Archive

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

by

Mario Roberto Granziera
Mujid S. Kazimi

Energy Laboratory Report No. MIT-EL 80-011

May 1980

REPORTS IN REACTOR THERMAL HYDARULICS RELATED TO THE MIT ENERGY LABORATORY ELECTRIC POWER PROGRAM

A. Topical Reports (For availability check Energy Laboratory Headquarters, Room E19-439, MIT, Cambridge, Massachusetts, 02139)

A.1 General Applications
A.2 PWR Applications
A.3 BWR Applications
A.4 LMFBR Applications

A.1 M. Massoud, "A Condensed Review of Nuclear Reactor Thermal-Hydraulic Computer Codes for Two-Phase Flow Analysis," MIT Energy Laboratory Report MIT-EL-79-018, February 1979.

J.E. Kelly and M.S. Kazimi, "Development and Testing of the Three Dimensional, Two-Fluid Code THERMIT for LWR Core and Subchannel Applications," MIT Energy Laboratory Report MIT-EL-79-046, December 1979.

A.2 P. Moreno, C. Chiu, R. Bowring, E. Khan, J. Liu, N. Todreas, "Methods for Steady-State Thermal/Hydraulic Analysis of PWR Cores," MIT Energy Laboratory Report MIT-EL-76-006, Rev. 1, July 1977 (Orig. 3/77).

J.E. Kelly, J. Loomis, L. Wolf, "LWR Core Thermal-Hydraulic Analysis--Assessment and Comparison of the Range of Applicability of the Codes COBRA-IIIC/MIT and COBRA IV-1," MIT Energy Laboratory Report MIT-EL-78-026, September 1978.

J. Liu, N. Todreas, "Transient Thermal Analysis of PWR's by a Single Pass Procedure Using a Simplified Model Layout," MIT Energy Laboratory Report MIT-EL-77-008, Final, February 1979, (Draft, June 1977).

J. Liu, N. Todreas, "The Comparison of Available Data on PWR Assembly Thermal Behavior with Analytic Predictions," MIT Energy Laboratory Report MIT-EL-77-009, Final, February 1979, (Draft, June 1977).

A.3 L. Guillebaud, A. Levin, W. Boyd, A. Faya, L. Wolf, "WOSUB-A Subchannel Code for Steady-State and Transient Thermal-Hydraulic Analysis of Boiling Water Reactor Fuel Bundles," Vol. II, Users Manual, MIT-EL-78-024. July 1977.

L. Wolf, A Faya, A. Levin, W. Boyd, L. Guillebaud, "WOSUB-A Subchannel Code for Steady-State and Transient Thermal-Hydraulic Analysis of Boiling Water Reactor Fuel Pin Bundles," Vol. III, Assessment and Comparison, MIT-EL-78-025, October 1977.

L. Wolf, A. Faya, A. Levin, L. Guillebaud, "WOSUB-A Subchannel Code for Steady-State Reactor Fuel Pin Bundles," Vol. I, Model Description, MIT-EL-78-023, September 1978.

A. Faya, L. Wolf and N. Todreas, "Development of a Method for BWR Subchannel Analysis," MIT-EL-79-027, November 1979.

A. Faya, L. Wolf and N. Todreas, "CANAL User's Manual," MIT-EL-79-028, November 1979.

A.4 W.D. Hinkle, "Water Tests for Determining Post-Voiding Behavior in the LMFBR," MIT Energy Laboratory Report MIT-EL-76-005, June 1976.

W.D. Hinkle, Ed., "LMFBR Safety and Sodium Boiling - A State of the Art Reprot," Draft DOE Report, June 1978.

M.R. Granziera, P. Griffith, W.D. Hinkle, M.S. Kazimi, A. Levin, M. Manahan, A. Schor, N. Todreas, G. Wilson, "Development of Computer Code for Multi-dimensional Analysis of Sodium Voiding in the LMFBR," Preliminary Draft Report, July 1979.

M. Granziera, P. Griffith, W. Hinkle (ed.), M. Kazimi, A. Levin, M. Manahan, A. Schor, N. Todreas, R. Vilim, G. Wilson, "Development of Computer Code Models for Analysis of Subassembly Voiding in the LMFBR," Interim Report of the MIT Sodium Boiling Project Covering Work Through September 30, 1979, MIT-EL-80-005.

A. Levin and P. Griffith, "Development of a Model to Predict Flow Oscillations in Low-Flow Sodium Boiling," MIT-EL-80-006, April 1980.

M.R. Granziera and M. Kazimi, "A Two Dimensional, Two Fluid Model for Sodium Boiling in LMFBR Assemblies," MIT-EL-80-011, May 1980.

G. Wilson and M. Kazimi, "Development of Models for the Sodium Version of the Two-Phase Three Dimensional Thermal Hydraulics Code THERMIT," MIT-EL-80-010, May 1980.

B. Papers

B.1 General Applications
B.2 PWR Applications
B.3 BWR Applications
B.4 LMFBR Applications

B.1 J.E. Kelly and M.S. Kazimi, "Development of the Two-Fluid Multi-Dimensional Code THERMIT for LWR Analysis," accepted for presentation 19th National Heat Transfer Conference, Orlando, Florida, August 1980.

J.E. Kelly and M.S. Kazimi, "THERMIT, A Three-Dimensional, Two-Fluid Code for LWR Transient Analysis," accepted for presentation at Summer Annual American Nuclear Society Meeting, Las Vegas, Nevada, June 1980.

B.2 P. Moreno, J. Kiu, E. Khan, N. Todreas, "Steady State Thermal Analysis of PWR's by a Single Pass Procedure Using a Simplified Method," American Nuclear Society Transactions, Vol. 26

P. Moreno, J. Liu, E. Khan, N. Todreas, "Steady-State Thermal Analysis of PWR's by a Single Pass Procedure Using a Simplified Nodal Layout," Nuclear Engineering and Design, Vol. 47, 1978, pp. 35-48.

C. Chiu, P. Moreno, R. Bowring, N. Todreas, "Enthalpy Transfer Between PWR Fuel Assemblies in Analysis by the Lumped Subchannel Model," Nuclear Engineering and Design, Vol. 53, 1979, 165-186.

B.3 L. Wolf and A. Faya, "A BWR Subchannel Code with Drift Flux and Vapor Diffusion Transport," American Nuclear Society Transactions, Vol. 28, 1978, p. 553.

B.4 W.D. Hinkle, (MIT), P.M Tschamper (GE), M.H. Fontana, (ORNL), R.E. Henry (ANL), and A. Padilla, (HEDL), for U.S. Department of Energy, "LMFBR Safety & Sodium Boiling," paper presented at the ENS/ANS International Topical Meeting on Nuclear Reactor Safety, October 16-19, 1978, Brussels, Belgium.

M.I. Autruffe, G.J. Wilson, B. Stewart and M. Kazimi, "A Proposed Momentum Exchange Coefficient for Two-Phase Modeling of Sodium Boiling," Proc. Int. Meeting Fast Reactor Safety Technology, Vol. 4, 2512-2521, Seattle, Washington, August 1979.

M.R. Granziera and M.S. Kazimi, "NATOF-2D: A Two Dimensional Two-Fluid Model for Sodium Flow Transient Analysis," Trans. ANS, 33, 515, November 1979.

NOTICE

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

by

Mario Roberto Granziera
Mujid S. Kazimi

Energy Laboratory and
Department of Nuclear Engineering

Massachusetts Institute of Technology
Cambridge, Massachusetts 02139

Topical Report of the
MIT Sodium Boiling Project

sponsored by:

U. S. Department of Energy,
General Electric Co. and
Hanford Engineering Development Laboratory

Energy Laboratory Report No. MIT-EL 80-011

May 1980

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 problems 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 problems caused by the strong disparity between the axial and radial dimensions of fuel assemblies. A solution to this problem was found which uses 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.

ACKNOWLEDGEMENT

Funding for this project was provided by the United States Department of Energy, the General Electric Co., and the Hanford Engineering Development Laboratory. Additional support was also provided to Mario R. Granziera by the Commisao Nacional de Energia Nuclear. This support was deeply appreciated.

The authors would also like to thank their co-workers on the MIT Sodium Boiling project for their help and contributions to this work.

A very special thanks is due to Bruce Stewart whose intimate knowledge of the concepts of numerical analysis was an invaluable resource.

The work described in this report was performed primarily by the principal author, Mario R. Granziera, who has submitted the same report in partial fulfillment for the PhD degree in Nuclear Engineering at MIT.

TABLE OF CONTENTS

Title Page 1

Abstract 2

Acknowledgement 3

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 W 1 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

No.		
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 Coefficienct 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

α	Void Fraction
ρ	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
$\Delta\gamma$	Radial Mesh Spacing
Δz	Axial Mesh Spacing
Δt	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
$\phi_A dA$	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

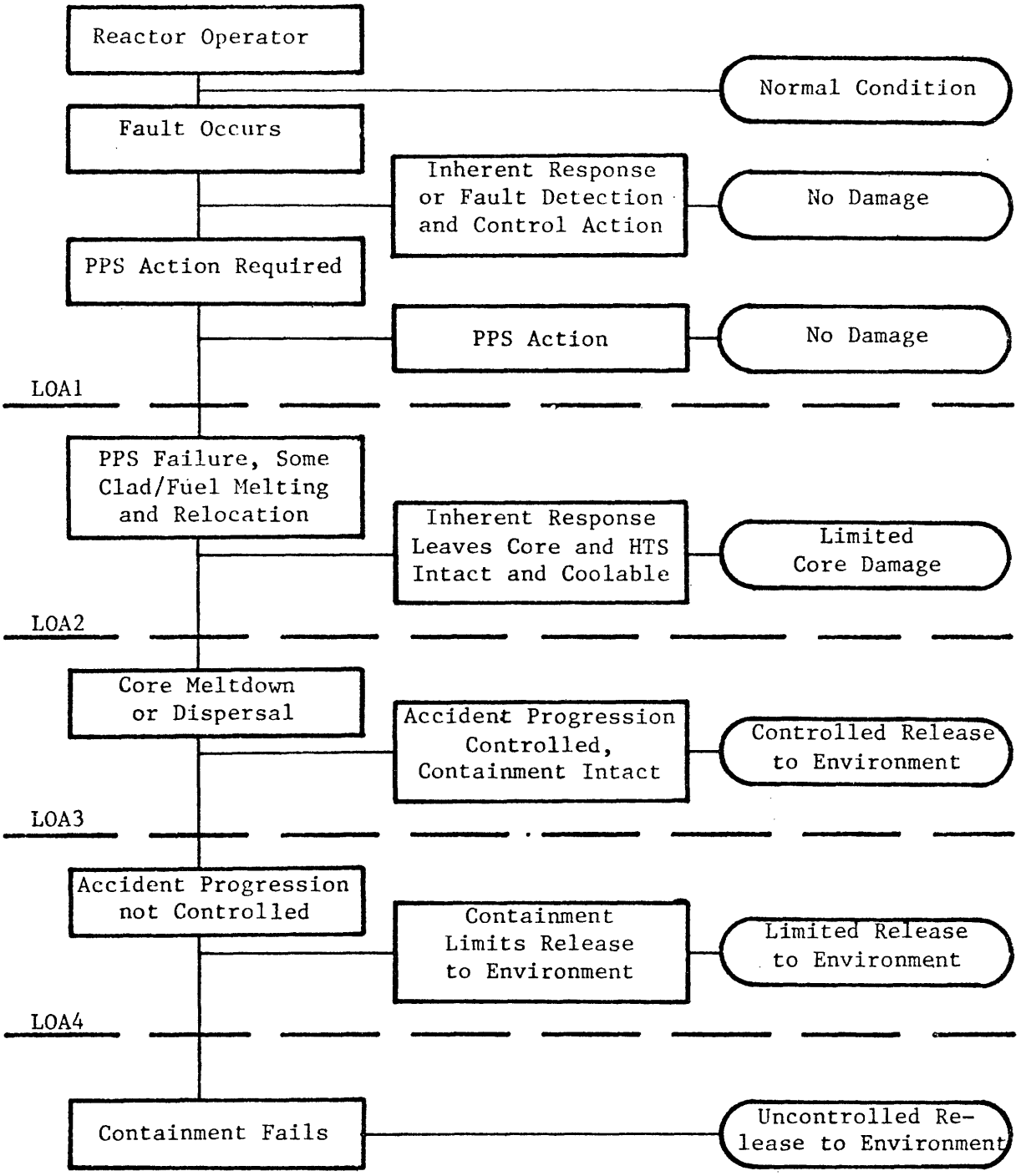

Figure 1.1

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

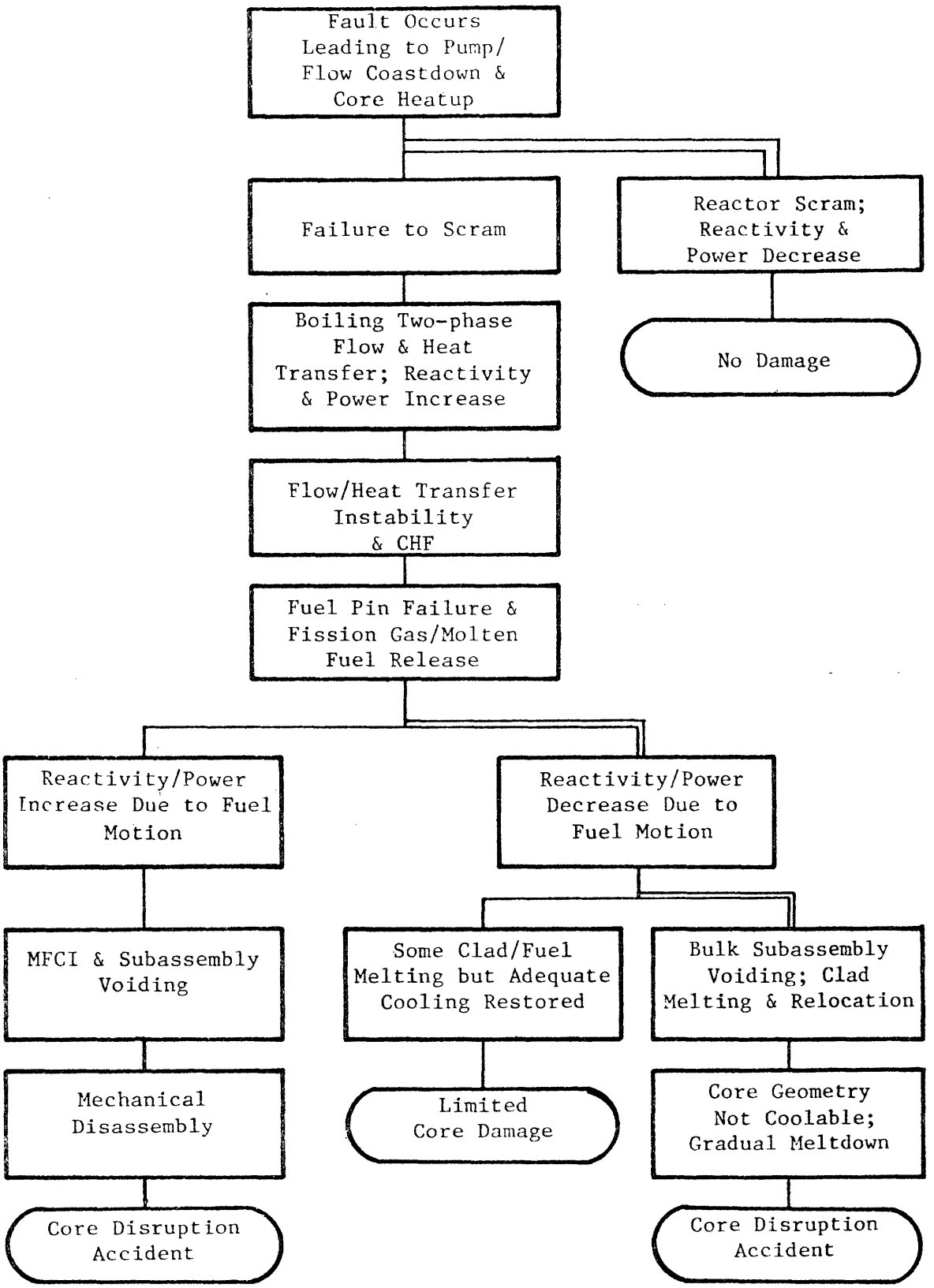

Figure 1.2

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

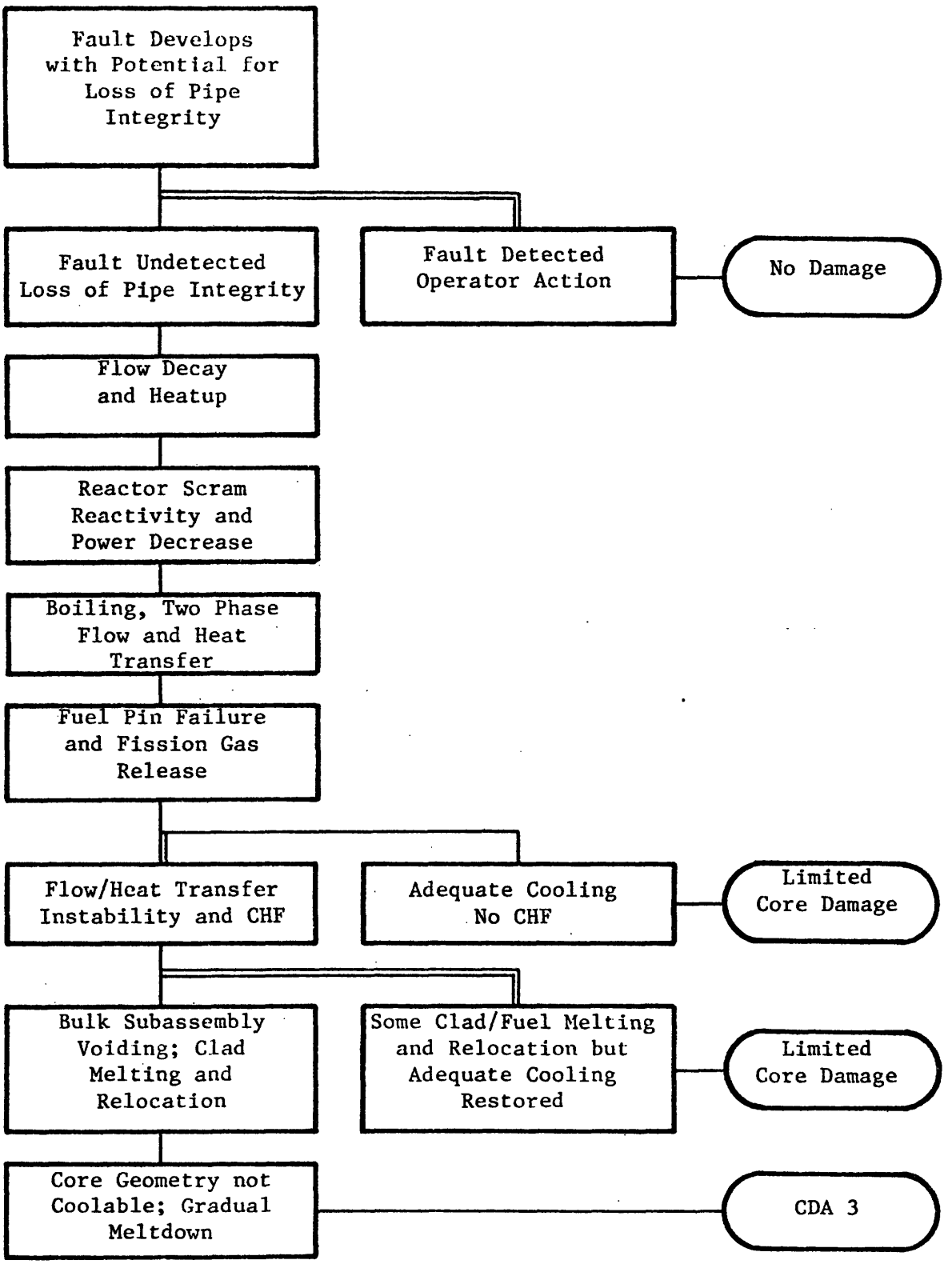

Figure 1.3

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

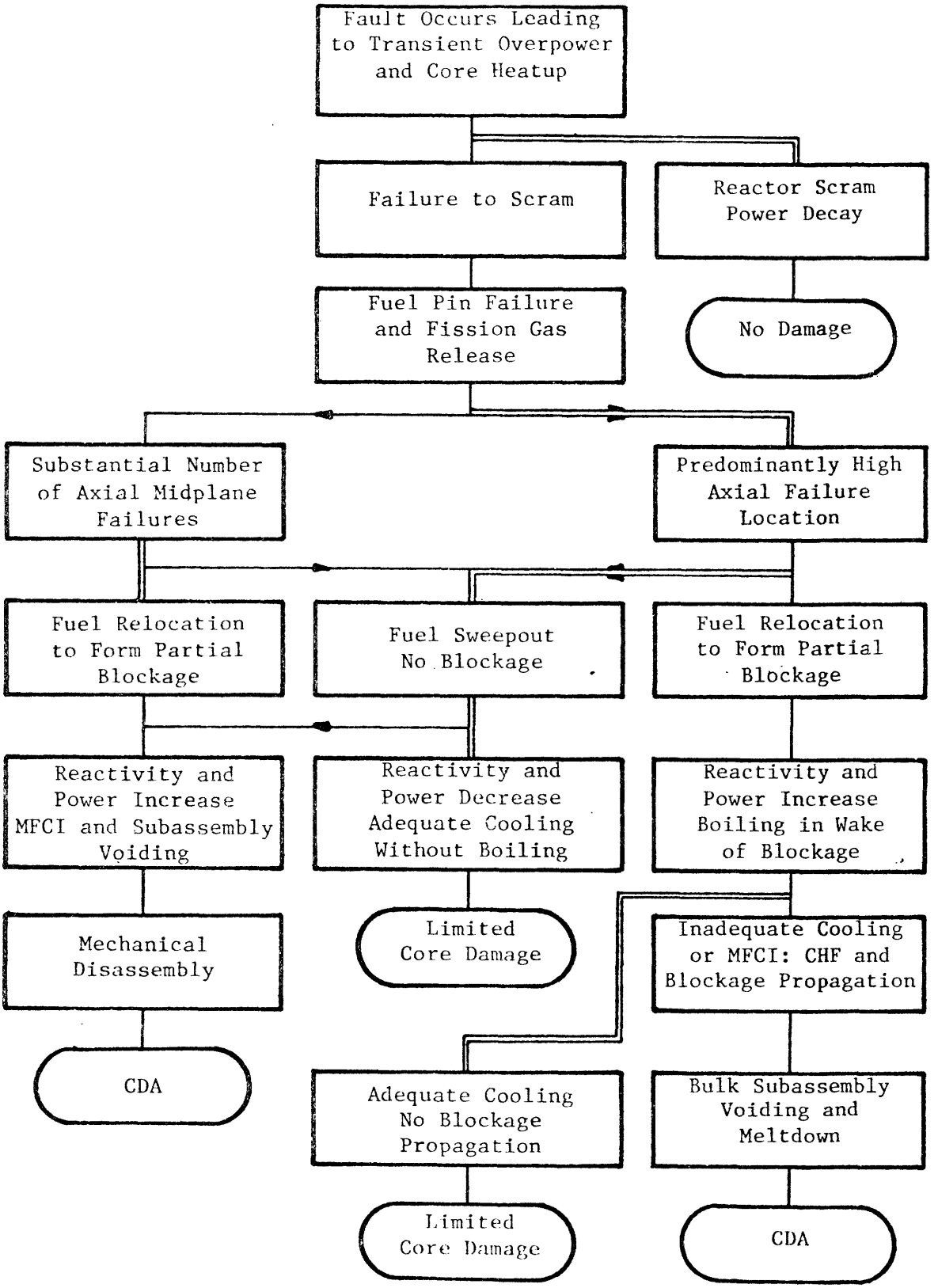

Figure 1.4

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

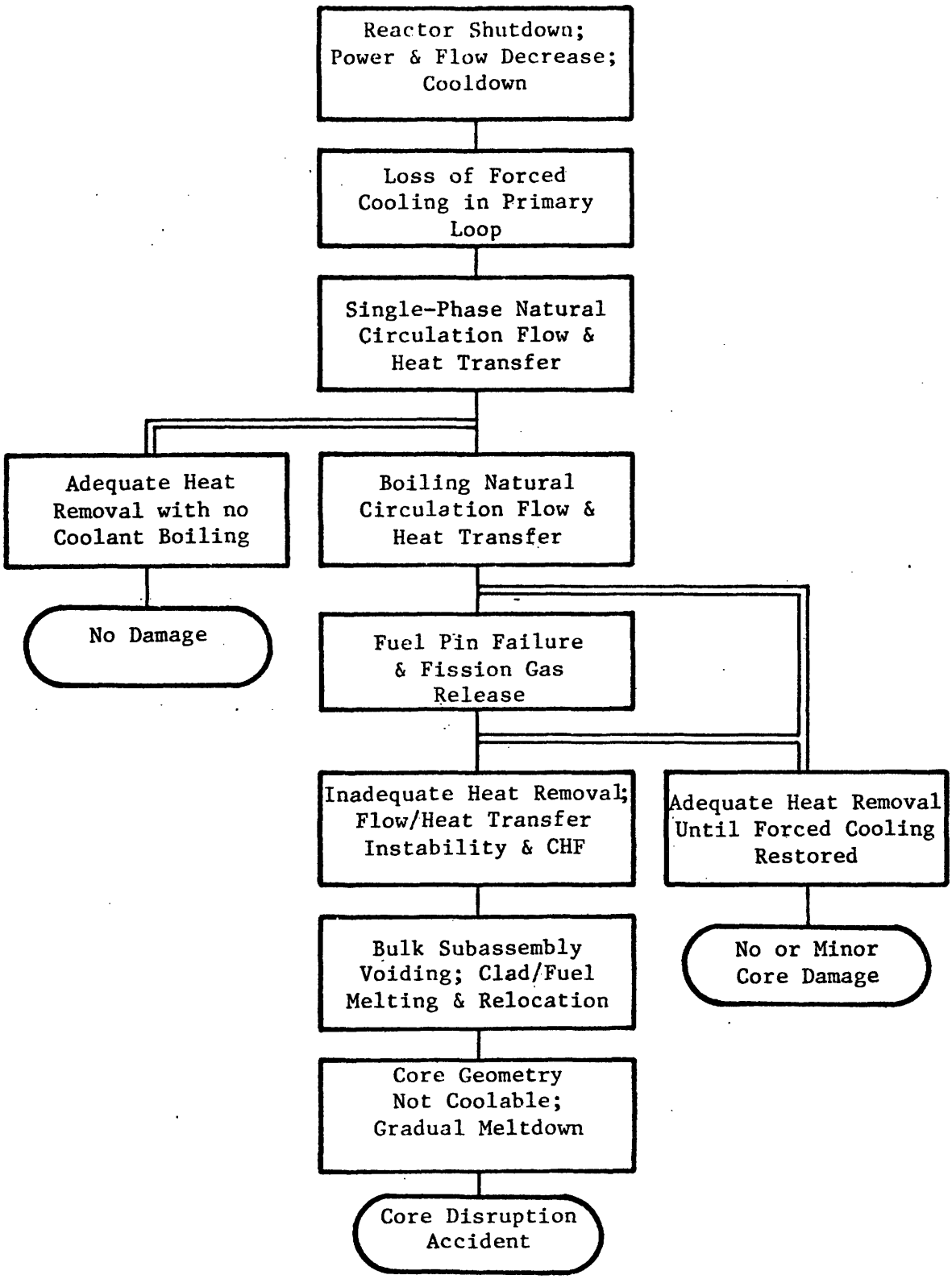

Figure 1.5

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

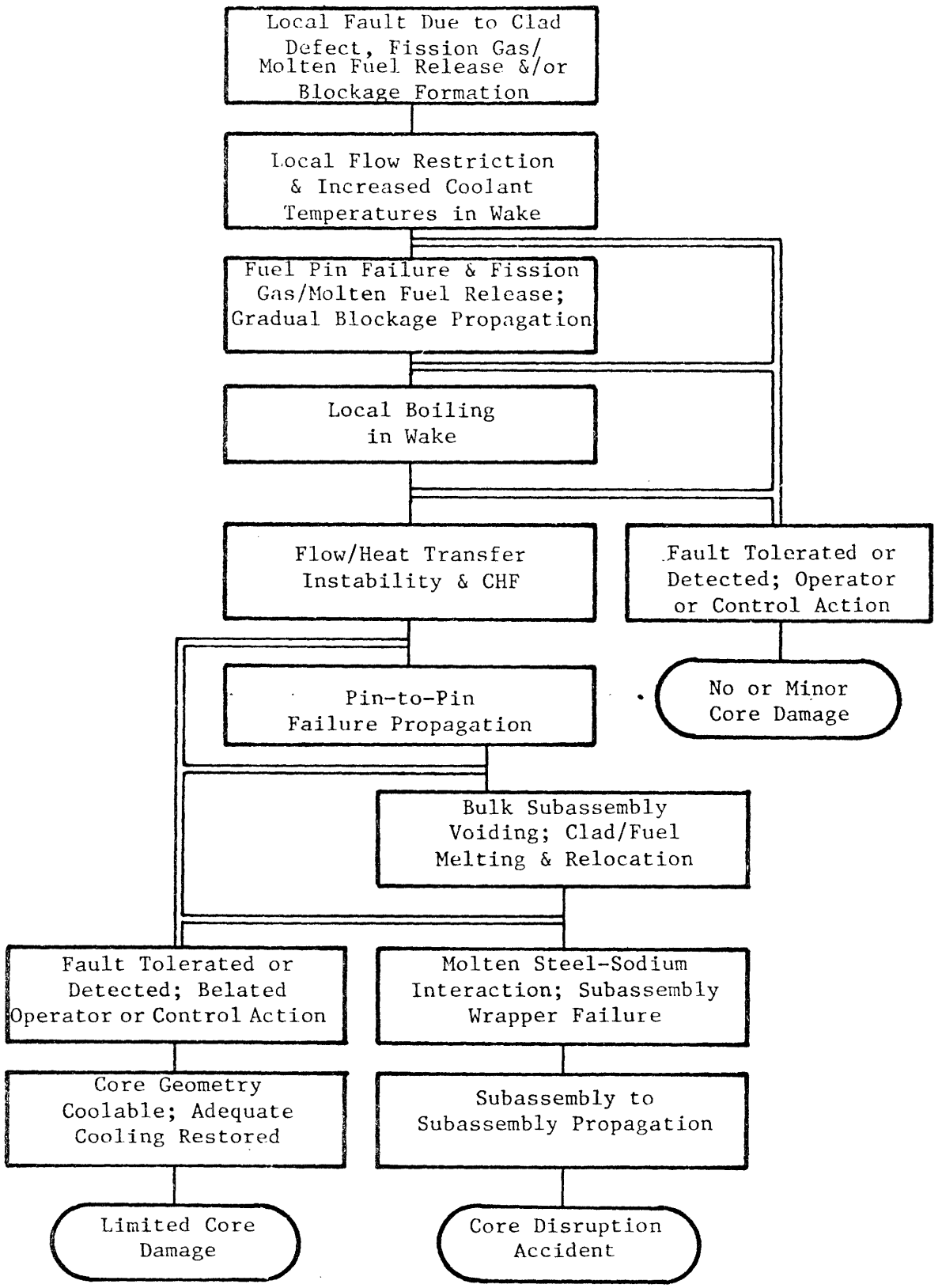

Figure 1.6

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 characterstics of numerical models relevant to LMFBR fuel assembly fluido-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.

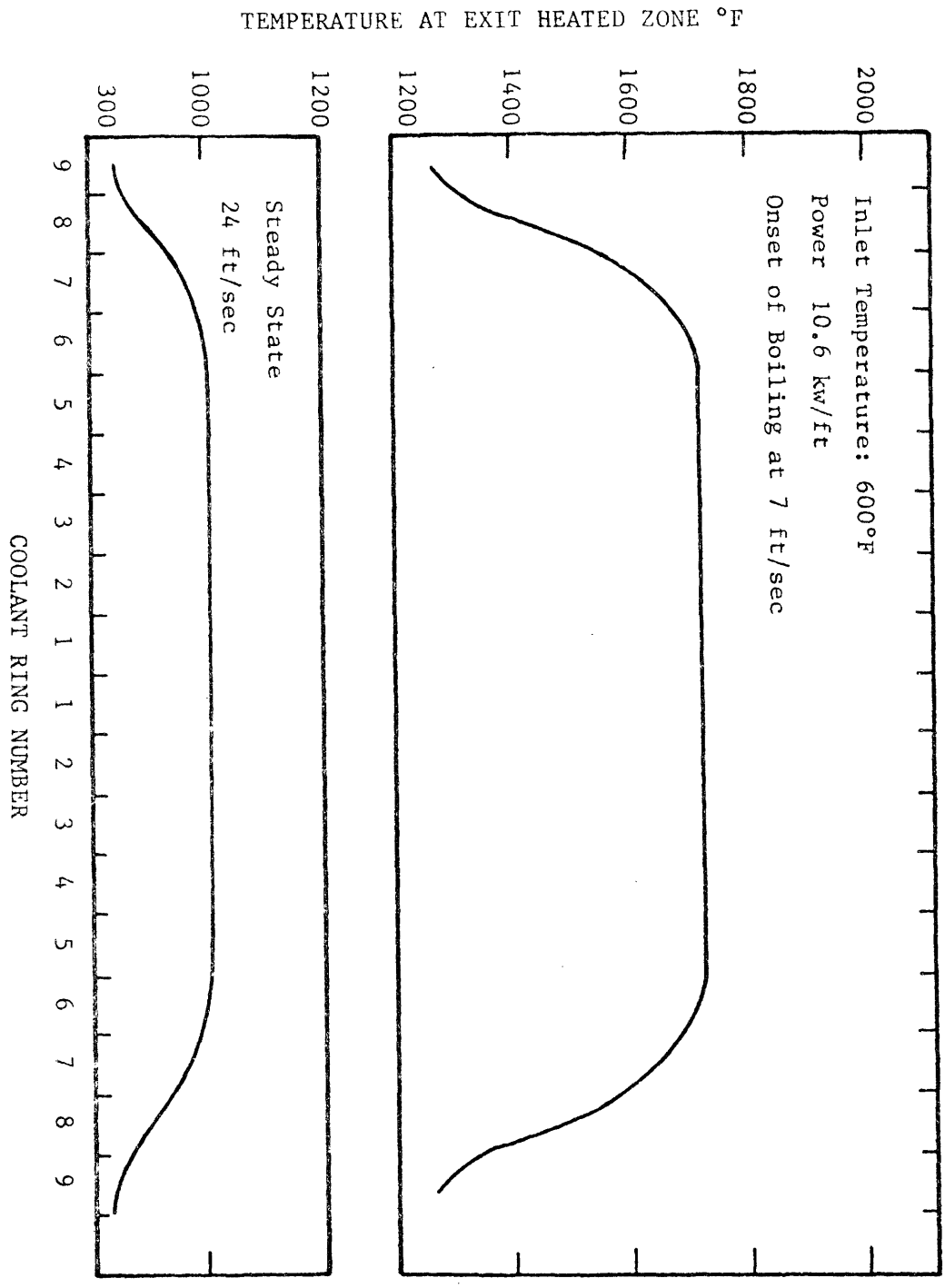

Figure 1.7

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-unform 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 fluido-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 fluido-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 be in principle 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 consevation 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 anlysis 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:

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

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:

$$\int_V \alpha \rho_v \, dV = < \rho_v > \int_V \alpha \, dV = < \rho_v > < \alpha > V \tag{2.2}$$

$$\int_A \alpha \rho_v U_v \, dA = < \alpha \rho_v U_v > A \tag{2.2}$$

$$\int_V (S_e - S_c) \, dV = (< S_e > - < S_c >) V \tag{2.3}$$

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

$$\frac{\partial}{\partial t} \left[< \alpha > < \rho_v > \right] + \frac{A_z}{V} \left[< \alpha \rho_v U_{vz} >_{A_{z+}} - < \alpha \rho_v U_{vz} >_{A_{z-}} \right.$$

$$\left. + \frac{A_{r+}}{V} < \alpha \rho_v U_{vr} >_{A_{r+}} - \frac{A_{r-}}{V} < \alpha \rho_v U_{vr} >_{A_{r-}} \right] = < S_e > - < S_c > \tag{2.5}$$

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 arbitrarely. 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:

$$\frac{\partial}{\partial t}\left[(1 - \langle \alpha \rangle)\,\rho_\ell\right] + \frac{A_z}{V}\left[\langle (1-\alpha)\,\rho_\ell\, U_{\ell z}\rangle_{A_{z+}}\right.$$

$$- \langle (1-\alpha)\,\rho_\ell\, U_{\ell z}\rangle_{A_{z-}} + \frac{A_{r+}}{V}\langle (1-\alpha)\,\rho_\ell\, U_{\ell r}\rangle_{A_{r+}}$$

$$\left. - \frac{A_{r-}}{V}\langle (1-\alpha)\,\rho_\ell\, U_{\ell r}\rangle_{A_{r-}}\right] = \langle S_c\rangle - \langle S_e\rangle \qquad (2.6)$$

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

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

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

Radial Direction

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

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

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

from equations 2.7 and 2.8 respectively:

$$\frac{\partial}{\partial t}\int_v \alpha\,\rho_v\,U_{vz}\,dV - \langle U_{vz}\rangle \frac{\partial}{\partial t}\int_v \alpha\,\rho_v\,dV + \int_{A_{z+}} - \int_{A_{z-}} \alpha\,\rho_v\,U_{vz}^2\,dA$$

$$- \langle U_{vz}\rangle \int_{A_{z+}} - \int_{A_{z-}} \alpha\,\rho_v\,U_{vz}\,dA + \int_{A_{r+}} - \int_{A_{r-}} \alpha\,\rho_v\,U_{vz}\,U_{vr}\,dA$$

$$- \langle U_{vz}\rangle \int_{A_{r+}} - \int_{A_{r-}} \alpha\,\rho_v\,U_{vr}\,dA - \oint_{A_v} P\,\hat{k}\cdot\hat{n}\,dA =$$

$$= -\int_{A_w} f_{vz}\,dA - g\int_v \alpha\,\rho_v\,dV + \int_v M_{vz}\,dV - \langle U_{vz}\rangle \int_v (S_e - S_c)\,dV \qquad (2.9)$$

$$\frac{\partial}{\partial t}\int_v \alpha\,\rho_v\,U_{vr}\,dV - \langle U_{vr}\rangle \frac{\partial}{\partial t}\int_v \alpha\,\rho_v\,dV + \int_{A_{z+}} - \int_{A_{z-}} \alpha\,\rho_v\,U_{vz}\,U_{vr}\,dA$$

$$- \langle U_{vr}\rangle \int_{A_{z+}} - \int_{A_{z-}} \alpha\,\rho_v\,U_{vz}\,dA + \int_{A_{r+}} - \int_{A_{r-}} \alpha\,\rho_v\,U_{vr}^2\,dA$$

$$- \langle U_{vr}\rangle \int_{A_{r+}} - \int_{A_{r-}} \alpha\,\rho_v\,U_{vr}\,dA - \int_{A_v} P\,\hat{r}\cdot\hat{n}\,dA =$$

$$= -\int_{A_w} f_{vr}\,dA + \int_v M_{vr}\,dV - \langle U_{vr}\rangle \int_v (s_e - S_c)\,dV \qquad (2.10)$$

