2) +
1681 + PO(KO)*(V04 - V02)
1682 F(5) = W(K(5)+1)*UVZN(KM) -W(K(6)+1)*ULZN(KM) + (PN(KM)-PN(KO))*
1683 * ALFAZ(KM)/DZ1(KM) + W(K(9)+1)
1684 F(6) = W(K(7)+1)*ULZN(KM) -W(K(8)+1)*UVZN(KM) + (PN(KM)-PN(KO))*
1685 * (ONE-ALFAZ(KM))/DZ1(KM) + W(K(10)+1)
1686 F(7) = W(K(16))*UVRN(KO) - W(K(17))*ULRN(KO) + (PN(KO+NI)-PN(KO))*
1687 * ALFAR(KO)/DR3(1) + W(K(20))
1688 F(8) = W(K(18))*ULRN(KO) - W(K(19))*UVRN(KO) + (PN(KO+NI)-PN(KO))*
1689 * (ONE-ALFAR(KO))/DR3(1) + W(K(21))
1690 C
1691 C
1692 A(1) = PROP(1,1)/DT - V09
1693 A(9) = (PROP(1,3)*PROP(1,1) + PQ(KO))/DT - S(5,2)
1694 A(17) = -PROP(1,2)/DT + V09
1695 A(25) = -(PROP(1,4)*PROP(1,2) + PQ(KO))/DT + S(5,2)
1696 C
1697 A(2) = PROP(2,1)*V03-V06
1698 A(10)=(PROP(1,1)*PROP(2,3)+PROP(1,3)*PROP(2,1))*V03 -
1699 - Q(2,1) - S(2,2)
1700 A(18)=V06
1701 A(26) = S(2,2) - Q(2,2)
1702 C
1703 A(3) = -V07
1704 A(11) = - S(3,2)
1705 A(19)=PROP(2,2)*V04+V07

```

```

1706      A(27)=(PROP(1,2)*PROP(2,4)+PROP(1,4)*PROP(2,2))*VQ4 -
1707      -      Q(3,2) + S(3,2)
1708 C
1709 C
1710      A(4) = ZERO
1711      A(12) = ZERO
1712      A(20) = ZERO
1713      A(28) = ZERO
1714 C
1715      A(5) = W(K(11))*V11
1716      A(13) = W(K(11))*V19
1717      A(21) = W(K(12))*V15
1718      A(29) = W(K(12))*V23
1719 C
1720      A(7) = W(K(11)+1)*V10
1721      A(15) = W(K(11)+1)*V18
1722      A(23) = W(K(12)+1)*V14
1723      A(31) = W(K(12)+1)*V22
1724      A(8) = W(K(22))*V12
1725      A(16) = W(K(22))*V20
1726      A(24) = W(K(23))*V16
1727      A(32) = W(K(23))*V24
1728 C
1729      A(6)=PROP(3,1)*V03-V08-A(5)-A(7)-A(8)
1730      A(14)=(PROP(1,1)*PROP(3,3)+PROP(1,3)*PROP(3,1))*V03-
1731      -      S(4,2) - A(13) - A(15) - A(16)
1732      A(22)=PROP(3,2)*V04+V08-A(21)-A(23)-A(24)
1733      A(30)=(PROP(1,2)*PROP(3,4)+PROP(1,4)*PROP(3,2))*V04 - Q(4,2)
1734      *      + S(4,2) - A(29) - A(31) - A(32)
1735 C
1736      IF(W(K(5)+1).GT.SMALL) GO TO 6
1737      FUVZN(KM) = ZERO
1738      FULZN(KM) = -F(6)/W(K(7)+1)
1739      GO TO 8
1740 6 IF(W(K(7)+1).GT.SMALL) GO TO 7
1741      FUVZN(KM) = -F(5)/W(K(5)+1)
1742      FULZN(KM) = ZERO
1743      GO TO 8
1744 7 CONTINUE
1745      FUVZN(KM) = -(W(K(7)+1)*F(5)+W(K(6)+1)*F(6))/W(K(13)+1)
1746      FULZN(KM) = -(W(K(8)+1)*F(5)+W(K(5)+1)*F(6))/W(K(13)+1)
1747 8 CONTINUE
1748      IF(W(K(16)).GT.SMALL) GO TO 9
1749      FUVRN(KO) = ZERO
1750      FULRN(KO) = -F(8)/W(K(18))

```

```

1751      GO TO 11
1752      9 CONTINUE
1753      IF(W(K(18)).GT.SMALL) GO TO 10
1754      FUVRN(KO) = -F(7)/W(K(16))
1755      FULRN(KO) = ZERO
1756      GO TO 11
1757      10 CONTINUE
1758      FUVRN(KO)=- (W(K(18))*F(7)+W(K(17))*F(8))/W(K(24))
1759      FULRN(KO)=- (W(K(19))*F(7)+W(K(16))*F(8))/W(K(24))
1760      11 CONTINUE
1761 C
1762      F(1)=-F(1)-FUVZN(KM)*V10+FUVZN(KO)*V11-FUVRN(KO)*V12
1763      F(2)=-F(2)-FUVZN(KM)*V18+FUVZN(KO)*V19-FUVRN(KO)*V20
1764      F(3)=-F(3)-FULZN(KM)*V14+FULZN(KO)*V15-FULRN(KO)*V16
1765      F(4)=-F(4)-FULZN(KM)*V22+FULZN(KO)*V23-FULRN(KO)*V24
1766 C
1767 C
1768 C
1769      DO 111 L = 1,27
1770      111 K(L) = L*NN + KO
1771      IX2 = 1
1772      DO 12 IX1 = 8,24,8
1773      AUX = A(IX1+1)/A(1)
1774      IX2 = IX2 + 1
1775      F(IX2) = F(IX2) - F(1)*AUX
1776      DO 12 IX3 = 2,8
1777      IX4 = IX1 + IX3
1778      12 A(IX4) = A(IX4) - A(IX3)*AUX
1779      DO 13 L = 1,7
1790      13 B(K(L)) = -A(L+1)/A(1)
1781      B(KO) = F(1)/A(1)
1782 C
1783      IF(DABS(A(10)).GT.SMALL) GO TO 16
1784 C
1785 C      ONLY LIQUID IN THE CELL
1786 C
1787      B(K(8)) = ZERO
1788      B(K(9)) = ONE
1789      DO 14 L = 10,14
1790      14 B(K(L)) = ZERO
1791      B(K(15)) = F(4)/A(27)
1792      DO 15 L = 16,20
1793      15 B(K(L)) = -A(L+12)/A(27)
1794 C
1795      AUX = A(19)/A(27)

```

```

1796      AUP = A(22) - A(30)*AUX
1797      A1(KP) = (A(20) - A(28)*AUX)/AUP
1798      A2(KP) = (A(21) - A(29)*AUX)/AUP
1799      A3(KP) = (A(23) - A(31)*AUX)/AUP
1800      A4(KP) = (A(24) - A(32)*AUX)/AUP
1801      YP(KP) = (F(3) - F(4)*AUX)/AUP
1802      GO TO 22
1803 C
1804      16 CONTINUE
1805      IF(DABS(A(27)).GT.SMALL) GO TO 18
1806 C
1807 C          ONLY VAPOR IN THE CELL
1808 C
1809      B(K(8)) = F(2)/A(10)
1810      B(K(15)) = B(K(8))
1811      B(K(9)) = ZERO
1812      DO 17 L = 10,14
1813      B(K(L)) = -A(L+2)/A(10)
1814      LL = L + 6
1815      17 B(K(LL)) = B(K(L))
1816 C
1817      AUX = A(18)/A(10)
1818      AUP = A(22) - A(14)*AUX
1819      A1(KP) = (A(20) - A(12)*AUX)/AUP
1820      A2(KP) = (A(21) - A(13)*AUX)/AUP
1821      A3(KP) = (A(23) - A(15)*AUX)/AUP
1822      A4(KP) = (A(24) - A(16)*AUX)/AUP
1823      YP(KP) = (F(3) - F(2)*AUX)/AUP
1824      GO TO 22
1825 C
1826 C          BOTH PHASES PRESENT
1827 C
1828      18 CONTINUE
1829      B(K(8)) = F(2)/A(10)
1830      DO 19 L = 9,14
1831      19 B(K(L)) = -A(L+2)/A(10)
1832 C
1833      IX2 = 2
1834      DO 20 IX1 = 18,26,8
1835      AUX = A(IX1)/A(10)
1836      IX2 = IX2 + 1
1837      F(IX2) = F(IX2) - F(2)*AUX
1838      DO 20 IX3 = 1,6
1839      IX4 = IX1 + IX3
1840      IX5 = IX3 + 10

```

```

1841 20 A(IX4) = A(IX4) - A(IX5)*AUX
1842 C
1843 C
1844 B(K(15)) = F(3)/A(19)
1845 DO 21 L = 16,20
1846 21 B(K(L)) = -A(L+4)/A(19)
1847 C
1848 AUX = A(27)/A(19)
1849 AUP = A(30) - A(22)*AUX
1850 A1(KP) = (A(28) - A(20)*AUX)/AUP
1851 A2(KP) = (A(29) - A(21)*AUX)/AUP
1852 A3(KP) = (A(31) - A(23)*AUX)/AUP
1853 A4(KP) = (A(32) - A(24)*AUX)/AUP
1854 YP(KP) = (F(4) - F(3)*AUX)/AUP
1855 C
1856 22 CONTINUE
1857 C
1858 DDT = DABS(A1(KP)) + DABS(A2(KP)) + DABS(A3(KP)) + DABS(A4(KP))
1859 IF(DDT.GT.ONE) GO TO 58
1860 C
1861 122 CONTINUE
1862 C
1863 C OUT OF THE BOUNDARIES
1864 C
1865 DO 46 J=NI,NNJ,NI
1866 JO=J/NI+1
1867 DO 46 I=2,NIM1
1868 KO=I+J
1869 KM = KO + 1
1870 KP = KO -1 - J/NI*2
1871 DO 23 L=1,27
1872 23 K(L) = (L-1)*NN+KO
1873 IO=I+1
1874 C
1875 CALL STATE(TVN(KO),TLN(KO),PN(KO),PROP,IFLAG)
1876 CALL NONEQ(ALFAO(KO),ALFAN(KO),TVN(KO),TLN(KO),PN(KO),
1877 * RHOV(KO),RHOL(KO),TS(KO),S,IFLAG)
1878 CALL CONDT(TVN(KO),TLN(KO),PN(KO),ALFAO(KO),TS(KO),TW(KP),
1879 * DTW(KP),HCONV(KP),HCONL(KP),HNB(KP),DV(KO),Q,KO)
1880 CALL IPHTC(HIF,ALFAN(KO))
1881 C
1882 V01 = ALFAO(KO)/DT
1883 V02 = (ONE-ALFAO(KO))/DT
1884 V03 = ALFAN(KO)/DT
1885 V04 = (ONE-ALFAN(KO))/DT

```

1886 V05 = S(1,1)  
 1887 V06 = S(2,1)  
 1888 V07 = S(3,1)  
 1889 V08 = S(4,1)  
 1890 V09 = S(5,1)  
 1891 V10 = W(K(3)+1)/DZ(I)  
 1892 V11 = W(K(3))/DZ(I)  
 1893 V12 = W(K(14))\*DR1(JO)  
 1894 V13 = W(K(14)-NI)\*DR2(JO)  
 1895 V14 = W(K(4)+1)/DZ(I)  
 1896 V15 = W(K(4))/DZ(I)  
 1897 V16 = W(K(15))\*DR1(JO)  
 1898 V17 = W(K(15)-NI)\*DR2(JO)  
 1899 V18 = HVZ(KM)\*V10 + PO(KO)/DZ(I)\*AVZD(KM)  
 1900 V19 = HVZ(KO)\*V11 + PO(KO)/DZ(I)\*AVZD(KO)  
 1901 V20 = HVR(KO)\*V12 + PO(KO)\*DR1(JO)\*AVRD(KO)  
 1902 V21 = HVR(KO-NI)\*V13 + PO(KO)\*DR2(JO)\*AVRD(KO-NI)  
 1903 V22 = HLZ(KM)\*V14 + PO(KO)\*ALZD(KM)/DZ(I)  
 1904 V23 = HLZ(KO)\*V15 + PO(KO)\*ALZD(KO)/DZ(I)  
 1905 V24 = HLR(KO)\*V16 + PO(KO)\*ALRD(KO)\*DR1(JO)  
 1906 V25 = HLR(KO-NI)\*V17 + PO(KO)\*ALRD(KO-NI)\*DR2(JO)  
 1907 V26 = (TVN(KO)-TLN(KO))\*HIF  
 1908 V27 = V03\*PROP(1,1)  
 1909 V28 = V04\*PROP(1,2)  
 1910 V29 = V01\*RHOV(KO)  
 1911 V30 = V02\*RHOL(KO)  
 1912 V31 = HV(KO)\*V29  
 1913 V32 = HL(KO)\*V30

1914 C  
 1915 C  
 1916 C

1917 F(1) = V27 - V29 + UVZN(KM)\*V10 - UVZN(KO)\*V11 +  
 + UVRN(KO)\*V12 - UVRN(KO-NI)\*V13 - V05  
 1918 F(2) = PROP(1,3)\*V27 - V31 + UVZN(KM)\*V18 -  
 - UVZN(KO)\*V19 + UVRN(KO)\*V20 - UVRN(KO-NI)\*V21 -  
 1919 S(1,2) - Q(1,1) + PO(KO)\*V03 - V01  
 1920 F(3) = V28 - V30 + ULZN(KM)\*V14 - ULZN(KO)\*V15 +  
 + ULRN(KO)\*V16 - ULRN(KO-NI)\*V17 + V05  
 1921 F(4) = PROP(1,4)\*V28 - V32 + ULZN(KM)\*V22 -  
 + ULZN(KO)\*V23 + ULRN(KO)\*V24 - ULRN(KO-NI)\*V25 +  
 + S(1,2) - Q(1,2) + PO(KO)\*V04 - V02  
 1922 F(5) = W(K(5)+1)\*UVZN(KM) - W(K(6)+1)\*ULZN(KM) +  
 + (PN(KM)-PN(KO))\*ALFAZ(KM)/DZ1(I+1) + W(K(9)+1)  
 1923 F(6) = W(K(7)+1)\*ULZN(KM) - W(K(8)+1)\*UVZN(KM) +  
 + (PN(KM)-PN(KO))\*ONE-ALFAZ(KM)/DZ1(I+1) + W(K(10)+1)  
 1924  
 1925  
 1926  
 1927  
 1928  
 1929  
 1930