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

$$\frac{\partial}{\partial t} \int_V \alpha \rho_v U_{vz} \, dV - \langle U_{vz} \rangle \frac{\partial}{\partial t} \int_V \alpha \rho_v \, dV =$$

$$= \int_V \left[\alpha \rho_v \frac{\partial U_{vz}}{\partial t} + (U_{vz} - \langle U_{vz} \rangle) \frac{\partial \alpha \rho_v}{\partial t} \right] dV$$

$$= \langle \alpha \rangle \langle \rho_v \rangle \frac{\partial}{\partial t} \langle U_{vz} \rangle V \qquad (2.11)$$

and

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

Next consider the convective terms. We define:

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

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

$$< \alpha \rho_v U_{vz} >_{A_z} = \frac{1}{A_z} \int_{A_z} \alpha \rho_v U_{vz} \, dA \tag{2.15}$$

$$< \alpha \rho_v U_{vr} >_{A_z} = \frac{1}{A_z} \int_{A_z} \alpha \rho_v U_{vr} \, dA \tag{2.16}$$

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:

$$< U_{vz} > = \frac{1}{2} \left(< U_{vz} >_{A_{z+}} + < U_{vz} >_{A_{z-}} \right) \tag{2.17}$$

$$< U_{vr} > = \frac{1}{2} \left(< U_{vr} >_{A_{z+}} + < U_{vr} >_{A_{z-}} \right) \tag{2.18}$$

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

The axial convective terms in both momentum equations become:

$$\int_{A_{z+}} - \int_{A_{z-}} \alpha \rho_v U_{vz}^2 \, dA - < U_{vz} > \int_{A_{z+}} - \int_{A_{z-}} \alpha \rho_v U_{vz} \, dA$$

$$= < \alpha > < \rho_v > < U_{vz} > \left[< U_{vz} >_{A_{z+}} - < U_{vz} >_{A_{z-}} \right] \tag{2.19}$$

$$\int_{A_{z+}} - \int_{A_{z-}} \alpha\, \rho_v\, U_{vz}\, U_{vr}\, dA \; - \; \langle U_{vr} \rangle \int_{A_{z+}} - \int_{A_{z-}} \alpha \;\; \rho_v\, U_{vz}\, dA =$$

$$= \langle \alpha \rangle \langle \rho_v \rangle \langle U_{vz} \rangle \left[\langle U_{vr} \rangle_{A_{z+}} - \langle U_{vr} \rangle_{A_{z-}} \right] A_z \qquad (2.20)$$

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

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

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

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:

$$< U_{vr} > = \frac{1}{2} \left(< U_{vr} >_{A_{r+}} + < U_{vr} >_{A_{r-}} \right) \tag{2.23}$$

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

$$< U_{vr} >_{A_r(r)} \cdot A_r(r) = U_r^*(r) A_r^* \cdot r \tag{2.24}$$

where $A_r^* \cdot r$ is the average linarly 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:

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

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

The volume V^k_{cell} is:

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

where A_{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:

$$A^*_r = \Delta z \quad \frac{\sqrt{3}}{2} \quad - \quad \frac{A_{pin}}{p^2} \tag{2.26}$$

We now can make the more acceptable assumption that $< U_{vr} >_{A_r} A_r(r)$ $= U^*_r(r)\ A^*_r\ .\ r$ = constant. It follows:

$$< U_{vr} > = \frac{1}{<\alpha> V} \int_V \alpha\ U_{vr}\ dV = \frac{1}{V} \int dr \int_{A_r} U_{vr}\ dA$$

$$= \frac{1}{V} \int_{r_-}^{r_+} < U_{vr} >_{A_r} A_r(r)\ dr$$

$$= U^*_r(r^*)\ .\ A^*_r\ .\ r^*\ .\ \frac{(r_+ - r_-)}{V} \tag{2.27}$$

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

$$A_r^* \, r^* \, (r_+ - r_-) = V \tag{2.28}$$

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

$$r^* = \frac{r_+ + r_-}{2} \tag{2.29}$$

Substituting for r* in equation 2.27 we have

$$< U_{vr} > = U_r^*(r^*) \tag{2.30}$$

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

$$< U_{vr} > = \frac{U^*(r_+) + U^*(r_-)}{2} \tag{2.31}$$

Going back to equation 2.24 we have:

$$< U_{vr} > = \frac{1}{2} \left[< U_{vr} >_{A_{r+}} \frac{A_{r+}}{A^* \, r_+} + < U_{vr} >_{A_{r-}} \frac{A_{r-}}{A^* \, r_-} \right] \tag{2.32}$$

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:

$$A_{r+} = A^* \, r+$$

$$\text{and} \qquad A_{r-} = A^* \, r- \tag{2.33}$$

then the desired equation 2.23 is obtained

$$< U_{vr} > = \frac{< U_{vr} >_{A_{r+}} + < U_{vr} >_{A_{r-}}}{2}$$

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

$$\int_{A_{r+}} - \int_{A_{r-}} \alpha \, \rho_v \, U_{vz} \, U_{vr} \, dA - < U_{vz} > \int_{A_{r+}} - \int_{A_{r-}} \alpha \, \rho_v \, U_{vr} \, dA =$$

$$= < \alpha > < \rho_v > < U_{vr} > \left(< U_{vz} >_{A_{r+}} - < U_{vz} >_{A_{r-}} \right) \frac{(A_{r+} + A_{r-})}{2} \tag{2.34}$$

$$\int_{A_{r+}} - \int_{A_{r-}} \alpha \, \rho_v \, U_{vr}^2 - < U_{vr} > \int_{A_{r+}} - \int_{A_{r-}} \alpha \, \rho_v \, U_{vr} \, dA =$$

$$< \alpha > < \rho_v > < U_{vr} > \left(< U_{vr} >_{A_{r+}} - < U_{vr} >_{A_{r-}} \right) \left(\frac{A_{r+} + A_{r-}}{2} \right) \tag{2.35}$$

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

$$\oint_{A_v} P \cdot \hat{k} \cdot \hat{n} \, dA = A_z \; < \alpha > \left(< P >_{A_{z+}} - < P >_{A_{z-}} \right) \tag{2.36}$$

$$\oint_{A_v} P \cdot \hat{r} \cdot \hat{n} \, dA = \left(\frac{A_{r+} + A_{r-}}{2} \right) < \alpha > \left(< P >_{A_{r+}} - < P >_{A_{r-}} \right) \tag{2.37}$$

$$\int_{A_w} f_{vz} \, dA = < f_{vz} >_{A_w} A_w$$

$$\int_{A_w} f_{vr} \, d \; = < f_{vr} >_{A_w} A_w$$

$$\int_v M_{vz} \, dV - < U_{vz} > \int_v (s_e - S_c) \, dV = < M'_{vz} > V$$

$$\int_v M_{vr} \, dV - < U_{vr} > \int_v (S_e - S_c) \, dV = <M'_{vr}> V$$

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 rewriten as:

$$< \alpha > < \rho_v > \left[\frac{\partial}{\partial t} < U_{vz} > + \frac{A_z}{V} < U_{vz} > \left(< U_{vz} >_{A_{z+}} - < U_{vz} >_{A_{z-}} \right) + \right.$$

$$\left. + \left(\frac{A_{r+} + A_{r-}}{2V} \right) < U_{vr} > \left(< U_{vz} >_{A_{r+}} - < U_{vz} >_{A_{r-}} \right) \right] -$$

$$- \frac{A_z}{V} < \alpha > \left(< P >_{A_{z+}} - < P >_{A_{z-}} \right)$$

$$= - \frac{A_w}{V} < f_{vz} > - < \alpha > < \rho_v > g - < M'_{vz} > \qquad (2.38)$$

$$< \alpha > < \rho_v > \left[\frac{\partial}{\partial t} < U_{vr} > + \frac{A_z}{V} < U_{vz} > \left(< U_{vr} >_{A_{z+}} - < U_{vr} >_{A_{z-}} \right) + \right.$$

$$\left. + \left(\frac{A_{r+} + A_{r-}}{2V} \right) < U_{vr} > \left(< U_{vr} >_{A_{r+}} - < U_{vr} >_{A_{r-}} \right) \right] -$$

$$- \left(\frac{A_{r+} + A_{r-}}{2V} \right) < \alpha > \left(< P >_{A_{r+}} - < P >_{A_{r-}} \right) = \frac{A_w}{V} < f_{vr} > - < M'_{vr} > \qquad (2.39)$$

Similarly for the liquid phase:

$$(1 - \langle \alpha \rangle) \langle \rho_\ell \rangle \left[\frac{\partial}{\partial t} \langle U_{\ell z} \rangle + \frac{A_z}{V} \langle U_z \rangle \left(\langle U_{\ell z} \rangle_{A_{z+}} - \langle U_{\ell z} \rangle_{A_{z-}} \right) + \right.$$

$$\left. + \left(\frac{A_{r+} + A_{r-}}{2V} \right) \langle U_{\ell r} \rangle \left(\langle U_{\ell z} \rangle_{A_{r+}} - \langle U_{\ell z} \rangle_{A_{r-}} \right) \right] -$$

$$- \frac{A_z}{2} \left(1 - \langle \alpha \rangle \right) \left(\langle P \rangle_{A_{z+}} - \langle P \rangle_{A_{z-}} \right) =$$

$$= \frac{A_w}{2} \langle f_{\ell z} \rangle - \left(1 - \langle \alpha \rangle \right) \langle \rho_\ell \rangle g - \langle M'_{\ell z} \rangle \qquad (2.40)$$

$$\left(1 - \langle \alpha \rangle \right) \langle \rho_\ell \rangle \left[\frac{\partial}{\partial t} \langle U_{\ell r} \rangle + \frac{A_z}{V} \langle U_{\ell z} \rangle \left(\langle U_{\ell r} \rangle_{A_{r+}} - \right. \right.$$

$$\left. \left. - \langle U_{\ell r} \rangle_{A_{r-}} \right) + \left(\frac{A_{r+} + A_{r-}}{2V} \right) \langle U_{\ell r} \rangle \left(\langle U_{\ell r} \rangle_{A_{r+}} - \langle U_{\ell r} \rangle_{A_{r-}} \right) - \right.$$

$$\left. - \left(\frac{A_{r+} + A_{r-}}{2V} \right) \left(1 - \langle \alpha \rangle \right) \left(\langle P \rangle_{A_{r+}} - \langle P \rangle_{A_{r-}} \right) \right] =$$

$$= \frac{A_w}{V} \langle f_{\ell r} \rangle - \langle M'_{\ell r} \rangle \qquad (2.41)$$

2.1.3 The Energy Equations

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

$$\frac{\partial}{\partial t}\int_V \alpha\,\rho_v\,(e_v + \tfrac{1}{2}\,U_v^2)\,dV + \int_{A_{z+}} - \int_{A_{z-}} \alpha\,\rho_v\,U_{vz}\,(e_v + \tfrac{1}{2}\,U_v^2)\,dA +$$

$$+\int_{A_{r+}} - \int_{A_{r-}} \alpha\,\rho_v\,U_{vr}\,(e_v + \tfrac{1}{2}\,U_v^2)\,dA =$$

$$= \int_V Q_v\,dV - \int_V \alpha\,\rho_v\,g\,U_{vz}\,dV - \int_{A_w} \vec{U}_v \cdot \vec{f}_v\,dA + \int_{A_v} P \cdot \hat{n}.\vec{U}_v\,dA -$$

$$- \int_V P\,\frac{\partial \alpha}{\partial t}\,dV \qquad (2.42)$$

Before proceding 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 $< U_{vz} >$ and 2.10 multiplied by $< U_{vr} >$, and rearranging the result it follows

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

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

$$+ \int_{A_{z+}} - \int_{A_{z-}} \alpha\rho_v \left[U_{vz} \tfrac{1}{2} U_v^2 - \langle U_{vz}\rangle U_{vz}^2 - \langle U_{vr}\rangle U_{vz} U_{vr}\right] dA +$$

$$+ \int_{A_{r+}} - \int_{A_{r-}} \alpha\rho_v \left[U_{vr} \tfrac{1}{2} U_v^2 - \langle U_{vz}\rangle U_{vz} U_{vr} - \langle U_{vr}\rangle U_{vr}^2\right] dA =$$

$$= \int_V Q_v\, dV - \int_V P \frac{\partial}{\partial t} dV + \oint_{A_v} \left[P\,\hat{n}\cdot\vec{u}_v - P\,\hat{n}\cdot\hat{k}\langle U_{vz}\rangle -\right.$$

$$\left. - P\,\hat{n}\cdot\hat{r}\langle U_{vr}\rangle\right] dA - \int_{A_w} \left[\vec{U}_v\cdot\vec{f}_v - \langle U_{vz}\rangle f_{vz} - \langle U_{vr}\rangle f_{vr}\right] dA -$$

$$\int_V \left[\alpha\rho_v g U_{vz} - \langle U_{vz}\rangle \alpha\rho_v g\right] dV - \int_V \left[\langle U_{vz}\rangle M_{vz} + \langle U_{vr}\rangle M_{vr}\right] dV$$

(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:

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

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

$$\left. + \frac{\partial \alpha \rho_v}{\partial t} \left[\tfrac{1}{2} U_v^2 - \langle U_{vz} \rangle U_{vz} - \langle U_{vr} \rangle U_{vr} \right] \right\} dV \qquad (2.44)$$

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:

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

then

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

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

would lead to:

$$\langle U_{vz}^2 \rangle \simeq \langle U_{vz} \rangle^2 \qquad (2.45)$$

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

$$1E = V < \alpha > < \rho_v > \left[\frac{\partial}{\partial t} \frac{1}{2} \left(< U_{vz}^2 > + < U_{vr}^2 > \right) - \right.$$

$$\left. - < U_{vz} > \frac{\partial}{\partial t} < U_{vz} > - < U_{vr} > \frac{\partial}{\partial t} < U_{vr} > \right] +$$

$$+ \left[1/2 \left(< U_{vz}^2 > + < U_{vr} >^2 \right) - < U_{vz} >^2 - < U_{vr} >^2 \right] \frac{\partial}{\partial t} \int_V \alpha \rho_v \, dV$$

and in view of equation 2.45 this becomes:

$$1E = - 1/2 \left(< U_{vz}^2 > + < U_{vr}^2 > \right) \frac{\partial}{\partial t} \int_V \alpha \rho_v \, dV \qquad (2.46)$$

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

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

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

Define:

$$< U_v^2 >_{A_z} = < U_{vz}^2 >_{A_z} + < U_{vr}^2 >_{A_z} = \int_{A_z} \alpha \rho_v U_{vz} U_v^2 \, dA \Big/ \int_{A_z} \alpha \rho_v U_{vz} \, dA$$

and equations 2.47 and 2.48 become:

$$2E = \left[\tfrac{1}{2} < U_v^2 >_{A_{z+}} - < U_{vz} > < U_{vz} >_{A_{z+}} - \right.$$

$$\left. - < U_{vr} > < U_{vr} >_{A_{z+}} \right] \int_{A_{z+}} \alpha \rho_v U_{vz} \, dA -$$

$$- \left[\tfrac{1}{2} < U_v^2 >_{A_{z-}} - < U_{vz} > < U_{vz} >_{A_{z-}} - \right.$$

$$\left. - < U_{vr} > < U_{vr} >_{A_{z-}} \right] \int_{A_{z-}} \alpha \rho_v U_{vz} \, dA \qquad (2.49)$$

$$3E = \left[\tfrac{1}{2} < U^2 >_{A_{r+}} - < U_{vr} > < U_{vr} >_{A_{r+}} - \right.$$

$$\left. - < U_{vz} > < U_{vz} >_{A_{r+}} \right] \int_{A_{r+}} \alpha \rho_v U_{vr} \, dA -$$

$$- \left[\tfrac{1}{2} < U_v^2 >_{A_{r-}} - < U_{vr} > < U_{vr} >_{A_{r-}} - \right.$$

$$\left. - < U_{vz} > < U_{vz} >_{A_{r-}} \right] \int_{A_{r-}} \alpha \rho_v U_{vr} \, dA \qquad (2.50)$$

and from equations 2.17, 2.18 and 2.23 23 get:

$$2E = -\tfrac{1}{2}\left[< U_{vz} >_{A_{z+}} < U_{vz} >_{A_{z-}} + < U_{vr} >_{A_{z+}} < U_{vr} >_{A_{z-}}\right]\int_{A_{z+}} - \int_{A_{z-}} \alpha\ \rho_v U_{vz} dA$$

$$3E = -\tfrac{1}{2}\left[< U_{vr} >_{A_{r+}} < U_{vr} >_{A_{r-}} + < U_{vz} >_{A_{r+}} < U_{vz} >_{A_{r-}}\right]\int_{A_{r+}} - \int_{A_{r-}} \alpha\ \rho_v U_{vr} dA$$

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

$$< U_{vz} >_{A_{z+}} < U_{vz} >_{A_{z-}} \simeq < U_{vz}^2 > \text{ etc, and it follows}$$

$$2E = -\tfrac{1}{2} < U_v^2 > \int_{A_{z+}} - \int_{A_{z-}} \alpha\ \rho_v\ U_{vz}\ dA \qquad (2.51)$$

$$3E = -\tfrac{1}{2} < U_v^2 > \int_{A_{r+}} - \int_{A_{r-}} \alpha\ \rho_v\ U_{vr}\ dA \qquad (2.52)$$

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

$$1E + 2E + 3E = -\tfrac{1}{2}\ (< S_e > - < S_c >) < U_v^2 > \qquad (2.53)$$

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

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

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

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

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:

$$\frac{\partial}{\partial t}\left[\langle \alpha \rangle \langle \rho_v \rangle \langle e_v \rangle\right] + \frac{A_z}{V}\left[\langle \alpha\ \rho_v\ e_v\ U_{vz} \rangle_{A_{z+}} - \langle \alpha\ \rho_v\ e_v\ U_{vz} \rangle_{A_{z-}}\right]$$

$$+ \frac{A_{r+}}{V} \langle \alpha\ \rho_v\ e_v\ U_{vr} \rangle_{A_{r+}} - \frac{A_{r-}}{V} \langle \alpha\ \rho_v\ e_v\ U_{vr} \rangle_{A_{r-}} =$$

$$= \langle Q_v \rangle - \langle P \rangle \frac{D\alpha}{Dt} - \langle Q_{\ell v} \rangle \qquad (2.57)$$

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

For the liquid phase the nergy equation is:

$$\frac{\partial}{\partial t}\left[(1 - \langle\alpha\rangle)\langle\rho_\ell\rangle\langle e_\ell\rangle\right] + \frac{A_z}{V}\left[\langle(1-\alpha)\rho_\ell\, e_\ell\, U_{\ell z}\rangle_{A_{z+}} - \langle(1-\alpha)\rho_\ell\, e_\ell\, U_{\ell z}\rangle_{A_{z-}}\right]$$

$$+ \frac{A_{r+}}{V}\langle(1-\alpha)\,\rho_\ell\, e_\ell\, U_{\ell r}\rangle_{A_{r+}} - \frac{A_{r-}}{V}\langle(1-\alpha)\,\rho_\ell\, e_\ell\, U_{\ell r}\rangle_{A_{r-}} =$$

$$= \langle Q_\ell \rangle + \langle P \rangle \frac{D\alpha}{Dt} + \langle Q_{\ell v} \rangle \qquad (2.58)$$

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 differen iation 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 $\underline{i}$ 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:

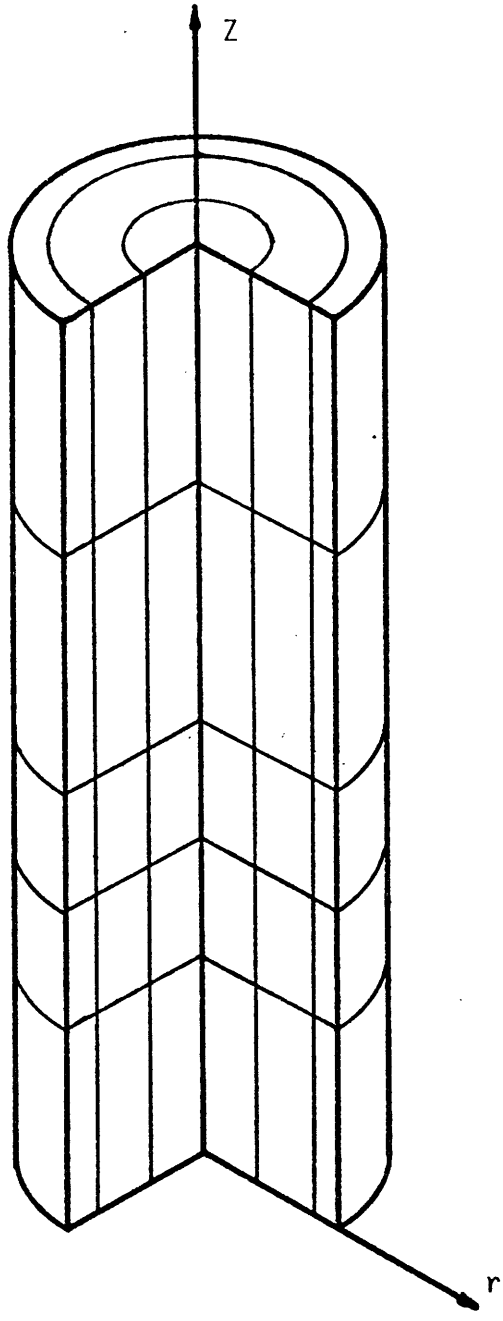

Figure 2.1 A Typical Cell Arrangement

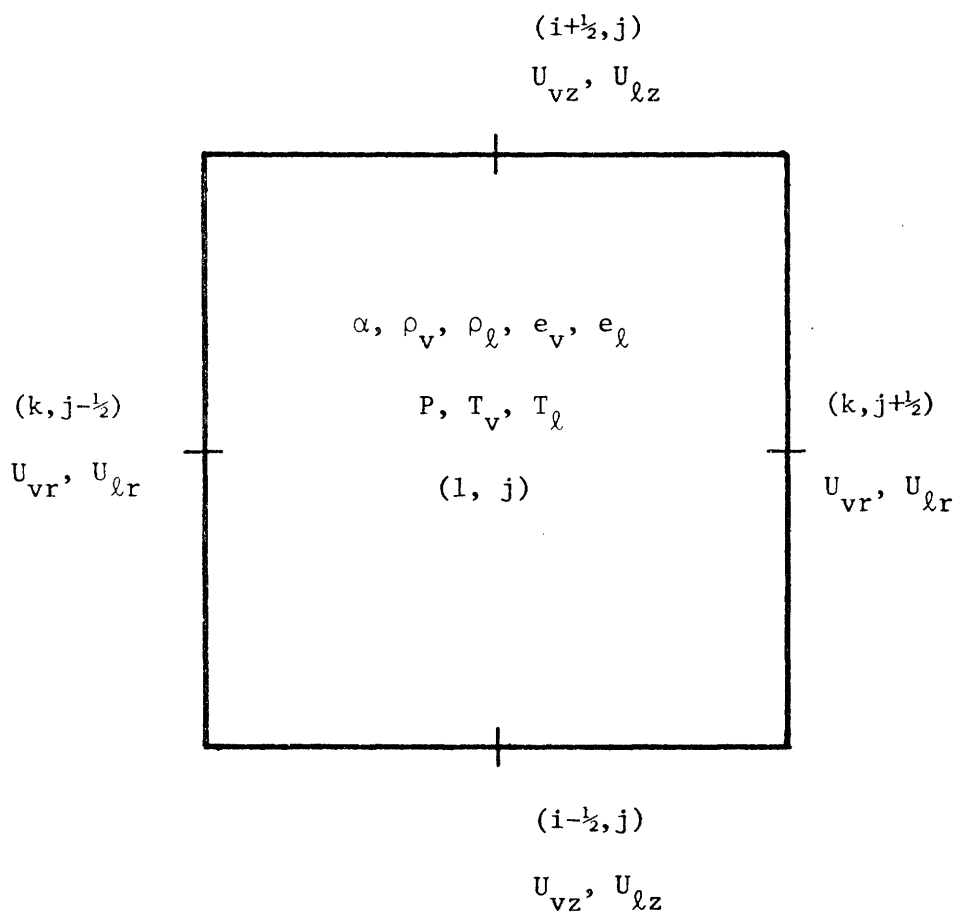

Figure 2.2 Position Evaluation of Variables

Vapor Mass:

$$\frac{\left(\alpha^{n+1}\ \rho_v^{n+1} - \alpha^n\ \rho_v^n\right)_{i,j}}{\Delta t} + \frac{\left(\alpha^n\ \rho_v^n\ U_{vz}^{n+1}\right)_{i+\frac{1}{2},j} - \left(\alpha^n\ \rho_v^n\ U_{vz}^{n+1}\right)_{i-\frac{1}{2},j}}{\Delta z_i} +$$

$$+ \left(A_r/V\ \alpha^n\ \rho_v^n\ U_{vr}^{n+1}\right)_{i,j+\frac{1}{2}} - \left(A_r/V\ \alpha^n\ \rho_v^n\ U_{vr}^{n+1}\right)_{i,j-\frac{1}{2}} = S_e - S_c \qquad (2.2.1)$$

Vapor Energy:

$$\frac{\left(\alpha^{n+1}\ \rho_v^{n+1}\ e_v^{n+1} - \alpha^n\ \rho_v^n\ e_v^n\right)_{i,j}}{\Delta t} + \frac{\left(\alpha^n\ \rho_v^n\ e_v^n\ U_{vz}^{n+1}\right)_{i+\frac{1}{2},j}}{\Delta z_i} -$$

$$- \frac{\left(\alpha^n\ \rho_v^n\ e_v^n\ U_{vz}^{n+1}\right)_{i-\frac{1}{2},j}}{\Delta z_i} + \left(A_r/V\ \alpha^n\ \rho_v^n\ e_v^n\ U_{vr}^{n+1}\right)_{i,j+\frac{1}{2}} -$$

$$- \left(A_r/V\ \alpha^n\ \rho_v^n\ e_v^n\ U_{vr}^{n+1}\right)_{i,j-\frac{1}{2}} + P_{ij}^n\left[\frac{\left(\alpha^{n+1} - \rho^n\right)_{i,j}}{\Delta t} +\right.$$

$$+ \frac{\left(\alpha^n\ U_{vz}^{n+1}\right)_{i+\frac{1}{2},j} - \left(\alpha^n \rho U_{vz}^{n+1}\right)_{i-\frac{1}{2},j}}{\Delta z_i} + \left(A_r/V\ \alpha^n\ U_{vr}^{n+1}\right)_{i,j+\frac{1}{2}} -$$

$$\left. - \left(A_r/V\ \alpha^n\ U_{vr}^{n+1}\right)_{i,j-\frac{1}{2}}\right] = Q_{wv} + Q_{\ell v} \qquad (2.2.2)$$

Liquid Mass:

$$\frac{\left[\left(1-\alpha^{n+1}\right)\rho_{\ell}^{n+1} - \left(1-\alpha^{n}\right)\rho_{\ell}^{n}\right]_{i,j}}{\Delta t} + \frac{\left[\left(1-\alpha^{n}\right)\rho_{\ell}^{n}\, U_{\ell z}^{n+1}\right]_{i+\frac{1}{2},j} - \left[\left(1-\alpha^{n}\right)\rho_{\ell}^{n}\, U_{\ell z}^{n+1}\right]_{i-\frac{1}{2},j}}{\Delta z_i} +$$

$$+ \left[A_r/V\left(1-\alpha^{n}\right)\rho_{\ell}^{n}\, U_{\ell r}^{n+1}\right]_{i,j+\frac{1}{2}} - \left[A_r/V\left(1-\alpha^{n}\right)\rho_{\ell}^{n}\, U_{\ell r}^{n+1}\right]_{i,j-\frac{1}{2}} = -S_e + S_c \qquad (2.2.3)$$

Liquid Energy:

$$\frac{\left[\left(1-\alpha^{n+1}\right)\rho_{\ell}^{n+1}\, e_{\ell}^{n+1} - \left(1-\alpha^{n}\right)\rho_{\ell}^{n}\, e_{\ell}^{n}\right]}{\Delta t} + \frac{\left[\left(1-\alpha^{n}\right)\rho_{\ell}^{n}\, e_{\ell}^{n}\, U_{\ell z}^{n+1}\right]_{i+\frac{1}{2},j}}{\Delta z_i} -$$

$$- \left[\left(1-\alpha^{n}\right)\rho_{\ell}^{n}\, e_{\ell}^{n}\, U_{\ell z}^{n+1}\right]_{i-\frac{1}{2},j} + \left[A_r/V\left(1-\alpha^{n}\right)\rho_{\ell}^{n}\, e_{\ell}^{n}\, U_{\ell r}^{n+1}\right]_{i,j+\frac{1}{2}} -$$

$$- \left[A_r/V\left(1-\alpha^{n}\right)\rho_{\ell}^{n}\, e_{\ell}^{n}\, U_{\ell r}^{n+1}\right]_{i,j-\frac{1}{2}} + P_{ij}^{n}\left\{-\frac{\left(\alpha^{n+1} - \alpha^{n}\right)_{ij}}{\Delta t} + \right.$$

$$+ \frac{\left[\left(1-\alpha^{n}\right) U_{\ell z}^{n+1}\right]_{i+\frac{1}{2},j} - \left[\left(1-\alpha^{n}\right) U_{\ell z}^{n+1}\right]_{i-\frac{1}{2},j}}{\Delta z_i} + \left[A_r/V\left(1-\alpha^{n}\right) U_{\ell r}^{n+1}\right]_{i,j+\frac{1}{2}}$$

$$\left. - \left[Ar/V\left(1-\alpha^{n}\right) U_{\ell r}^{n+1}\right]_{i,j-\frac{1}{2}}\right\} = Q_{w\ell} - Q_{\ell v} \qquad (2.2.4)$$

Some variables in the above equations are used in a location other than the place where they are primarely defined (see figure 2.2.2). For instance, the void fraction α, 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 α, ρ_v, ρ_ℓ, e_v, e_ℓ. The face centered value $X_{i+\frac{1}{2}}$ will be given by:

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

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 primarely 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:

$$(\alpha\rho_v)^n_{i+\frac{1}{2},j}\left[\frac{\left(U^{n+1}_{vz}-U^n_{vz}\right)_{i+\frac{1}{2},j}}{\Delta t}+U^n_{vz\ i+\frac{1}{2},j}\frac{\left(\Delta_z U^n_{vz}\right)_{i+\frac{1}{2},j}}{\Delta z}+\right.$$

$$\left.+U^n_{vr\ i+\frac{1}{2},j}\frac{\left(\Delta_r U^n_{vz}\right)_{i+\frac{1}{2},j}}{\Delta r}\right]+\alpha^n_{i+\frac{1}{2},j}\frac{\left(P^{n+1}_{i+1,j}-P^{n+1}_{ij}\right)}{\Delta z_{i+\frac{1}{2}}}+$$

$$+(\alpha\rho_v)^n_{i+\frac{1}{2},j}\ g=-\left(M_{wzv}+M_{\ell vz}\right)_{i+\frac{1}{2},j} \qquad (2.2.6)$$

$$(\alpha\rho_v)^n_{i+\frac{1}{2},j}\left[\frac{(U^{n+1}_{vr}-U^n_{vr})_{i,j+\frac{1}{2}}}{\Delta t}+U^n_{vz\ i,j+\frac{1}{2}}\frac{(\Delta_z U^n_{vr})_{i,j+\frac{1}{2}}}{\Delta z}+\right.$$

$$\left.+U^n_{vr\ i,j+\frac{1}{2}}\frac{(\Delta_r U^n_{vr})_{i,j+\frac{1}{2}}}{\Delta r}\right]+{}^{n}_{i,j+\frac{1}{2}}\frac{(P^{n+1}_{i,j+1}-P^{n+1}_{ij})}{\Delta r_{j+\frac{1}{2}}}=$$

$$=-(M_{wrv}+M_{\ell vr})_{i,j+\frac{1}{2}} \qquad (2.2.7)$$

And the liquid momentum equations:

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

$$\left.+U^n_{\ell r\ i+\frac{1}{2},j}\frac{(\Delta_r U^n{}_z)_{i+\frac{1}{2},j}}{\Delta r}\right]+(1-\alpha^n_{i+\frac{1}{2},j})\frac{(P^{n+1}_{i+1,j}-P^{n+1}_{i,j})}{\Delta z_{i+\frac{1}{2}}}+$$

$$+\left[(1-\alpha)\rho_\ell\right]^n_{i+\frac{1}{2},j}\ g=-(M_{wz\ell}-M_{\ell vz})_{i+\frac{1}{2},j} \qquad (2.2.8)$$

$$\left[(1-\alpha)\rho_\ell\right]^n_{i,j+\frac{1}{2}}\left[\frac{(U^{n+1}_{\ell r}-U^n_{\ell r})_{i,j+\frac{1}{2}}}{\Delta t}+U^n_{\ell z\ i,j+\frac{1}{2}}\frac{(\Delta_z U^n_{\ell r})_{i,j+\frac{1}{2}}}{\Delta z}+\right.$$

$$+ \; U^{n}_{\ell r\; i,j+\frac{1}{2}} \frac{(\Delta_r U^{n}_{r})_{i,j+\frac{1}{2}}}{\Delta r}\Bigg] + (1-\alpha^{n}_{ij+\frac{1}{2}}) \frac{(P^{n+1}_{i,j+1} - P^{n+1}_{i,j})}{\Delta r_{j+\frac{1}{2}}} =$$

$$= - (M_{wr\ell} - M_{\ell vr})_{i,j+\frac{1}{2}} \qquad (2.2.9)$$

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 α and the densities ρ_v and ρ_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:

$$X_{i+\frac{1}{2}} = \frac{X_{i+1}\,\Delta Z_{i+1} + X_i\,\Delta Z_i}{\Delta Z_{i+1} + \Delta Z_i} \qquad (2.2.10)$$

where X stands for the void fraction α and the two densities ρ_v and ρ_ℓ. A similar rule is used to transfer the variables to the faces $j+\frac{1}{2}$ in the radial direction, with Δr replacing ΔZ.