```

1931 C
1932 IF(J.LT.NNJ) GO TO 24
1933 CALL HEXCAN(TCAN(I),TCAN(NI+1),TVN(KO),TLN(KO),HCONV(KP),
1934 * HCONL(KP),QVC,QLC,DQCDTV,DQCDTL)
1935 F(2) = F(2) + QVC
1936 F(4) = F(4) + QLC
1937 F(7) = ZERO
1938 F(8) = ZERO
1939 GO TO 25
1940 24 CONTINUE
1941 F(7) = W(K(16))*UVRN(KO) -W(K(17))*ULRN(KO) + (PN(KO+NI)-PN(KO))*
1942 * ALFAR(KO)/DR3(JD) + W(K(20))
1943 F(8) = W(K(18))*ULRN(KO) - W(K(19))*UVRN(KO) + (PN(KO+NI)-PN(KO))*
1944 * (ONE-ALFAR(KO))/DR3(JD) + W(K(21))
1945 25 CONTINUE
1946 C
1947 C
1948 C
1949 A(1)=PROP(1,1)/DT - V09
1950 A(2)=PROP(2,1)*V03-V06
1951 A(3)=-V07
1952 A(4)=W(K(22)-NI)*V13
1953 C
1954 A(9)=(PROP(1,3)*PROP(1,1) + PO(KO))/DT - S(5,2)
1955 A(10)=(PROP(1,1)*PROP(2,3)+PROP(1,3)*PROP(2,1))*V03 -
1956 - Q(2,1) - S(2,2)
1957 A(11)= - S(3,2)
1958 A(12)=W(K(22)-NI)*V21
1959 C
1960 A(17)=-PROP(1,2)/DT + V09
1961 A(18)=V06
1962 A(19)=PROP(2,2)*V04+V07
1963 A(20)=W(K(23)-NI)*V17
1964 C
1965 A(25) = -(PROP(1,4)*PROP(1,2) + PO(KO))/DT + S(5,2)
1966 A(26)= S(2,2)
1967 A(27)=(PROP(1,2)*PROP(2,4)+PROP(1,4)*PROP(2,2))*V04 -
1968 - Q(3,2) + S(3,2)
1969 A(28)=W(K(23)-NI)*V25
1970 C
1971 A(5) = W(K(11))*V11
1972 A(13) = W(K(11))*V19
1973 A(21) = W(K(12))*V15
1974 A(29) = W(K(12))*V23
1975 C

```

```

1976 A(7) = W(K(11)+1)*V10
1977 A(15) = W(K(11)+1)*V18
1978 A(23) = W(K(12)+1)*V14
1979 A(31) = W(K(12)+1)*V22
1980 C
1981 C
1982 IF(J.GE.NNJ) GO TO 125
1983 A(8) = W(K(22))*V12
1984 A(16) = W(K(22))*V20
1985 A(24) = W(K(23))*V16
1986 A(32) = W(K(23))*V24
1987 C
1988 GO TO 225
1989 C
1990 125 A(8) = ZERO
1991 A(16) = ZERO
1992 A(24) = ZERO
1993 A(32) = ZERO
1994 A(10) = A(10) + DQCDTV
1995 A(27) = A(27) + DQCDTL
1996 225 CONTINUE
1997 C
1998 C
1999 C
2000 A(6) = PROP(3,1)*V03 - V08 - A(4) - A(5) - A(7) - A(8)
2001 A(14) = (PROP(1,1)*PROP(3,3)+PROP(1,3)*PROP(3,1))*V03 -
- S(4,2) - A(12) - A(13) - A(15) - A(16)
2002 A(22) = PROP(3,2)*V04+V08-A(20)-A(21)-A(23)-A(24)
2003 A(30) = (PROP(1,2)*PROP(3,4)+PROP(1,4)*PROP(3,2))*V04-Q(4,2)-
* A(28)-A(29)-A(31)-A(32) + S(4,2)
2006 C
2007 C
2008 C
2009 IF(W(K(5)+1).GT.SMALL) GO TO 26
2010 FUVZN(KM) = ZERO
2011 FULZN(KM) = -F(6)/W(K(7)+1)
2012 GO TO 28
2013 26 IF(W(K(7)+1).GT.SMALL) GO TO 27
2014 FUVZN(KM) = -F(5)/W(K(5)+1)
2015 FULZN(KM) = ZERO
2016 GO TO 28
2017 27 CONTINUE
2018 FUVZN(KM) = -(W(K(7)+1)*F(5)+W(K(6)+1)*F(6))/W(K(13)+1)
2019 FULZN(KM) = -(W(K(8)+1)*F(5)+W(K(5)+1)*F(6))/W(K(13)+1)
2020 28 CONTINUE

```

```

2021      IF(JO.EQ.NJ) GO TO 31
2022      IF(W(K(16)).GT.SMALL) GO TO 29
2023      FUVRN(KO) = ZERO
2024      FULRN(KO) = -F(8)/W(K(18))
2025      GO TO 31
2026      29 CONTINUE
2027      IF(W(K(18)).GT.SMALL) GO TO 30
2028      FUVRN(KO) = -F(7)/W(K(16))
2029      FULRN(KO) = ZERO
2030      GO TO 31
2031      30 CONTINUE
2032      FUVRN(KO)=- (W(K(18))*F(7)+W(K(17))*F(8))/W(K(24))
2033      FULRN(KO)=- (W(K(19))*F(7)+W(K(16))*F(8))/W(K(24))
2034      31 CONTINUE
2035      C
2036      F(1)=-F(1)-FUVZN(KM)*V10+FUVZN(KO)*V11-FUVRN(KO)*V12
2037      +      + FUVRN(KO-NI)*V13
2038      F(2)=-F(2)-FUVZN(KM)*V18+FUVZN(KO)*V19-FUVRN(KO)*V20
2039      +      + FUVRN(KO-NI)*V21
2040      F(3)=-F(3)-FULZN(KM)*V14+FULZN(KO)*V15-FULRN(KO)*V16
2041      +      + FULRN(KO-NI)*V17
2042      F(4)=-F(4)-FULZN(KM)*V22+FULZN(KO)*V23-FULRN(KO)*V24
2043      +      + FULRN(KO-NI)*V25
2044      C
2045      C
2046      C
2047      DO 32 L = 1,27
2048      32 K(L) = L*NN + KO
2049      IX2 = 1
2050      DO 33 IX1 = 8,24,8
2051      AUX = A(IX1+1)/A(1)
2052      IX2 = IX2 + 1
2053      F(IX2) = F(IX2) - F(1)*AUX
2054      DO 33 IX3 = 2,8
2055      IX4 = IX1 + IX3
2056      33 A(IX4) = A(IX4) - A(IX3)*AUX
2057      DO 34 L = 1,7
2058      34 B(K(L)) = -A(L+1)/A(1)
2059      B(KO) = F(1)/A(1)
2060      C
2061      IF(DABS(A(10)).GT.SMALL) GO TO 37
2062      C
2063      C      ONLY LIQUID IN THE CELL
2064      C
2065      B(K(8)) = ZERO

```

```

2066      B(K(9)) = ONE
2067      DO 35 L = 10,14
2068      35 B(K(L)) = ZERO
2069      B(K(15)) = F(4)/A(27)
2070      DO 36 L = 16,20
2071      36 B(K(L)) = -A(L+12)/A(27)
2072 C
2073      AUX = A(19)/A(27)
2074      AUP = A(22) - A(30)*AUX
2075      A1(KP) = (A(20) - A(28)*AUX)/AUP
2076      A2(KP) = (A(21) - A(29)*AUX)/AUP
2077      A3(KP) = (A(23) - A(31)*AUX)/AUP
2078      A4(KP) = (A(24) - A(32)*AUX)/AUP
2079      YP(KP) = (F(3) - F(4)*AUX)/AUP
2080      GO TO 43
2081 C
2082      37 CONTINUE
2083      IF(DABS(A(27)).GT.SMALL) GO TO 39
2084 C
2085 C          ONLY VAPOR IN THE CELL
2086 C
2087      B(K(8)) = F(2)/A(10)
2088      B(K(15)) = B(K(8))
2089      B(K(9)) = ZERO
2090      DO 38 L = 10,14
2091      B(K(L)) = -A(L+2)/A(10)
2092      LL = L + 6
2093      38 B(K(LL)) = B(K(L))
2094 C
2095      AUX = A(18)/A(10)
2096      AUP = A(22) - A(14)*AUX
2097      A1(KP) = (A(20) - A(12)*AUX)/AUP
2098      A2(KP) = (A(21) - A(13)*AUX)/AUP
2099      A3(KP) = (A(23) - A(15)*AUX)/AUP
2100      A4(KP) = (A(24) - A(16)*AUX)/AUP
2101      YP(KP) = (F(3) - F(2)*AUX)/AUP
2102      GO TO 43
2103 C
2104 C          BOTH PHASES PRESENT
2105 C
2106      39 CONTINUE
2107      B(K(8)) = F(2)/A(10)
2108      DO 40 L = 9,14
2109      40 B(K(L)) = -A(L+2)/A(10)
2110 C

```

```

2111      IX2 = 2
2112      DO 41 IX1 = 18,26,8
2113      AUX = A(IX1)/A(10)
2114      IX2 = IX2 + 1
2115      F(IX2) = F(IX2) - F(2)*AUX
2116      DO 41 IX3 = 1,6
2117      IX4 = IX1 + IX3
2118      IX5 = IX3 + 10
2119      41 A(IX4) = A(IX4) - A(IX5)*AUX
2120 C
2121 C
2122      B(K(15)) = F(3)/A(19)
2123      DO 42 L = 16,20
2124      42 B(K(L)) = -A(L+4)/A(19)
2125 C
2126      AUX = A(27)/A(19)
2127      AUP = A(30) - A(22)*AUX
2128      A1(KP) = (A(28) - A(20)*AUX)/AUP
2129      A2(KP) = (A(29) - A(21)*AUX)/AUP
2130      A3(KP) = (A(31) - A(23)*AUX)/AUP
2131      A4(KP) = (A(32) - A(24)*AUX)/AUP
2132      YP(KP) = (F(4) - F(3)*AUX)/AUP
2133 C
2134      43 CONTINUE
2135 C
2136      DDT = DABS(A1(KP)) + DABS(A2(KP)) + DABS(A3(KP)) + DABS(A4(KP))
2137      IF(DDT.GT.ONE) GO TO 58
2138 C
2139      46 CONTINUE
2140 C
2141      CALL GAUSIE(A1,A2,A3,A4,YP,DPN,BETA,GAMMA,NN)
2142 C
2143 C          CELL (2,1)
2144 C
2145      KO = 2
2146      KP = KO - 1
2147      KQ = KP + NIM2
2148      DO 47 L = 1,27
2149      M(L) = (L-1)*NN + KO
2150      47 K(L) = L*NN+KO
2151 C
2152 C
2153      DTL = B(K(15)) +
2154      +      B(K(18))*DPN(KP) + B(K(19))*DPN(KO) + B(K(20))*DPN(KQ)
2155      DTV = B(K(8)) + B(K(9))*DTL +

```

```

2156 + B(K(12))*DPN(KP) + B(K(13))*DPN(KO) +
2157 + B(K(14))*DPN(KQ)
2158 DAL = B(KO) + B(K(1))*DTV + B(K(2))*DTL +
2159 + B(K(5))*DPN(KP) + B(K(6))*DPN(KO) +
2160 + B(K(7))*DPN(KQ)
2161 PN(KO) = PN(KO) + DPN(KP)
2162 IF(PN(KO).LT.1.D+04) GO TO 59
2163 IF(PN(KO).GT.4.D+07) GO TO 60
2164 TLN(KO) = TLN(KO) + DTL
2165 TVN(KO) = TVN(KO) + DTV
2166 ALFAN(KO) = ALFAN(KO) + DAL
2167 TX = SAT(PN(KO))
2168 DTS = TX - TS(KO)
2169 TS(KO) = TX
2170 TW(KP) = TW(KP) + (HCONV(KP)*DTV + HCONL(KP)*DTL +
2171 + HNB(KP)*DTS)*DTW(KP)
2172 C
2173 C
2174 UVZN(KO) = W(M(11))*DPN(KP) + FUVZN(KO) + UVZN(KO)
2175 ULZN(KO) = W(M(12))*DPN(KP) + FULZN(KO) + ULZN(KO)
2176 UVRN(KO) = W(M(22))*(DPN(KO)-DPN(KP)) + FUVRN(KO) + UVRN(KO)
2177 ULRN(KO) = W(M(23))*(DPN(KO)-DPN(KP)) + FULRN(KO) + ULRN(KO)
2178 UVZN(1) = UVZN(KO)
2179 ULZN(1) = ULZN(KO)
2180 C
2181 C CELLS (I,1) , I=3,NI-2
2182 C
2183 DO 49 I = 3,NIM2
2184 KO = I
2185 KP = KO - 1
2186 KM = KO
2187 KQ = KP + NIM2
2188 KR = KP - 1
2189 DO 48 L = 1,27
2190 M(L) = (L-1)*NN + KO
2191 48 K(L) = L*NN+KO
2192 C
2193 C
2194 DTL = B(K(15)) + B(K(17))*DPN(KR) +
2195 + B(K(18))*DPN(KP) + B(K(19))*DPN(KM) + B(K(20))*DPN(KQ)
2196 DTV = B(K(8)) + B(K(9))*DTL +
2197 + B(K(11))*DPN(KR) + B(K(12))*DPN(KP) + B(K(13))*DPN(KM) +
2198 + B(K(14))*DPN(KQ)
2199 DAL = B(KO) + B(K(1))*DTV + B(K(2))*DTL +
2200 + B(K(4))*DPN(KR) + B(K(5))*DPN(KP) + B(K(6))*DPN(KM) +

```

```

2201 + B(K(7))*DPN(KQ)
2202 PN(KO) = PN(KO) + DPN(KP)
2203 IF(PN(KO).LT.1.D+04) GO TO 59
2204 IF(PN(KO).GT.4.D+07) GO TO 60
2205 TLN(KO) = TLN(KO) + DTL
2206 TVN(KO) = TVN(KO) + DTV
2207 ALFAN(KO) = ALFAN(KO) + DAL
2208 TX = SAT(PN(KO))
2209 DTS = TX - TS(KO)
2210 TS(KO) = TX
2211 TW(KP) = TW(KP) + (HCONV(KP)*DTV + HCONL(KP)*DTL +
2212 + HNB(KP)*DTS)*DTW(KP)
2213 C
2214 C
2215 UVZN(KO) = W(M(11))*(DPN(KP)-DPN(KR)) + FUVZN(KO) + UVZN(KO)
2216 ULZN(KO) = W(M(12))*(DPN(KP)-DPN(KR)) + FULZN(KO) + ULZN(KO)
2217 UVRN(KO) = W(M(22))*(DPN(KQ)-DPN(KP)) + FUVRN(KO) + UVRN(KO)
2218 ULRN(KO) = W(M(23))*(DPN(KQ)-DPN(KP)) + FULRN(KO) + ULRN(KO)
2219 49 CONTINUE
2220 C
2221 C CELL (NIM1,1)
2222 C
2223 KO = NIM1
2224 KP = KO - 1
2225 KQ = KP + NIM2
2226 KR = KP - 1
2227 DO 148 L = 1,27
2228 M(L) = (L-1)*NN + KO
2229 148 K(L) = L*NN+KO
2230 C
2231 C
2232 DTL = B(K(15)) + B(K(17))*DPN(KR) +
2233 + B(K(18))*DPN(KP) + B(K(20))*DPN(KQ)
2234 DTV = B(K(8)) + B(K(9))*DTL +
2235 + B(K(11))*DPN(KR) + B(K(12))*DPN(KP) +
2236 + B(K(14))*DPN(KQ)
2237 DAL = B(K(3)) + B(K(1))*DTV + B(K(2))*DTL +
2238 + B(K(4))*DPN(KR) + B(K(5))*DPN(KP) +
2239 + B(K(7))*DPN(KQ)
2240 PN(KO) = PN(KO) + DPN(KP)
2241 IF(PN(KO).LT.1.D+04) GO TO 59
2242 IF(PN(KO).GT.4.D+07) GO TO 60
2243 TLN(KO) = TLN(KO) + DTL
2244 TVN(KO) = TVN(KO) + DTV
2245 ALFAN(KO) = ALFAN(KO) + DAL

```

```

2246 TX = SAT(PN(KO))
2247 DTS = TX - TS(KO)
2248 TS(KO) = TX
2249 TW(KP) = TW(KP) + (HCONV(KP)*DTV + HCONL(KP)*DTL +
2250 + HNB(KP)*DTS)*DTW(KP)
2251 C
2252 C
2253 UVZN(KO) = W(M(11))*(DPN(KP)-DPN(KR)) + FUVZN(KO) + UVZN(KO)
2254 ULZN(KO) = W(M(12))*(DPN(KP)-DPN(KR)) + FULZN(KO) + ULZN(KO)
2255 UVRN(KO) = W(M(22))*(DPN(KQ)-DPN(KP)) + FUVRN(KO) + UVRN(KO)
2256 ULRN(KO) = W(M(23))*(DPN(KQ)-DPN(KP)) + FULRN(KO) + ULRN(KO)
2257 C
2258 C CELLS (2,J) , J = 2,NU-1
2259 C
2260 C
2261 DO 51 J = NI,NNJJ,NI
2262 KO = J+2
2263 KP = KO - 1 - 2*J/NI
2264 KM = KP + 1
2265 KQ = KP + NIM2
2266 KR = KP - 1
2267 KS = KP - NIM2
2268 DO 50 L = 1,27
2269 M(L) = (L-1)*NN + KO
2270 50 K(L) = L*NN+KO
2271 C
2272 C
2273 DTL = B(K(15)) + B(K(16))*DPN(KS) +
2274 + B(K(18))*DPN(KP) + B(K(19))*DPN(KM) + B(K(20))*DPN(KQ)
2275 DTV = B(K(8)) + B(K(9))*DTL + B(K(10))*DPN(KS) +
2276 + B(K(12))*DPN(KP) + B(K(13))*DPN(KM) +
2277 + B(K(14))*DPN(KQ)
2278 DAL = B(K(0)) + B(K(1))*DTV + B(K(2))*DTL + B(K(3))*DPN(KS) +
2279 + B(K(5))*DPN(KP) + B(K(6))*DPN(KM) +
2280 + B(K(7))*DPN(KQ)
2281 PN(KO) = PN(KO) + DPN(KP)
2282 IF(PN(KO).LT.1.D+04) GO TO 59
2283 IF(PN(KO).GT.4.D+07) GO TO 60
2284 TLN(KO) = TLN(KO) + DTL
2285 TVN(KO) = TVN(KO) + DTV
2286 ALFAN(KO) = ALFAN(KO) + DAL
2287 TX = SAT(PN(KO))
2288 DTS = TX - TS(KO)
2289 TS(KO) = TX
2290 TW(KP) = TW(KP) + (HCONV(KP)*DTV + HCONL(KP)*DTL +

```

```

2291 + HNB(KP)*DTS)*DTW(KP)
2292 C
2293 C
2294 UVZN(KO) = W(M(11))*DPN(KP) + FUVZN(KO) + UVZN(KO)
2295 ULZN(KO) = W(M(12))*DPN(KP) + FULZN(KO) + ULZN(KO)
2296 UVRN(KO) = W(M(22))*(DPN(KQ)-DPN(KP)) + FUVRN(KO) + UVRN(KO)
2297 ULRN(KO) = W(M(23))*(DPN(KQ)-DPN(KP)) + FULRN(KO) + ULRN(KO)
2298 UVZN(KO-1) = UVZN(KO)
2299 ULZN(KO-1) = ULZN(KO)
2300 51 CONTINUE
2301 C
2302 C CELLS (I,J) , I=3,NI-2 , J=2,NJ-1
2303 C
2304 DO 53 J = NI,NNJJ,NI
2305 DO 53 I = 3,NIM2
2306 KO = I+J
2307 KP = KO - 1 - 2*J/NI
2308 KM = KP + 1
2309 KQ = KP +NIM2
2310 KR = KP - 1
2311 KS = KP - NIM2
2312 DO 52 L = 1,27
2313 M(L) = (L-1)*NN + KO
2314 52 K(L) = L*NN+KO
2315 C
2316 C
2317 DTL = B(K(15)) + B(K(16))*DPN(KS) + B(K(17))*DPN(KR) +
2318 + B(K(18))*DPN(KP) + B(K(19))*DPN(KM) + B(K(20))*DPN(KQ)
2319 DTV = B(K(8)) + B(K(9))*DTL + B(K(10))*DPN(KS) +
2320 + B(K(11))*DPN(KR) + B(K(12))*DPN(KP) + B(K(13))*DPN(KM) +
2321 + B(K(14))*DPN(KQ)
2322 DAL = B(KO) + B(K(1))*DTV + B(K(2))*DTL + B(K(3))*DPN(KS) +
2323 + B(K(4))*DPN(KR) + B(K(5))*DPN(KP) + B(K(6))*DPN(KM) +
2324 + B(K(7))*DPN(KQ)
2325 PN(KO) = PN(KO) + DPN(KP)
2326 IF(PN(KO).LT.1.D+04) GO TO 59
2327 IF(PN(KO).GT.4.D+07) GO TO 60
2328 TLN(KO) = TLN(KO) + DTL
2329 TVN(KO) = TVN(KO) + DTV
2330 ALFAN(KO) = ALFAN(KO) + DAL
2331 TX = SAT(PN(KO))
2332 DTS = TX - TS(KO)
2333 TS(KO) = TX
2334 TW(KP) = TW(KP) + (HCONV(KP)*DTV + HCONL(KP)*DTL +
2335 + HNB(KP)*DTS)*DTW(KP)

```