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

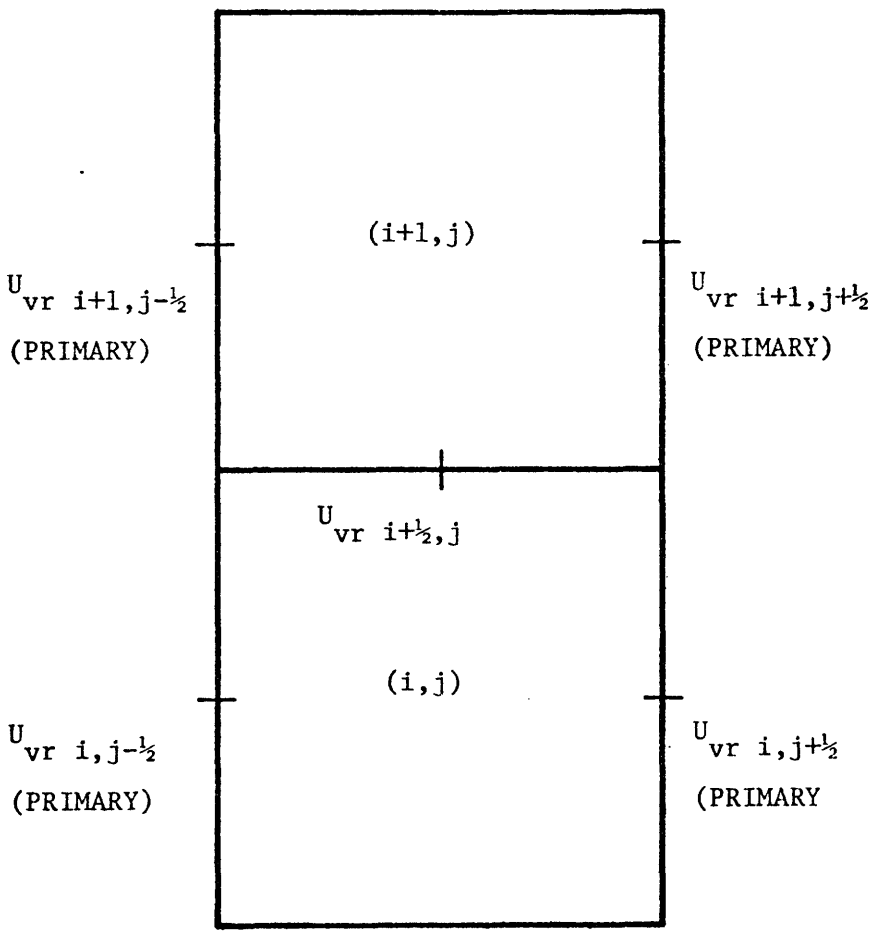

Figure 2.3 Different Positions for the Radial Velocity

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

$$U_{vz\ i,j+\frac{1}{2}} = \tfrac{1}{4}\left[U_{vz\ i+\frac{1}{2},j} + U_{vz\ i-\frac{1}{2},j} + U_{vz\ i+\frac{1}{2},j+1} + U_{vz\ i-\frac{1}{2},j+1}\right] \tag{2.2.11}$$

$$U_{vr\ i+\frac{1}{2},j} = \tfrac{1}{4}\left[U_{vr\ i,j+\frac{1}{2}} + U_{vr\ i,j-\frac{1}{2}} + U_{vr\ i+1,j+\frac{1}{2}} + U_{vr\ i+\frac{1}{2},j+\frac{1}{2}}\right] \tag{2.2.12}$$

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:

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

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

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

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

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

$$\Delta z_{i+\frac{1}{2}} = \frac{\Delta z_{i+1} + \Delta z_i}{2} \tag{2.2.17}$$

$$\Delta r_{j+\frac{1}{2}} = \frac{\Delta r_{j+1} + \Delta r_j}{2} \tag{2.2.18}$$

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 determine 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:

$$X = \left[\alpha,\ P,\ T_v,\ T_\ell,\ U_{vz},\ U_{vr},\ U_{\ell z},\ U_{\ell r}\right]^{n+1} \qquad (2.3.1)$$

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

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

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 necessarely equal to zero. Then, let us make a Taylor expansion of F(X) around the point X^k:

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

$$P = 1, 8$$

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

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

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:

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

$$- \frac{\partial S}{\partial T}\,\delta T_{\ell} + \frac{\left(\alpha\rho_v\right)_{i+\frac{1}{2},j}}{\Delta zi}\,\delta U_{vz\ i+\frac{1}{2},j} - \frac{\left(\alpha\rho_v\right)_{i-\frac{1}{2},j}}{\Delta zi}\,\delta U_{vz\ i-\frac{1}{2},j} +$$

$$+ \left(Ar/V\alpha\rho_v\right)_{i\ j+\frac{1}{2}}\,\delta U_{vr\ i\ j+\frac{1}{2}} - \left(Ar/V\ \alpha\rho_v\right)_{i,j-\frac{1}{2}}\,\delta U_{vr\ i,j-\frac{1}{2}} = - F_1^k \qquad (2.3.5)$$

$$\left[\frac{\rho_v^k\, e_v^k + P^k}{\Delta t} - \frac{\partial Q_v}{\partial \alpha}\right]\delta\alpha + \left[\left(\alpha^k\,\rho_v^k\,\frac{\partial e_v}{\partial P} + \alpha^k\, e_v^k\,\frac{\partial \rho_v}{\partial P}\right)\frac{1}{\Delta t} - \frac{\partial Q_v}{\partial P}\right]\delta P_{ij} +$$

$$+ \left[\frac{\alpha^k}{\Delta t}\left(\rho_v^k\,\frac{\partial e_v}{\partial T_v} + e_v^k\,\frac{\partial \rho_v}{\partial T_v}\right) - \frac{\partial Q_v}{\partial T_v}\right]\delta T_v - \frac{\partial Q_v}{\partial T}\,\delta T_{\ell} +$$

$$+\left[\frac{\alpha_{i+\frac{1}{2},j}}{\Delta zi}\left(P_{ij} + \left(\rho_v\, e_v\right)_{i+\frac{1}{2},j}\right)\right]\delta U_{vz\ i+\frac{1}{2},j} - \left[\frac{\alpha_{i-\frac{1}{2},j}}{\Delta zi}\left(P_{ij} +\right.\right.$$

$$\left.\left.+\left(\rho_v\, e_v\right)_{i-\frac{1}{2},j}\right)\right]\delta U_{vz\ i-\frac{1}{2},j} + \left[\left(Ar/V\,\alpha\right)_{i,j+\frac{1}{2}}\left(P_{ij} + \left(\rho_v e_v\right)_{i,j+\frac{1}{2}}\right)\right]\delta\, U_{vr\ i,j+\frac{1}{2}}$$

$$- \left[\left(Ar/V\ \alpha\right)_{i,j-\frac{1}{2}}\left(P_{ij} + \left(\rho_v\ e_v\right)_{i,j-\frac{1}{2}}\right)\right]\delta U_{vr\ i,j-\frac{1}{2}} = - F_2^k \qquad (2.3.6)$$

$$- \left[\frac{\rho_\ell^k}{\Delta t} - \frac{\partial S}{\partial \alpha}\right] \delta\alpha + \left[\frac{(1-\alpha^k)}{\Delta t} \frac{\partial \rho_\ell}{\partial P} + \frac{\partial S}{\partial P}\right] \delta P_{ij} + \frac{\partial S}{\partial T_v} \delta T_v +$$

$$+ \left[\frac{(1-\alpha^k)}{\Delta t} \frac{\partial \rho_\ell}{\partial T_\ell} + \frac{\partial S}{\partial T_\ell}\right] \partial T_\ell + \left[(1-\alpha)\rho_\ell\right]_{i+\frac{1}{2},j} \delta U_{\ell z\ i+\frac{1}{2},j} -$$

$$- \left[(1-\alpha)\rho_\ell\right]_{i-\frac{1}{2},j} \delta U_{\ell\ z\ i-\frac{1}{2},j} + \left[Ar/V\ (1-\alpha)\rho_\ell\right]_{i,j+\frac{1}{2}} \delta U_{\ell r\ i,j+\frac{1}{2}} -$$

$$- \left[Ar/V\ (1-\alpha)\rho_\ell\right]_{i,j-\frac{1}{2}} \delta U_{vr\ i,j-\frac{1}{2}} = - F_3^k \qquad (2.3.7)$$

$$- \left[\frac{\rho_\ell^k\ e_\ell^k + P}{\Delta t} - \frac{\partial Q}{\partial \alpha}\right] \delta\alpha + \left[\frac{(1-\alpha)}{\Delta t}\left(\rho_\ell^k \frac{\partial e_\ell^k}{\partial P} + e_\ell^k \frac{\partial \rho_\ell}{\partial P}\right) - \frac{\partial Q_\ell}{\partial P}\right]\delta P_{ij} -$$

$$- \frac{\partial Q_\ell}{\partial T_v} \delta T_v + \left[\frac{(1-\alpha^k)}{\Delta t}\left(\rho_\ell^k \frac{\partial e_\ell}{\partial T_\ell} + e_\ell^k \frac{\partial \rho_\ell}{\partial T_\ell}\right) - \frac{\partial Q_\ell}{\partial T\ell}\right]\delta T_\ell +$$

$$+ \left[\frac{\left(1-\alpha_{i+\frac{1}{2},j}\right)}{\Delta z_i}\left(P_{ij} + (\rho_\ell\ e_\ell)_{i+\frac{1}{2},j}\right)\right] \delta\ U_{\ell z\ i+\frac{1}{2},j} - \frac{\left(1-\alpha_{i-\frac{1}{2},j}\right)}{\Delta z_i}\left[\left(P_{ij} +\right.\right.$$

$$\left.\left. + (\rho_\ell\ e_\ell)_{i-\frac{1}{2},j}\right)\right] \delta U_{\ell z\ i-\frac{1}{2},j} + \left[\left(Ar/V(1-\alpha)\right)_{i,j+\frac{1}{2}}\left(P_{ij} + (\rho_\ell e_\ell)_{i,j+\frac{1}{2}}\right)\right]\delta U_{\ell r\ i,j+\frac{1}{2}}$$

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

$$\left[\frac{(\alpha\rho_v)_{i+\frac{1}{2},j}}{\Delta t} + \frac{\partial M_v}{\partial U_{vz}}\right]\delta\ U_{vz\ i,+\frac{1}{2},j} + \frac{\partial M_v}{\partial U_{\ell z}}\ \delta\ U_{\ell z\ i+\frac{1}{2},j} +$$

$$+ \frac{\alpha_{i+\frac{1}{2},j}}{\Delta z_{i+\frac{1}{2}}}\ (\delta\ P_{i+1,j} - \delta\ P_{ij}) = - F_5^k \tag{2.3.9}$$

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

$$+ \frac{\left(1-\alpha_{i+\frac{1}{2},j}\right)}{\Delta z_{i+\frac{1}{2}}}\ (\delta\ P_{i+1,j} - \delta\ P_{ij}) = - F_6^k \tag{2.3.10}$$

$$\left[\frac{(\alpha\rho_v)_{i,j+\frac{1}{2}}}{\Delta t} + \frac{\partial M_v}{\partial U_{vr}}\right]\delta\ U_{vr\ i,j+\frac{1}{2}} + \frac{\partial M_v}{\partial U_{\ell r}}\ \delta\ U_{\ell r\ i,j+\frac{1}{2}} +$$

$$+ \frac{\alpha_{i,j+\frac{1}{2}}}{\Delta r_{j+\frac{1}{2}}}\ (\delta\ P_{i,j+1} - \delta\ P_{ij}) = F_7^k \tag{2.3.11}$$

$$\left[\frac{(1-\alpha)\rho_{\ell\ i,j+\frac{1}{2}}}{\Delta t} + \frac{\partial M_\ell}{\partial U_{\ell r}}\right]\delta\ U_{\ell r\ i,j+\frac{1}{2}} + \frac{\partial M_\ell}{\partial U_{vr}}\ \delta\ U_{vr\ i,j+\frac{1}{2}} +$$

$$+ \frac{\left(1-\alpha_{i,j+\frac{1}{2}}\right)}{\Delta r_{j+\frac{1}{2}}}\ (\delta\ P_{i,j+1} - \delta\ P_{ij}) = - F_8^k \tag{2.3.12}$$

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:

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

$$\delta U_{\ell z\ i+\frac{1}{2},j} = W_{\ell z\ i+\frac{1}{2},j} (\delta P_{i+1,j} - \delta P_{ij}) + f_{u\ell z} \qquad (2.3.14)$$

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

$$\delta U_{\ell r\ i,j+\frac{1}{2}} = W_{\ell r\ i,j+\frac{1}{2}} (\delta P_{i,j+1} - \delta P_{ij}) + f_{u\ell r} \qquad (2.3.16)$$

where the coefficients W are given by:

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

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

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 matrice form:

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \times \begin{bmatrix} \delta\alpha \\ \delta T_v \\ \delta T_\ell \\ \delta P \end{bmatrix}_{i,j} + \quad .$$

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

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 $\underline{a}$ in equation 2.3.18 into an upper triangular matrix, this equation becomes:

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

(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 $\underline{a}$ 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 α 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:

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

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 $\underline{a}$ 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;

$$M_{\ell v} = K\ (U_{\ell}^{n+1} - U_{v}^{n+1})$$

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 ot 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 $U_{ri,\frac{1}{2}}$ 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 = ½ and i = I + ½ the following conditions are imposed

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

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

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.

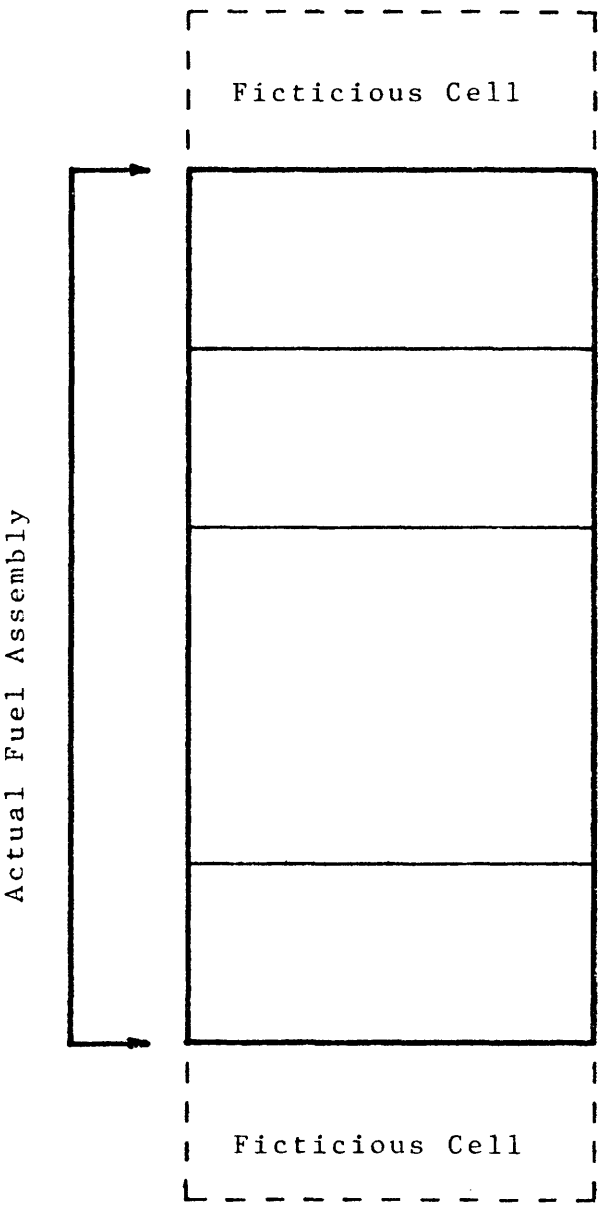

Figure 2.4 The Ficticious Cells

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:

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

To perform the kth 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 tope of cell (i,j).

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

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 centimenter.

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 atenuating 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:

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

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

and the equation of state:

$$\frac{\partial\rho}{\partial P} = \frac{1}{c^2} \tag{2.4.5}$$

with c being the sonic speed.

Applying the differentiating scheme to these equations we get:

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

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

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

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

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

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

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:

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

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

$$C_{ij} = -A_{ij} - B_{ij} - D_{ij} - E_{ij} + \frac{1}{c^2} \qquad (2.4.11c)$$

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

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

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, Δz is ten or more times larger than Δr, 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 that 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 loosing 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:

$$\frac{\partial}{\partial t}\alpha\rho_v + \frac{\partial}{\partial z}\alpha\rho_v U_v = S \tag{2.5.1}$$

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

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

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

and the equations of state:

$$\frac{\partial \rho_v}{\partial P} = \frac{1}{c_v^2} \tag{2.5.5}$$

$$\frac{\partial \rho_\ell}{\partial P} = \frac{1}{c_\ell^2} \tag{2.5.6}$$

In the canonical form the above equations would appear as:

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

with

$$X = [\alpha, P, U_v, U_\ell]^T \tag{2.5.8}$$

$$A = \begin{bmatrix} \rho_v & \alpha/c_v^2 & 0 & 0 \\ -\rho_\ell & (1-\alpha)/c_\ell^2 & 0 & 0 \\ 0 & 0 & \alpha\rho_v & 0 \\ 0 & 0 & 0 & (1-\alpha)\rho_\ell \end{bmatrix} \tag{2.5.9}$$

$$B = \begin{bmatrix} \rho_v U_v & \alpha U_v/c_v^2 & \alpha\rho_v & 0 \\ -\rho_\ell U_\ell & (1-\alpha)U_\ell/c_\ell & 0 & (1-\alpha)\rho_\ell \\ 0 & \alpha & \alpha_v U_v & 0 \\ 0 & (1-\alpha) & 0 & (1-\alpha)\rho_\ell U_\ell \end{bmatrix} \tag{2.5.10}$$

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

$$\det[B - \lambda A] = 0 \tag{2.5.11}$$

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

$$\alpha\rho_\ell(U_\ell-\lambda)^2 + (1-\alpha)\rho_v(U_v-\lambda)^2 - \left[\frac{\alpha\rho_\ell}{c_v^2} + \frac{(1-\alpha)\rho_v}{c_\ell^2}\right](U_v-\lambda)^2(U_\ell-\lambda)^2 = 0 \tag{2.5.12}$$

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 ρ_v, and two real roots are obtained, which are approximately:

$$\lambda \simeq U_v \pm c_v \tag{2.5.13}$$

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 $1/c_\ell^2$ are neglected, the two other roots become:

$$\lambda \simeq \frac{U_\ell + \varepsilon^2 U_v}{1+\varepsilon^2} \pm \frac{i\varepsilon(U_v - U_\ell)}{1+\varepsilon^2} \qquad (2.5.14)$$

with

$$\varepsilon^2 = \frac{(1-\alpha)\rho_v}{\alpha\rho_\ell} \qquad (2.5.15)$$

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:

$$\frac{\alpha_j^{n+1}\rho_{vj}^{n+1} - \alpha_j^n\rho_{vj}^n}{\Delta t} + \frac{\alpha_j^n\rho_{vj}^n U_{vj+\frac{1}{2}}^{n+1} - \alpha_{j-1}^n\rho_{vj-1}^n U_{vj-\frac{1}{2}}^{n+1}}{\Delta z} = S \qquad (2.5.16)$$

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

$$\alpha_j^n \rho_{vj}^n \left[\frac{U_{vj+\frac{1}{2}}^{n+1} - U_{vj+\frac{1}{2}}^n}{\Delta t} + U_{vj+\frac{1}{2}}^n \frac{(U_{vj+\frac{1}{2}}^n - U_{vj-\frac{1}{2}}^n)}{\Delta z}\right] +$$

$$+ \alpha_j^n \frac{(P_{j+1}^{n+1} - P_j^{n+1})}{\Delta z} = k_{j+\frac{1}{2}}(U_{\ell j+\frac{1}{2}}^{n+1} - U_{vj+\frac{1}{2}}^{n+1}) \tag{2.5.18}$$

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

$$+ (1-\alpha_j^n)\frac{(P_{j+1}^{n+1} - P_j^{n+1})}{\Delta z} = k_{j+\frac{1}{2}}(U_{vj+\frac{1}{2}}^{n+1} - U_{\ell j+\frac{1}{2}}^{n+1}) \tag{2.5.19}$$

The convective terms in the mass and momentum equations involve donor cell differenciating, so the above equations are written for both U_v and U_ℓ 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:

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

where

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

Applying this formalism to equations 2.5.16 through 2.5.19 it follows:

$$\frac{\alpha}{\Delta t}(\xi-1)\varepsilon^{n}_{\rho vj} + \frac{\rho_v}{\Delta t}(\xi-1)\varepsilon^{n}_{\alpha j} + \frac{\alpha\rho_v}{\Delta z}(1-e^{-i\theta})\xi\varepsilon^{n}_{Uvj+\frac{1}{2}} + \frac{\alpha U_v}{\Delta z}(1-e^{-i\theta})\varepsilon^{n}_{\rho vj} +$$

$$+ \frac{\rho_v U_v}{\Delta z}(1-e^{-i\theta})\varepsilon^{n}_{\alpha j} = 0 \qquad (2.5.20)$$

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

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

$$(1-\alpha)\rho_\ell\left[\frac{(\xi-1)}{\Delta t}\varepsilon^{n}_{U\ell j+\frac{1}{2}} + \frac{U_\ell}{\Delta z}(1-e^{-i\theta})\varepsilon^{n}_{U\ell j+\frac{1}{2}}\right] + \frac{(1-\alpha)C_v^2}{\Delta z}(e^{i\theta}-1)\xi\varepsilon^{n}_{\rho vj} =$$

$$= k\xi(\varepsilon^{n}_{Uvj+\frac{1}{2}} - \varepsilon^{n}_{U\ell j+\frac{1}{2}}) \qquad (2.5.23)$$

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

$$/A \times E = 0$$

with

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

and

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

Where we have abbreviated

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

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

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

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

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

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

In order for the errors in the basic variables not to grow geometrically, the absolute value of the eigenvalues ξ 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

$$\xi^2 C_m^2[(\xi-1+\tilde{U}_v+2\xi\kappa)(\xi-1+\tilde{U}_v)\varepsilon^2 + (\xi-1+\tilde{U}_\ell)(\xi-1+U_\ell+2\xi\kappa) +$$

$$+ (\xi-1+\tilde{U}_v)(\xi-1+\tilde{U}_\ell)[(\xi-1+\tilde{U}_v+\xi\kappa)(\xi-1+U_\ell+\xi\kappa\rho_v/\rho_\ell) - (\xi\kappa)^2\ \rho_v/\rho_\ell] = 0 \qquad (2.5.24)$$

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

$$\xi^2 C_m^2 [(\xi-1+\tilde{U}_v)^2\varepsilon^2 + (\xi-1+\tilde{U}_\ell)^2] + (\xi-1+\tilde{U}_v)^2(\xi-1+\tilde{U}_\ell)^2 = 0 \qquad (2.5.25)$$

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

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

$C_m^2 >> 1$. It follows that equation 2.5.25 will have two roots of magnitude approximately $\xi \simeq \pm 1/C_m$, which are smaller than one. The other two roots approximately satisfy:

or

$$\xi \simeq \frac{1 - \tilde{U}_\ell(1 \pm i\varepsilon \tilde{U}_v/\tilde{U}_\ell)}{1 \pm i\varepsilon} \qquad (2.5.26)$$

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

$$\frac{U_\ell \Delta t}{\Delta z} \leq 1 \qquad (2.5.27)$$

and

$$\frac{U_v \Delta t}{\Delta z} \leq 1 \qquad (2.5.28)$$

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 $m \to \infty$, $C_m \to 0$) and in the limit the roots of 2.5.25 will be:

$$\xi = \pm 1 - \tilde{U}_v \tag{2.5.29}$$

and

$$\xi = \pm 1 - \tilde{U}_\ell \tag{2.5.30}$$

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

$$\xi = 1 - \tilde{U}_\ell + \delta \tag{2.5.31}$$

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

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

and the roots of this equation are:

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

and again using the fact that $(U_v - U_\ell) / C_v << 1$ we can write the expression for the characteristic root ξ as:

$$\xi \simeq (1-\tilde{U}_\ell)\left[1 - \frac{(\tilde{U}_v-\tilde{U}_\ell)\epsilon(\epsilon \pm i)}{1+\epsilon^2}\right] \tag{2.5.34}$$

Since $|1 - \tilde{U}_\ell|$ and $|1 + (\tilde{U}_v - \tilde{U}_\ell)|$ are of the same order of magnitude for some value of ε one root ξ 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 $m \to \infty$ and then introduce a perturbation of order 1/m. Then, for $m \to \infty$ equation 2.5.24 becomes:

$$(\xi-1+\tilde{U}_v)(\xi-1+\tilde{U}_\ell)_x[(\xi-1+\tilde{U}_v+\xi\kappa)(\xi-1+\tilde{U}_\ell+\xi\kappa\ \rho_v/\rho_\ell)-\xi^2\kappa^2\rho_v/\rho_\ell] = 0 \tag{2.5.35}$$

and the roots of this equation are:

$$\xi = 1-\tilde{U}_v \tag{2.5.36a}$$

$$\xi = 1-\tilde{U}_\ell \tag{2.5.36b}$$

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

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

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 δ 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 ξ, 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 δ. This will give a first order equation, and the single root of this equation gives an expression for ξ as:

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

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

$$\rho_\ell \, |\tilde{U}_\ell - \tilde{U}_v| \ll \rho_v K \qquad (2.5.38)$$

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 Tckar 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

Property	SI Units	Equal to
Temperature	°K	°C + 273.15
Pressure	Pa	14.05×10^{-5} lbf/in^2
Density	kg/m^3	0.06243 lbm/ft^3
Internal Energy	J/kg	4.2992×10^{-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	2.3884×10^{-4} BTU/lbm °F
Surface Tension	N/m	—

The expression is:

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

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:

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

with

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

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

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

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:

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

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 approximatly 900°C, and the expression for the liquid density becomes:

$$\rho_\ell(P,T) = r\ell_o + r\ell_1 T + r\ell_2 T^2 + r\ell_3 T^3 + r\ell_4 P$$

with

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

$$r\ell_1 = - 0.2205$$

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

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

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

which is valid in the range

$$100 < T < 1370°C$$

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:

$$h_\ell(T) = h\ell_o + h\ell_1 T + h\ell_2 T^2 + h\ell_3 T^3$$

with:

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

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

$$h\ell_2 = - 0.41672$$

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

valid in the range

$$100 < T < 1500°C$$

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:

$$h_v\ (T) = h\ell\ (T) + hv_o + hv_1\ T$$

with

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

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

valid for $600° < T < 1200°C$

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

$$K_\ell(T) = C\ell_o + C\ell_1\ T + C\ell_2\ T^2$$

with

$$C\ell_o = 1.0969 \times 10^2$$

$$C\ell_1 = -\ 6.4494 \times 10^{-2}$$

$$C\ell_2 = 1.1727 \times 10^{-5}$$

valid for

$100 < T < 1370°C$

Vapor Thermal Conductivity

$$K_v(T) = Cv_o + Cv_1 T + Cv_2 T^2$$

with

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

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

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

for

$700 < T < 5000°C$

Liquid Viscosity

$$(T) = \exp\left[v\ell_o + \frac{v\ell_1}{T} + v\ell_2 \ell n\ T\right]$$

with

$$v\ell_o = -5.732$$

$$v\ell_1 = 508.7$$

$$v\ell_2 = -0.4925$$

for

$100 < T < 1370°C$

Vapor Viscosity

$$\eta_v(T) = vv_o + vv_1 T$$

with

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

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

for

$700 < T < 5000°C$

Liquid Specific Heat

$$C_{p\ell}(T) = C_{p\ell_o} + C_{p\ell_1} T + C_{p\ell_2} T^2$$

with

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

$C_{p\ell_1} = -0.83344$

$C_{p\ell_2} = 4.6281 \times 10^{-4}$

for

$100 < T < 1500°C$

Vapor Specific Heat

$$C_{pv}(T) = C_{pv_o} + C_{pv_1} T + C_{pv_2} T^2$$

with

$C_{pv_o} = 0.5871 \times 10^3$

$C_{pv_1} = -0.83344$

$C_{pv_2} = 4.6281 \times 10^{-4}$

for

$600 < T < 1200°C$

Surface Tension

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

with

$st_o = 0.18$

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

in the range

$100 < T < 1370°C$

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 fited

$$Prv\ (T) = pv_o + pv_1\ (T - pv_2)^2$$

with

$$pv_o = 0.7596$$

$$pv_1 = 0.810 \times 10^{-6}$$

$$pv_2 = 844.4$$

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

$$600 < T < 1200°C$$

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, differenciable 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:

$$j = \sqrt{\frac{M}{2\pi R}} \left(\frac{Pv}{\sqrt{T_v}} - \frac{P\ell}{\sqrt{T_\ell}} \right) \tag{3.2.1}$$

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:

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

the Clayperon equation

$$\left(\frac{dP}{dT} \right)_{sat} = \frac{h_{fg}}{T\, v_{fg}} \tag{3.2.3}$$

is used to eliminate ΔP in equation 3.2.2 leading to:

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

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 $\rho\ell >> \rho v$, thus:

$$v_{fg} = \frac{1}{\rho_v} - \frac{1}{\rho_\ell} \simeq \frac{1}{\rho_v}$$

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

$$\frac{h_{fg}}{P\, v_{fg}} >> \frac{1}{2}$$

Therefore equation 3.2.4 becomes:

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

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:

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

Figure 3.1 reproduced from reference 41 shows the value of σ 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 σ is relatively small, thus the simplification can be made:

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

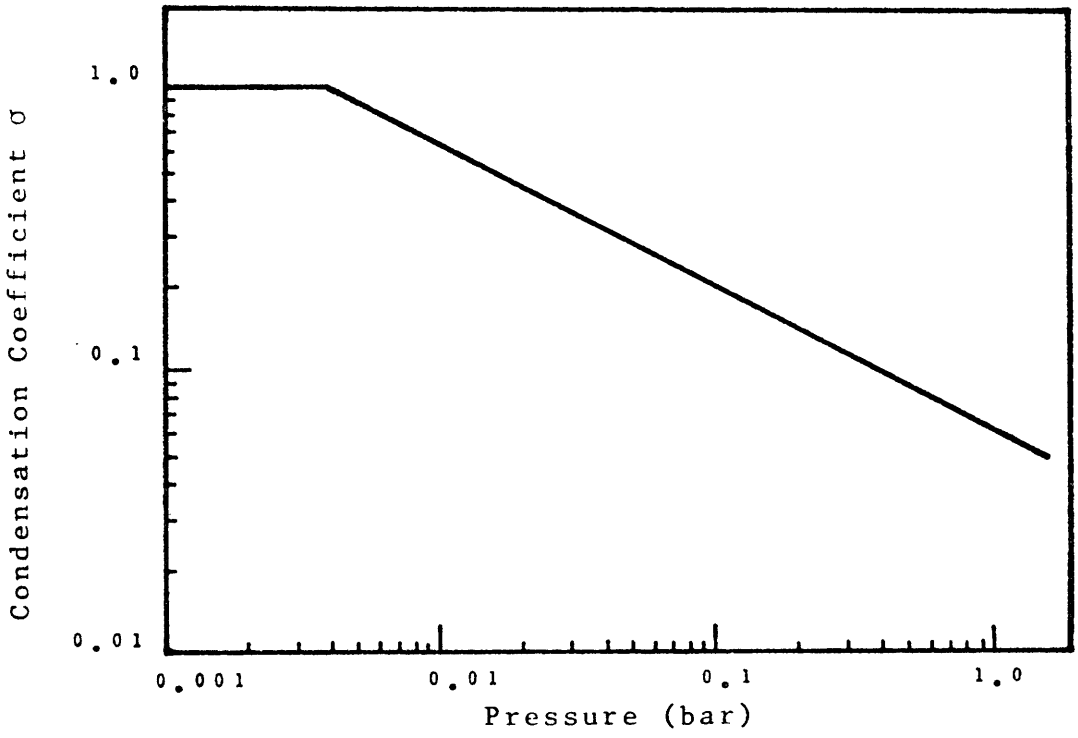

Figure 3.1 Condensation Coefficient as a Function of Pressure

(From Reference 41)

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

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

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 existance of unreasonalby small vapor bubbles, thus overestimating the specific area. To correct this we introduced a minimum value for the bubble radius, so that for

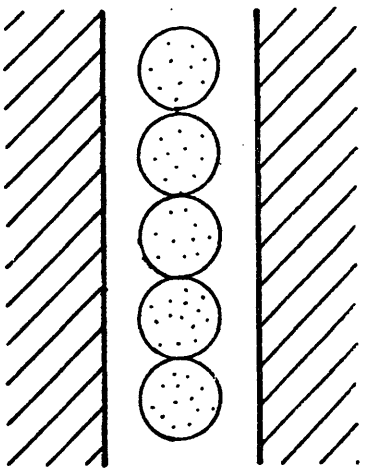

Figure 3.2 Bubbly Flow Representation

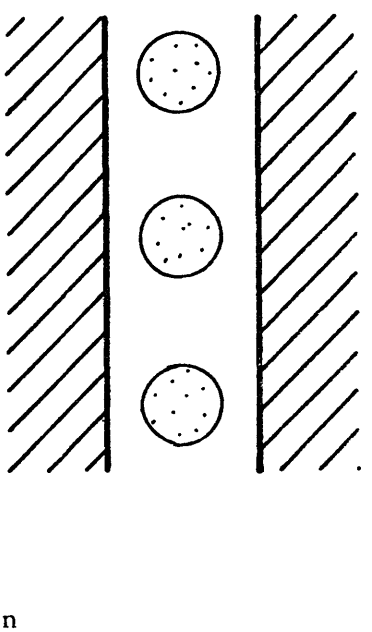

Figure 3.3 Low Void Fraction Bubbly Flow Representation

small void fractions the model would be pictured as in Figure 3.3. The expression for the specific area becomes:

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

where

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

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

For the anular 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:

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

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:

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

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 choosen 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 α is substituted in equations 3.2.8 and 3.2.10 by $\hat{\alpha}$ which is defined as:

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

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

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:

$$S = S_e - S_c \tag{3.2.11}$$

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

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

where

$$\sigma_e = \begin{cases} 0 & \text{if} \quad T_\ell < T_s \\ 1.0 & \text{if} \quad T_\ell \geq T_s \end{cases} \tag{3.2.14}$$

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

$$A = \frac{3\,\hat{\alpha}}{r_m} \qquad \text{for} \quad \alpha < \alpha_m \tag{3.2.16}$$

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

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

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

$$0.6 < \alpha < 0.957$$

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

$$\alpha > 0.957$$

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

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

$\frac{R}{M} = 361.30$ J/kg °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

$$F_{\ell z} = \left[\frac{0.18}{2D_H} Re_\ell^{-.2} \rho_\ell |U_{\ell z}|\right]^n U_{\ell z}^{n+1} \quad \alpha < \alpha \tag{3.3.1}$$

$$F_{\ell z} = \left[\frac{0.18}{2D_H} Re_\ell^{-.2} \rho_\ell |U_{\ell z}| \frac{(1-\alpha)}{(1-\alpha_{dry})}\right]^n U_{\ell z}^{n+1} \quad \alpha \geq \alpha_{dry} \tag{3.3.2}$$

with

$$Re_\ell = \frac{(1-\alpha)\rho_\ell |U_{\ell z}| D_H}{\eta_\ell} \tag{3.3.3}$$

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

$$\alpha_{dry} = 0.957$$

Vapor wall friction: axial direction

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

$$F_{vz} = \left[\frac{0.2}{2D_H} Re_v^{-.2} \alpha\rho_v |U_{vz}|\right]^n U_{vz}^{n+1} \qquad \alpha > \alpha_{dry} \tag{3.3.4}$$

with

$$Re_v = \frac{\alpha\rho_v |U_{vz}| D_H}{\eta_v} \tag{3.3.5}$$

Interphase momentum exchange: axial direction

$$M_z = K_z^n (U_{vz} - U_{\ell z})^{n+1} \tag{3.3.6}$$

with

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

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:

$$M_z = (K_z + S)^n (U_{vz} - U_{\ell z})^{n+1} \tag{3.3.8}$$

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:

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

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 differenciates 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

$$F_{\ell r} = \left[\frac{f_{\ell r}}{2D_H} \rho_\ell \left|U^m_{\ell r}\right|\right]^n \left(U^m_{\ell r}\right)^{n+1} \tag{3.3.10}$$

where

$$f_{\ell r} = \begin{cases} \dfrac{180}{Re_{\ell r}} & Re_{\ell r} \leq 202.5 \\ 1.92\ Re_{\ell r}^{-.145} & Re_{\ell r} \geq 202.5 \end{cases} \tag{3.3.11}$$

$$Re_{\ell r} = \frac{\rho_\ell \left|U^m_{\ell r}\right| D_H}{\eta_\ell} \tag{3.3.12}$$

and $U^m_{\ell r}$ 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

$$\begin{aligned} F_{vr} &= 0 && \alpha \leq \alpha_{dry} \\ F_{vr} &= \left[\frac{f_{vr}}{2D_H} \rho_v \left|U^m_{vr}\right|\right]^n \left(U^{n+1}_{vr}\right)^{n+1} && \alpha > \alpha_{dry} \end{aligned} \tag{3.3.13}$$

with

$$f_{vr} = \begin{cases} \dfrac{180}{Re_{vr}} & Re_{vr} \leq 202.5 \\ 1.92\ Re_{vr}^{-.145} & Re_{vr} > 202.5 \end{cases} \tag{3.3.14}$$

with

$$Re_{vr} = \frac{\rho_v |U^m_{vr}| D_H}{\eta_v} \tag{3.3.15}$$

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

Interphase momentum exchange: radial direction

$$M_r = K_r^n (U^m_{vr} - U^m_{\ell r})^{n+1} \tag{3.3.16}$$

with

$$K_r = \frac{4.31}{2D_H} \rho_v |U^m_{vr} - U^m_{\ell r}| [(1-\alpha)(1+75(1-\alpha))]^{.95} \tag{3.3.17}$$

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:

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

$$\langle U_r \rangle = \frac{U^m_r A^m_r}{V} (r_{k+1} - r_k)$$

or

$$U^m_r = \frac{V}{A^m_r (r_{k+1} - r_k)} \langle U_r \rangle \tag{3.3.18}$$

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:

$$\rho C_p \frac{\partial T}{\partial t} - \frac{1}{r}\frac{\partial}{\partial r}\left(rK\frac{\partial T}{\partial t}\right) = q''' \qquad (3.4.1)$$

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:

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

Using the approximation:

$$<\rho C_p>_k = \frac{r_{k+\frac{1}{2}}^2 - r_k^2}{2}(\rho C_p)_{k+\frac{1}{2}} + \frac{r_k^2 - r_{k-\frac{1}{2}}^2}{2}(\rho C_p)_{k-\frac{1}{2}} \qquad (3.4.3)$$

and

$$\int_{r_{k-\frac{1}{2}}}^{r_{k+\frac{1}{2}}} r\rho C_p \frac{\partial T}{\partial t}dr = <\rho C_p>_k \frac{\partial T_k}{\partial t} \qquad (3.4.4)$$

Also in equation 3.4.2 we have:

$$\int_{r_{k-\frac{1}{2}}}^{r_{k+\frac{1}{2}}} [\frac{\partial}{\partial r}rK\frac{\partial T}{\partial t}]dr = [rK\frac{\partial T}{\partial r}]_{r_{k-\frac{1}{2}}}^{r_{k+\frac{1}{2}}}$$

$$= (rK)_{k+\frac{1}{2}} \frac{T_{k+1} - T_k}{\Delta r_{k+\frac{1}{2}}} - (rK)_{k-\frac{1}{2}} \frac{T_k - T_{k-1}}{\Delta r_{k-\frac{1}{2}}} \qquad (3.4.5)$$

Finally, the right hand side of equation 3.4.2 becomes:

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

and the difference equation corresponding to equation 3.4.2 becomes:

$$\langle \rho C_p \rangle_k^n \left(\frac{T_k^{n+1} - T_k^n}{\Delta t}\right) - \left(\frac{rK}{\Delta r}\right)_{k+\frac{1}{2}}^n (T_{k+1}^{n+1} - T_k^{n+1}) + \left(\frac{rK}{\Delta r}\right)_{k-\frac{1}{2}}^n (T_k^{n+1} - T_{k-1}^{n+1})$$

$$= \left[\frac{r_{k+\frac{1}{2}}^2 - r_k^2}{2} q'''_{k+\frac{1}{2}} + \frac{r_k^2 - r_{k-\frac{1}{2}}^2}{2} q'''_{k-\frac{1}{2}}\right]^n \qquad (3.4.7)$$

There are four locations where equation 3.4.7 must be modified to accomodate 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:

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

with

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

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:

$$\langle \rho C_p \rangle_N \left(\frac{T_N^{n+1} - T_N^n}{\Delta t}\right) + \left(\frac{rK}{\Delta r}\right)_{N-\frac{1}{2}}^n (T_N^{n+1} - T_{N-1}^{n+1}) + q'' r_N$$

$$= \frac{r_N^2 - r_{N-\frac{1}{2}}^2}{2} q'''_{N-\frac{1}{2}} \qquad (3.4.10)$$

with

$$\langle \rho C_p \rangle_N = \frac{r_N^2 - r_{N-\frac{1}{2}}^2}{2} (\rho C_p)^n_{N-\frac{1}{2}} \qquad (3.4.11)$$

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:

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

where

$T_w = T_N$ = outside clad temperature

T_ℓ, T_v, T_s = liquid, vapor and saturation temperatures

h_ℓ, 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:

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

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 $\mathbb{A}$) by a product:

$$\mathbb{A} = \mathbb{C} \times B$$

with

$$\mathbb{C} = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 & 0 \\ C_{21} & 1 & 0 & \cdots & 0 & 0 \\ 0 & C_{31} & 1 & 0 & \cdots & \\ \vdots & 0 & & & & \\ \vdots & \vdots & & & & \\ 0 & 0 & \cdots & & C_{N,N-1} & 1 \end{bmatrix}$$

and

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

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

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

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

$$\vdots$$

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

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

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

Now define a vector X such that:

$$\mathbb{F} = \mathbb{C} \times X$$

where $\mathbb{F}$ 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:

$$\mathbb{B} \times \mathbb{T} = X \qquad (3.4.14)$$

where $\mathbb{B}$ is an upper triangular matrix, and once we have gotten the value of $T^{n+1}_{N,N}$, 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:

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

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:

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

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

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:

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

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

with

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

θ_{pu} = fraction of PuO_2 in the mixed oxide fuel

θ_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 /:

$$h_{Gap} = h_c + h_r \tag{3.4.20}$$

with

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

and

$$h_c = [\frac{dg + 1.32 \times 10^{-4}}{Cg} + 0.61 \times 10^{-4}]^{-1} + 1.8 \times 10^3 \tag{3.4.22}$$

with

$$Cg = 15. \times 2^{dil}$$

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:

$$h_{NB} = 0.00122 \frac{K_\ell^{.79} C_{p\ell}^{.45} \rho_\ell^{.49} \Delta p^{.75} S_f}{\sigma^{.5} \eta_\ell^{.29}} \left(\frac{\Delta T}{h_{\ell g} \rho_v}\right)^{.24} \qquad (3.4.23)$$

where S_f the suppression factor defined as:

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

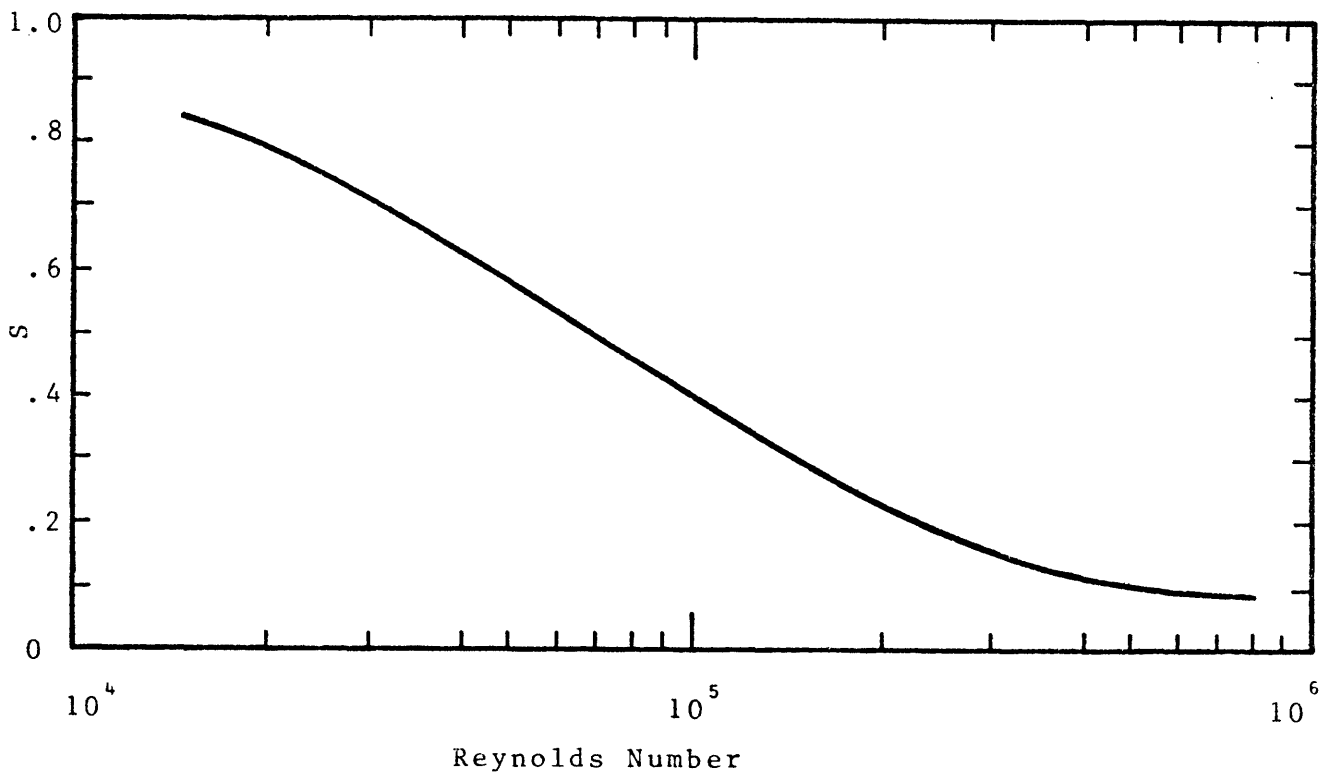

Figure 3.4 Supression Factor vs. Reynolds Number

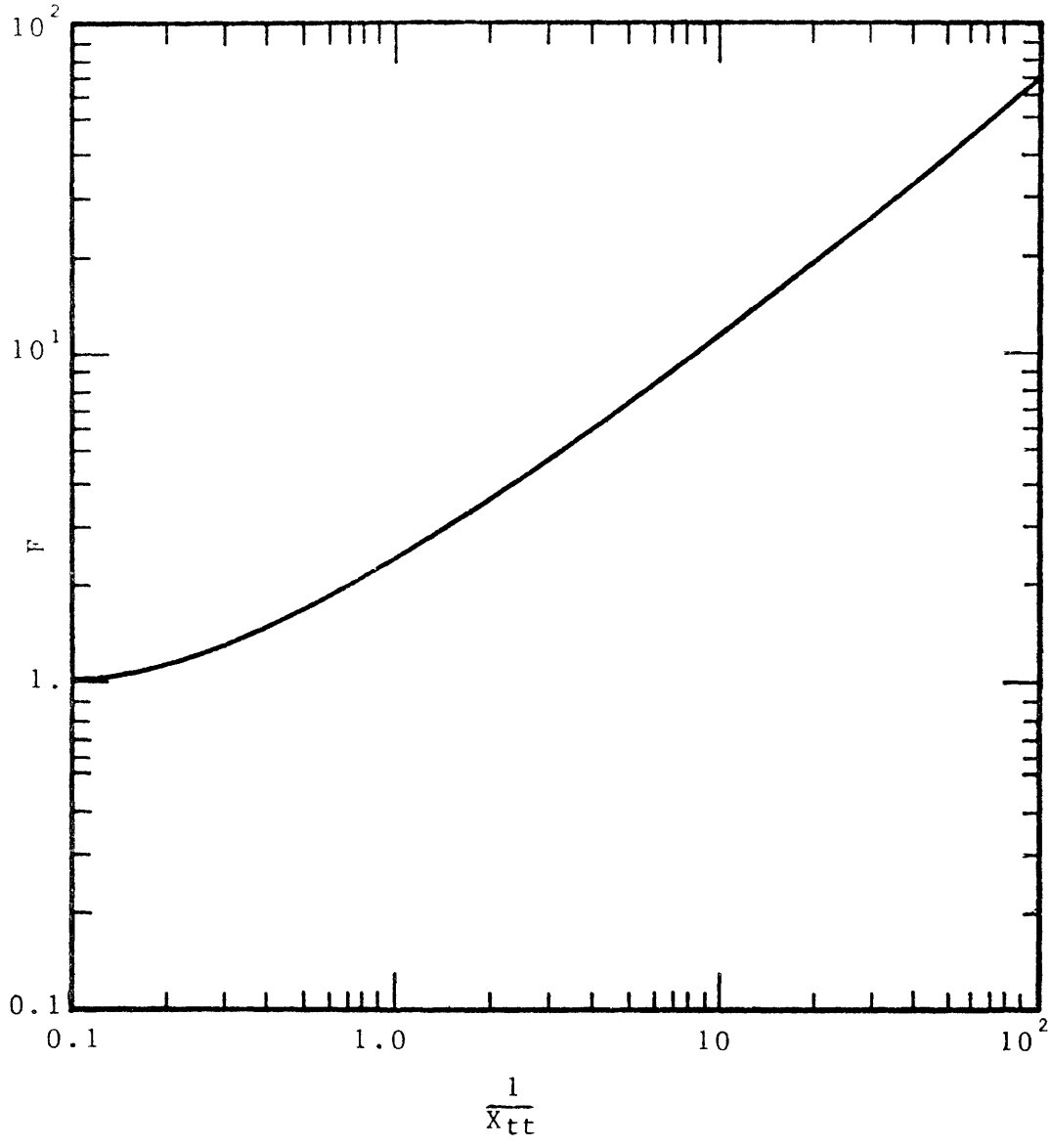

Figure 3.5 The Reynolds Number Factor

with

ΔT_e = effective superheat for bubble growth in annular liquid

ΔT = difference between wall temperature and saturation temperature

Δp = 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:

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

and the two phase Reynolds number is defined as:

$$Re_{TP} = F^{1.25}\ \frac{(1-\alpha)\rho_\ell U_\ell D_H}{\eta_\ell} \qquad (3.4.25)$$

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

$$\begin{aligned} F &= 2.35(.213 + \frac{1}{X_{tt}})^{.736} && X_{tt} < 10. \\ F &= 1.0 && X_{tt} \geq 10. \end{aligned} \qquad (3.4.26)$$

and

X_{tt} is the Martinelli parameter:

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

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

$$h_\ell = F^{.375} h_{\ell sp} \tag{3.4.28}$$

whre F is the same Reynolds number factor used for the microscopic heat tranfer 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

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

with

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

and Pe is the Perclet number = RePr

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

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

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 accomodate changes which would consider a heat sink outside the hex can.

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

$$(\rho C_p)_c \left(\frac{T_c^{n+1} - T_c^n}{\Delta t}\right) + h_\ell^n (T_c^{n+1} - T_\ell^{n+1}) + h_v^n (T_c^{n+1} - T_v^{n+1})$$

$$+ h_{NB}^n (T_c^{n+1} - T_s^{n+1}) = 0 \qquad (3.4.32)$$

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:

$$q_{\ell v} = S_e^{n+1} h_{v s}^{n+1} - S_c h_{\ell s}^{n+1} + HA(T_\ell^{n+1} - T_v^{n+1}) \qquad (3.4.33)$$

where

S_e = evaporation rate

S_c = condensation rate

$h_{v s}$ = enthalpy for the saturated vapor

$h_{\ell s}$ = 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_{ℓ} = liquid thermal conductivity

D_H = hydraulic diamter

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 says 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:

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

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 once 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

Geometry

		(British Units)
Number of Pins	37	.1945 in
Fuel Pellet OD (m)	4.94×10^{-3}	
Clad OD (m)	5.842×10^{-3}	.230 in
Clad ID (m)	5.080×10^{-3}	.200 in
Wire Wrap OD (m)		
inner pins	1.422×10^{-3}	.056 in
outer pins	7.11×10^{-4}	.028 in
Flat to Flat (m)	4.501×10^{-2}	1.772 in
Duct Wall Thickness (m)	3.048×10^{-3}	.12 in
Length of Fuel		36.0 in
Inlet to Bottom of Fuel (m)	1.857×10^{-1}	7.31 in
Top of Fuel to End Cap (m)	5.334	210.0 in
Wire Wrap Lead (m)	3.048×10^{-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

Pin Number*	Power Factor
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

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

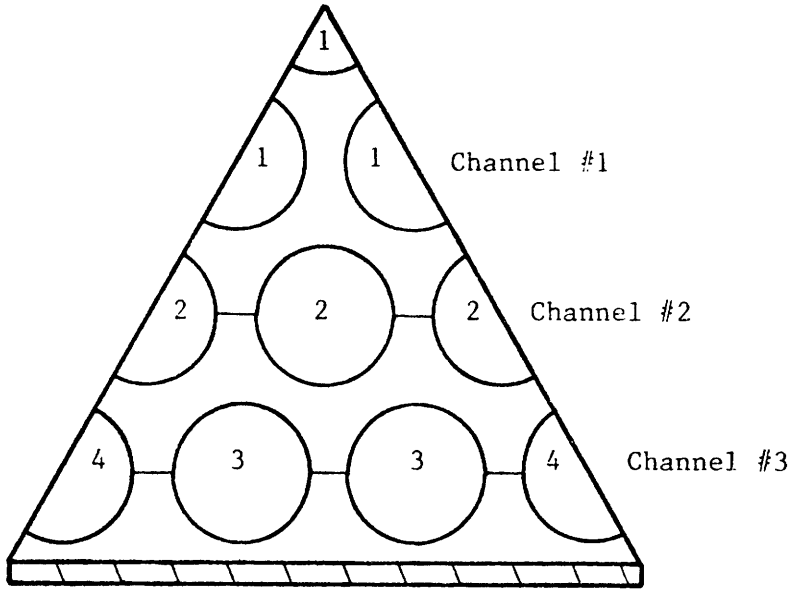

Figure 4.1

Pin Number Location

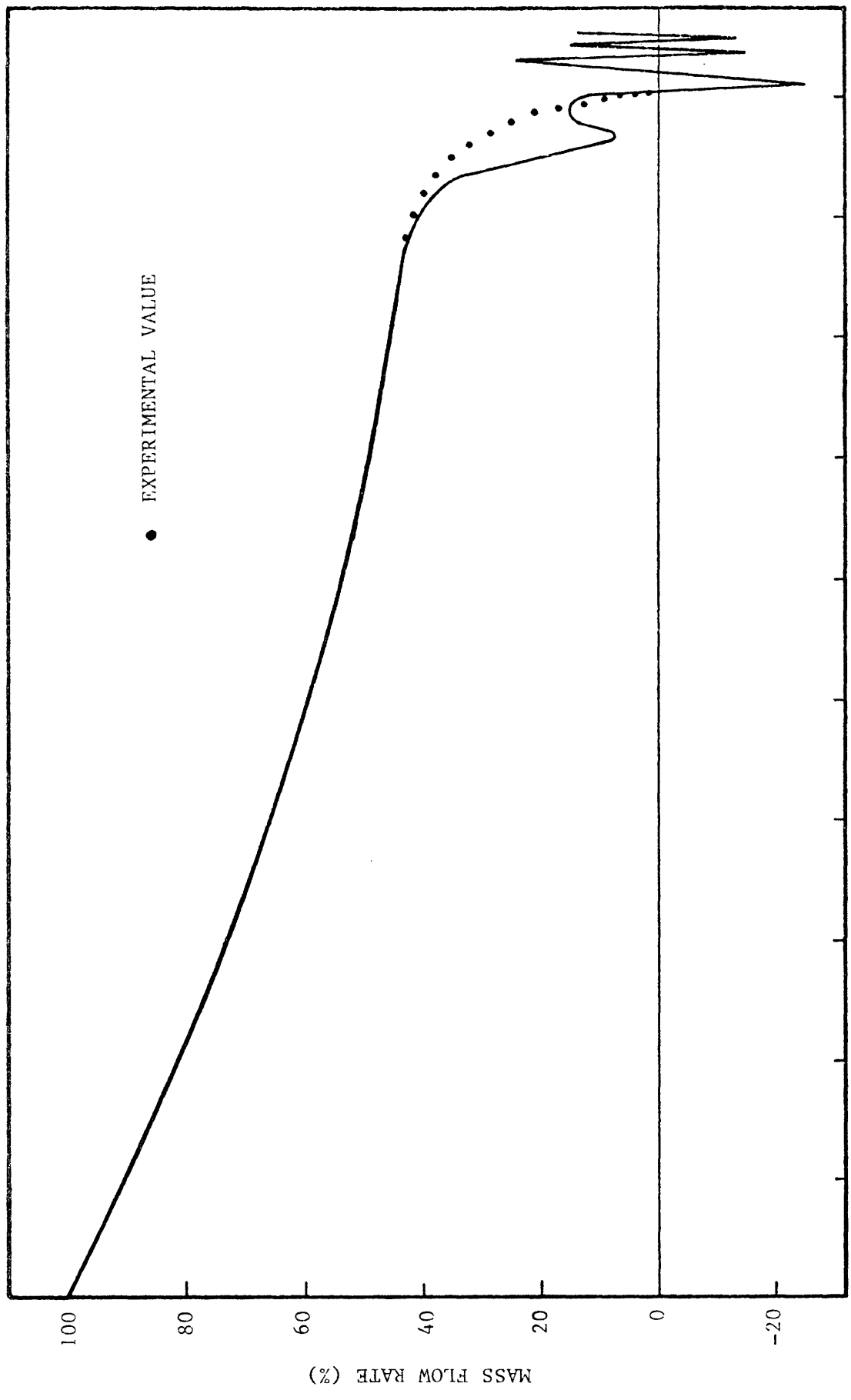

Figure 4.2: P3A: Mass Flow Rate Vs. Time

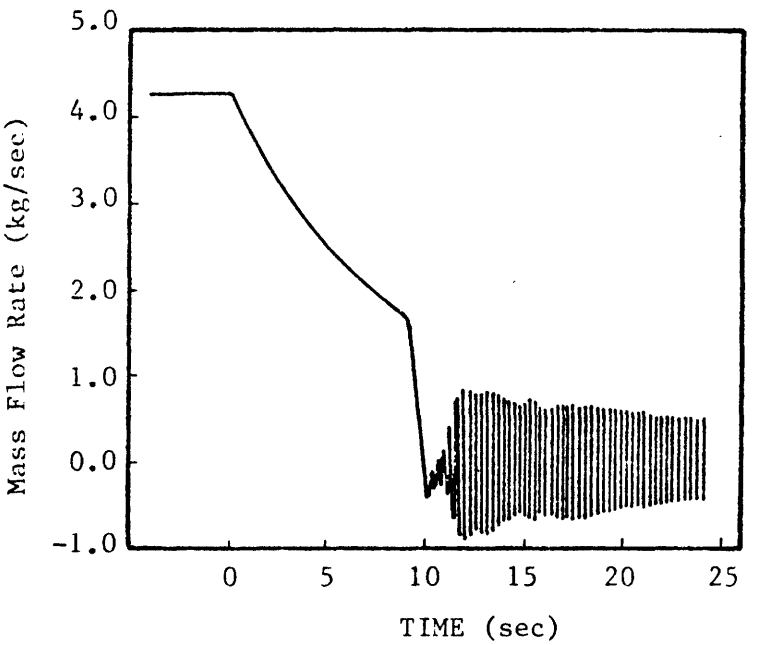

Figure 4.3

Experimental Inlet Mass Flow Rate

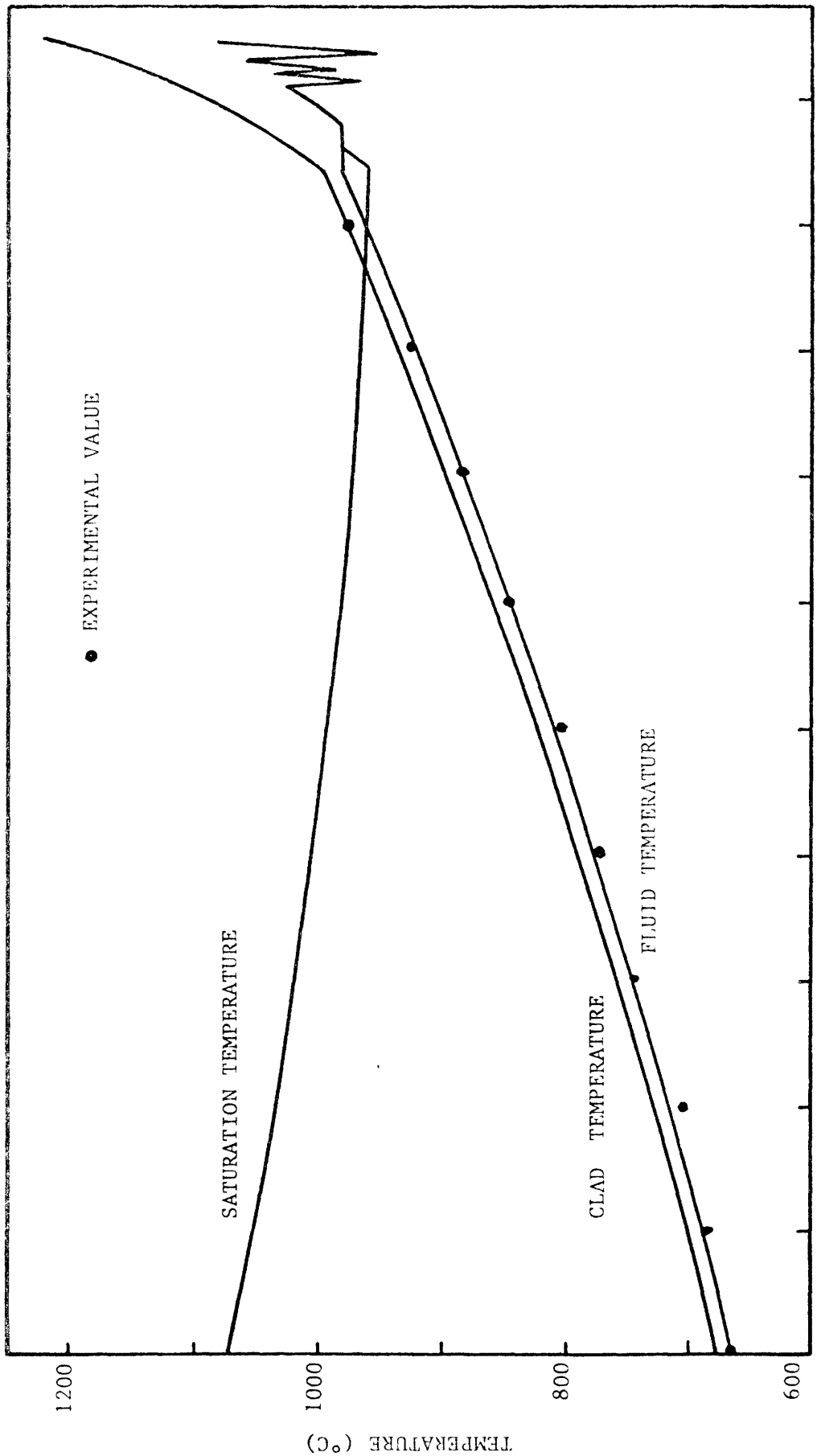

Figure 4.4: P3A - Temperature Vs. Time: Central Channel

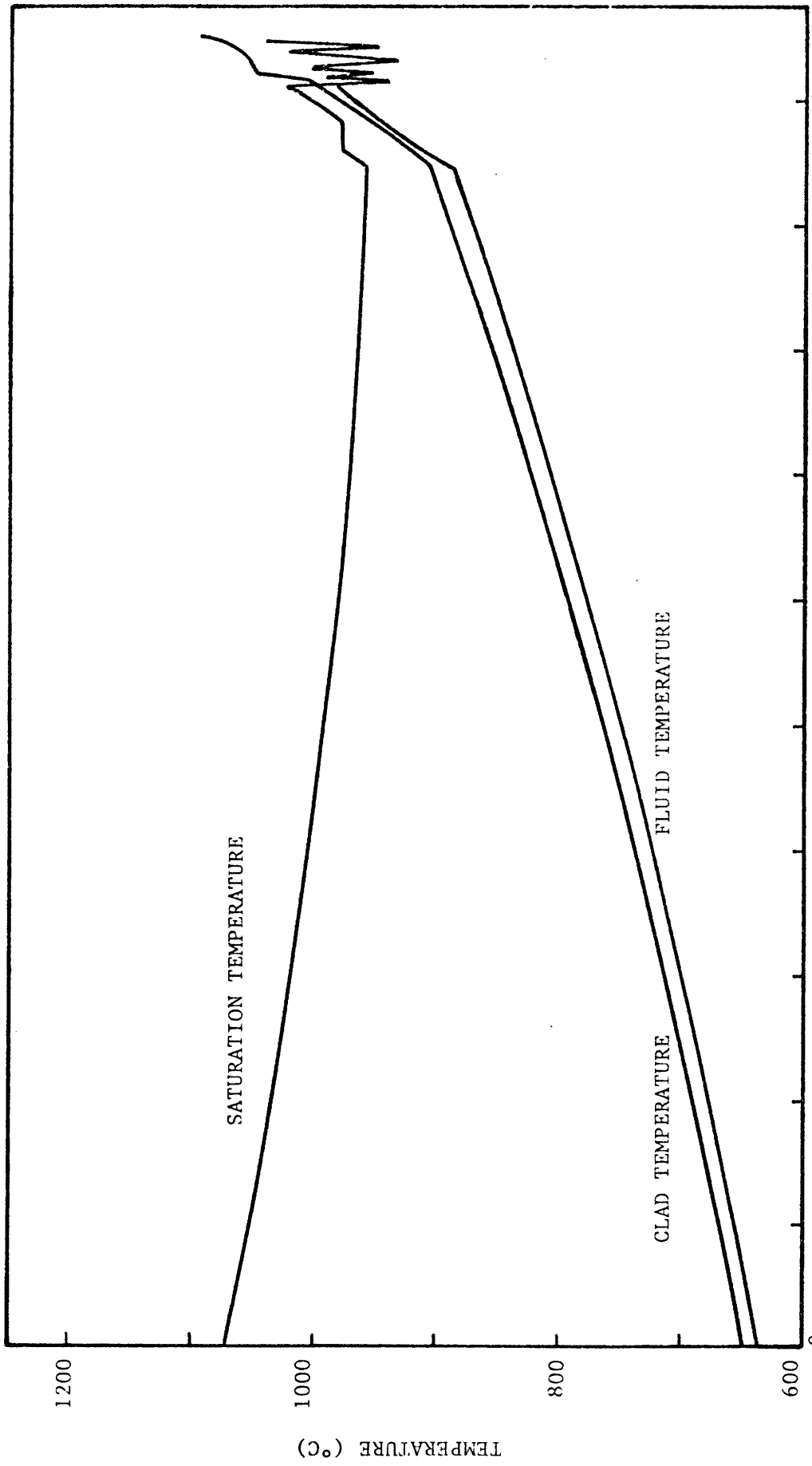

Figure 4.5: P3A — Temperature Vs. Time Edge Channel

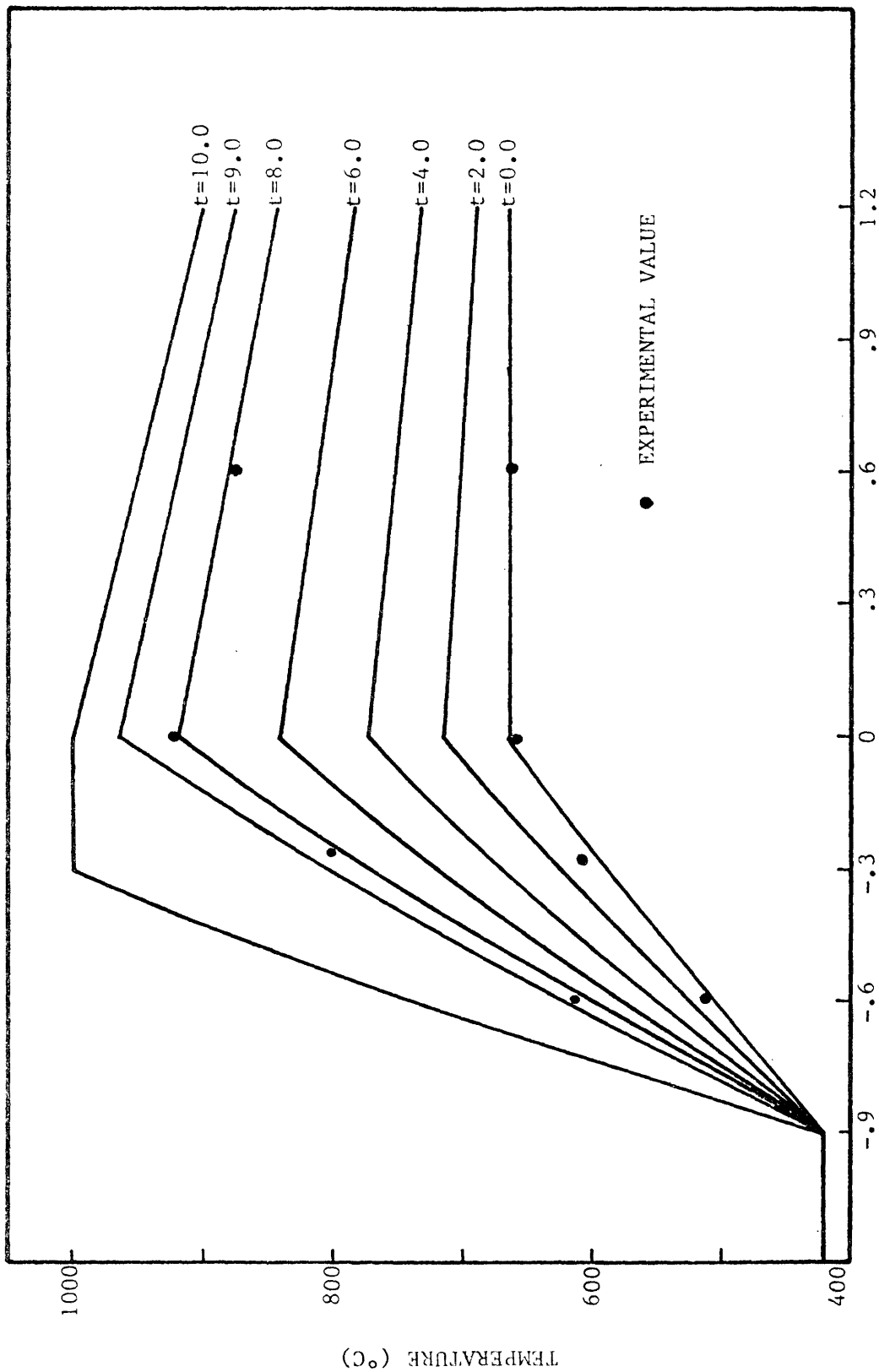

Figure 4.6: P3A: Axial Temperature Profile

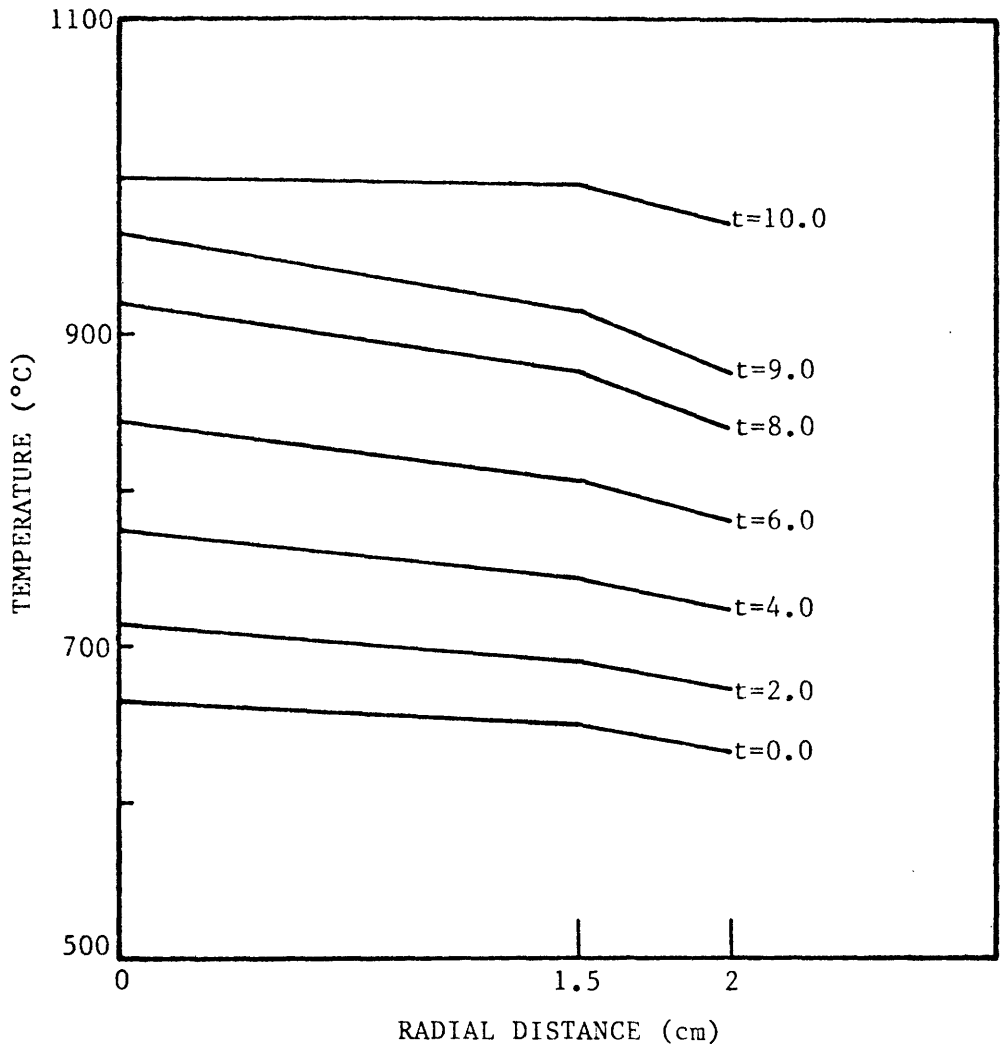

Figure 4.7

P3A: Radial Temperature Profile

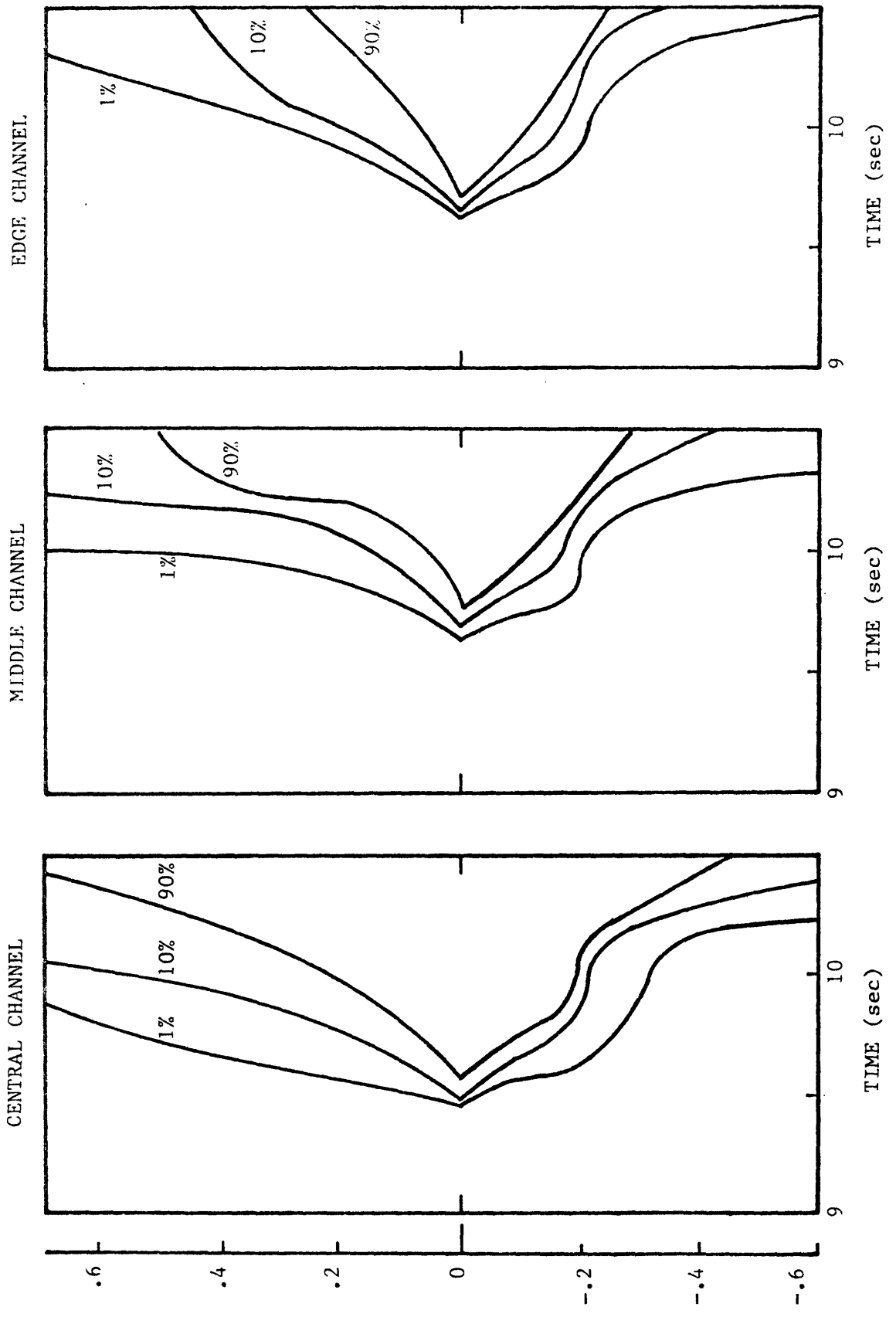

Figure 4.8: Void Fraction Maps for the P3A Experiments

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 an 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.