```

2336 C
2337 C
2338 UVZN(KO) = W(M(11))*(DPN(KP)-DPN(KR)) + FUVZN(KO) + UVZN(KO)
2339 ULZN(KO) = W(M(12))*(DPN(KP)-DPN(KR)) + FULZN(KO) + ULZN(KO)
2340 UVRN(KO) = W(M(22))*(DPN(KQ)-DPN(KP)) + FUVRN(KO) + UVRN(KO)
2341 ULRN(KO) = W(M(23))*(DPN(KQ)-DPN(KP)) + FULRN(KO) + ULRN(KO)
2342 53 CONTINUE
2343 C
2344 C CELLS (NIM1,J) , J=2,NJ-1
2345 C
2346 DO 153 J = NI,NNJJ,NI
2347 I = NIM1
2348 KO = I+J
2349 KP = KO - 1 - 2*J/NI
2350 KQ = KP +NIM2
2351 KR = KP - 1
2352 KS = KP - NIM2.
2353 DO 152 L = 1,27
2354 M(L) = (L-1)*NN + KO
2355 152 K(L) = L*NN+KO
2356 C
2357 C
2358 DTL = B(K(15)) + B(K(16))*DPN(KS) + B(K(17))*DPN(KR) +
2359 + B(K(18))*DPN(KP) + B(K(20))*DPN(KQ)
2360 DTV = B(K(8)) + B(K(9))*DTL + B(K(10))*DPN(KS) +
2361 + B(K(11))*DPN(KR) + B(K(12))*DPN(KP) +
2362 + B(K(14))*DPN(KQ)
2363 DAL = B(KO) + B(K(1))*DTV + B(K(2))*DTL + B(K(3))*DPN(KS) +
2364 + B(K(4))*DPN(KR) + B(K(5))*DPN(KP) +
2365 + B(K(7))*DPN(KQ)
2366 PN(KO) = PN(KO) + DPN(KP)
2367 IF(PN(KO).LT.1.D+04) GO TO 59
2368 IF(PN(KO).GT.4.D+07) GO TO 60
2369 TLN(KO) = TLN(KO) + DTL
2370 TVN(KO) = TVN(KO) + DTV
2371 ALFAN(KO) = ALFAN(KO) + DAL
2372 TX = SAT(PN(KO))
2373 DTS = TX - TS(KO)
2374 TS(KO) = TX
2375 TW(KP) = TW(KP) + (HCONV(KP)*DTV + HCONL(KP)*DTL +
2376 + HNB(KP)*DTS)*DTW(KP)
2377 C
2378 C
2379 UVZN(KO) = W(M(11))*(DPN(KP)-DPN(KR)) + FUVZN(KO) + UVZN(KO)
2380 ULZN(KO) = W(M(12))*(DPN(KP)-DPN(KR)) + FULZN(KO) + ULZN(KO)

```

```

2381      UVRN(KO) = W(M(22))*(DPN(KQ)-DPN(KP)) + FUVRN(KO) + UVRN(KO)
2382      ULRN(KO) = W(M(23))*(DPN(KQ)-DPN(KP)) + FULRN(KO) + ULRN(KO)
2383      153 CONTINUE
2384 C
2385 C          CELLS (I,NJ) , I=3,NI-1
2386 C
2387      DO 55 I = 3,NIM1
2388      KO = I + NNJ
2389      KP = KO + 1 - 2*NJ
2390      KM = KP + 1
2391      KQ = KP + NIM2
2392      KR = KP - 1
2393      KS = KP - NIM2
2394      DO 54 L = 1,27
2395      M(L) = (L-1)*NN + KO
2396      54 K(L) = L*NN+KO
2397 C
2398 C
2399      DTL = B(K(15)) + B(K(16))*DPN(KS) + B(K(17))*DPN(KR) +
2400      +      B(K(18))*DPN(KP) + B(K(19))*DPN(KM)
2401      DTV = B(K(8)) + B(K(9))*DTL + B(K(10))*DPN(KS) +
2402      +      B(K(11))*DPN(KR) + B(K(12))*DPN(KP) + B(K(13))*DPN(KM)
2403      DAL = B(KO) + B(K(1))*DTV + B(K(2))*DTL + B(K(3))*DPN(KS) +
2404      +      B(K(4))*DPN(KR) + B(K(5))*DPN(KP) + B(K(6))*DPN(KM)
2405      PN(KO) = PN(KO) + DPN(KP)
2406      IF(PN(KO).LT.1.D+04) GO TO 59
2407      IF(PN(KO).GT.4.D+07) GO TO 60
2408      TLN(KO) = TLN(KO) + DTL
2409      TVN(KO) = TVN(KO) + DTV
2410      ALFAN(KO) = ALFAN(KO) + DAL
2411      TX = SAT(PN(KO))
2412      DTS = TX - TS(KO)
2413      TS(KO) = TX
2414      TW(KP) = TW(KP) + (HCONV(KP)*DTV + HCONL(KP)*DTL +
2415      +      HNB(KP)*DTS)*DTW(KP)
2416      TCAN(I) = TCAN(I) + TCAN(NI + 1)*(HCONV(KP)*DTV +
2417      +      HCONL(KP)*DTL)
2418 C
2419 C
2420      UVZN(KO) = W(M(11))*(DPN(KP)-DPN(KR)) + FUVZN(KO) + UVZN(KO)
2421      ULZN(KO) = W(M(12))*(DPN(KP)-DPN(KR)) + FULZN(KO) + ULZN(KO)
2422      UVRN(KO) = ZERO
2423      ULRN(KO) = ZERO
2424      55 CONTINUE
2425 C

```

```

2426 C          CELLS (NI,J) , J=1,NJ
2427 C
2428 DO 57 KO = NI,NN,NI
2429 KR = KO -2*KO/NI
2430 DO 56 L = 1,27
2431 M(L) = (L-1)*NN + KO
2432 56 K(L) = L*NN+KO
2433 C
2434 C
2435 UVZN(KO) = FUVZN(KO) - W(M(11))*DPN(KR) + UVZN(KO)
2436 ULZN(KO) = FULZN(KO) - W(M(12))*DPN(KR) + ULZN(KO)
2437 TLN(KO) = TLN(KO-1)
2438 TVN(KO) = TVN(KO-1)
2439 ALFAN(KO) = ALFAN(KO-1)
2440 57 CONTINUE
2441 C
2442 C          CELL (2,NJ)
2443 KO = NNJ + 2
2444 KP = KO + 1 - 2*NJ
2445 KR = KP - 1
2446 KS = KP - NIM2
2447 KM = KP + 1
2448 DO 561 L = 1,27
2449 M(L) = (L-1)*NN + KO
2450 561 K(L) = L*NN+KO
2451 C
2452 C
2453 DTL = B(K(15)) + B(K(16))*DPN(KS) +
+      B(K(18))*DPN(KP) + B(K(19))*DPN(KM)
2454 DTV = B(K(8)) + B(K(9))*DTL + B(K(10))*DPN(KS) +
+      B(K(13))*DPN(KM) + B(K(12))*DPN(KP)
2455 DAL = B(KO) + B(K(11))*DTV + B(K(2))*DTL + B(K(3))*DPN(KS) +
+      B(K(6))*DPN(KM) + B(K(5))*DPN(KP)
2456 PN(KO) = PN(KO) + DPN(KP)
2457 IF(PN(KO).LT.1.D+04) GO TO 59
2458 IF(PN(KO).GT.4.D+07) GO TO 60
2459 TLN(KO) = TLN(KO) + DTL
2460 TVN(KO) = TVN(KO) + DTV
2461 ALFAN(KO) = ALFAN(KO) + DAL
2462 TX = SAT(PN(KO))
2463 DTS = TX - TS(KO)
2464 TS(KO) = TX
2465 TW(KP) = TW(KP) + (HCONV(KP)*DTV + HCONL(KP)*DTL) +
+      HNB(KP)*DTS)*DTW(KP)
2466 TCAN(2) = TCAN(2) + TCAN(NI + 2)*(HCONV(KP)*DTV +

```

```

2471      +          HCONL(KP)*DTL)
2472 C
2473 C
2474      UVZN(KO) = W(M(11))*DPN(KP) + FUVZN(KO) + UVZN(KO)
2475      ULZN(KO) = W(M(12))*DPN(KP) + FULZN(KO) + ULZN(KO)
2476      UVRN(KO) = ZERO
2477      ULRN(KO) = ZERO
2478      UVZN(KO-1) = UVZN(KO)
2479      ULZN(KO-1) = ULZN(KO)
2480 C
2481      DO 357 KO = 1,NN
2482      IF(ALFAN(KO).GE.ZERO) GO TO 257
2483      IF(ALFAN(KO).LT.-1.D-05) IERR = 3
2484      ALFAN(KO) = ZERO
2485 257 CONTINUE
2486      IF(ALFAN(KO).LE.ONE) GO TO 2257
2487      IF(ALFAN(KO).GT.1.00001) IERR = 3
2488      ALFAN(KO) = ONE
2489 2257 CONTINUE
2490      IF(TVN(KO).LT.4.D+02) IERR = 14
2491      IF(TVN(KO).GT.3.D+03) IERR = 15
2492      IF(TLN(KO).LT.4.D+02) IERR = 16
2493      IF(TLN(KO).GT.3.D+03) IERR = 17
2494 357 CONTINUE
2495      RETURN
2496 58 IERR = 2
2497      RETURN
2498 59 IERR = 12
2499      RETURN
2500 60 IERR = 13
2501      RETURN
2502      END

```

Subroutine ccoeff

```

2503 SUBROUTINE COEFF(TV,TL,UVZ,UVR,ULZ,ULR,ALFAZ,ALFAR,
2504 * RHOVZ,RHOVR,RHOLZ,RHOLR,DH,DV,QSI,
2505 * SPPD,WZ1,WZ2,WR1,WR2,FRVZ,FRLZ,FRVR,
2506 * FRLR,C1Z,C1R)
2507 IMPLICIT REAL*8 (A-H,O-Z)
2508 COMMON /NUMBER/ ZERO,ONE,BIG,SMALL
2509 DATA TWO,PTWO,ADRY,CADRY/2.D0,.2D0,.957D0,0.043D0/
2510 C
2511 C SUBROUTINE COEFF CALCULATES THE MOMENTUM EXCHANGE
2512 C COEFFICIENTS.
2513 C C1. ARE THE INTERPHASE MOMENTUM EXCHANGE COEFFICIENTS
2514 C FOR THE TWO DIRECTIONS.
2515 C FR.. ARE THE WALL FRICTION COEFFICIENTS FOR BOTH PHASES
2516 C AND DIRECTIONS.
2517 C
2518 VV = VISCV (TV)
2519 VL = VISCL (TL)
2520 C
2521 AUVZ = DABS (UVZ)
2522 AUVR = DABS (UVR)
2523 AULZ = DABS (ULZ)
2524 AULR = DABS (ULR)
2525 C
2526 REVZ = WZ1*AUVZ*DH/VV +SMALL
2527 RELZ = RHOLZ*AULZ*DH/VL + SMALL
2528 REVR = WR1*AUVR*QSI*DH/VV + SMALL
2529 RELR = WR2*AULR*QSI*DH/VL + SMALL
2530 C
2531 FVZ = 0.180D0/REVZ**PTWO + SPPD*DH
2532 FLZ = 0.180D0/RELZ**PTWO + SPPD*DH
2533 FVR = PTWO/REVR**PTWO
2534 FLR = PTWO/RELR**PTWO
2535 C
2536 FRVZ = (ALFAZ - ADRY)/CADRY*RHOVZ*AUVZ+FVZ/TWO/DH
2537 FRVR = (ALFAR - ADRY)/CADRY*180.*VV/(DH*DH)*QSI
2538 FRLZ = RHOLZ*AULZ*FLZ/TWO/DH
2539 FRLR =180.*VL/(DH*DH)*QSI
2540 XZ = (ONE - ALFAZ)/CADRY
2541 XR = (ONE - ALFAR)/CADRY
2542 C
2543 IF(ALFAZ.GT.ADRY) GO TO 1
2544 FRVZ = ZERO
2545 XZ = ONE

```