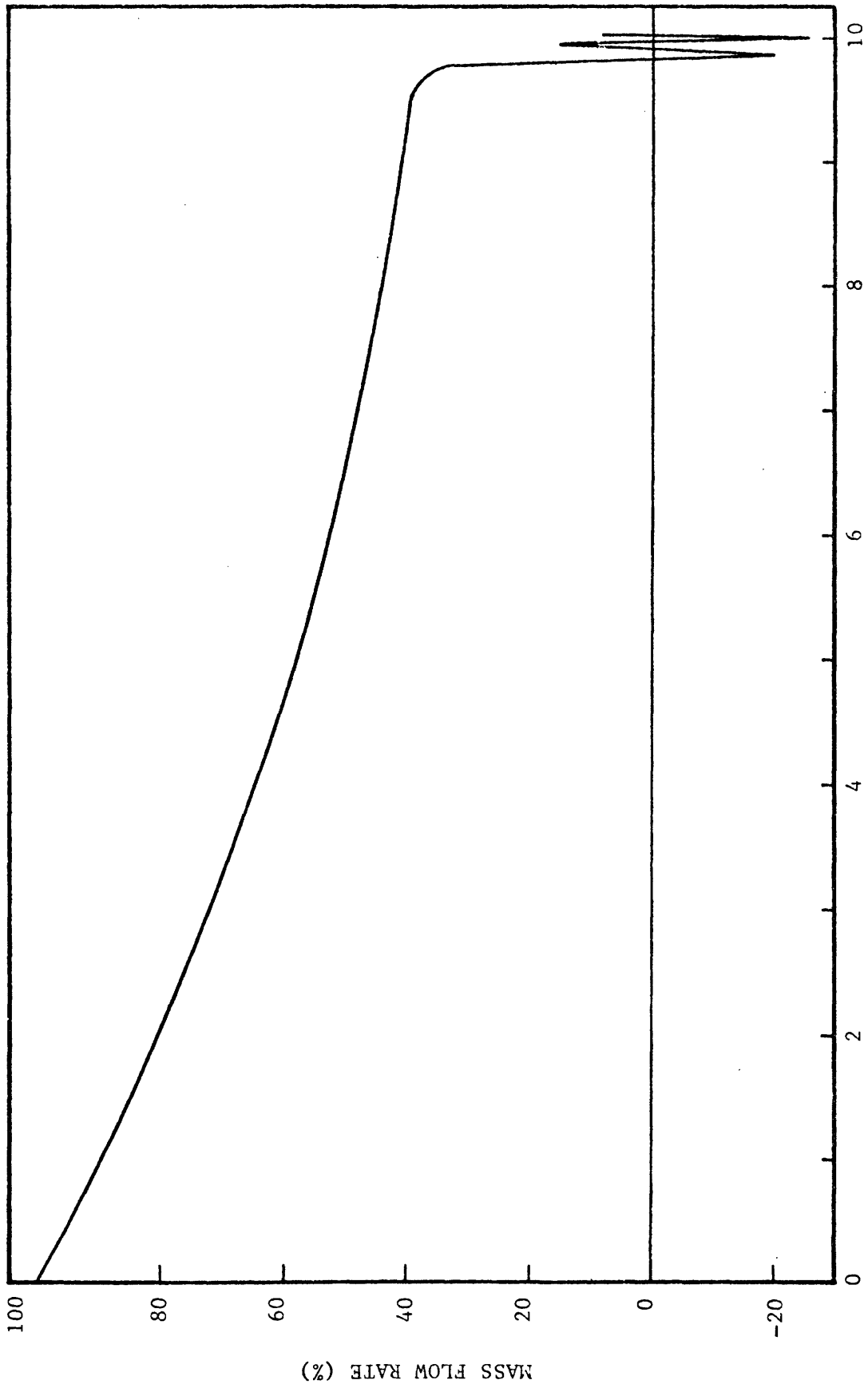

Figure 4.9: P3A — 1D: Mass Flow Rate Vs. Time

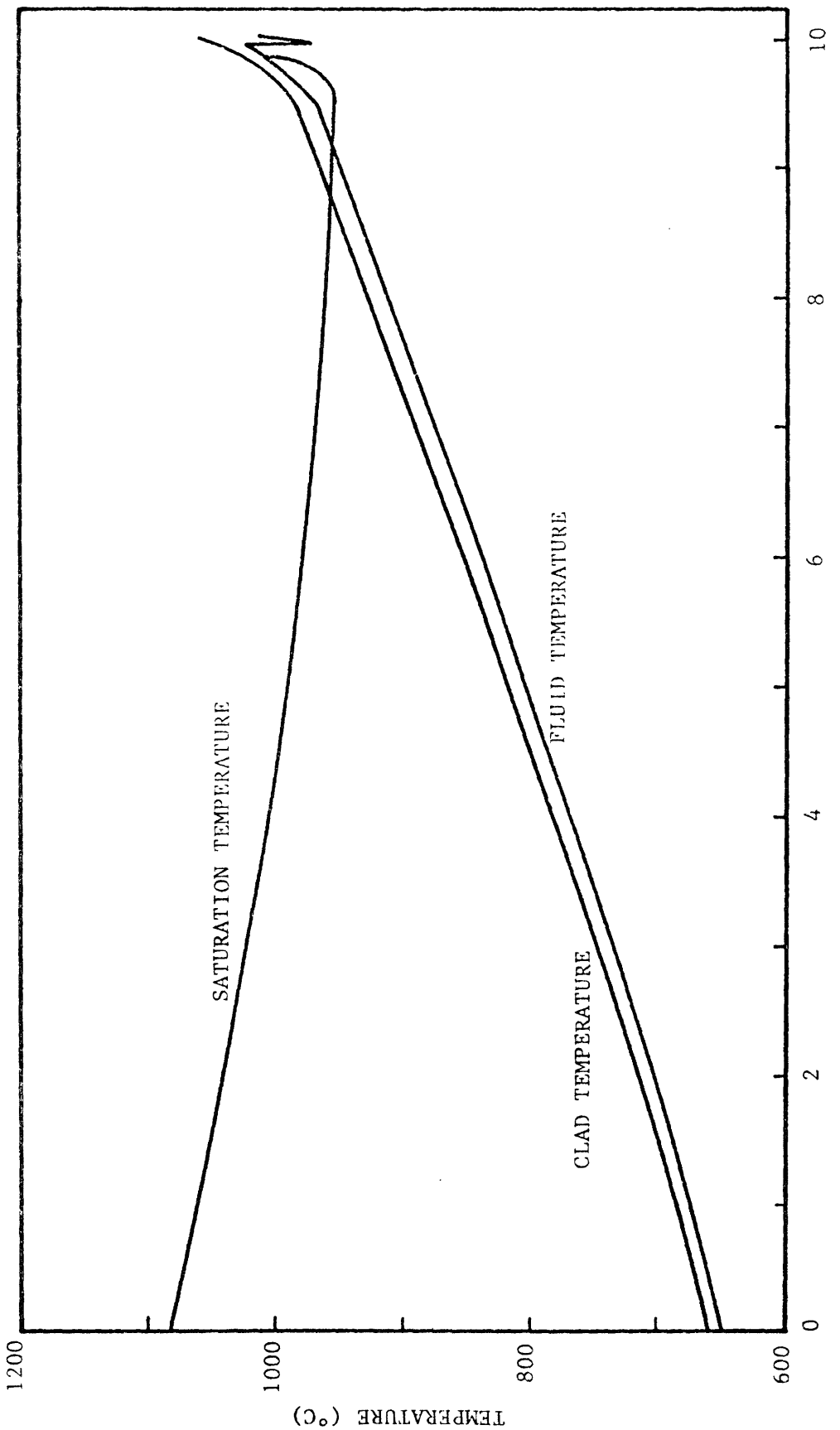

Figure 4.10: P3A — 1D: Temperature Vs. Time

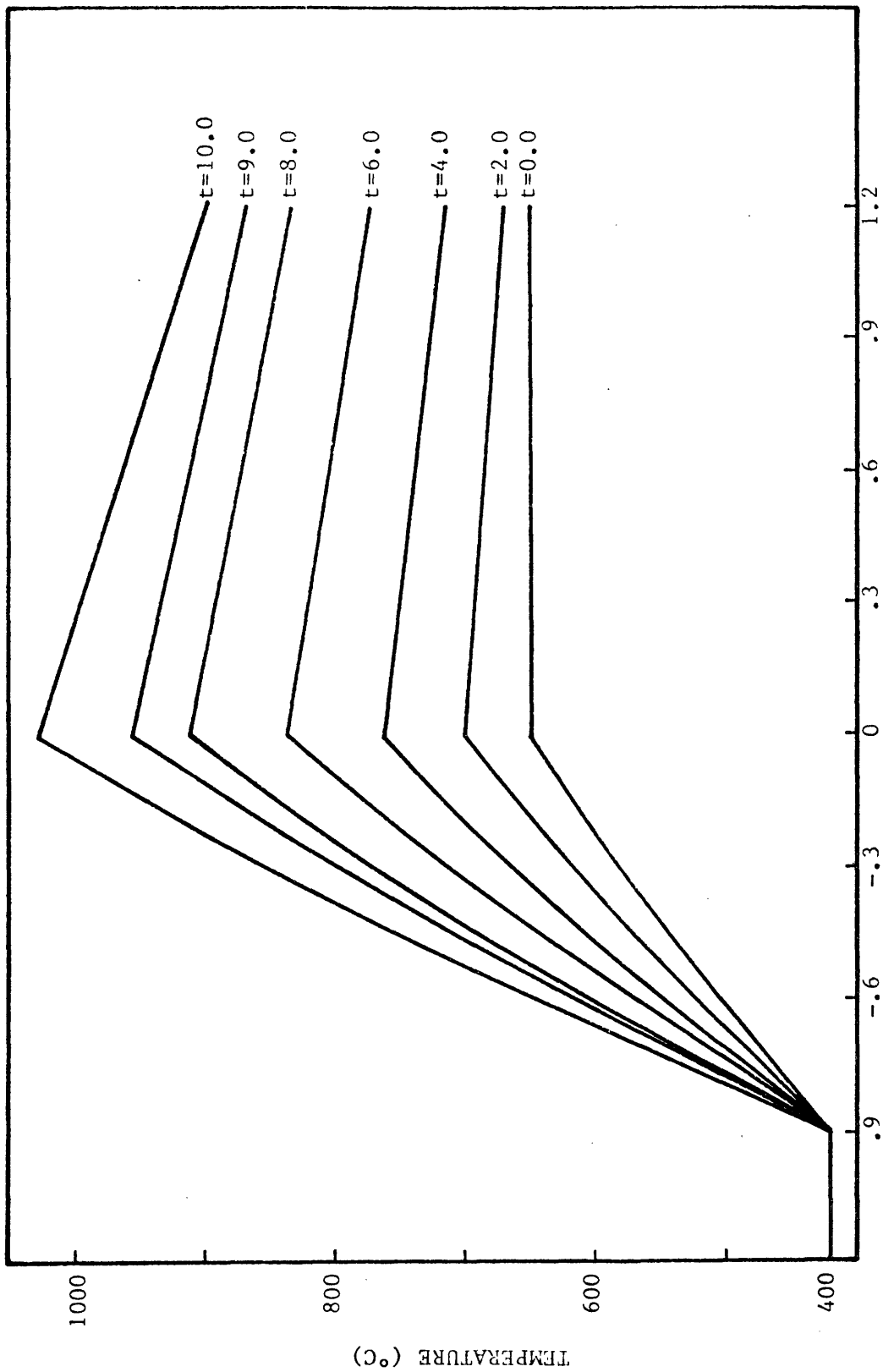

Figure 4.11: P3A-1D: Temperature Profiles

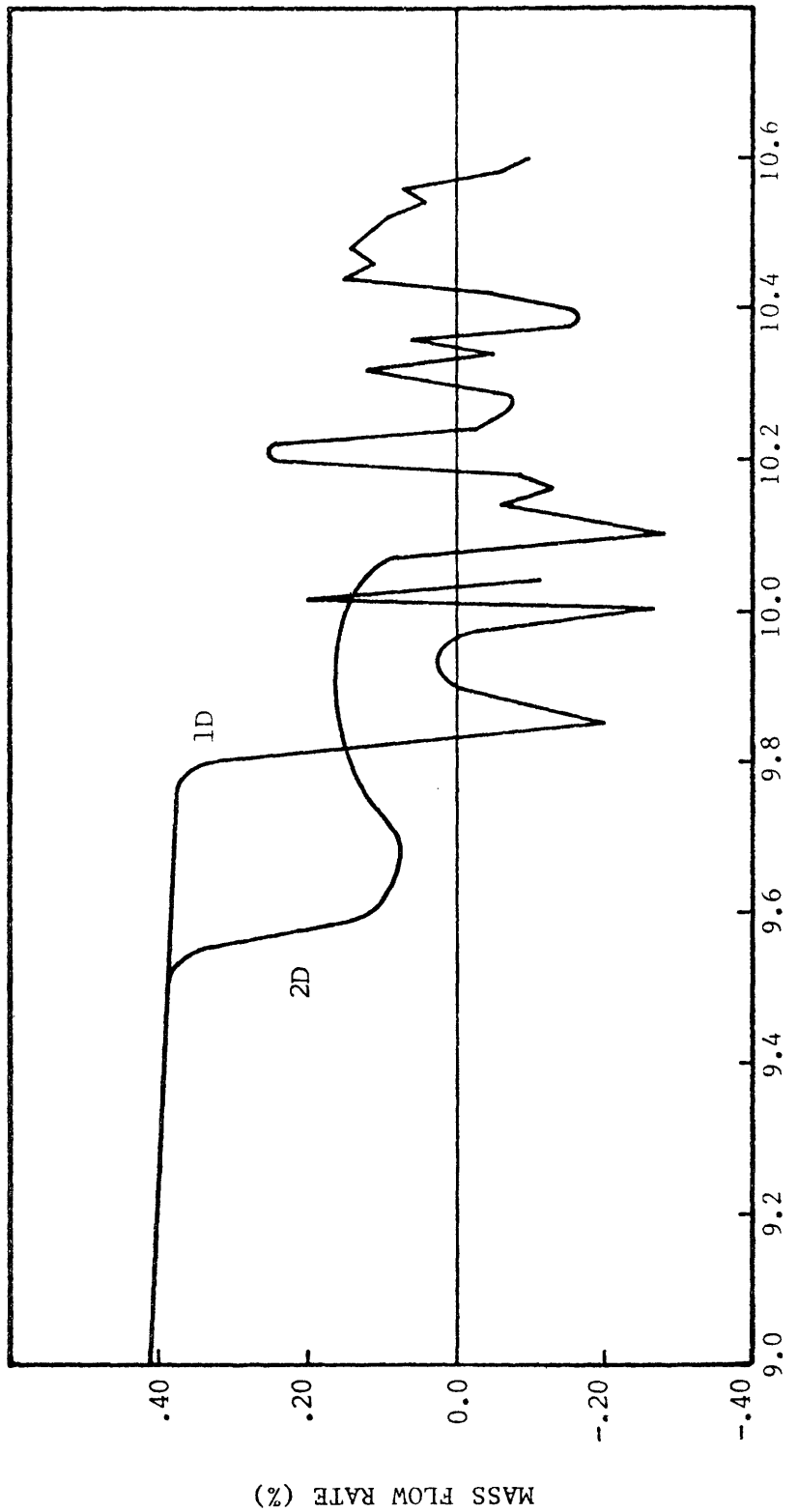

Figure 4.12: P3A: Comparison Between 1D and 2D: Mass Flow Rate

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 Data

Geometry

Number of Pins	19	
Fuel Pellet OD (m)	4.94×10^{-3}	.1945 in
Clad OD (m)	5.842×10^{-3}	.230 in
Clad ID (m)	5.030×10^{-3}	.200 in
Wire Wrap OD (m)		
inner pins	1.422×10^{-3}	.056 in
outer pins	7.11×10^{-4}	.028 in
Flat to Flat (m)	3.26×10^{-2}	1.283 in
Duct Wall Thickness (m)	1.016×10^{-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

	Percentage of Full Flow	Δtz	Test Sequence
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

	Percentage of Full Flow	Δtz	Test Sequence
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

	Percentage of Full Flow	Δtz	Test Sequence
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′

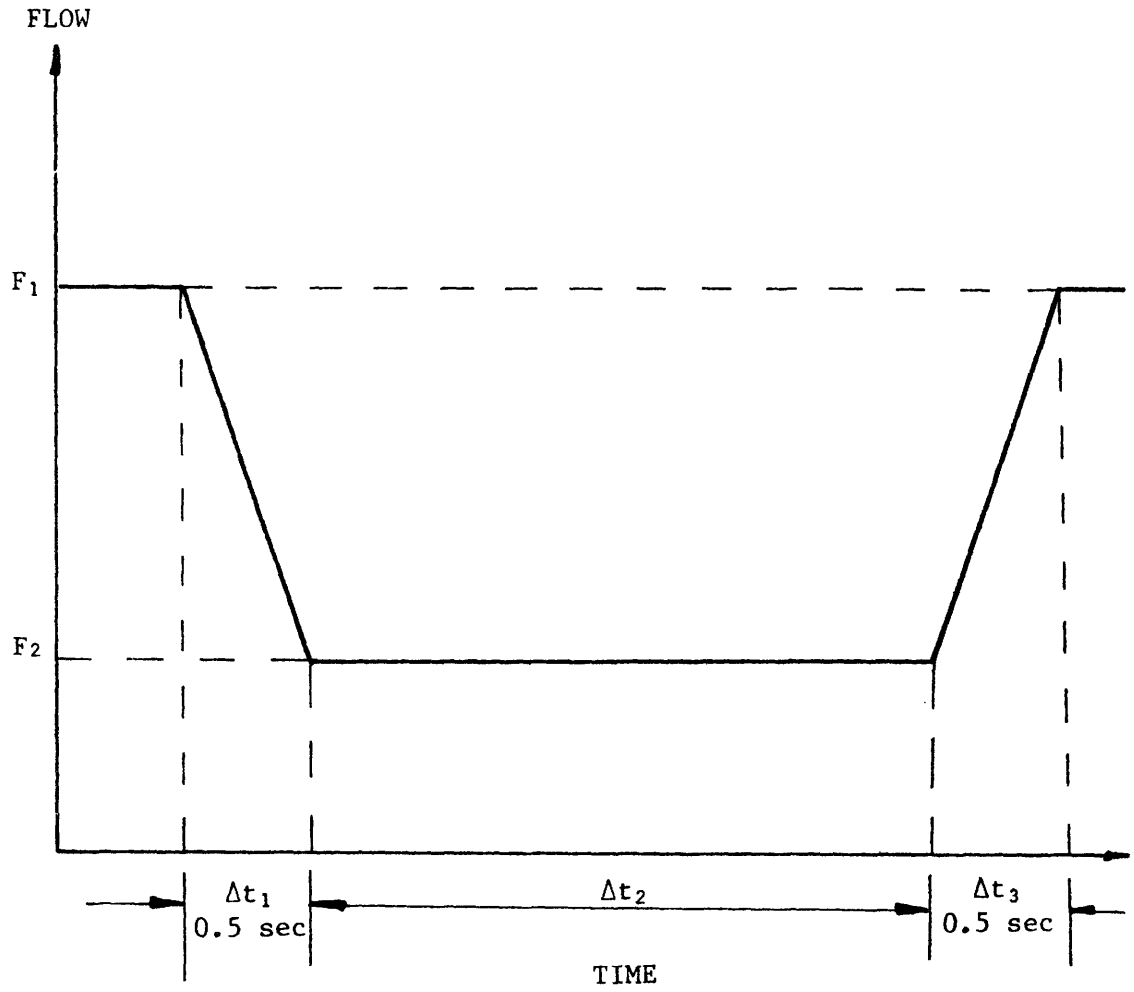

Figure 4.13

Typical Boiling Window Flow Decay for the W1 Test

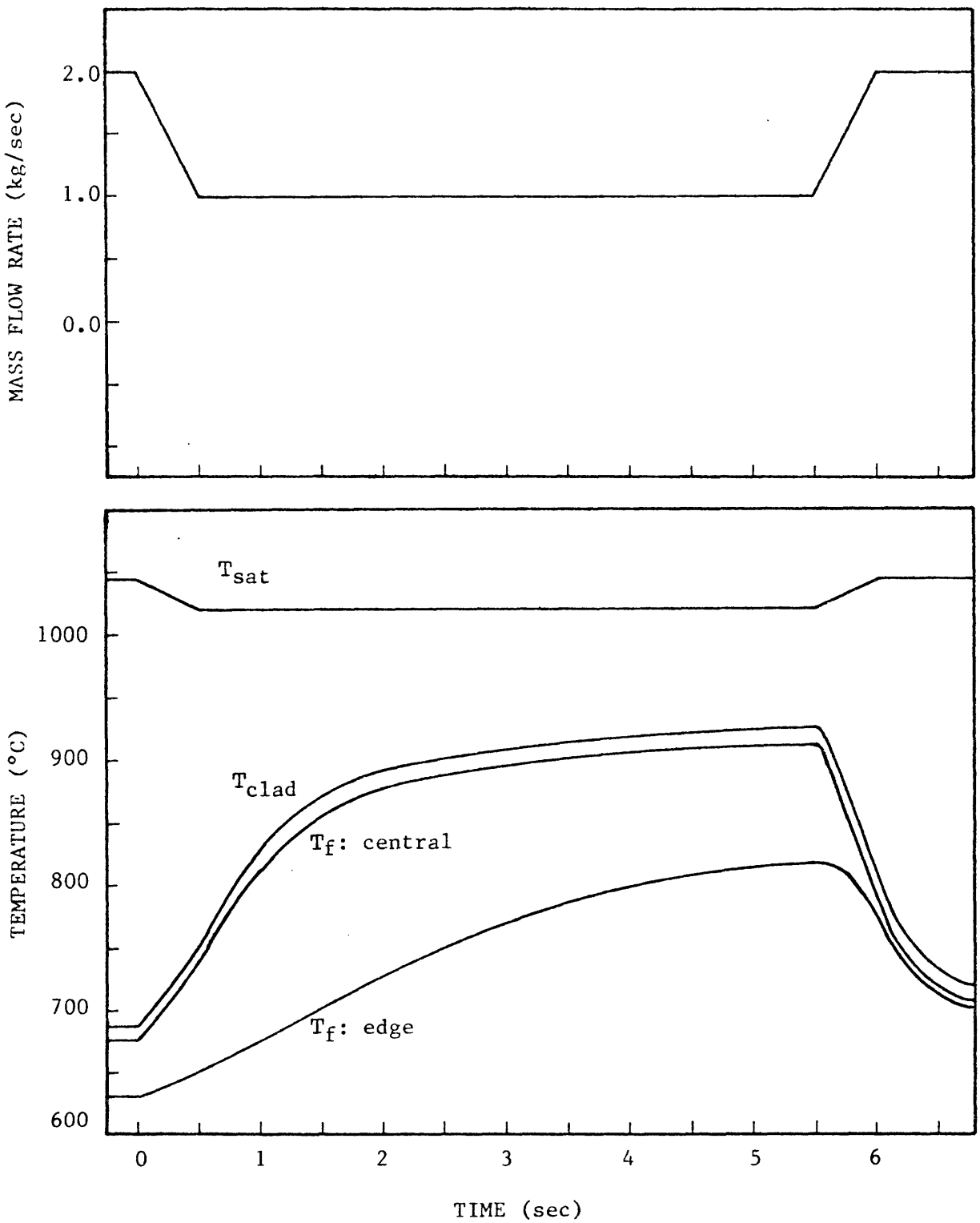

Figure 4.14: W1: Temperatures and Mass Flow Rate for the Sequence 5

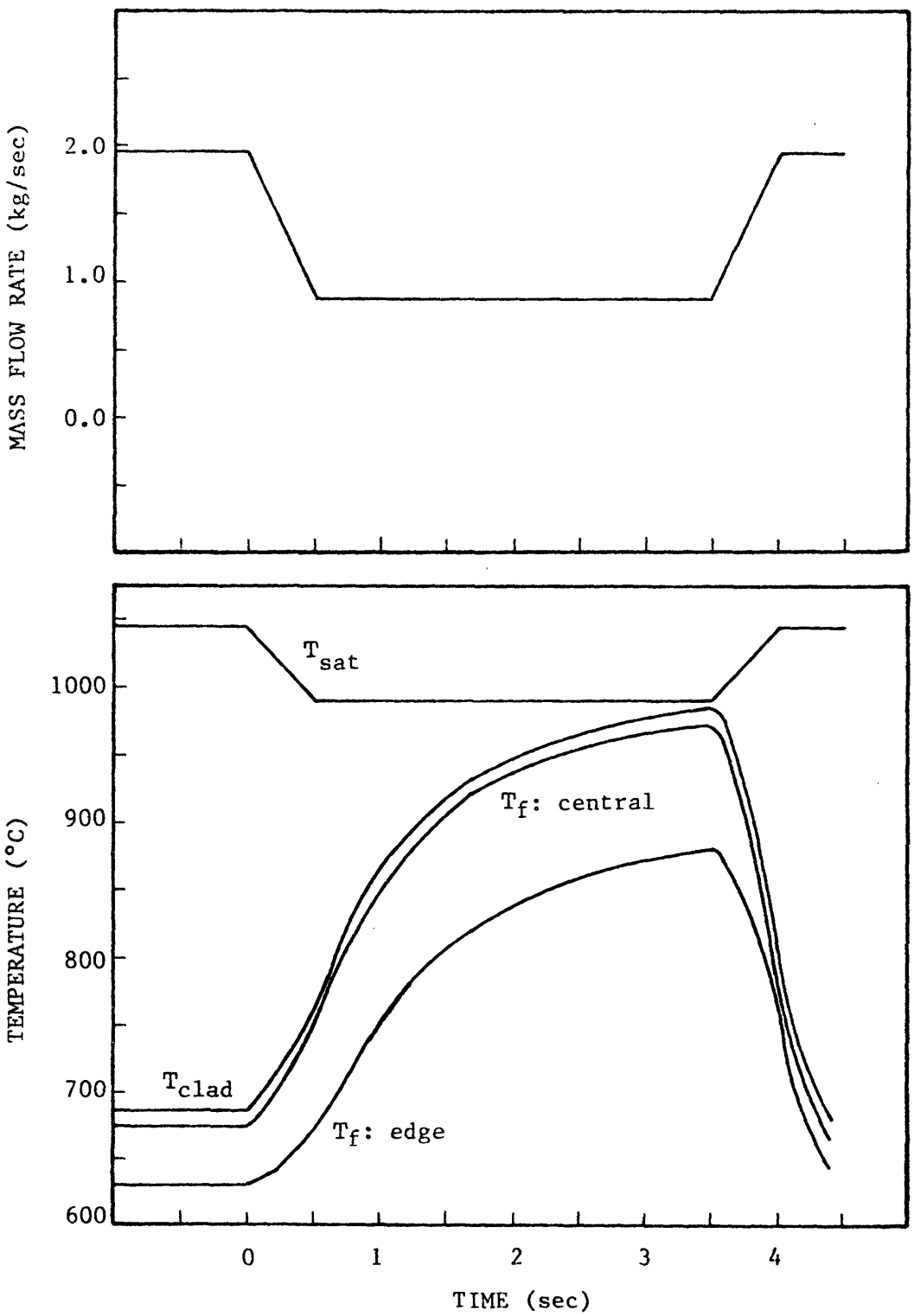

Figure 4.15: W1: Temperature and Mass Flow Rate for Sequence 6

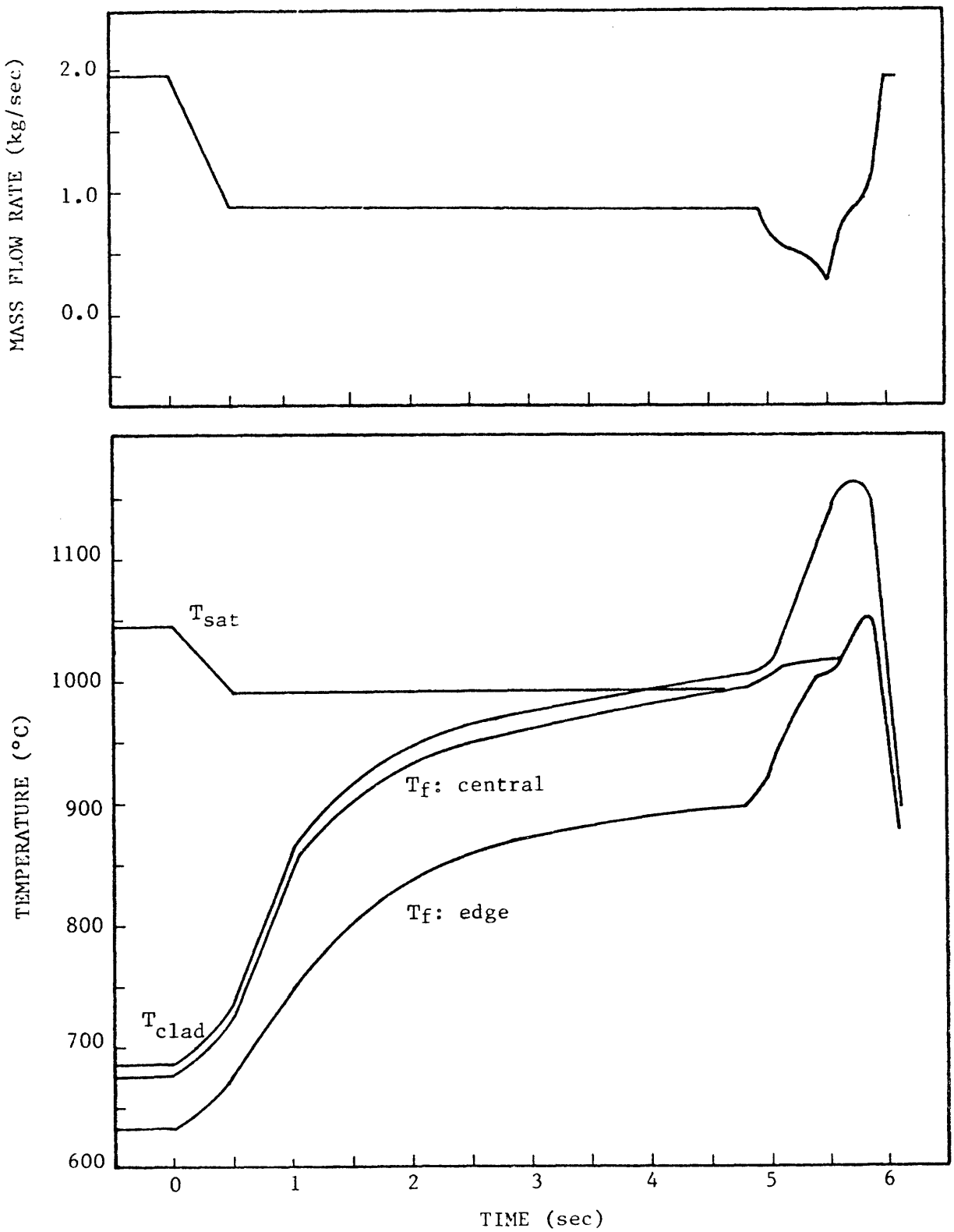

Figure 4.16: W1: Temperatures and Mass Flow Rate for the Sequence 6a

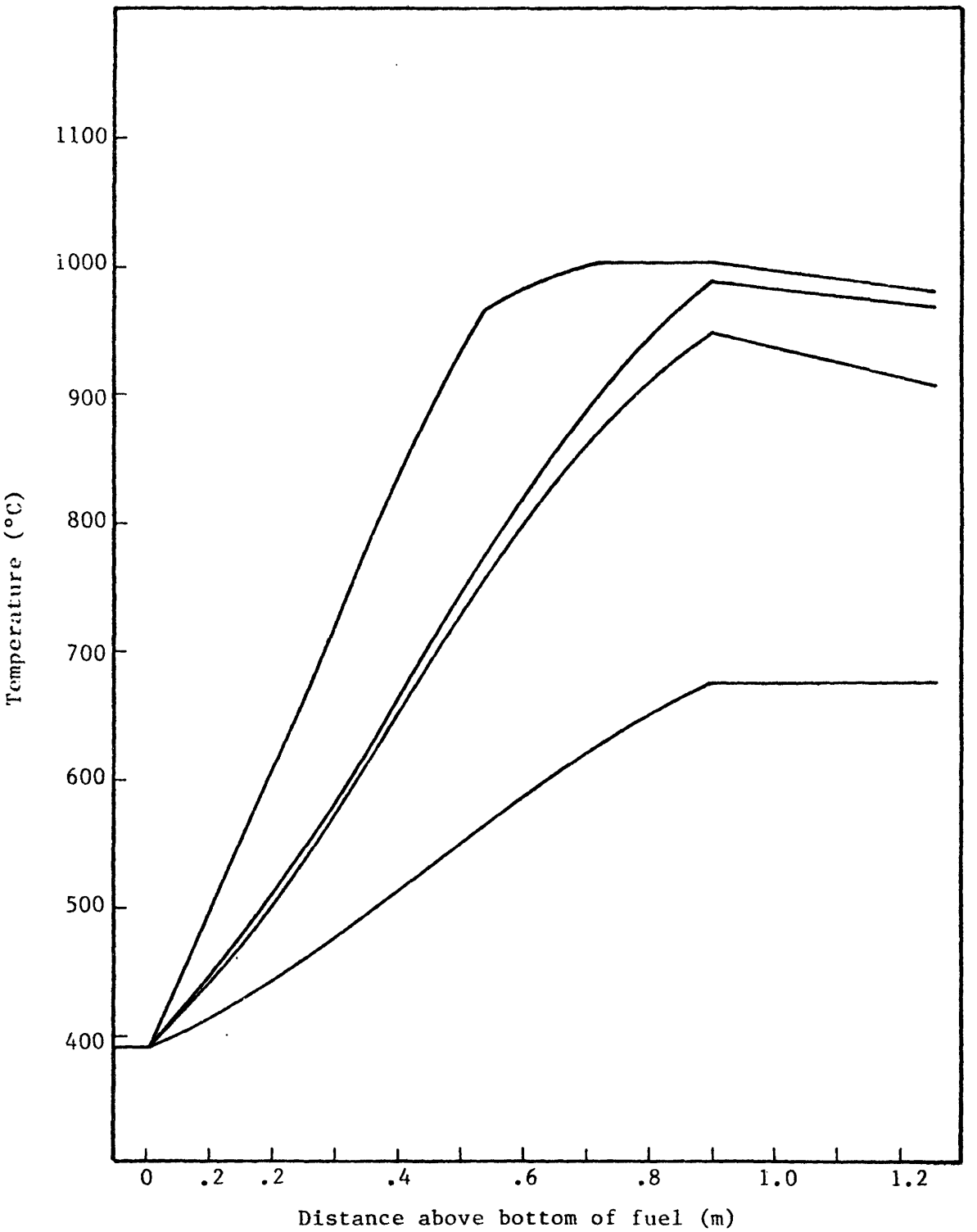

Figure 4.17: W1: Axial Temperature Profile for Sequence 6a

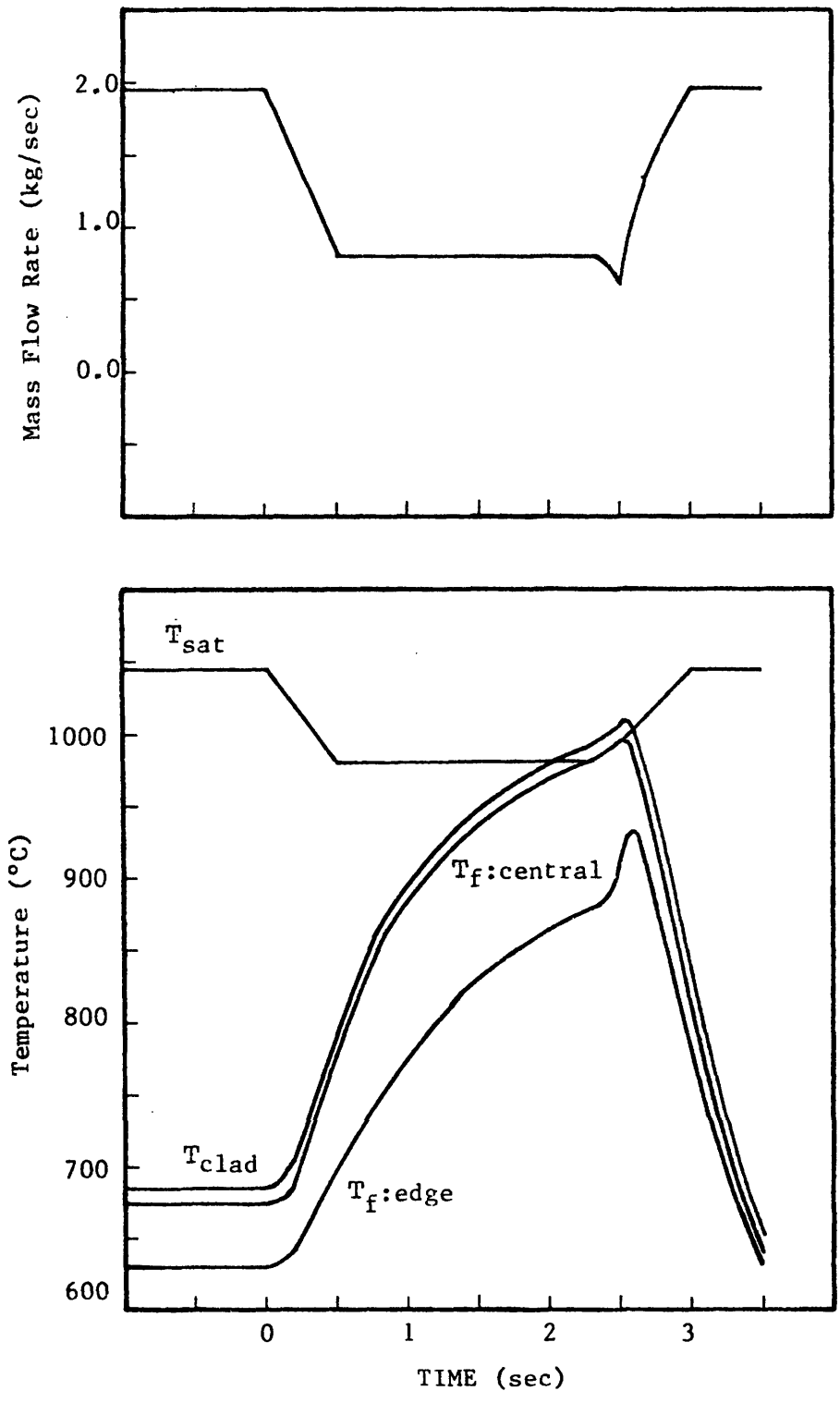

Figure 4.18: W1: Temperature and Mass Flow Rate for Sequence 7

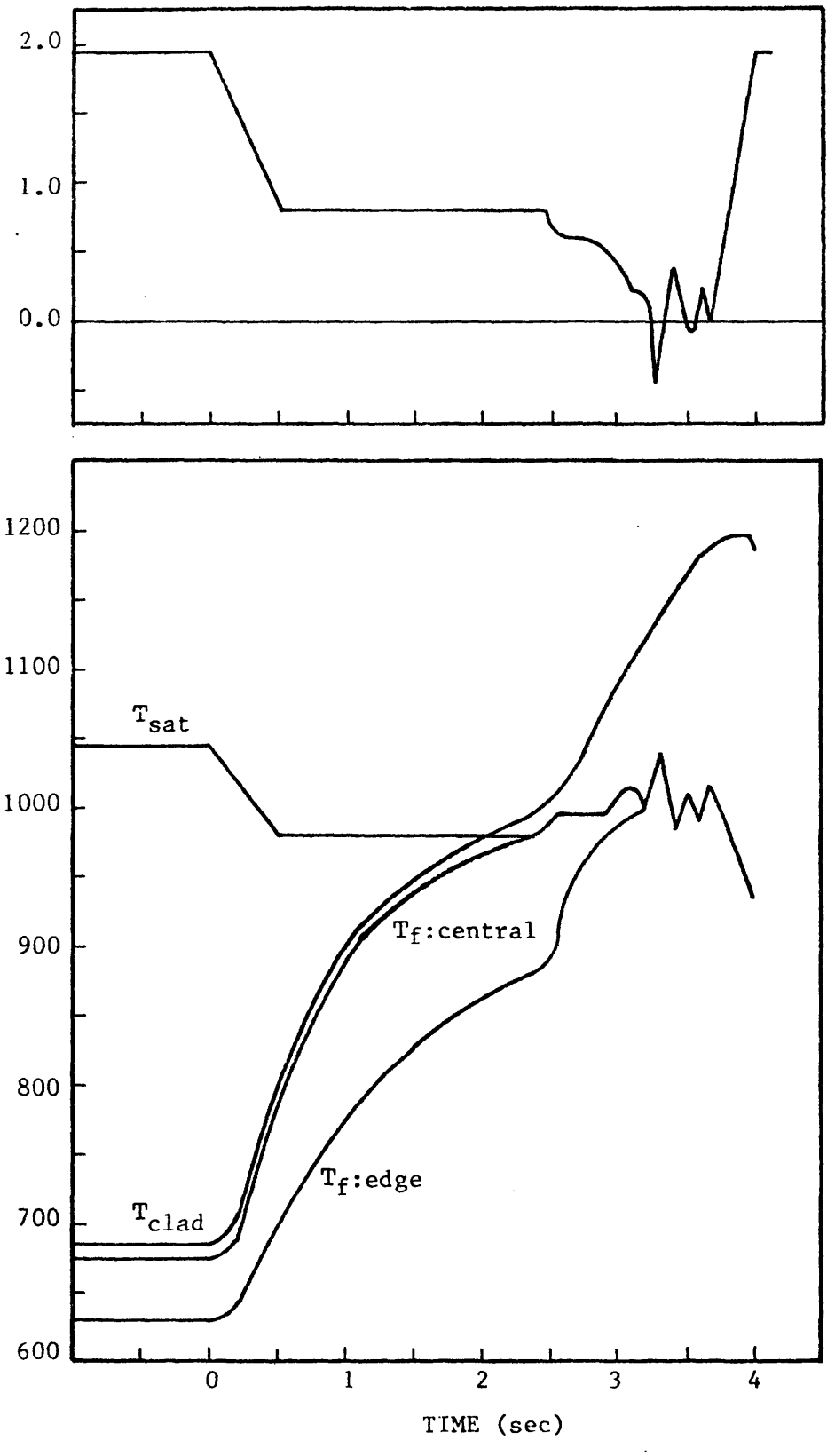

Figure 4.19: W1: Temperature and Mass Flow Rate for Sequence 7a

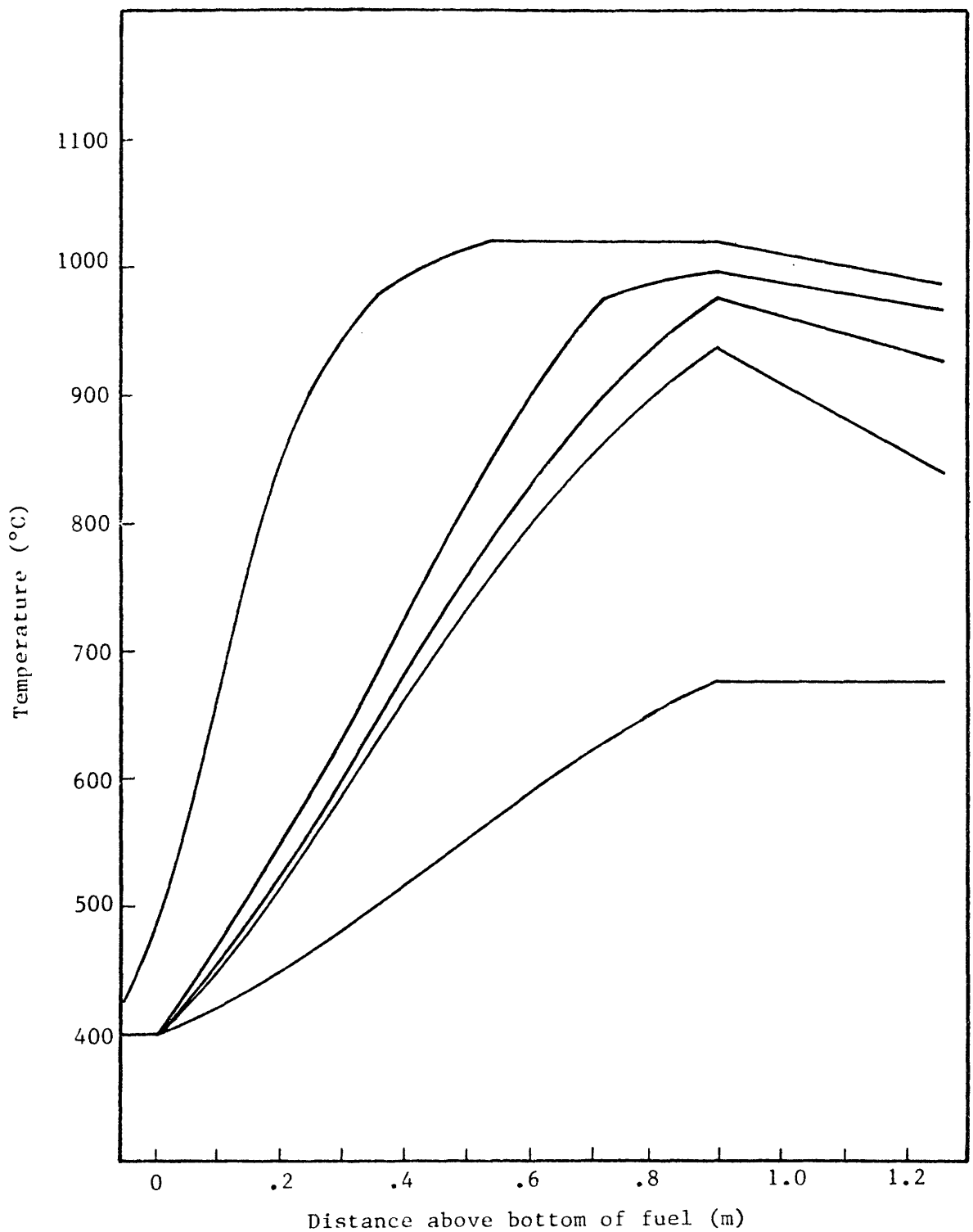

Figure 4.20: W1: Axial Temperature Profile for Sequence 7a

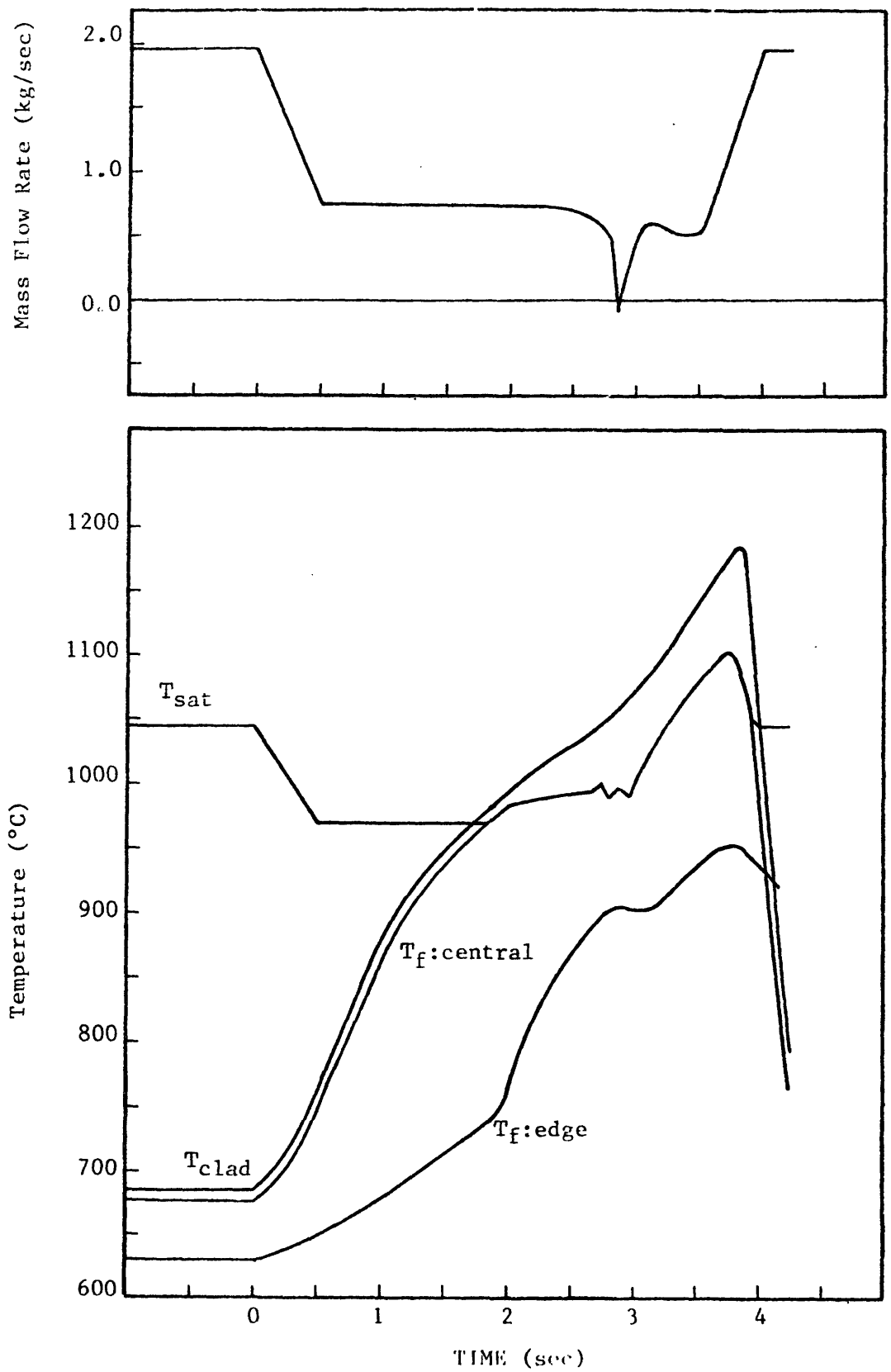

Figure 4.21: W1: Temperatures and Mass Flow Rate for Sequence 7a'

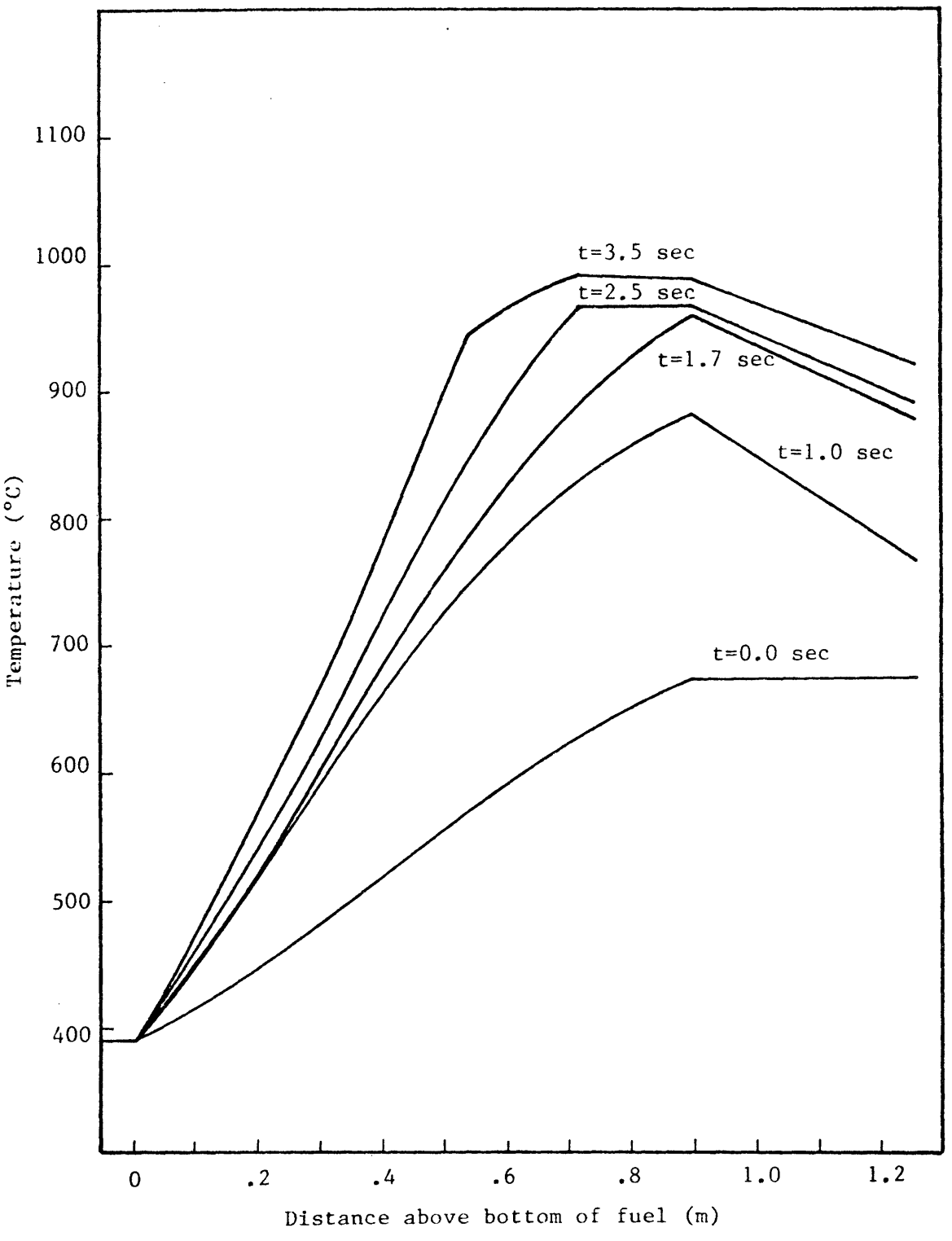

Figure 4.22: Axial Temperature Profile for Sequence 7a'

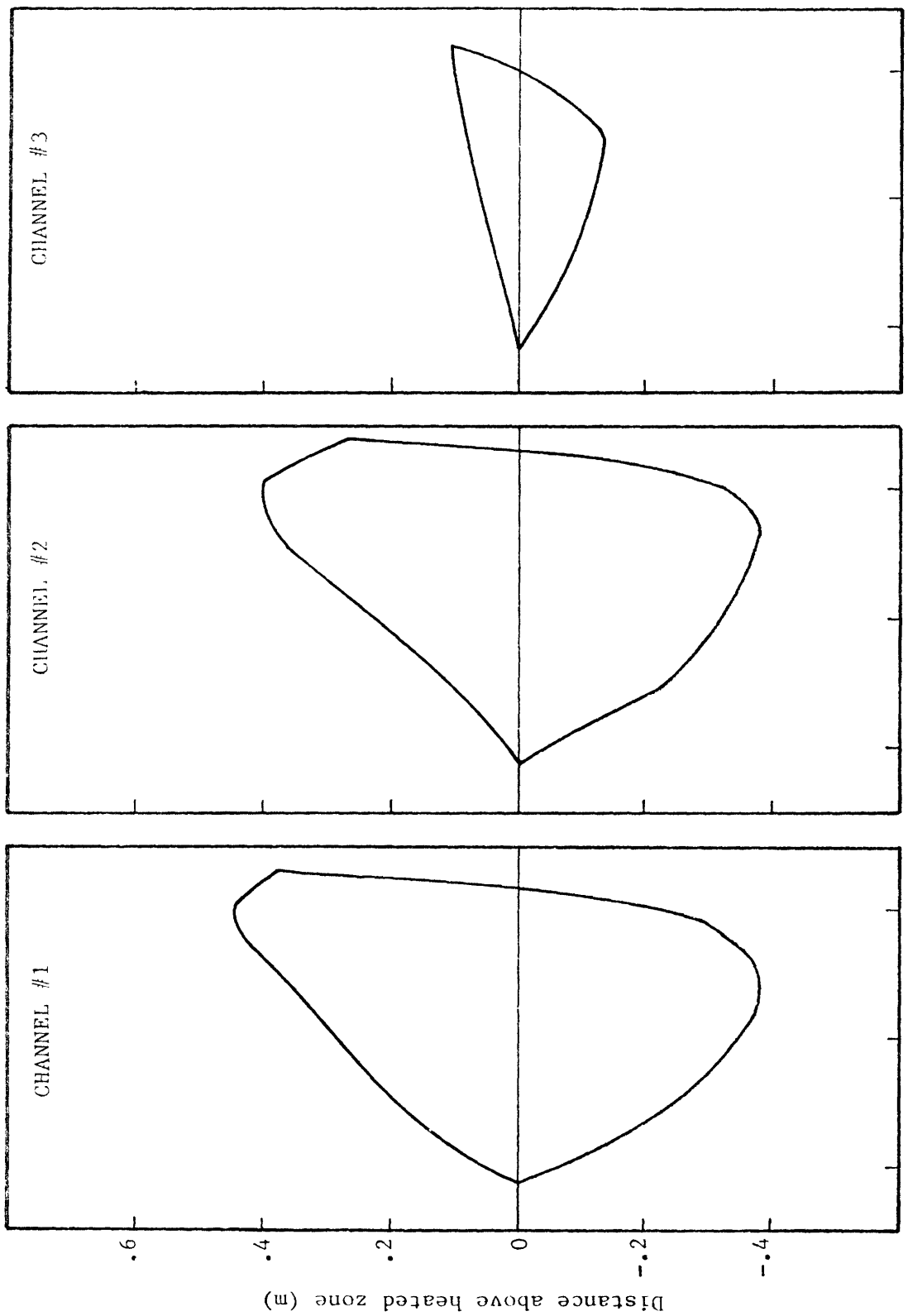

Figure 4.23: W1: Void Maps for Sequence 7a'

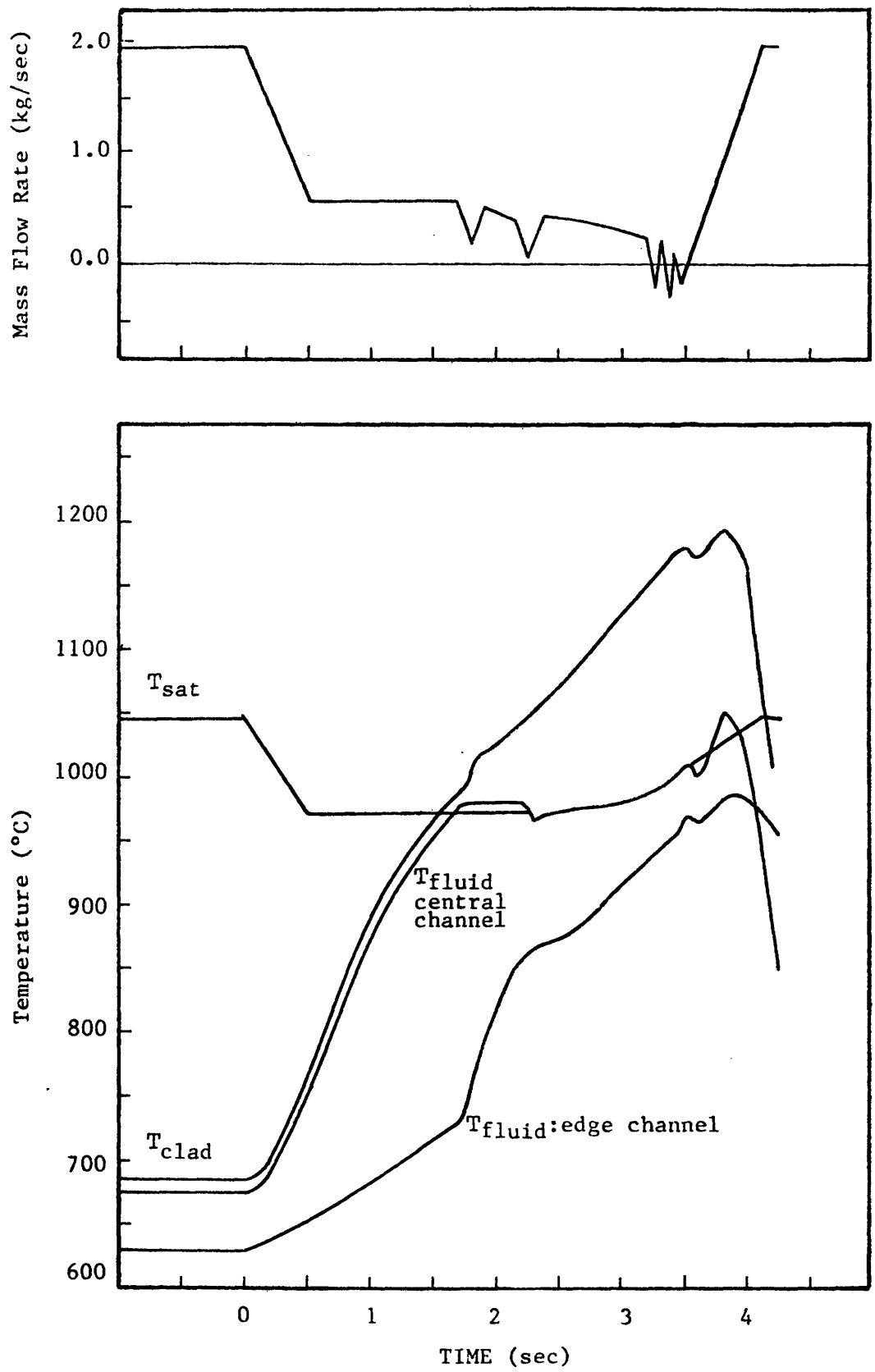

Figure 4.24: W1: Temperatures and Mass Flow Rate for Sequence 7b'