```

2546      1 CONTINUE
2547      IF(ALFAR.GT.ADRY) GO TO 2
2548      FRVR = ZERO
2549      XR = ONE
2550      2 CONTINUE
2551 C
2552      FRLZ = FRLZ*XZ
2553      FRLR = FRLR*XR
2554 C
2555      X = (ONE + (ONE-ALFAZ)*75.00)**.95*4.31
2556 C
2557      C1Z = ((ONE - ALFAZ)*DABS(UVZ - ULZ)*RHOVZ/TWO +
2558 + VL/DH)*X/DH
2559      C1R = ((ONE - ALFAR)*DABS(UVR - ULR)*RHOVR/TWO +
2560 + VL/DH)*X*QSI*QSI/DH
2561 C
2562      RETURN
2563      END

```

Subroutine bc

```

2564 SUBROUTINE BC(P,TV,TL,ALFA,TIME,UL,NN,NI,NIM1)*
2565 IMPLICIT REAL*8 (A-H,O-Z)
2566 LOGICAL LP
2567 COMMON /BCX/ ULO
2568 COMMON /BCOND/ TB(51),PNB1(51),PNB2(51),PNB3(51),OMP(51),
2569 * PNT1(51),PNT2(51),PNT3(51),OMT(51),ALB1(51),
2570 * ALB2(51),ALB3(51),OMA(51),TVB1(51),TVB2(51),
2571 * TVB3(51),OMV(51),TLB1(51),TLB2(51),TLB3(51),
2572 * OML(51),HNW1(51),HNW2(51),HNW3(51),OMH(51),
2573 * LMAX,LP(51)
2574 DIMENSION P(NN),TV(NN),TL(NN),ALFA(NN)
2575 C
2576 C
2577 L = 2
2578 1 CONTINUE
2579 IF(TIME.LE.TB(L)) GO TO 2
2580 L = L + 1
2581 IF(L.GT.LMAX) RETURN
2582 GO TO 1
2583 2 CONTINUE
2584 DTIME = TIME - TB(L-1)
2585 C
2586 PNB = PNB1(L)*DTIME + PNB2(L)
2587 PNT = PNT1(L)*DTIME + PNT2(L)
2588 ALB = ALB1(L)*DTIME + ALB2(L)
2589 TVB = TVB1(L)*DTIME + TVB2(L)
2590 TLB = TLB1(L)*DTIME + TLB2(L)
2591 C
2592 IF(LP(L)) GO TO 3
2593 C
2594 PNB = DEXP(OMP(L)*DTIME)*PNB + PNB3(L)
2595 PNT = DEXP(OMT(L)*DTIME)*PNT + PNT3(L)
2596 ALB = DEXP(OMA(L)*DTIME)*ALB + ALB3(L)
2597 TVB = DEXP(OMV(L)*DTIME)*TVB + TVB3(L)
2598 TLB = DEXP(OML(L)*DTIME)*TLB + TLB3(L)
2599 C
2600 3 CONTINUE
2601 DO 4 J = NI,NN,NI
2602 KO = J - NIM1
2603 C
2604 P(KO) = PNB
2605 P(J) = PNT
2606 ALFA(KO) = ALB

```

2607 TV(KO) = TVB  
2608 TL(KO) = TLB  
2609 4 CONTINUE  
2610 RETURN  
2611 END

Function viscl

```
2612 FUNCTION VISCL(T)
2613 IMPLICIT REAL*8 (A-H,O-Z)
2614 C
2615 C     FUNCTION VISCL RETURNS THE SODIUM LIQUID VISCOSITY
2616 C     IN (KG/M/SEC),AS A FUNCTION OF THE TEMPERATURE
2617 C     IN DEGREE CELSIUS
2618 C
2619     TK = T
2620     VISCL = DEXP(508.07/TK - 5.7316 - .4925*DLOG(TK))
2621     RETURN
2622     END
```

Function viscv

```
2623 FUNCTION VISCV(T)
2624 IMPLICIT REAL*8 (A-H,O-Z)
2625 C
2626 C FUNCTION VISCV RETURNS THE SODIUM VAPOR VISCOSITY
2627 C IN (KG/M/SEC),AS A FUNCTION OF THE TEMPERATURE
2628 C IN DEGREE CELSIUS
2629 C
2630 TK = T
2631 VISCV = 6.085D-09*TK + 1.261D-05
2632 RETURN
2633 END
```

Function surten

```
2634 FUNCTION SURTEN (T)
2635 IMPLICIT REAL*8 (A-H,O-Z)
2636 C
2637 C      FUNCTION SURTEN RETURNS THE SURFACE TENSION OF LIQUID
2638 C      SODIUM IN NEWTON/METER
2639 C      CORRELATION FROM GOLDEN AND TOKAR.
2640 C
2641 TC = T - 273.14
2642 SURTEN = 2.067D-01 - 1.0D-04*TC
2643 IF(SURTEN.LT.0.D0) SURTEN = 0.D0
2644 RETURN
2645 END
```

Function sat

```
2646 FUNCTION SAT(P)  
2647 IMPLICIT REAL*8 (A-H,O-Z)  
2648 C  
2649 SAT = 12020./{(21.9358 - DLOG(P))  
2650 RETURN  
2651 END
```

Function dtsdp

```
2652     FUNCTION DTSDP(P)
2653     IMPLICIT REAL*8 (A-H,O-Z)
2654     C
2655     C           CALCULATES THE DERIVATIVE OF THE SATURATION
2656     C           TEMPERATURE WITH RESPECT TO THE PRESSURE
2657     C
2658     X = 21.9358 - DLOG(P)
2659     DTSDP = 12020./((X*X*P)
2660     RETURN
2661     END
```

Function cand1

```
2662 FUNCTION CONDL(T)
2663 IMPLICIT REAL*8 (A-H,O-Z)
2664 DATA A1,A2,A3,X1,X2,X3 /54.306,-1.878D-02,2.0914D-06,1.8D0,
2665 * 459.67D0,1.7307D0/
2666 C
2667 TF = X1*T - X2
2668 T2 = TF*TF
2669 C = A1 + A2*TF + A3*T2
2670 CONDL = C*X3
2671 RETURN
2672 END
```

Function condv

```
2673 FUNCTION COND(V,T)
2674 IMPLICIT REAL*8 (A-H,O-Z)
2675 DATA A1,A2,A3,X1,X2,X3 /16.39D-04,3.977D-05,-9.697D-09,
2676 * 1.8D0,459.67D0,1.7307D0/
2677 C
2678 TF = X1*T - X2
2679 T2 = TF*TF
2680 C = A1 + A2*TF + A3*T2
2681 COND = X3*C
2682 RETURN
2683 END
```

Function cql

```
2684 FUNCTION CPL(T)
2685 IMPLICIT REAL*8 (A-H,O-Z)
2686 DATA A1,A2,A3,X1,X2 / .389352D0,1.10599D-04,3.41178D-08,
2687 * 1.8D0,4.1869D+03/
2688 C
2689 TR = T*X1
2690 T2 = TR*TR
2691 CP = A1 - A2*TR + A3*T2
2692 CPL = X2*CP
2693 RETURN
2694 END
```

Function prv

```
2695 FUNCTION PRV(T)
2696 IMPLICIT REAL*8 (A-H, O-Z)
2697 C
2698 TX = T - 844.1
2699 PRV = .759600 + .810D-06*TX*TX
2700 RETURN
2701 END
```

Function prl

```
2702 FUNCTION PRL(T)
2703 IMPLICIT REAL*8 (A-H,O-Z)
2704 C
2705 PRL = CPL(T)*VISCL(T)/CONDL(T)
2706 RETURN
2707 END
```

Function hfg

```
2708 FUNCTION HFG(P)
2709 IMPLICIT REAL*8 (A-H,O-Z)
2710 C
2711 T = SAT(P)
2712 HFG = 5.0890+06 - 1.0430+03*T
2713 RETURN
2714 END
```

Subroutine htcf

```

2715 SUBROUTINE HTCFC (P,TV,TL,ALFA,RHOV,RHOL,HV,HL,DH,TS,TW,
2716 * HCONV,HCONL,HNB,UV,UL)
2717 IMPLICIT REAL*8 (A-H,O-Z)
2718 COMMON /NUMBER/ ZERO,ONE,BIG,SMALL
2719 COMMON /POVERD/ R
2720 C
2721 HCONV = ZERO
2722 HCONL = ZERO
2723 HNB = ZERO
2724 C
2725 VV = VISCV(TV)
2726 VL = VISCL(TL)
2727 PV = PRV(TV)
2728 PL = PRL(TL)
2729 CV = CONDV(TV)
2730 CL = CONDL(TL)
2731 AUV = DABS(UV)
2732 AUL = DABS(UL)
2733 SIG = SURTEN(TL)
2734 C
2735 C COMPUTE QUALITY
2736 C
2737 GV = ALFA*RHOV*AUV
2738 GL = (ONE-ALFA)*RHOL*AUL
2739 G = GV + GL
2740 IF((UV-UL)*UL.LE.ZERO) GO TO 1
2741 X = GV/G
2742 GO TO 2
2743 1 CONTINUE
2744 X = ALFA*RHOV/(ALFA*RHOV + (ONE-ALFA)*RHOL)
2745 2 CONTINUE
2746 C
2747 C SINGLE PHASE : DITTUS-BOELTER CORRELATION (VAPOR)
2748 C
2749 IF(ALFA.LE.0.96) GO TO 3
2750 REV = RHOV*AUV*DH/VV
2751 HCONV = 0.023*REV**0.8*PV**0.4*CV/DH
2752 RETURN
2753 3 CONTINUE
2754 C
2755 C SINGLE PHASE : SCHAD CORRELATION (LIQUID)
2756 C
2757 REL = RHOL*AUL*DH/VL

```