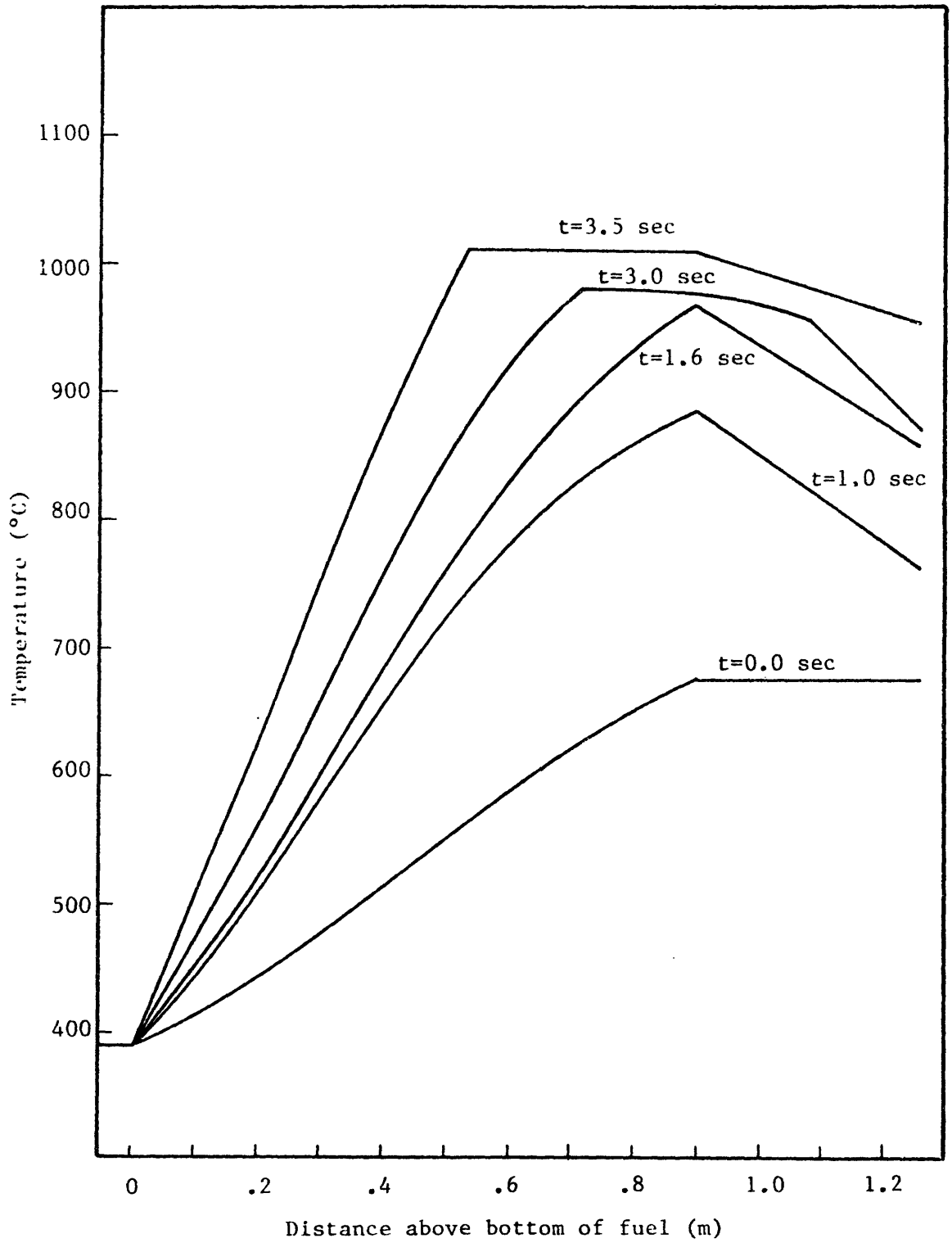

Figure 4.25: W1: Axial Temperature Profile for Sequence 7b´

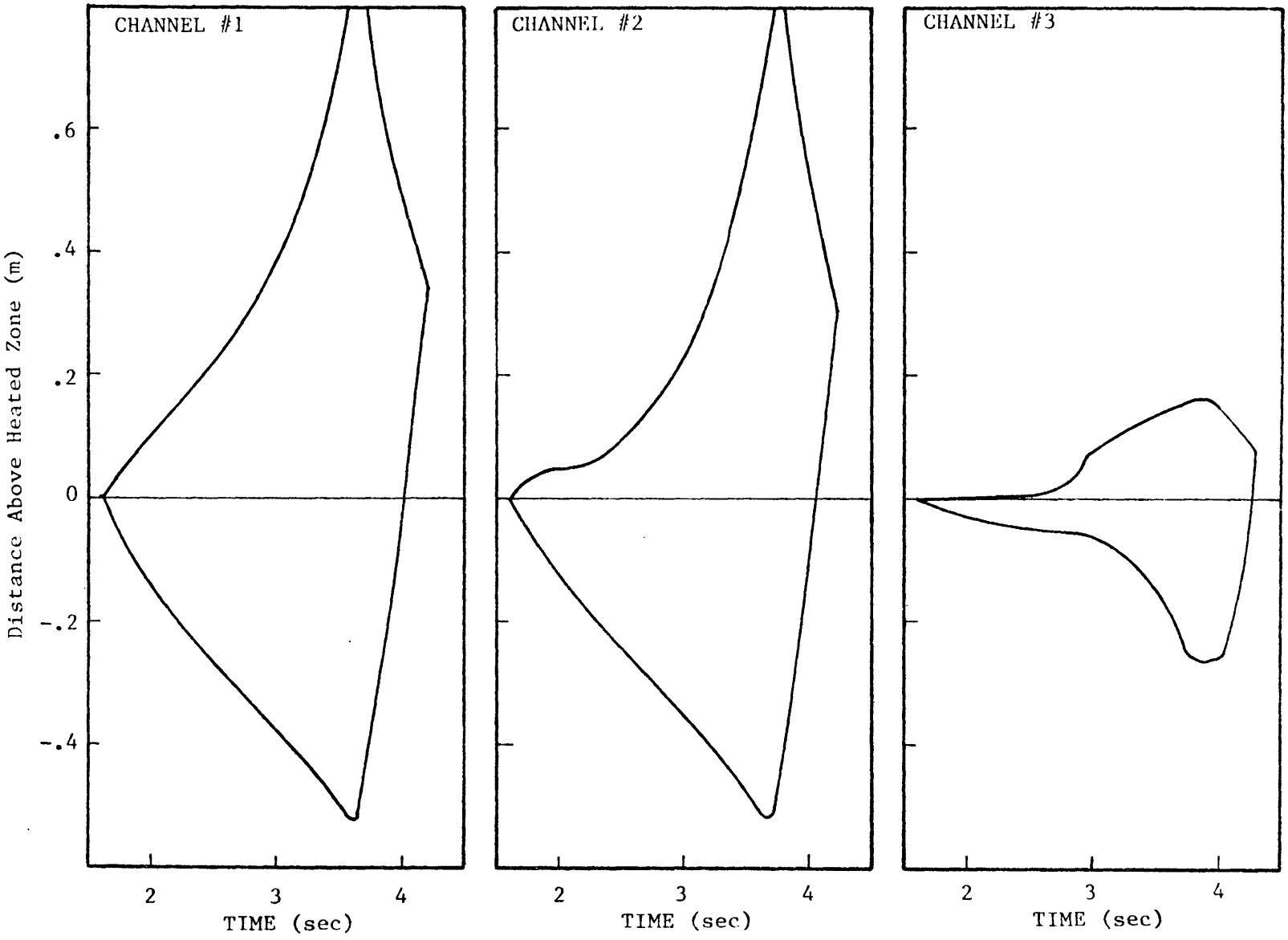

Figure 4.26: W1: Void Maps for Sequence 7b´

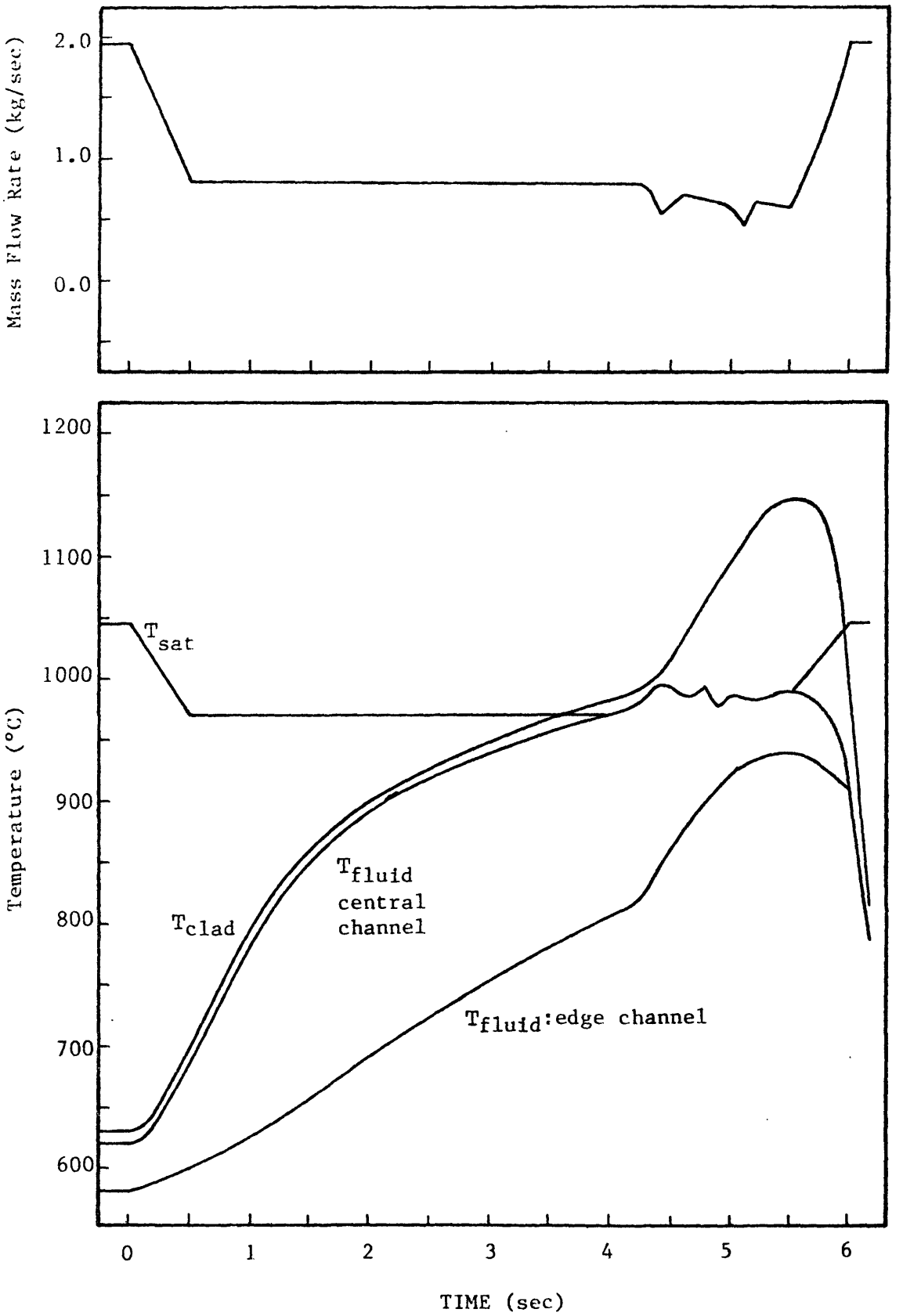

Figure 4.27: W1: Temperature and Mass Flow Rate for Sequence 3

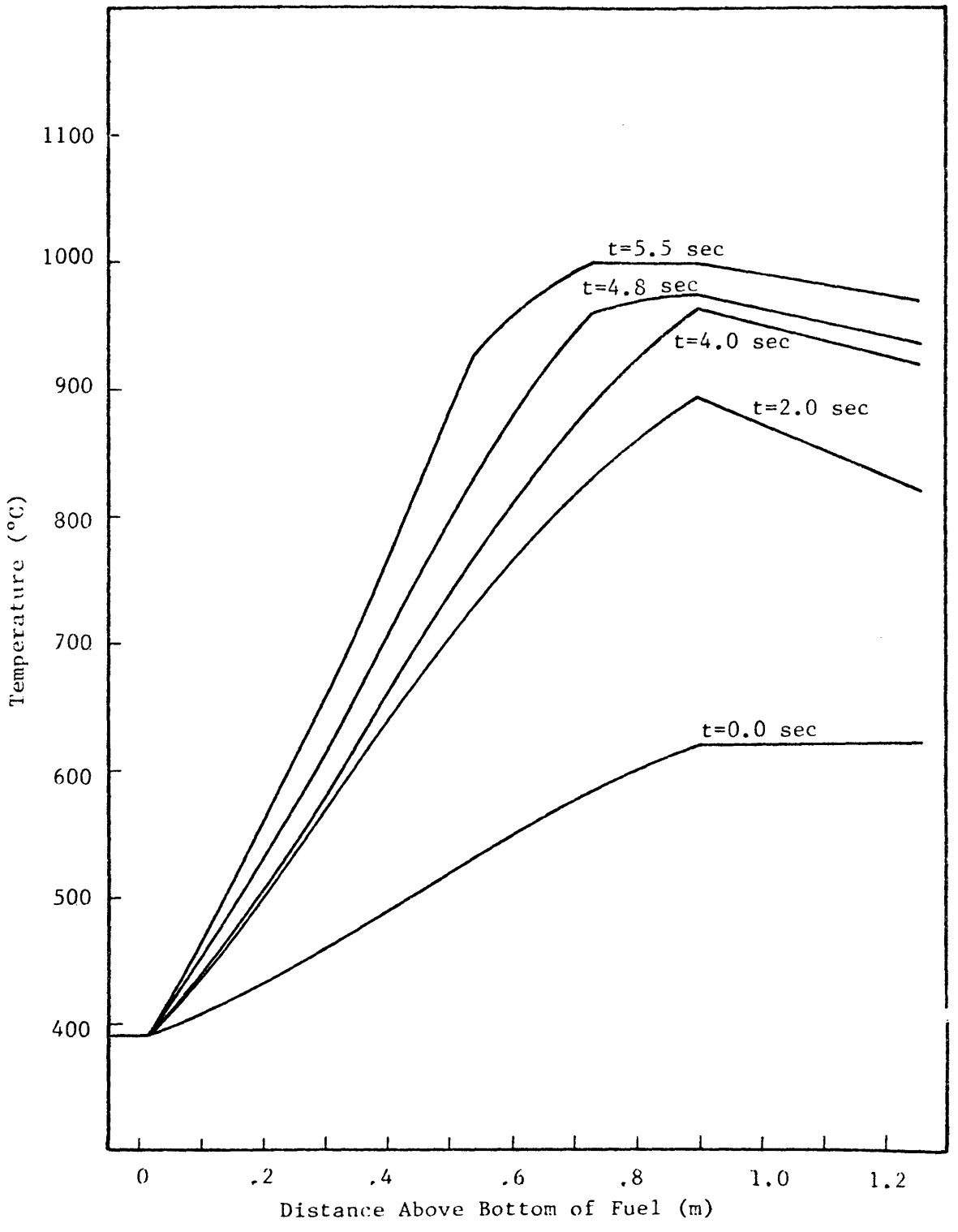

Figure 4.28: W1: Axial Temperature Profile for Sequence 3

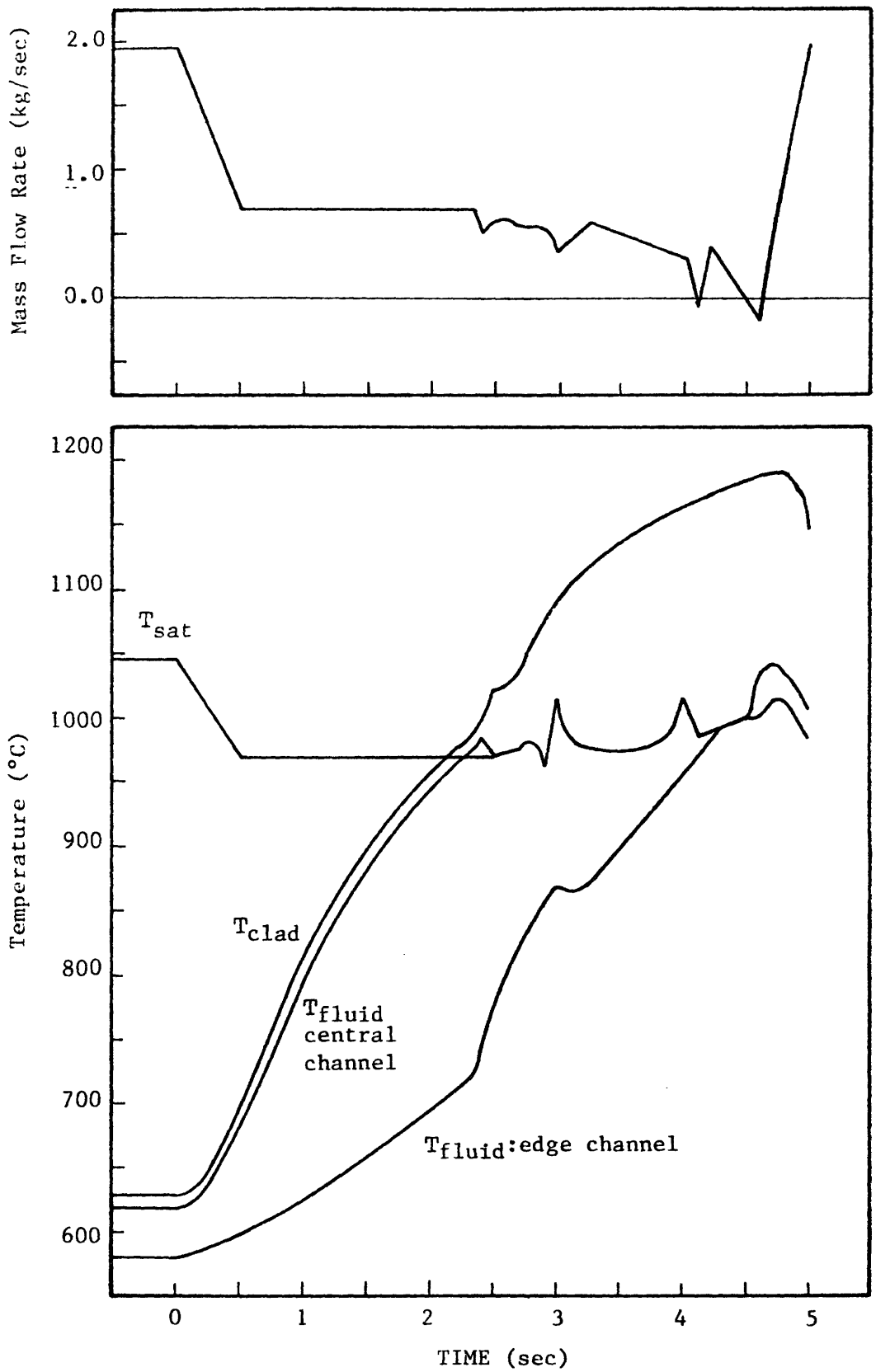

Figure 4.29: W1: Temperature and Mass Flow Rate for Sequence 4

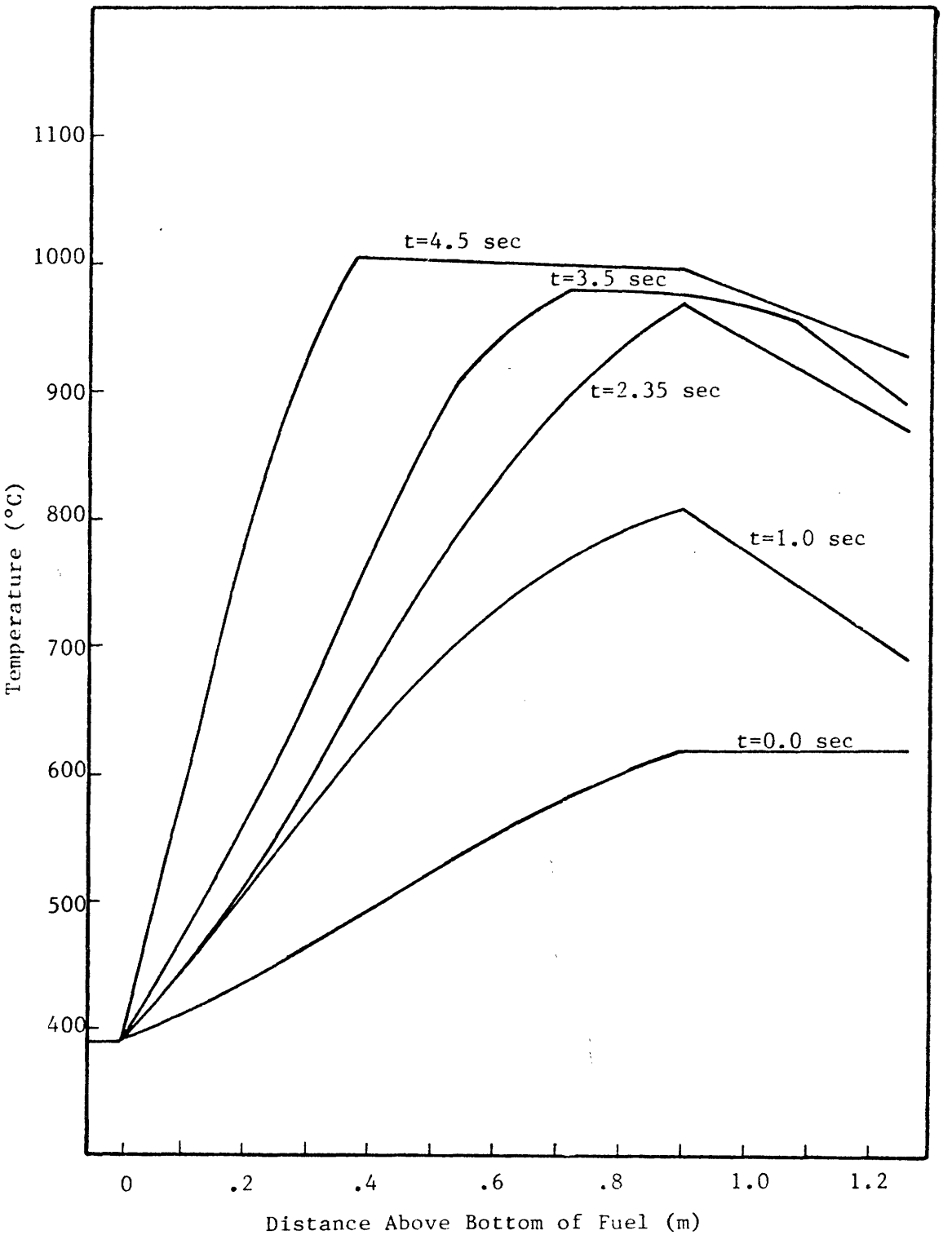

Figure 4.30: W1: Axial Temperature Profile for Sequence 4

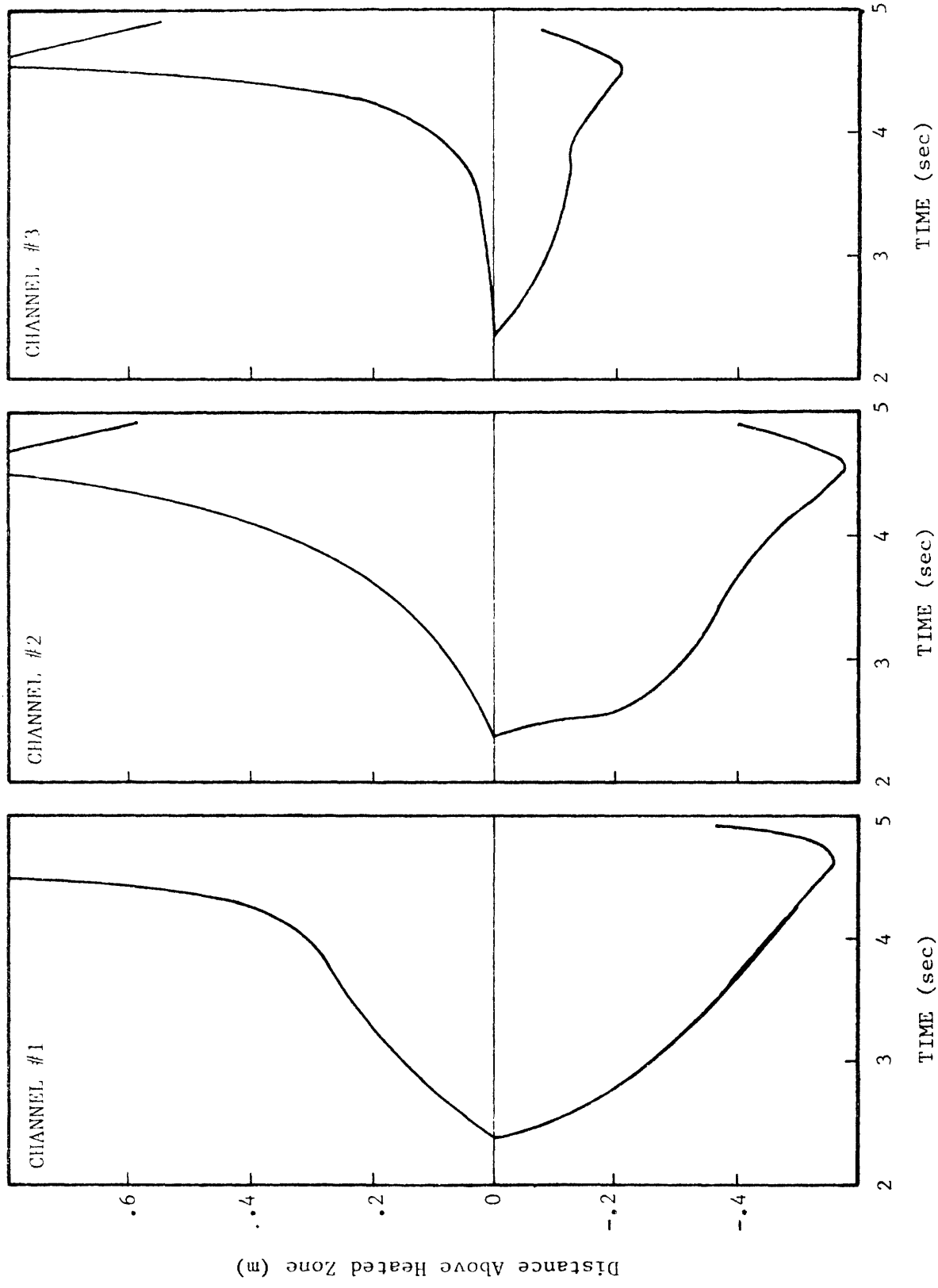

Figure 4.31: W1: Void Maps for Sequence 4

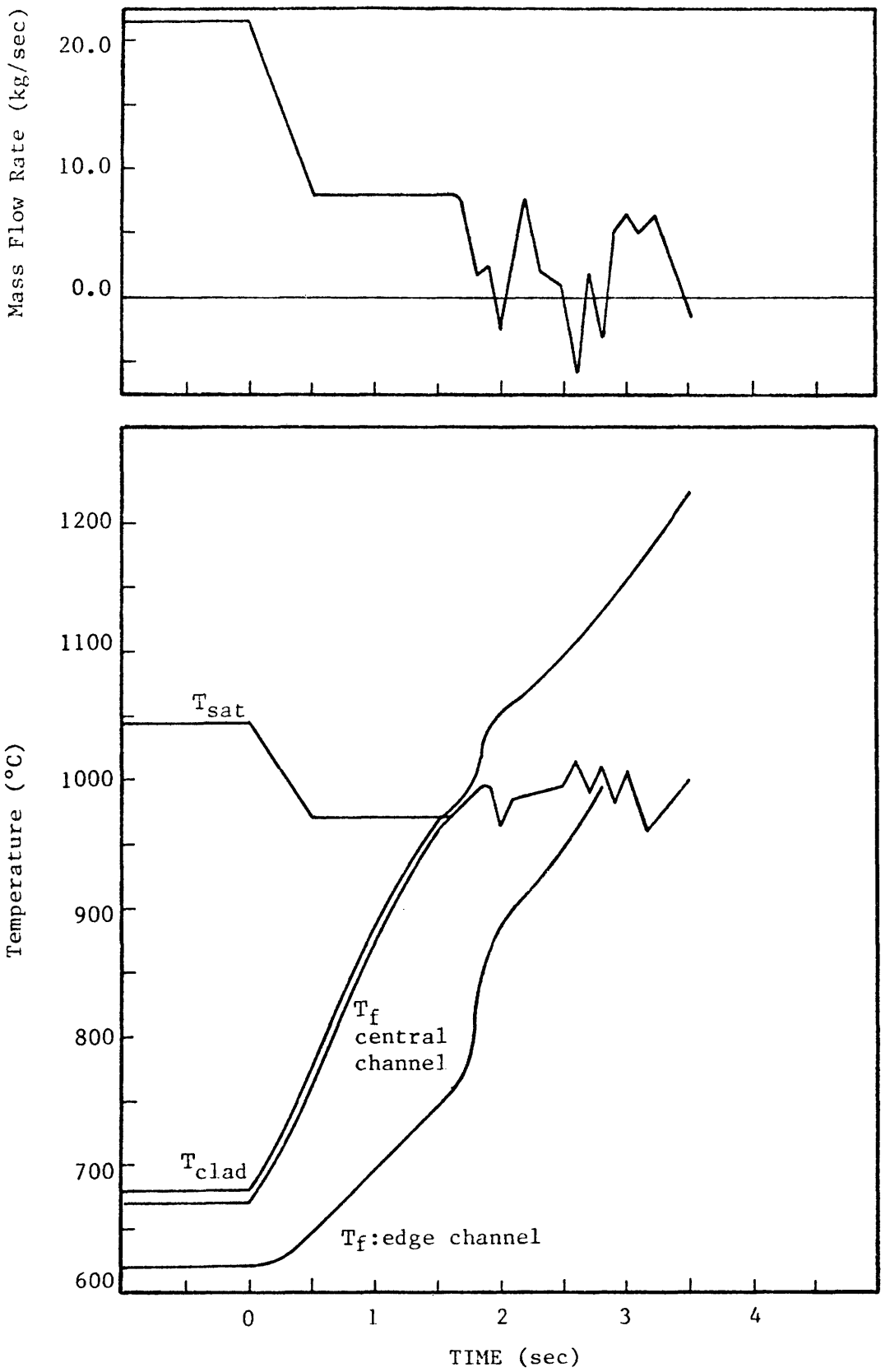

Figure 4.32: Temperature and Mass Flow Rate for 217-Pin Bundle Under Sequence 7b´ Conditions

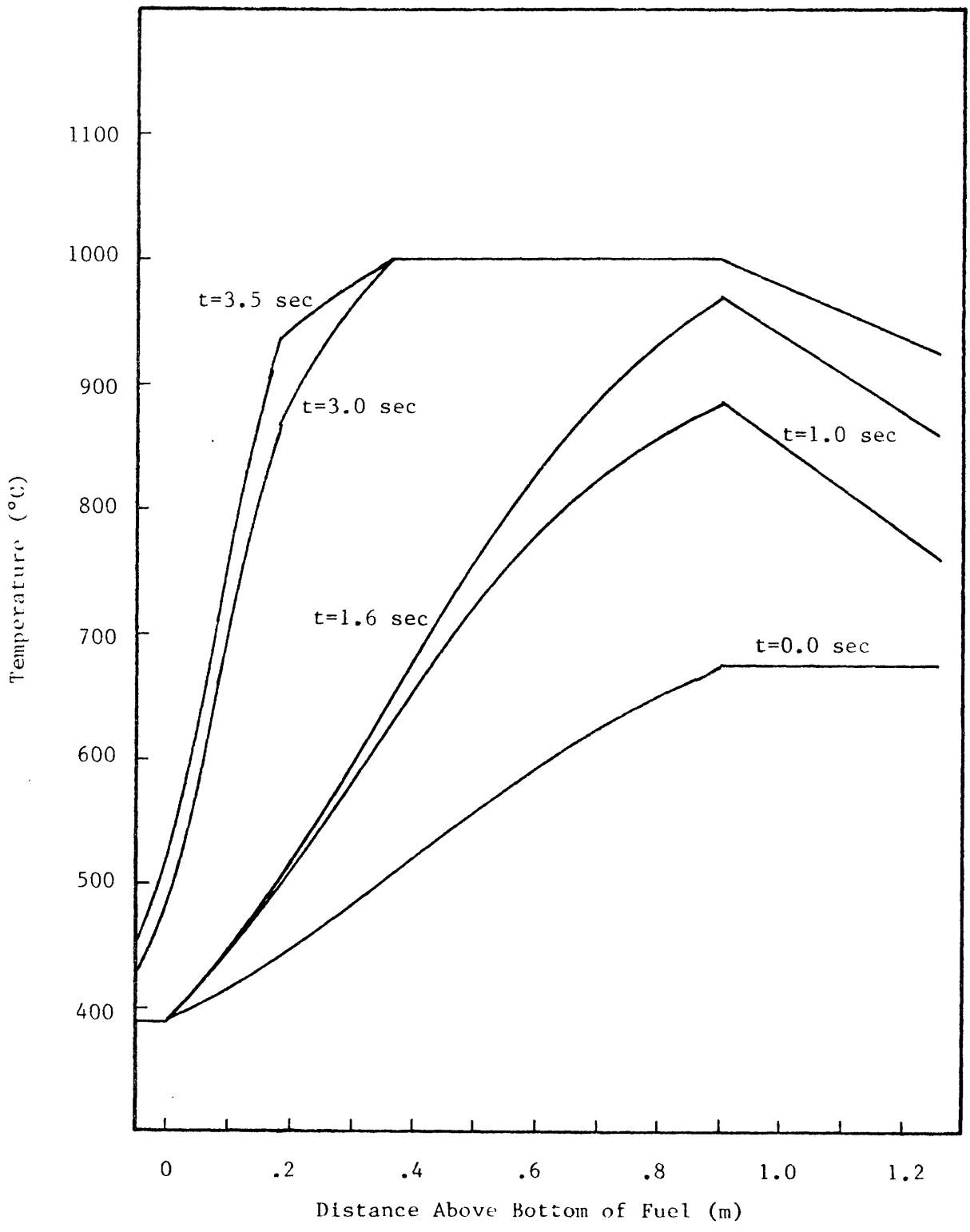

Figure 4.33: Axial Temperature Profile for 217-Pin Bundle Under Sequence 7b′ Conditions

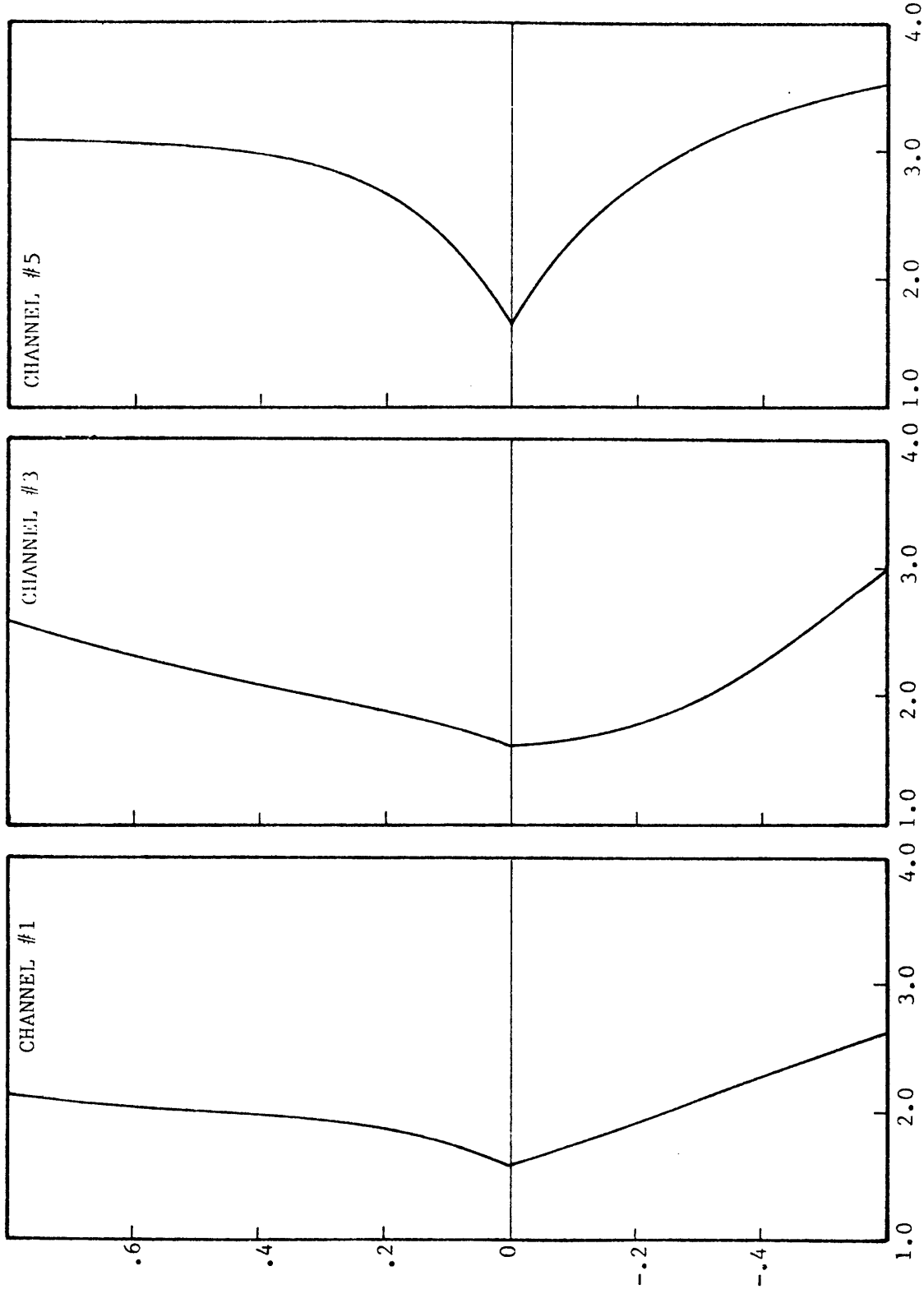

Figure 4.34: Void Maps for 217-Pin Bundle Under Sequenc 7b′ Conditions

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)	8.65×10^{-3}
Heated Length (m)	0.6
Downstream Unheated Length (m)	0.494
Upstream Unheated Length (m)	0.12
Wire Wrap OD (m)	1.28×10^{-3}
Flat to Flat (m)	4.58×10^{-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

FLOW (kg/sec)	T_{max}(°C) (MEASURED)	T_{max}(°C) (NATOF-2D)
.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

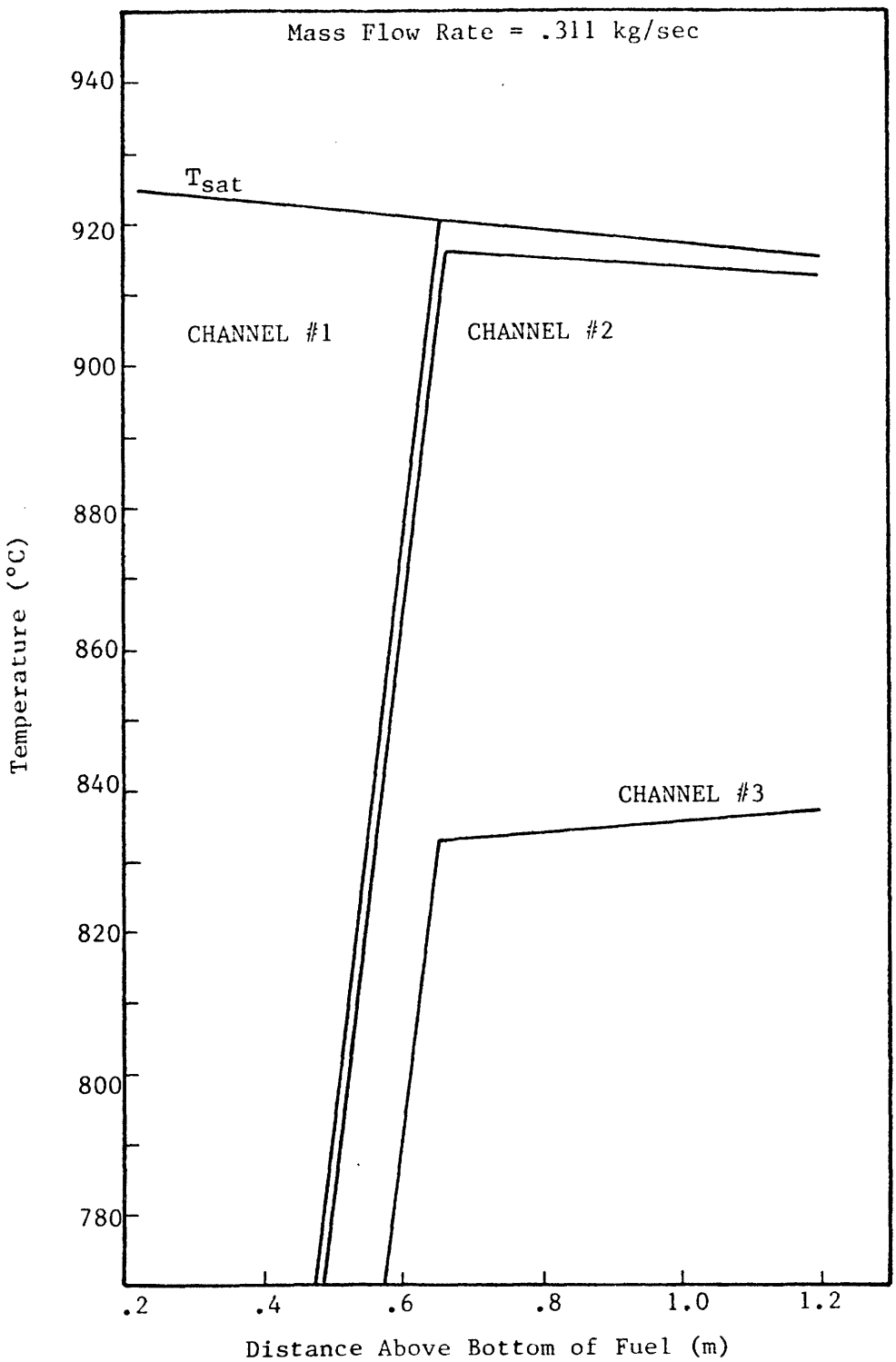

Figure 4.35: GR19: Temperature Profiles
For .311 kg/sec Mass Flow Rate

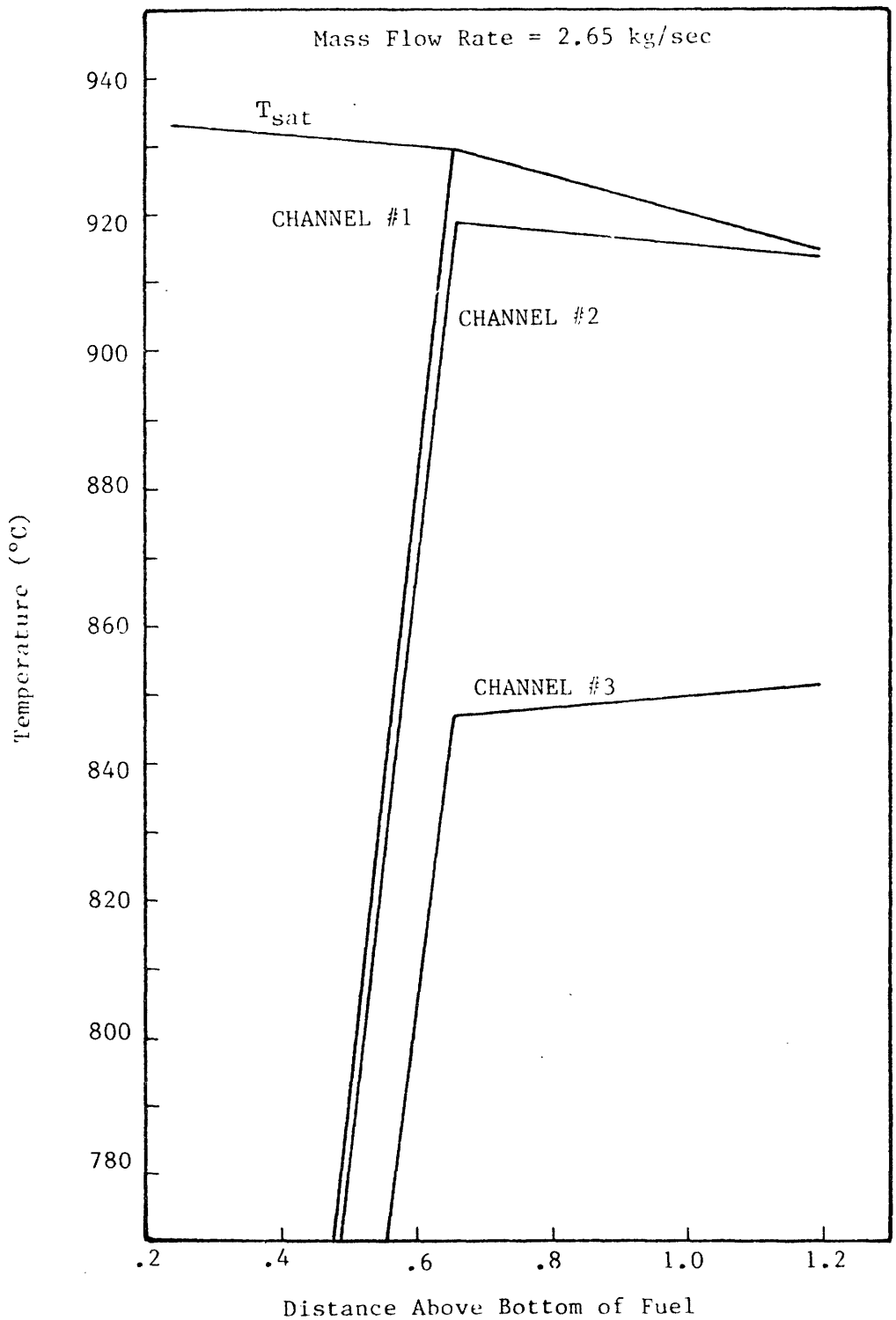

Figure 4.36: GR19: Temperature Profiles
For .265 kg/sec Mass Flow Rate

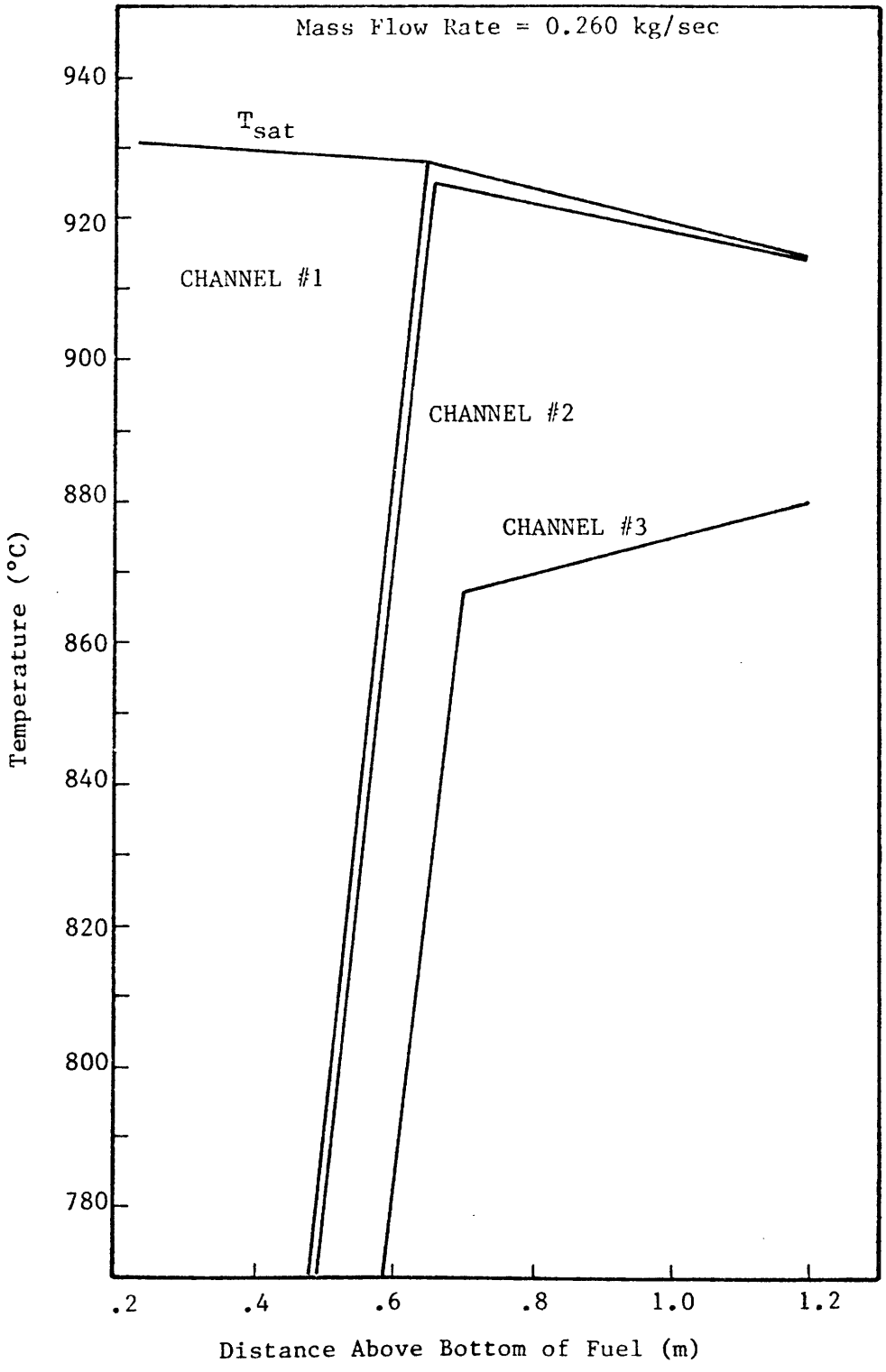

Figure 4.37: GR19: Temperature Profiles
For .260 kg/sec Mass Flow Rate

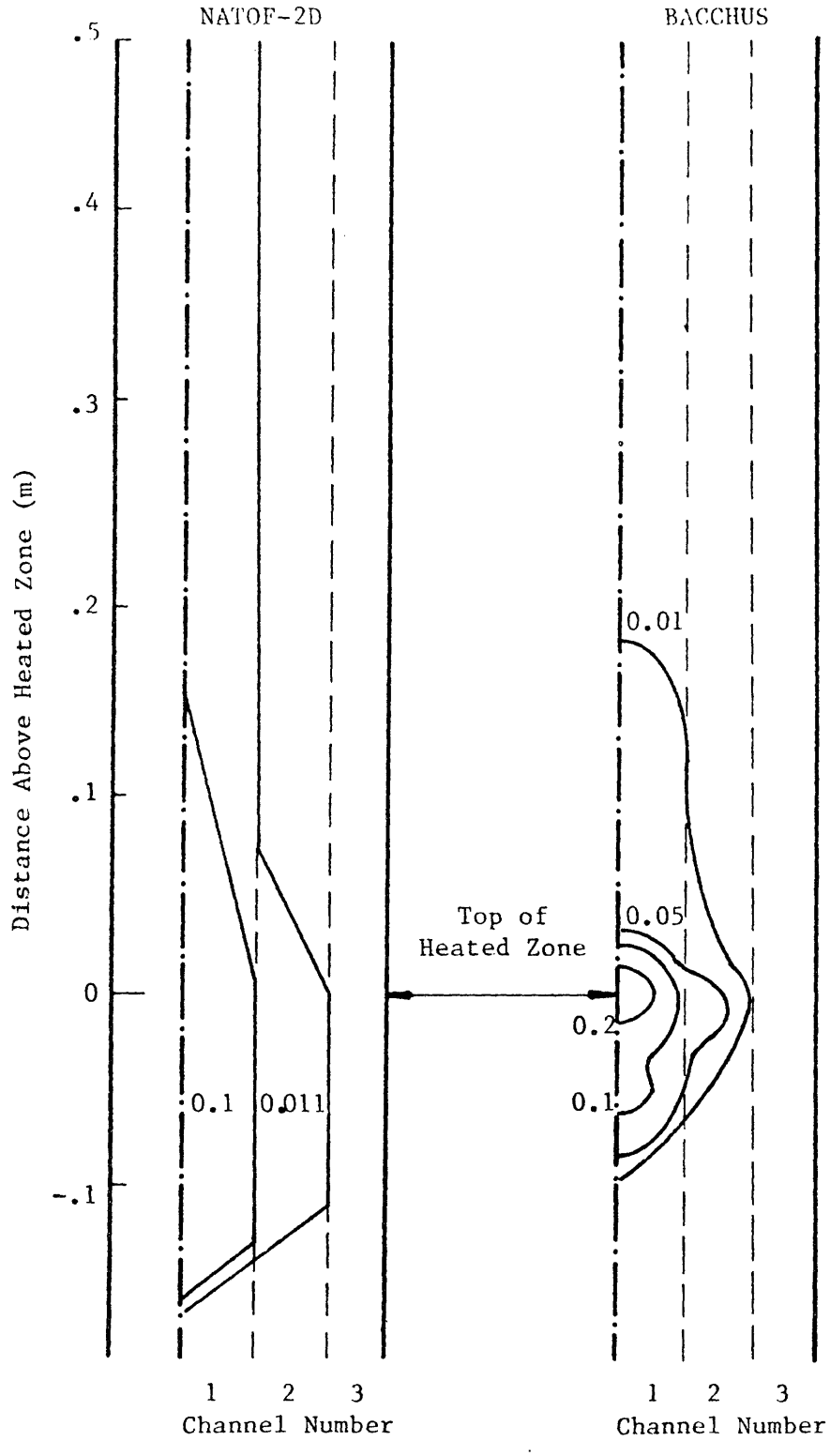

Figure 4.38

GR19 Quality Contours for 0.265 kg/sec Mass Flow Rate

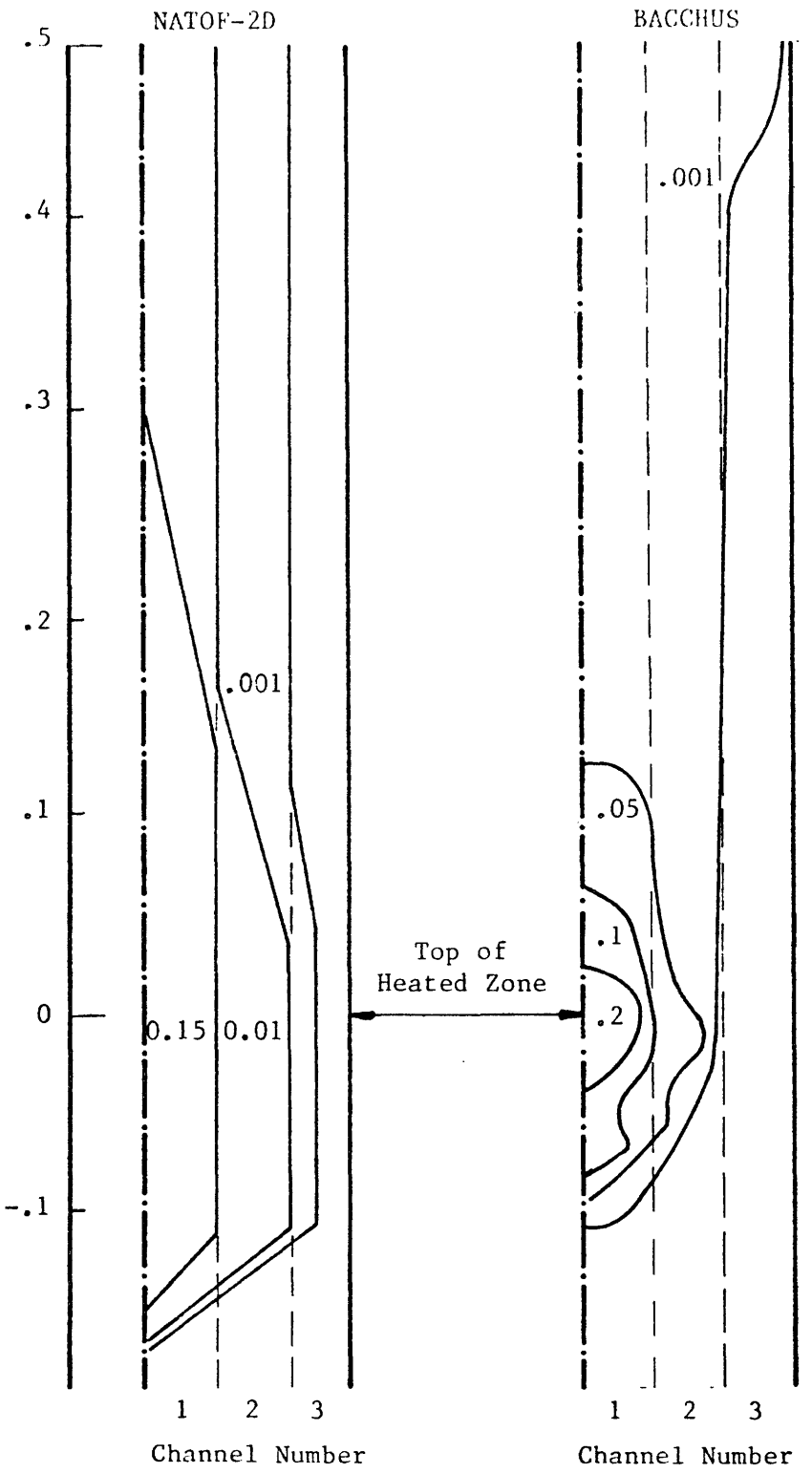

Figure 4.39

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

CHAPTER 5

CONCLUSIONS AND RECOMMENDATIONS

5.1 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 beahvior, especially those characterizing the explosive volume change associated with sodim 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 this work.