```

2758      PEL = REL*PL
2759      IF(PEL.LE.150.) GO TO 4
2760      HCONL = PEL**0.3*R*CL/DH
2761      GO TO 5
2762      4 CONTINUE
2763      HCONL = 4.5*R*CL/DH
2764      5 CONTINUE
2765 C
2766 C          TWO PHASES : CHEN CORRELATION
2767 C
2768      XTTI = (X/(ONE-X))**0.9*(RHOL/RHOV)**0.5*(VV/VL)**0.1
2769      F = (XTTI + .213)**0.736*2.35D0
2770      IF(F.LT.ONE) RETURN
2771      HCONL = F**0.375*HCONL
2772 C
2773      IF(TW.LE.TL) GO TO 7
2774 C
2775      FX = ONE
2776      GX = G
2777      IF(TL.LT.TS) GO TO 7
2778      IF(XTTI.GT.0.1) FX = F
2779      GX = GL
2780      6 CONTINUE
2781      REL = GX*DH/VL
2782      RETP = REL*FX**1.25*1.D-04
2783      S = 0.1D0
2784      IF(RETP.LT.70.D0.AND.RETP.GE.32.5D0) S = ONE/
2785      /      (ONE + RETP**0.78*0.42D0)
2786 C
2787      IF(RETP.LT.32.5D0) S = ONE/(ONE + .12D0*RETP**1.14)
2788 C
2789      HS = 1.22D-03*S*DSQRT(CL*CPL(TL)/SIG)/PL**.29*
2790      *      RHOL**.25*(CPL(TL)*RHOL/RHOV/HFG(P))**.24
2791 C
2792      PWALL = DEXP(21.9358D0 - 12020.D0/TW)
2793      Z = DABS(PWALL - P)
2794 C
2795      HNB = HS*(TW - TS)**.24*Z**.75
2796      7 CONTINUE
2797      IF(ALFA.LE.0.88) RETURN
2798 C
2799      FAL = 12.D0 - 12.5D0*ALFA
2800      FAL = FAL*FAL*FAL
2801      REV = RHOV*AUV*DH/VV
2802      HCV = 0.023*(REV*REV*PV)**0.4*CV/DH

```

```
2803 HCONL = HCONL*FAL + HCV  
2804 HNS = ZERO  
2805 RETURN  
2806 END
```

Subroutine iphtc

```
2807 SUBROUTINE IPHTC (HIF,ALFA)
2808 IMPLICIT REAL*8 (A-H,O-Z)
2809 COMMON /NUMBER/ ZERO,ONE,BIG,SMALL
2810 C
2811 HIF = 5.D+08
2812 RETURN
2813 END
```

Subroutine state

```

2814 SUBROUTINE STATE (TV,TL,P,PROP,IFLAG)
2815 IMPLICIT REAL*8 (A-H,O-Z)
2816 COMMON /ERROR/ IERR
2817 COMMON /NUMBER/ ZERO,ONE,BIG,SMALL
2818 DIMENSION PROP(3,4)
2819 DATA RV0,RV1,RV2,RV22 /1.605D-02,2.51D-06,-3.23D-13,-6.46D-13/
2820 DATA RLO,RL1,RL2,RL3,RLP,RL22,RL33 /1.0116D+03,-0.2205,
2821 1 -1.9224D-05,5.6377D-09,2.26D-07,-3.8448D-05,
2822 2 1.69131D-08/
2823 DATA EVO,EV1,EV2,EV3,EV22,EV33 /5.0215D+06,5.8714D+02,
2824 1 -.41672,1.54272D-04,-.83344,4.62816D-04/
2825 DATA ELO,EL1,EL2,EL3,EL22,EL33 /-6.75075D+04,1.63014D+03,
2826 1 -.41672,1.54272D-04,-.83344,4.62816D-04/
2827 C
2828 C ALL PROPERTIES IN SI UNITS
2829 C PROPERTIES BASED IN
2830 C GOLDEN,G.H. AND TOKAR,J.V.,
2831 C THERMOPHYSICAL PROPERTIES OF SODIUM, ANL-7323
2832 C WITH THE ADDITION OF PRESSURE DEPENDENCE IN THE
2833 C LIQUID DENSITY.
2834 C THIS ADDITION WAS MADE BECAUSE THE NUMERICAL
2835 C STABILITY OF THE MODEL REQUIRES A NON ZERO,
2836 C POSITIVE VALUE OF THE PRESSURE DERIVATIVE OF
2837 C THE DENSITY .
2838 C
2839 C ALSO A REQUIREMENT FOR THE NUMERICAL CONVERGENCE
2840 C IS THE DERIVATIVES OF PROPERTIES WITH RESPECT TO
2841 C TEMPERATURE AND PRESSURE BEING THE MATHEMATICAL
2842 C DERIVATIVES OF THE EXPRESSIONS FOR THE PROPERTIES
2843 C
2844 C
2845 C TS = SAT(P)
2846 C X1 = (RV2*P + RV1)*P + RV0
2847 C PROP(1,1) = X1*TS/TV
2848 C PROP(1,2) = ((RL3*TL + RL2)*TL + RL1)*TL + RLO + RLP*P
2849 C PROP(1,3) = ((EV3*TV + EV2)*TV + EV1)*TV + EVO - P/PROP(1,1)
2850 C PROP(1,4) = ((EL3*TL + EL2)*TL + EL1)*TL + ELO
2851 C
2852 C PROP(2,1) = -PROP(1,1)/TV
2853 C PROP(2,2) = (RL33*TL + RL22)*TL + RL1
2854 C PROP(2,3) = (EV33*TV + EV22)*TV + EV1
2855 C PROP(2,4) = (EL33*TL + EL22)*TL + EL1
2856 C

```

```
2857     PROP(3,1) = (X1*DTSDP(P) + (RV22*P + RV1)*TS)/TV
2858     PROP(3,2) = RLP
2859     PROP(3,3) = (P/PROP(1,1)*PROP(3,1) - ONE)/PROP(1,1)
2860     PROP(3,4) = ZERO
2861     RETURN
2862     END
```

Subroutine noneq

```

2863 SUBROUTINE NONEQ(ALFAO,ALFA,TV,TL,P,RHOV,RHOL,TS,S,IFLAG)
2864 IMPLICIT REAL*8 (A-H,O-Z)
2865 COMMON /ERROR/ IERR
2866 COMMON /NUMBER/ ZERO,ONE,BIG,SMALL
2867 COMMON /PD/ D4,POD2
2868 DIMENSION S(5,2)
2869 DATA AN,RGAS /1.3333333D+07,.14469D+03/,HALF /0.5D0/
2870 DATA PI,SR3,CADRY,ADRY /3.141592654,3.464101616,0.043,0.957/
2871 DATA H0,H1 /5.089D+06,-.1043D+04/
2872 DATA RNU /6.D+03/
2873 DATA HL0,HL1,HL2,HL3 /-6.75075D+04,1.63014D+03,
2874 * -.41672D0,1.54272D-04/
2875 C
2876 C SUBROUTINE NONEQ CALCULATES THE MASS AND ENERGY EXCHANGE RATES
2877 C AND ITS DERIVATIVES.
2878 C AN = 4/3*N, N = 1.0D+07 BUBLES/CUBIC METER
2879 C RGAS = SQUARE ROOT OF GAS CONSTANT FOR SODIUM OVER 2*PI
2880 C POD2 = PITCH TO DIAMETER RATIO SQUARED
2881 C
2882 C S(1, ) = EXCHANGE RATE S( .1) = MASS
2883 C S(2, ) = D/DTV S( .2) = ENERGY
2884 C S(3, ) = D/DYL
2885 C S(4, ) = D/DP
2886 C S(5, ) = D/DALFA
2887 C
2888 AX = ALFAO
2889 IF(ALFAO.LT.1.D-04) AX = 1.D-04
2890 IF(ALFAO.GT.0.9999) AX = 0.9999
2891 C
2892 TS = SAT(P)
2893 HLG = H1*TS + H0
2894 X = ONE/(SR3*POD2 - PI)
2895 C
2896 AM = 1.2D-07*PI*X*D4*D4
2897 IF(ALFAO.GT.0.6) GO TO 10
2898 C
2899 XX = 3.*PI*AX*X
2900 GO TO 20
2901 10 CONTINUE
2902 Y = ONE
2903 IF(AX.GT.ADRY) Y = (ONE - AX)/CADRY
2904 XK = 1.8/(SR3*POD2*X - 0.6)
2905 XX = (SR3*POD2*X - AX)*X*Y*PI*XK

```

INSTITUTO DE INVESTIGACIONES Y ESTUDIOS EN CIENCIAS

```

2906 20 CONTINUE
2907 A = DSQRT(XX)*D4
2908 C
2909 30 CONTINUE
2910 CE = ARGAS*RHOV*RHOV
2911 CC = CE*(ONE - AX)
2912 CE = CE*AX
2913 C
2914 EL = ZERO
2915 CL = ZERO
2916 IF(TL.GT.TS) EL = -1.0D0
2917 IF(TS.GT.TV) CL = 5.D-03
2918 C
2919 CE = CE*EL
2920 CC = CC*CL
2921 C
2922 DDP = DTSDP(P)
2923 SRTS = DSQRT(TS)
2924 DTL = (TL - TS)/SRTS
2925 DTV = (TS - TV)/SRTS
2926 C
2927 C MASS EXCHANGE RATE
2928 C
2929 SE = DTL*CE*(ONE - ALFA)
2930 SC = DTV*CC*ALFA
2931 S(1,1) = SE - SC
2932 C
2933 C DERIVATIVES
2934 C
2935 S(2,1) = CC*ALFA/SRTS
2936 S(3,1) = CE*(ONE-ALFA)/SRTS
2937 DSEVAP = CE*(ALFA-ONE)*(TS+TL)/TS/SRTS+HALF*DDP
2938 DSCOND = CC*ALFA*(TS+TV)/TS/SRTS+HALF*DDP
2939 S(4,1) = DSEVAP - DSCOND
2940 S(5,1) = -CE*DTL - CC*DTV
2941 C
2942 C ENERGY EXCHANGE RATE
2943 C
2944 U = A*CONDL(TV)*RNU*D4
2945 HL = ((HL3*TS + HL2)*TS + HL1)*TS + HLO
2946 HV = HL + HLG
2947 DHLDP = ((3.*HL3*TS + 2.*HL2)*TS + HL1)*DDP
2948 DHVDP = DHLDP + H1*DDP
2949 C
2950 S(1,2) = SE*HV - SC*HL + U*(TL - TV)

```

2951 C  
2952 C  
2953 C  
2954  
2955  
2956  
2957  
2958  
2959

DERIVATIVES

$S(2,2) = S(2,1)*HL - U$   
 $S(3,2) = S(3,1)*HV + U$   
 $S(4,2) = DSEVAP*HV + SE*DHVDP - DSCOND*HL - SC*DHLDP$   
 $S(5,2) = -CE*DTL*HV - CC*DTV*HL$   
RETURN  
END

Subroutine condt

```
2960 SUBROUTINE CONDT(TV,TL,P,ALFA,TS,TW,DTW,  
2961 * HCONV,HCONL,HNB,DV,Q,KO)  
2962 IMPLICIT REAL*8 (A-H,O-Z)  
2963 LOGICAL LSS  
2964 COMMON /STST/ TAFP,LSS  
2965 COMMON /ERROR/ IERR  
2966 COMMON /NUMBER/ ZERO,ONE,BIG,SMALL  
2967 DIMENSION Q(4,2)  
2968 C  
2969 Q(1,1) = (TW - TV)*HCONV*DV  
2970 Q(1,2) = ((TW - TL)*HCONL + (TW - TS)*HNB)*DV  
2971 Q(2,1) = (DTW*HCONV - 1)*HCONV*DV  
2972 Q(2,2) = ZERO  
2973 Q(3,1) = ZERO  
2974 Q(3,2) = ((HCONL + HNB)*DTW - 1)*HCONL*DV  
2975 Q(4,1) = ZERO  
2976 Q(4,2) = ((HCONL + HNB)*DTW - 1)*HNB*DV*DTSDP(P)  
2977 RETURN  
2978 END
```

Subroutine hexcan

```
2979 SUBROUTINE HEXCAN(TCAN,DTC,TV,TL,HCONV,HCONL,QV,QL,  
2980 * DQDTV,DQDTL)  
2981 IMPLICIT REAL*8 (A-H,O-Z)  
2982 COMMON /NUMBER/ ZERO,ONE,BIG,SMALL  
2983 COMMON /HXCN/ ACOV  
2984 C  
2985 C SUBROUTINE HEXCAN CALCULATES THE HEAT TRANSFERED TO  
2986 C THE HEXCAN AND ITS DERIVATIVES.  
2987 C  
2988 QV = ACOV*HCONV*(TV - TCAN)  
2989 QL = ACOV*HCONL*(TL - TCAN)  
2990 DQDTV = ACOV*HCONV*(ONE - DTC*HCONV)  
2991 DQDTL = ACOV*HCONL*(ONE - DTC*HCONL)  
2992 RETURN  
2993 END
```

Subroutine fprop

```

2994 SUBROUTINE FPROP(TRN,NPIN,NPM1,I)
2995 IMPLICIT REAL*8 (A-H,O-Z)
2996 COMMON /NUMBER/ ZERO,ONE,BIG,SMALL
2997 COMMON /PIN1/ CPIN(20),ROCP(20)
2998 COMMON /ICONST/ NCF,NCC,NG
2999 DIMENSION TRN(NPIN)
3000 C
3001 C FUEL PROPERTIES
3002 C
3003 DO 1 K = 1,NCF
3004 T = (TRN(K+1) + TRN(K))/2.DO
3005 CALL FUEL (T,K,I)
3006 1 CONTINUE
3007 C
3008 C CLAD PROPERTIES
3009 C
3010 DO 2 K = NCC,NPM1
3011 T = (TRN(K+1) + TRN(K))/2.DO
3012 CALL CLAD (T,K)
3013 2 CONTINUE
3014 C
3015 C GAP CONDUCTIVITY
3016 C
3017 T = (TRN(NG+1) + TRN(NG))/2.DO
3018 CALL GAP (T,TRN(NG),TRN(NG+1),NG)
3019 RETURN
3020 END

```

Subroutine fuel

```
3021 SUBROUTINE FUEL (T,K,I)
3022 IMPLICIT REAL*8 (A-H,O-Z)
3023 COMMON /NUMBER/ ZERO,ONE,BIG,SMALL
3024 COMMON /PIN1/ CPIN(20),ROCP(20)
3025 COMMON /FCONST/ A0,A1,A2,A3,
3026 * B0,B1,B2,AD,APU,LPLNM(40)
3027 C
3028 T2 = T*T
3029 T3 = T*T2
3030 X = 2.74D0 - 5.8D-04*T
3031 C
3032 CPIN(K) = (B0 + B1*T + B2*T2)*(ONE - (ONE - AD)*X)
3033 ROCP(K) = (A0 + A1*T + A2*T2 + A3*T3)*AD*(ONE + 0.045*APU)
3034 IF(LPLNM(I).EQ.0) ROCP(K) = 1.D+04
3035 RETURN
3036 END
```

Subroutine clad

```
3037 SUBROUTINE CLAD (T,K)
3038 IMPLICIT REAL*8 (A-H,O-Z)
3039 COMMON /NUMBER/ ZERO,GNE,BIG,SMALL
3040 COMMON /PIN1/ CPIN(20),ROCP(20)
3041 COMMON /CCONST/ A0,A1,A2,A3,B0,B1,B2,B3
3042 C
3043 T2 = T*T
3044 T3 = T*T2
3045 C
3046 CPIN(K) = B0 + B1*T + B2*T2 + B3*T3
3047 ROCP(K) = A0 + A1*T + A2*T2 + A3*T3
3048 RETURN
3049 END
```

Subroutine gap

```
3050 SUBROUTINE GAP (T,TF,TC,NG)
3051 IMPLICIT REAL*8 (A-H,O-Z)
3052 COMMON /NUMBER/ ZERO,ONE,BIG,SMALL
3053 COMMON /PIN1/ CPIN(20),ROCP(20)
3054 COMMON /GCONST/ DIL,RADFU,RADCL
3055 C
3056 DATA ESB,HMIN /1.7D-08,3.705D+03/
3057 DATA C1,C2 /2.D0,1.5D+01/
3058 DATA G1,G2,G3 /1.32D-04,0.61D-04,1.8D+03/
3059 C
3060 C CONDUCTION HEAT TRANSFER
3061 C
3062 DGAP = RADCL - RADFU
3063 CG = C2**DIL*C1
3064 HG = ONE/((DGAP + G1)/CG + G2) + G3
3065 C
3066 C RADIATION HEAT TRANSFER
3067 C
3068 HR = (TF*TF + TC*TC)*(TF + TC)*ESB
3069 HGAP = HG + HR
3070 IF(HGAP.LT.HMIN) HGAP = HMIN
3071 C
3072 ROCP(NG) = ZERO
3073 CPIN(NG) = HGAP
3074 RETURN
3075 END
```

Subroutine fpin

```

3076 SUBROUTINE FPIN(TV,TL,TS,TW,DTW,HCONV,HCONL,HNB,
3077 * TR,DTR,DT,NPIN,NPM1,KO)
3078 IMPLICIT REAL*8 (A-H,O-Z)
3079 LOGICAL LSS
3080 COMMON /NUMBER/ ZERO,ONE,BIG,SMALL
3081 COMMON /PINO/ RDR(20),VP(20),VM(20),RADR,PPP(20)
3082 COMMON /PINI/ CPIN(20),ROCP(20)
3083 COMMON /STST/ TAFP,LSS
3084 DIMENSION A1(20),A2(20),A3(20),B1(20)
3085 DIMENSION TR(NPIN),DTR(NPIN)
3086 C
3087 CALL POWER(HEAT,KO)
3088 C
3089 DTI = ONE/DT
3090 IF(LSS) DTI = ZERO
3091 C
3092 A1(1) = ZERO
3093 A2(1) = RDR(1)*CPIN(1) + VP(1)*ROCP(1)*DTI
3094 B1(1) = VP(1)*HEAT*PPP(1) + VP(1)*ROCP(1)*TR(1)*DTI
3095 DO 1 K = 2,NPM1
3096 KM1 = K - 1
3097 A1(K) = -RDR(KM1)*CPIN(KM1)
3098 A2(K) = -A1(K) + RDR(K)*CPIN(K) + (VP(K)*ROCP(K) +
3099 + VM(K)*ROCP(KM1))*DTI
3100 B1(K) = VP(K)*HEAT*PPP(K) + VM(K)*HEAT*PPP(KM1) +
3101 + (VP(K)*ROCP(K) + VM(K)*ROCP(KM1))*TR(K)*DTI
3102 1 CONTINUE
3103 C
3104 A1(NPIN) = -RDR(NPM1)*CPIN(NPM1)
3105 A2(NPIN) = -A1(NPIN) + VM(NPIN)*ROCP(NPM1)*DTI +
3106 + RADR*(HCONV + HCONL + HNB)
3107 B1(NPIN) = VM(NPIN)*ROCP(NPM1)*TR(NPIN)*DTI +
3108 + RADR*(HCONV*TV + HCONL*TL + HNB*TS) +
3109 + VM(NPIN)*HEAT*PPP(NPM1)
3110 C
3111 A1(NPIN+1) = ZERO
3112 C
3113 A2(1) = ONE/A2(1)
3114 A3(1) = A1(2)*A2(1)
3115 B1(1) = B1(1)*A2(1)
3116 C
3117 DO 2 K = 2,NPIN
3118 KM1 = K - 1

```

```
3119      A2(K) = ONE/(A2(K) - A1(K)*A3(KM1))
3120      A3(K) = A1(K+1)*A2(K)
3121      B1(K) = (B1(K) - A1(K)*B1(KM1))*A2(K)
3122      2 CONTINUE
3123 C
3124      TW = B1(NPIN)
3125      DTW = A2(NPIN)*RADR
3126      DD 3 K = 1,NPM1
3127      TR(K) = B1(K)
3128      DTR(K) = A3(K)
3129      3 CONTINUE
3130      RETURN
3131      END
```

Subroutine ftp

```

3132 SUBROUTINE FTP(TV,TL,TS,TW,HCONV,HCONL,HNB,TR,DTR,QPP,
3133 * NI,NJ,NN,NP,NTR,NPM1,NIM2,NPIN)
3134 IMPLICIT REAL*8 (A-H,O-Z)
3135 COMMON /NUMBER/ ZERO,ONE,BIG,SMALL
3136 DIMENSION TR(NTR),DTR(NTR),TW(NP),TS(NN),TV(NN),TL(NN),
3137 * HCONV(NP),HCONL(NP),HNB(NP),QPP(NN)
3138 C
3139 TWMAX = ZERO
3140 TRMAX = ZERO
3141 C
3142 DO 3 I = 1,NIM2
3143 DO 3 J = 1,NJ
3144 KO = (J-1)*NI + I + 1
3145 KP = (J-1)*NIM2 + I
3146 KR = KP*NPIN
3147 C
3148 TR(KR) = TW(KP)
3149 C
3150 DO 1 KK = 1,NPM1
3151 KTR = KR - KK
3152 C
3153 TR(KTR) = TR(KTR) - DTR(KTR)*TR(KTR+1)
3154 IF(TRMAX.GT.TR(KTR)) GO TO 1
3155 TRMAX = TR(KTR)
3156 KTRMAX = KTR
3157 1 CONTINUE
3158 C
3159 IF(TWMAX.GT.TW(KP)) GO TO 2
3160 TWMAX = TW(KP)
3161 KTWMAX = KO
3162 2 CONTINUE
3163 C
3164 QPP(KP) = HCONV(KP)*(TW(KP) - TV(KO)) + HCONL(KP)*
3165 * (TW(KP)-TL(KO)) + HNB(KP)*(TW(KP)-TS(KO))
3166 3 CONTINUE
3167 RETURN
3168 END

```



Subroutine thxcn0

```
3197 SUBROUTINE THXCNO(TCAN,NCAN,NI)
3198 IMPLICIT REAL*8 (A-H,O-Z)
3199 DIMENSION TCAN(NCAN)
3200 C
3201 C SUBROUTINE THXCNO TRANSFERS THE NEW VALUE OF THE HEXCAN
3202 C TEMPERATURE TO THE OLD HEXCAN TEMPERATURE ARRAY.
3203 C
3204 TCAN(1) = TCAN(2)
3205 TCAN(NI) = TCAN(NI-1)
3206 DO 10 I = 1,NI
3207 K3 = 2*NI + I
3208 TCAN(K3) = TCAN(I)
3209 10 CONTINUE
3210 RETURN
3211 END
```

Subroutine power

```

3212 SUBROUTINE POWER (HEAT,KO)
3213 IMPLICIT REAL*8 (A-H,O-Z)
3214 LOGICAL LP
3215 COMMON /ERROR/ IERR
3216 COMMON /NUMBER/ ZERO,ONE,BIG,SMALL
3217 COMMON /PSHAPE/ SHAPE(100)
3218 COMMON /TEMPO/ TIME,DT,DT0,DTLS,NOT
3219 COMMON /BCOND/ TB(51),PNB1(51),PNB2(51),PNB3(51),OMP(51),
3220 * PNT1(51),PNT2(51),PNT3(51),OMT(51),ALB1(51),
3221 * ALB2(51),ALB3(51),OMA(51),TVB1(51),TVB2(51),
3222 * TVB3(51),OMV(51),TLB1(51),TLB2(51),TLB3(51),
3223 * OML(51),HNW1(51),HNW2(51),HNW3(51),OMH(51),
3224 * LMAX,LP(51)
3225 C
3226 C
3227 L = 2
3228 1 CONTINUE
3229 IF(TIME.LE.TB(L)) GO TO 2
3230 L = L + 1
3231 IF(L.GT.LMAX) RETURN
3232 GO TO 1
3233 2 CONTINUE
3234 DTIME = TIME - TB(L-1)
3235 HEAT = HNW1(L)*DTIME + HNW2(L)
3236 IF(LP(L)) GO TO 3
3237 HEAT = DCDS(OMH(L)*DTIME)*HEAT + HNW3(L)
3238 3 CONTINUE
3239 HEAT = SHAPE(KO)*HEAT
3240 RETURN
3241 END

```

Subroutine gaussie

```

3242 SUBROUTINE GAUSIE (A1,A2,A3,A4,F,X,BETA,GAMMA,NC)
3243 IMPLICIT REAL*8 (A-H,O-Z)
3244 COMMON /NUMBER/ ZERO,ONE,BIG,SMALL
3245 COMMON /GAUSS/ NZ,NR,NZM1
3246 COMMON /ERROR/ IERR
3247 COMMON /CNTRL/ EPS1,EPS2,RES,IT1,IT2,IT3,ITM1,ITM2,ITRMAX
3248 DIMENSION A1(NC),A2(NC),A3(NC),A4(NC),F(NC),X(NC),
3249 *          BETA(NC),GAMMA(NC)
3250 C
3251 ITR = 0
3252 1 CONTINUE
3253 C
3254 C      NEW SOLUTION AT THE BOTTON
3255 C
3256 I = 1
3257 GAMMA(1) = F(I) - A3(I)*X(I+1)
3258 BETA(1) = ONE
3259 C
3260 DO 2 J = 2,NR
3261 K = (J-1)*NZ + I
3262 K1 = K - NZ
3263 C
3264 BETA(J) = ONE - A1(K)*A4(K1)/BETA(J-1)
3265 GAMMA(J) = (F(K)-A3(K)*X(K+1) - A1(K)*GAMMA(J-1))/BETA(J)
3266 2 CONTINUE
3267 C
3268 K = (NR-1)*NZ + I
3269 CONV = DABS(X(K) - GAMMA(NR))
3270 X(K) = GAMMA(NR)
3271 DO 3 J = 2,NR
3272 K = NR - J + 1
3273 KX = (K-1)*NZ + I
3274 XA = GAMMA(K) - A4(KX)*X(KX+NZ)/BETA(K)
3275 DX = DABS(X(KX) - XA)
3276 IF(DX.GT.CONV) CONV = DX
3277 X(KX) = XA
3278 3 CONTINUE
3279 C
3280 C      NEW SOLUTION OUT OF THE BOUNDARIES
3281 C
3282 DO 6 I = 2,NZM1
3283 C
3284 GAMMA(1) = F(I) - A2(I)*X(I-1) - A3(I)*X(I+1)

```

```

3285 DO 4 J = 2, NR
3286 K = (J-1)*NZ + I
3297 K1 = K - NZ
3288 C
3289 BETA(J) = ONE - A1(K)*A4(K1)/BETA(J-1)
3290 GAMMA(J) = (F(K) - A2(K)*X(K-1) - A3(K)*X(K+1) -
3291 - A1(K)*GAMMA(J-1))/BETA(J)
3292 4 CONTINUE
3293 C
3294 K = (NR-1)*NZ + I
3295 DX = DABS(X(K) - GAMMA(NR))
3296 IF(DX.GT.CONV) CONV = DX
3297 X(K) = GAMMA(NR)
3298 C
3299 DO 5 J = 2, NR
3300 K = NR - J + 1
3301 KX = (K-1)*NZ + I
3302 XA = GAMMA(K) - A4(KX)*X(KX+NZ)/BETA(K)
3303 DX = DABS(X(KX) - XA)
3304 IF(DX.GT.CONV) CONV = DX
3305 X(KX) = XA
3306 5 CONTINUE
3307 6 CONTINUE
3308 C
3309 C NEW SOLUTION AT THE TOP
3310 C
3311 I = NZ
3312 GAMMA(1) = F(I) - A2(I)*X(I-1)
3313 DO 7 J = 2, NR
3314 K = (J-1)*NZ + I
3315 K1 = K - NZ
3316 C
3317 BETA(J) = ONE - A1(K)*A4(K1)/BETA(J-1)
3318 GAMMA(J) = (F(K) - A2(K)*X(K-1) - A1(K)*GAMMA(J-1))/
3319 BETA(J)
3320 7 CONTINUE
3321 C
3322 X = (NR-1)*NZ + I
3323 DX = DABS(X(K) - GAMMA(NR))
3324 IF(DX.GT.CONV) CONV = DX
3325 X(K) = GAMMA(NR)
3326 C
3327 DO 8 J = 2, NR
3328 K = NR - J + 1
3329 KX = (K-1)*NZ + I

```

```

3330      XA = GAMMA(K) - A4(KX)*X(KX+NZ)/BETA(K)
3331      DX = DABS(X(KX) - XA)
3332      IF(DX.GT.CONV) CONV = DX
3333      X(KX) = XA
3334      8 CONTINUE
3335 C
3336 C          CONVERGENCE TEST
3337 C
3338      IF(CONV - EPS2) 11,11,9
3339      9 IF(ITR - ITRMAX) 1,10,10
3340      10 IERR = 1
3341      11 CONTINUE
3342      RES = ZERO
3343      DO 12 L = 1,NC
3344      XX = DABS(X(L))
3345      IF(XX.GT.RES) RES = XX
3346      12 CONTINUE
3347      RETURN
3348      END

```

Subroutine errmes

```
3349 SUBROUTINE ERRMES(TIME)
3350 C
3351 C SUBROUTINE ERRMES PRINTS THE ERROR MESSAGES
3352 C WHENEVER THE EXECUTION OF THE PROGRAM HAS
3353 C BEEN TERMINATED DUE TO NUMERICAL ERRORS SUCH
3354 C AS INSTABILITY,VARIABLES OUT OF RANGE ETC.
3355 C
3356 C
3357 C IMPLICIT REAL*8 (A-H,O-Z)
3358 C COMMON /ERROR/ IERR
3359 C
3360 C WRITE(6,1100) TIME
3361 C
3362 C IF(IERR - 2) 1,2,100
3363 100 IF(IERR - 4) 3,4,101
3364 101 IF(IERR - 22) 21,22,102
3365 102 IF(IERR - 24) 23,24,103
3366 103 IF(IERR - 26) 25,26,104
3367 104 IF(IERR - 28) 27,50,50
3368 C
3369 1 WRITE(6,1001)
3370 GO TO 200
3371 2 WRITE(6,1002)
3372 GO TO 200
3373 3 WRITE(6,1003)
3374 GO TO 200
3375 4 WRITE(6,1004)
3376 GO TO 200
3377 21 WRITE(6,1021)
3378 GO TO 200
3379 22 WRITE(6,1022)
3380 GO TO 200
3381 23 WRITE(6,1023)
3382 GO TO 200
3383 24 WRITE(6,1024)
3384 GO TO 200
3385 25 WRITE(6,1025)
3386 GO TO 200
3387 26 WRITE(6,1026)
3388 GO TO 200
3389 27 WRITE(6,1027)
3390 GO TO 200
3391 50 WRITE(6,1050)
```

```

3392 C
3393   200 CONTINUE
3394   WRITE(6,1101)
3395  1100 FORMAT(1H1,35(' *'))//10X,'EXECUTION TERMINATED ON ERROR',
3396 *' CONDITION AT TIME ',F10.4//)
3397  1001 FORMAT(1X,'THE PRESSURE MATRIX INVERSION DOES NOT CONVERGE'/
3398 *      1X,'IN THE MAXIMUM NUMBER OF ITERATIONS ALLOWED'//
3399 *      1X,'ERROR CONDITION NUMBER = 1'//)
3400  1002 FORMAT(1X,'THE PRESSURE MATRIX IS NOT DIAGONAL DOMINANT'//
3401 *      1X,'ERROR CONDITION NUMBER = 2'//)
3402  1003 FORMAT(1X,'THE VOID FRACTION TOOK A VALUE EITHER LOWER THAN'/
3403 *      1X,'ZERO OR GREATER THAN ONE'//
3404 *      1X,'ERROR CONDITION NUMBER = 3'//)
3405  1004 FORMAT(1X,'THE INITIAL CONDITIONS INPUT DATA IS NOT IN THE'/
3406 *      1X,'PROPER ORDER'//
3407 *      1X,'ERROR CONDITION NUMBER = 4'//)
3408  1021 FORMAT(1X,'THE TIME STEP SIZE TOOK A VALUE TOO SMALL'/
3409 *      1X,'ERROR CONDITION NUMBER = 21'//)
3410  1022 FORMAT(1X,'THE PRESSURE TOOK A VALUE TOO SMALL'/
3411 *      1X,'ERROR CONDITION NUMBER = 22'//)
3412  1023 FORMAT(1X,'THE PRESSURE TOOK A VALUE TOO HIGH'/
3413 *      1X,'ERROR CONDITION NUMBER = 23'//)
3414  1024 FORMAT(1X,'THE VAPOR TEMPERATURE TOOK A VALUE TOO SMALL'/
3415 *      1X,'ERROR CONDITION NUMBER = 24'//)
3416  1025 FORMAT(1X,'THE VAPOR TEMPERATURE TOOK A VALUE TOO HIGH'/
3417 *      1X,'ERROR CONDITION NUMBER = 25'//)
3418  1026 FORMAT(1X,'THE LIQUID TEMPERATURE TOOK A VALUE TOO SMALL'/
3419 *      1X,'ERROR CONDITION NUMBER = 26'//)
3420  1027 FORMAT(1X,'THE LIQUID TEMPERATURE TOOK A VALUE TOO HIGH'/
3421 *      1X,'ERROR CONDITION NUMBER = 27'//)
3422  1050 FORMAT(1X,'A QUIT SIGNAL WAS ISSUED BY THE TERMINAL OPERATOR'/
3423 *      1X,'ERROR CONDITION NUMBER = 50'//)
3424  1101 FORMAT(1X,35(' *'))
3425   RETURN
3426   END

```

Subroutine saver

```

3427   SUBROUTINE SAVER(P,TV,TL,ALFA,UVZ,ULZ,UVR,ULR,TR,TCAN,
3428   *           TIME,NTR,NN,NCAN,NI)
3429   IMPLICIT REAL*8 (A-H,O-Z)
3430   LOGICAL LDATA
3431   DIMENSION P(NN),TV(NN),TL(NN),ALFA(NN),UVZ(NN),ULZ(NN),
3432   *           UVR(NN),TCAN(NCAN),TR(NTR),ULR(NN)
3433   DIMENSION XOUT(5)
3434   LDATA = .FALSE.
3435   WRITE(7,103) LDATA,TIME
3436   DO 1 KO = 1,NN
3437   WRITE(7,100) KO,TV(KO),TL(KO),P(KO),ALFA(KO)
3438   WRITE(7,100) KO,UVZ(KO),ULZ(KO),UVR(KO),ULR(KO)
3439   1 CONTINUE
3440   LDATA = .TRUE.
3441   KRES = 0
3442   2 CONTINUE
3443   DO 3 K = 1,5
3444   KM = KRES + K
3445   IF(KM.GT.NTR) GO TO 4
3446   XOUT(K) = TR(KM)
3447   3 CONTINUE
3448   WRITE(7,101) LDATA,(XOUT(KL),KL=1,5)
3449   KRES = KRES + 5
3450   GO TO 2
3451   4 CONTINUE
3452   WRITE(7,101) LDATA,(XOUT(KL),KL=1,5)
3453   LDATA = .FALSE.
3454   WRITE(7,102) LDATA
3455   LDATA = .TRUE.
3456   KRES = 2*NI
3457   K3 = 3*NI
3458   5 CONTINUE
3459   DO 6 K = 1,5
3460   KM = KRES + K
3461   IF(KM.GT.K3) GO TO 7
3462   XOUT(K) = TCAN(KM)
3463   6 CONTINUE
3464   WRITE(7,101) LDATA,(XOUT(KL),KL=1,5)
3465   KRES = KRES + 5
3466   GO TO 5
3467   7 CONTINUE
3468   WRITE(7,101) LDATA,(XOUT(KL),KL=1,5)
3469   LDATA = .FALSE.

```

```
3470     WRITE(7,102) LDATA
3471 100  FORMAT(I5,4D15.9)
3472 101  FORMAT(L1,5D15.9)
3473 102  FORMAT(L1)
3474 103  FORMAT(L1,D15.9)
3475     RETURN
3476     END
```

M16494