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, SAS1A, "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. Fabic, 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," Kalsruhe, 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 LOD1 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 Peppler, 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," Chelesea 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.

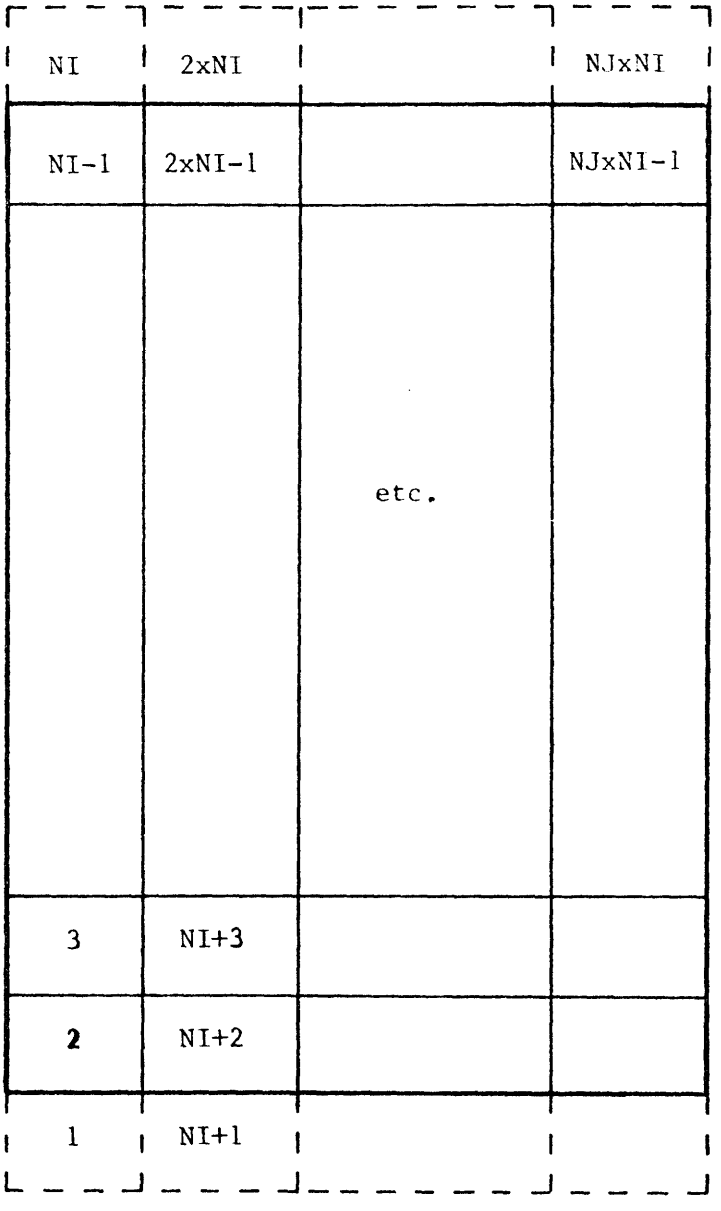

Figure A1. Cell Arrangement in the R-Z Plane

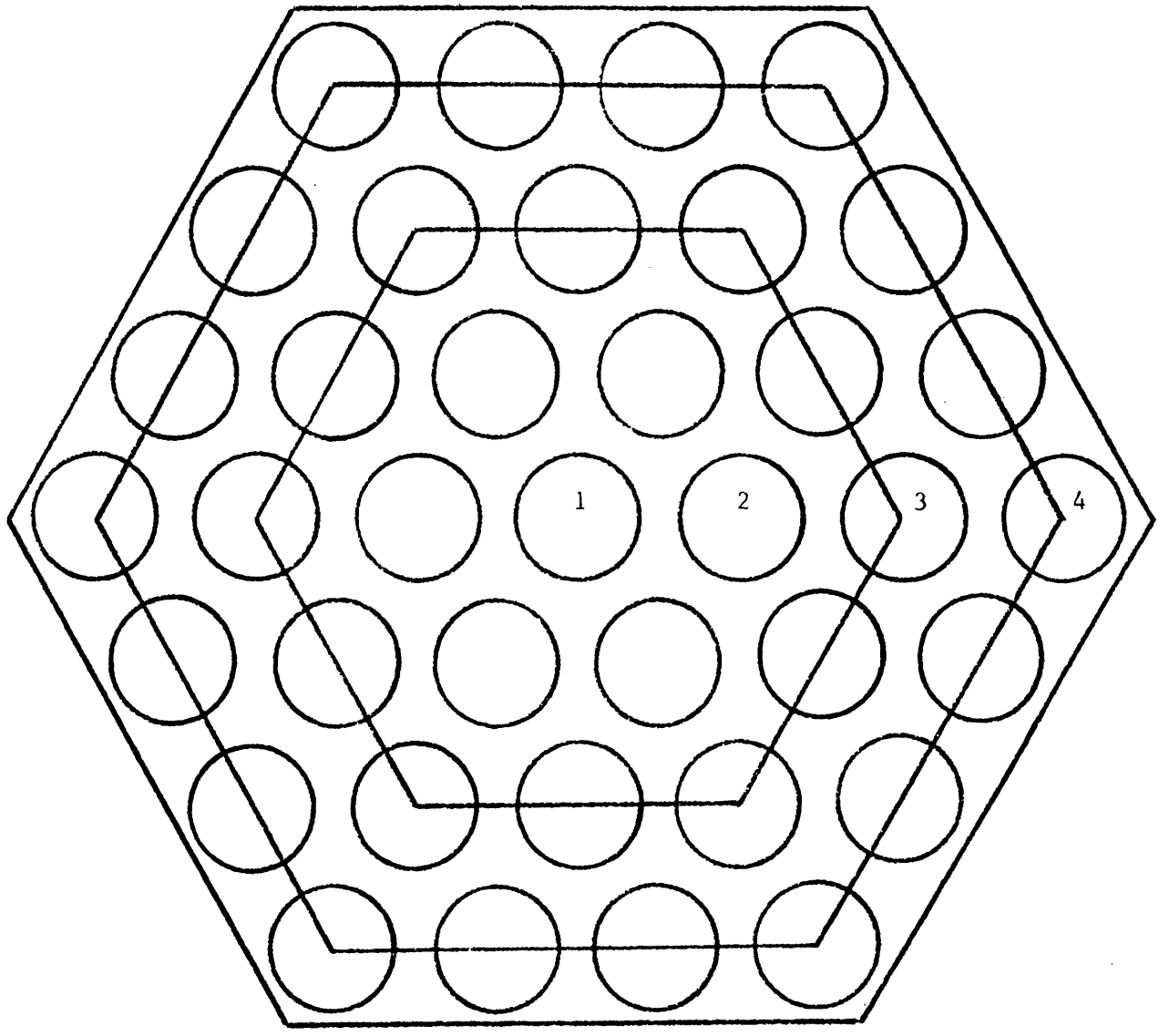

Figure A.2 Fuel Pin Numbering

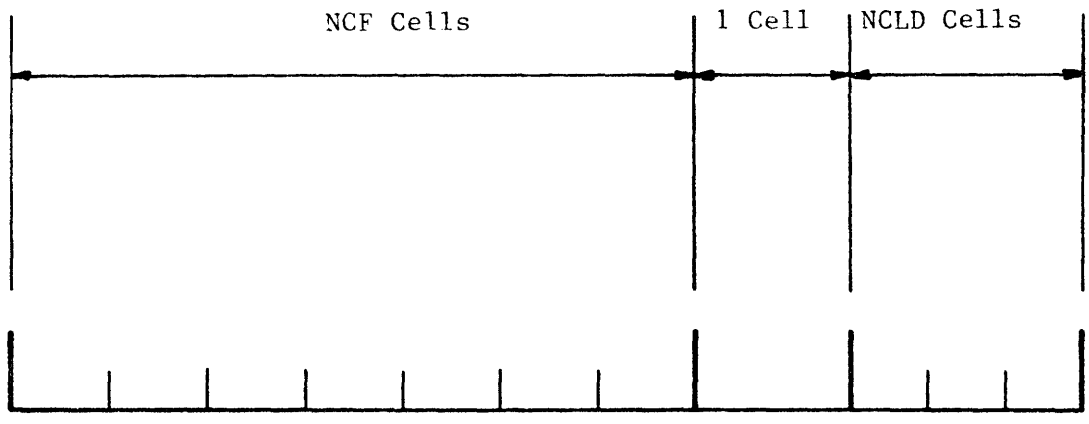

Figure A3. Cell Arrangement for Fuel Pin Heat Conduction

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/m2.

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_1, X_2, X_3, OMX = Input parameters

and X stands for:

PNB = Pressure at the bottom of fuel assembly (N/m^2)

PNT = Pressure at the top of fuel assembly (N/m^2)

ALB = Void fraction at the inlet of fuel assembly.

TVB = Vapor temperature at inlet (°K).

TLB = Liquid temperature at inlet (°K).

HNW = Power density in fuel pins (W/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:

$$X = X_1(L)*DTIME + X_2(L)$$

1st CARD:	LP, TB	(L1, F15.5)
2nd CARD:	PNB1, PND2, 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 ($J/m^2{}^\circ K$). 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 full.

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_2. 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 (°K)

TAV = An estimate of the average temperature in fuel assembly (°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 (°K)

TL = Liquid temperature (°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 (°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 (°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.

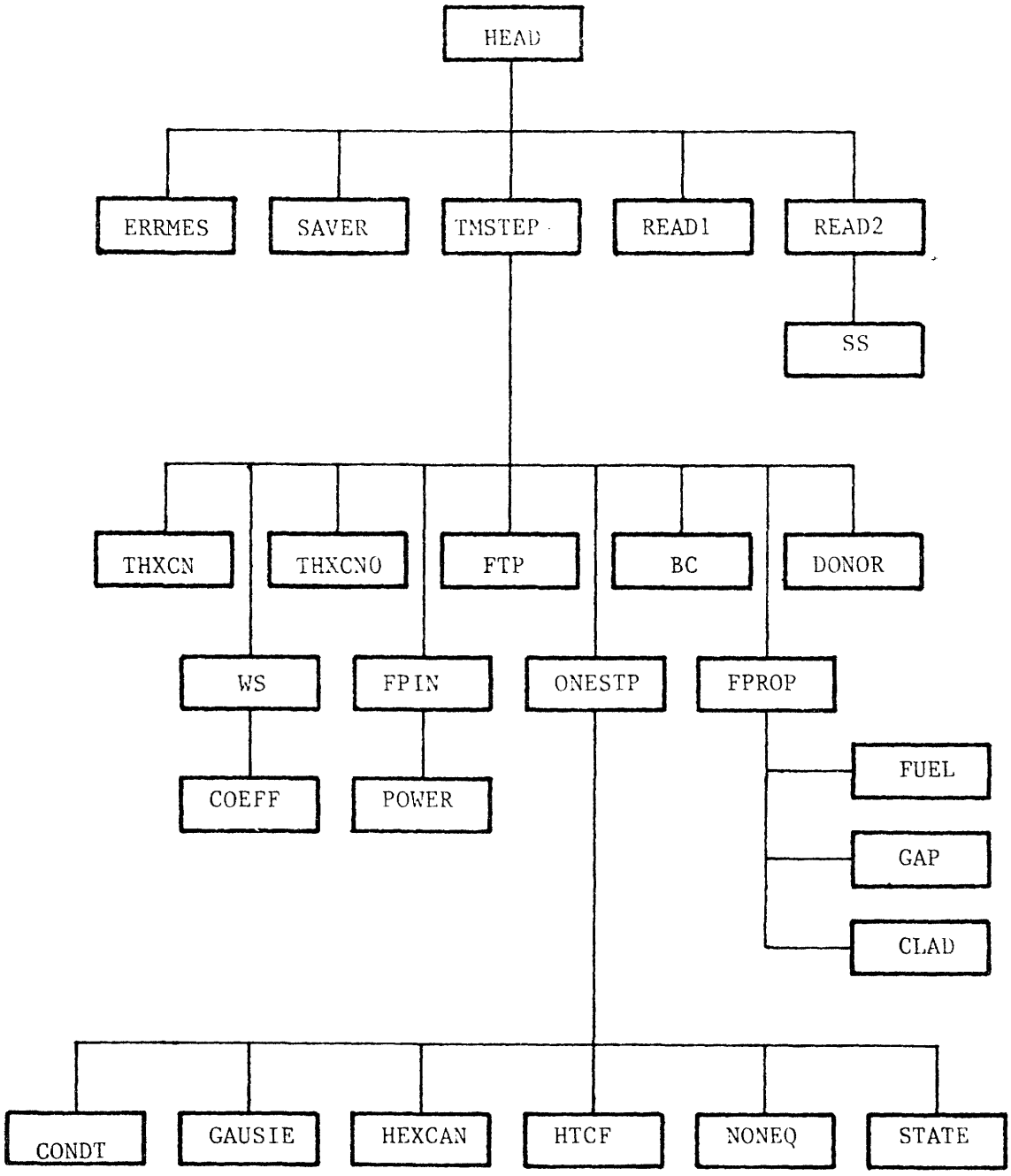

Figure B1. NATOF-2D Subroutine Structure

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 coefficienct

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 connective 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

```
   12    5    4    2
    20.150000000D+01
    00.000000000D+00
         8        5000.100000000D+000.100000000D-010.100000000D-03
T        4.40000
0.000000000D+000.675750000D+060.000000000D+000.000000000D+00
0.000000000D+000.27760000CD+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+000.244508686D+100.000000000D+000.000000000D+00
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   0
TO.182900000D+000.182900000D+000.182900000D+000.182900000D+000.182900000D+00
TO.182900000D+000.182900000D+000.182900000D+000.182900000D+000.182900000D+00
TO.182900000D+000.182900000D+000.000000000D+000.000000000D+000.000000000D+00
FO.000000000D+000.000000000D+000.000000000D+000.000000000D+000.000000000D+00
TO.850000000D+040.850000000D+040.850000000D+040.850000000D+040.850000000D+04
TO.850000000D+040.850000000D+040.000000000D+000.000000000D+000.000000000D+00
TO.000000000D+000.000000000D+000.000000000D+000.000000000D+000.000000000D+00
FO.000000000D+000.000000000D+000.000000000D+000.000000000D+000.000000000D+00
TO.000000000D+000.000000000D+000.677403000D+000.838702000D+000.100000000D+01
TO.838702000D+000.677403000D+000.000000000D+000.000000000D+000.000000000D+00
TO.000000000D+000.000000000D+000.000000000D+000.000000000D+000.677403000D+00
TO.838702000D+000.100000000D+010.838702000D+000.677403000D+000.000000000D+00
TO.000000000D+000.000000000D+000.000000000D+000.000000000D+000.000000000D+00
TO.000000000D+000.677403000D+000.838702000D+000.100000000D+010.838702000D+00
TO.677403000D+000.000000000D+000.000000000D+000.000000000D+000.000000000D+00
TO.000000000D+000.000000000D+000.000000000D+000.677403000D+000.838702000D+00
TO.100000000D+010.838702000D+000.677403000D+000.000000000D+000.000000000D+00
TO.000000000D+000.000000000D+000.000000000D+000.000000000D+000.000000000D+00
TO.677403000D+000.838702000D+000.100000000D+010.838702000D+000.677403000D+00
T
FO.000000000D+000.000000000D+000.000000000D+000.000000000D+000.000000000D+00
TO.000000000D+000.610000000D+020.000000000D+000.000000000D+000.000000000D+00
TO.000000000D+000.000000000D+000.000000000D+000.000000000D+000.000000000D+00
TO.000000000D+000.000000000D+000.000000000D+000.610000000D+020.000000000D+00
TO.000000000D+000.000000000D+000.000000000D+000.000000000D+000.000000000D+00
TO.000000000D+000.000000000D+000.000000000D+000.000000000D+000.000000000D+00
TO.610000000D+020.000000000D+000.000000000D+000.000000000D+000.000000000D+00
TO.000000000D+000.000000000D+000.000000000D+000.000000000D+000.000000000D+00
TO.000000000D+000.000000000D+000.610000000D+020.000000000D+000.000000000D+00
TO.000000000D+000.000000000D+000.000000000D+000.000000000D+000.000000000D+00
TO.000000000D+000.000000000D+000.000000000D+000.000000000D+000.869000000D+02
```

```
T
T
F0.000000000D+000.000000000D+000.000000000D+000.000000000D+000.000000000D+00
T0.100000000D+010.100000000D+010.100000000D+010.100000000D+010.000000000D+00
T0.000000000D+000.000000000D+000.000000000D+000.000000000D+000.000000000D+00
F0.000000000D+000.000000000D+000.000000000D+000.000000000D+000.000000000D+00
0.954000000D+000.250000000D+000.900000000D+00
 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0.292100000D-020.381000000D-030.600000000D-04
T0.000000000D+00
0.675750000D+060.277600000D+060.661140000D+030.900000000D+03
```

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
-.464566510D+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   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
```

```
T0.000000000D+000.000000000D+000.000000000D+000.000000000D+000.677403000D+00
T0.838702000D+000.100000000D+010.838702000D+000.677403000D+000.000000000D+00
T0.000000000D+000.000000000D+000.000000000D+000.000000000D+000.000000000D+00
T0.000000000D+000.677403000D+000.838702000D+000.100000000D+010.838702000D+00
T0.677403000D+000.000000000D+000.000000000D+000.000000000D+000.000000000D+00
T0.000000000D+000.000000000D+000.000000000D+000.677403000D+000.838702000D+00
T0.100000000D+010.838702000D+000.677403000D+000.000000000D+000.000000000D+00
T0.000000000D+000.000000000D+000.000000000D+000.000000000D+000.000000000D+00
T0.677403000D+000.838702000D+000.100000000D+010.838702000D+000.677403000D+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+000.610000000D+020.000000000D+000.000000000D+000.000000000D+00
T0.000000000D+000.000000000D+000.000000000D+000.000000000D+000.000000000D+00
T0.000000000D+000.000000000D+000.000000000D+000.610000000D+020.000000000D+00
T0.000000000D+000.000000000D+000.000000000D+000.000000000D+000.000000000D+00
T0.000000000D+000.000000000D+000.000000000D+000.000000000D+000.000000000D+00
T0.610000000D+020.000000000D+000.000000000D+000.000000000D+000.000000000D+00
T0.000000000D+000.000000000D+000.000000000D+000.000000000D+000.000000000D+00
T0.000000000D+000.000000000D+000.610000000D+020.000000000D+000.000000000D+00
T0.000000000D+000.000000000D+000.000000000D+000.000000000D+000.000000000D+00
T0.000000000D+000.000000000D+000.000000000D+000.000000000D+000.869000000D+02
T0.000000000D+000.000000000D+000.000000000D+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
T0.100000000D+010.100000000D+010.100000000D+010.100000000D+010.000000000D+00
T0.000000000D+000.000000000D+000.000000000D+000.000000000D+000.000000000D+00
F0.000000000D+000.000000000D+000.000000000D+000.000000000D+000.000000000D+00
0.954000000D+000.250000000D+000.900000000D+00
 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0.292100000D-020.381000000D-030.600000000D-04
F0.000000000D+00
    10.661140000D+030.661140000D+030.675750000D+060.000000000D+00
    10.645364280D+010.645364280D+010.000000000D+000.000000000D+00
    20.661139976D+030.661139976D+030.457953361D+060.000000000D+00
    20.645364280D+010.645364280D+010.944392050D-030.944392050D-03
    30.709018194D+030.709018194D+030.440294610D+060.000000000D+00
    30.644028454D+010.644028454D+010.624381194D-030.624381194D-03
    40.768816955D+030.768816955D+030.422317592D+060.000000000D+00
    40.651813704D+010.651813704D+010.360111639D-030.360111639D-03
    50.840612692D+030.840612693D+030.404311805D+060.000000000D+00
    50.662503059D+010.662503059D+010.268289802D-030.268289802D-03
    60.901049391D+030.901049391D+030.386078761D+060.000000000D+00
    60.676132297D+010.676132297D+010.124816471D-030.124816471D-03
    70.949882204D+030.949882204D+030.367828564D+060.000000000D+00
    70.688248670D+010.688248670D+010.267044809D-040.267044809D-04
```

```
 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 .295536496D+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 .676259771D+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 .331413956D+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+01-.419776596D-03-.419776596D-03
330.949827962D+030.949827962D+030.331414574D+060.000000000D+00
330.698806424D+010.698806424D+01-.177593108D-03-.177593108D-03
340.949827916D+030.949827916D+030.313474560D+060.000000000D+00
340.698963622D+010.698963622D+01-.718543316D-04-.718548316D-04
350.949827871D+030.949827871D+030.295536625D+060.000000000D+00
350.699031921D+010.699031921D+01-.227451686D-04-.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+030.457928041D+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+010.726055283D-030.726055283D-03
410.840580594D+030.840580594D+030.404305235D+060.000000000D+00
410.662617911D+010.662617911D+010.579300377D-030.579300377D-03
420.901009888D+030.901009888D+030.386075914D+060.000000000D+00
420.676208049D+010.676208049D+010.294186726D-030.294186726D-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+01-.461539051D-03-.461539051D-03
450.949658968D+030.949658968D+030.331415063D+060.000000000D+00
450.698663932D+010.698663932D+01-.207414256D-03-.207414256D-03
460.949634152D+030.949634152D+030.313474756D+060.000000000D+00
460.698828621D+010.698828621D+01-.922642260D-04-.922642260D-04
470.949624131D+030.949624131D+030.295536684D+060.000000000D+00
470.698923257D+010.698923257D+01-.371568794D-04-.371568794D-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+030.440272850D+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
    530.569885826D+010.569885826D+010.000000000D+000.000000000D+00
    540.851601173D+030.851601173D+030.386075215D+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+030.177692822D+040.169945358D+04
T0.148493467D+040.119305137D+040.887163285D+030.762281343D+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+030.780962846D+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+040.153484906D+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+030.739639288D+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
T0.886799659D+030.855643684D+030.249908909D+040.242051161D+040.215139534D+04
T0.165022069D+040.112241561D+040.968174625D+030.940155089D+030.914024510D+03
T0.223801488D+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+030.949827916D+03
T0.949827916D+030.949827916D+030.949827916D+030.949827916D+030.949827916D+03
T0.949827916D+030.949827916D+030.949827871D+030.949827871D+030.949827871D+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
T0.886807497D+030.855651522D+030.249910223D+040.242052567D+040.215141208D+04
T0.165023657D+040.112242552D+040.968184552D+030.940165015D+030.914034436D+03
T0.223803277D+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+030.949634152D+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+030.758024583D+030.265753179D+04
T0.257657755D+040.228898081D+040.169079942D+040.106588978D+040.881819019D+03
T0.848410804D+030.817254929D+030.242750750D+040.234398530D+040.206164460D+04
T0.156873596D+040.107147044D+040.917126727D+030.889107191D+030.862976612D+03
T0.213527169D+040.205587076D+040.181242410D+040.144635784D+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
outlet mass flow rate = 0.914200D+01 kg/sec
total heat transfered = 0.272580D+07 watt

inlet enthalpy flow = 0.234219D+07 watt
outlet enthalpy flow = 0.131079D+08 watt

channel number 1

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	0.010783	851.598	851.598	1005.330	849.091	3.43381	3.37281	0.00000	0.00000
11	1.9192	0.010783	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	tsat	twall	uvz (m/sec)	ulz (m/sec)	uvr (m/sec)	ulr (m/sec)
			(degree celsius)							
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	1047.662	4.36154	1.16028	1.11469	1.10838
6	2.4263	0.000089	984.672	984.672	987.277	993.807	1.05234	1.05201	0.04685	0.04685
5	2.4627	0.000000	885.371	885.371	989.251	897.267	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	980.801	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

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	991.181	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

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 inlet enthalpy flow = -.248236D+07 watt
outlet mass flow rate = 0.754283D+01 kg/sec outlet enthalpy flow = 0.109099D+08 watt
total heat transfered = -.562540D+06 watt

channel number 1

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	0.303258	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	634.489	-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	tsat	twall	uvz (m/sec)	ulz (m/sec)	uvr (m/sec)	ulr (m/sec)
			(degree celsius)							
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	3.35960	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	960.787	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	-0.86065	-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	tsat	twall	uvz (m/sec)	ulz (m/sec)	uvr (m/sec)	ulr (m/sec)
			(degree celsius)							
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	-0.29487
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	0.40286
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	tsat	twall	uvz (m/sec)	ulz (m/sec)	uvr (m/sec)	ulr (m/sec)
			(degree celsius)							
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	1.8862	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	tsat	twall	tcan	uvz (m/sec)	ulz (m/sec)
			(degree celsius)						
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

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       IMPLICIT REAL*8 (A-H,O-Z)
19 C
20       DIMENSION ORBI(12000)
21       NORBI = 12000
22       DO 10 II = 1,NORBI
23    10 ORBI(II) = 0.D0
24 C
25       CALL HEAD(ORBI,NORBI)
26       STOP
27       END
```

```
28          BLOCK DATA
29          COMMON /NUMBER/ ZERO,ONE,BIG,SMALL
30          REAL*8 ZERO/0.D0/,ONE/1.D0/,BIG/1.D+07/,SMALL/1.D-08/
31          END
```

```
      SUBROUTINE HEAD(ORBI,NORBI)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /BCX/ ULO
      COMMON /NUMBER/ ZERO,ONE,BIG,SMALL
      COMMON /ERROR/ IERR
      COMMON /RHEA/ TSET(40),TSHSET(40),DTMAX,DTM1
      COMMON /REA/ NN,NP,NB,NW,NTR,NPIN,NPM1,NSET(40),NSHSET(40)
      COMMON /DIM/ DZ(40),DZ1(40),DRO(40),DR1(40),DR2(40),DR3(40),
     *             DR4(40),NI,NJ,NIM1,NIM2,NJM1,NNI,NNJ,NNJJ
      COMMON /CNTRL/ EPS1,EPS2,RES,IT1,IT2,IT3,ITM1,ITM2,IGAUSS
      COMMON /TEMPO/ TIME,DT,DTO,DTLS,NDT
      COMMON /PNTR1/ K(100),M(100)
      DIMENSION ORBI(NORBI)
C
C          THE MATRIX M CONTAINS POINTERS TO THE MATRIX ORBI
C          WHICH CORRESPOND TO THE FIRST ELEMENT OF THE VARIABLE
C          DIMENSIONED ARAYS IN THE FOLLOWING EQUIVALENCE :
C
C     M( 1) = P              M( 2) = PO
C     M( 3) = TV             M( 4) = TVO
C     M( 5) = TL             M( 6) = TLO
C     M( 7) = ALFAN          M( 8) = ALFAO
C     M( 9) = ALFAZ          M(10) = ALFAR
C     M(11) = RHOV           M(12) = RHOL
C     M(13) = RHOVZ          M(14) = RHOLZ
C     M(15) = RHOVR          M(16) = RHOLR
C     M(17) = HV             M(18) = HL
C     M(19) = HVZ            M(20) = HLZ
C     M(21) = HVR            M(22) = HLR
C     M(23) = UVZN           M(24) = ULZN
C     M(25) = UVRN           M(26) = ULRN
C     M(27) = UVZO           M(28) = ULZO
C     M(29) = UVRO           M(30) = ULRO
C     M(31) = UVRZ           M(32) = ULRZ
C     M(33) = UVZR           M(34) = ULZR
C     M(35) = FUVZN          M(36) = FULZN
C     M(37) = FUVRN          M(38) = FULRN
C     M(39) = W(K( 1))       M(40) = W(K( 3))
C     M(41) = W(K( 3))       M(42) = W(K( 4))
C     M(43) = W(K( 5))       M(44) = W(K( 6))
C     M(45) = W(K( 7))       M(46) = W(K( 8))
C     M(47) = W(K( 9))       M(48) = W(K(10))
C     M(49) = W(K(11))       M(50) = W(K(12))
```

```
C      M(51) = W(K(13))        M(52) = W(K(14))
C      M(53) = W(K(15))        M(54) = W(K(16))
C      M(55) = W(K(17))        M(56) = W(K(18))
C      M(57) = W(K(19))        M(58) = W(K(20))
C      M(59) = W(K(21))        M(60) = W(K(22))
C      M(61) = W(K(23))        M(62) = W(K(24))
C      M(63) = DH              M(64) = DHR
C      M(65) = DV              M(66) = QSI
C      M(67) = TS              M(68) = TW
C      M(69) = DTW             M(70) = HCONV
C      M(71) = HCONL           M(72) = HNB
C      M(73) = DPN             M(74) = A1
C      M(75) = A2              M(76) = A3
C      M(77) = A4              M(78) = YP
C      M(79) = B               M(80) = TR
C      M(81) = DTR             M(82) = TWOLD
C      M(83) = BETA            M(84) = GAMMA
C      M(85( = QPP      C      M(86) = AVZD
C      M(87) = ALZD     C      M(88) = AVRD
C      M(89) = ALRD            M(90) = SPPD
C      M(91) = TCAN            M(92) =
C
       CALL READ1
C
       M(1) = 1
       DO 1001 L = 2,79
 1001  M(L) = M(L-1) + NN
       M(80) = M(79) + NB
       M(81) = M(80) + NTR
       M(82) = M(81) + NTR
       M(83) = M(82) + NN
       M(84) = M(83) + NN
       M(85) = M(84) + NN
       M(86) = M(85) + NN
       M(87) = M(86) + NN
       M(88) = M(87) + NN
       M(89) = M(88) + NN
       M(90) = M(89) + NN
       M(91) = M(90) + NN
       NCAN = 4*NI
       M(92) = M(91) + NCAN
C
C
C
       TIME = ZERO
```

```
      CALL READ2(ORBI(M(1)),ORBI(M(3)),ORBI(M(5)),ORBI(M(7)),
     *           ORBI(M(23)),ORBI(M(24)),ORBI(M(25)),ORBI(M(26)),
     *           ORBI(M(63)),ORBI(M(65)),ORBI(M(66)),ORBI(M(80)),
     *           ORBI(M(81)),TINIT,ORBI(M(68)),ORBI(M(90)),ORBI(M(91)),
     *           NP,NTR,NPIN,NPM1,NN,NCAN)
      DO 104 KO = 1,NN
      KL = KO - 1
      DO 103 L = 1,67
  103 K(L) = M(L) + KL
C
      ORBI(K(67)) = SAT(ORBI(K(1)))
      ORBI(K(2)) = ORBI(K(1))
      ORBI(K(4)) = ORBI(K(3))
      ORBI(K(6)) = ORBI(K(5))
      ORBI(K(8)) = ORBI(K(7))
      ORBI(K(30)) = ORBI(K(26))
      ORBI(K(27)) = ORBI(K(23))
      ORBI(K(28)) = ORBI(K(24))
      ORBI(K(29)) = ORBI(K(25))
  104 CONTINUE
C
C
C
C
      NPRI = 0
      TPRI = TINIT + TSET(1)
      TSHPRI = TINIT + TSHSET(1)
      LSH = 1
      L = 1
      NTS = 0
      NIT = 0
      TIME = TINIT
      LPRI = 0
      LSHPRI = 0
    1 CONTINUE
      NDT = 0
      IT3 = 0
      CALL TMSTEP(ORBI,NORBI,
     *            NN,NP,NB,NW,NTR,NPIN,NPM1,NCAN)
C
      NIT = NIT + IT3
      NTS = NTS + 1
C
      IF(IERR.NE.0) GO TO 7
      IF(TIME.LT.TPRI) GO TO 1
```

```
C
      GO TO 8
    2 CONTINUE
C
C
      LPRI = LPRI + 1
      IF(LPRI-NSET(L))3,4,4
    3 TPRI = TPRI + TSET(L)
      GO TO 1
    4 L = L + 1
      NPRI = NPRI + LPRI
      IF(NSET(L))6,6,5
    5 LPRI = 0
      TPRI = TPRI + TSET(L)
      GO TO 1
    6 CALL SAVER(ORBI(M(1)),ORBI(M(3)),ORBI(M(5)),ORBI(M(7)),
     *           ORBI(M(23)),ORBI(M(24)),ORBI(M(25)),ORBI(M(26)),
     *           ORBI(M(80)),ORBI(M(91)),TIME,NTR,NN,NCAN,NI)
      RETURN
    7 CALL ERRMES(TIME)
    8 CONTINUE
C
      QT = ZERO
      FMI = ZERO
      FME = ZERO
      FHI = ZERO
      FHE = ZERO
      KP = 0
      DO 9 J = 1,NJ
      KI = (J-1)*NI + 1
      KE = J*NI
      IF(ORBI(M(24)+KI-1).LT.ZERO) KI = KI + 1
C
      TV = ORBI(M(3)+KI-1)
      TL = ORBI(M(5)+KI-1)
      PP = ORBI(M(1)+KI-1)
      UV = ORBI(M(23)+KI-1)
      UL = ORBI(M(24)+KI-1)
      AA = ORBI(M(7)+KI-1)
      RV = ORBI(M(11)+KI-1)
      RL = ORBI(M(12)+KI-1)
      EV = ORBI(M(17)+KI-1)
      EL = ORBI(M(18)+KI-1)
C
C
```

```
      FMVJ = AA*RV*UV*DR4(J)
      FMLJ = (ONE - AA)*RL*UL*DR4(J)
      HV = EV + PP/RV
      HL = EL + PP/RL
C
      FMI = FMI + FMVJ + FMLJ
      FHI = FHI + FMVJ*HV + FMLJ*HL
C
      TV = ORBI(M(3)+KE-1)
      TL = ORBI(M(5)+KE-1)
      PP = ORBI(M(1)+KE-1)
      AA = ORBI(M(7)+KE-2)
      UV = ORBI(M(23)+KE-1)
      UL = ORBI(M(24)+KE-1)
      RV = ORBI(M(11)+KE-1)
      RL = ORBI(M(12)+KE-1)
      EV = ORBI(M(17)+KE-1)
      EL = ORBI(M(18)+KE-1)
C
C
      FMVJ = AA*RV*UV*DR4(J)
      FMLJ = (ONE - AA)*RL*UL*DR4(J)
      HV = EV + PP/RV
      HL = EL + PP/RL
C
      FME = FME + FMVJ + FMLJ
      FHE = FHE + FMVJ*HV + FMLJ*HL
C
      DO 9 I = 1,NIM2
      KP = KP + 1
      KO = (J-1)*NI + I
      QT = QT + ORBI(M(85)+KP-1)*ORBI(M(65)+KO)*DZ(I+1)*DR4(J)
    9 CONTINUE
      WRITE(6,200) TIME
      WRITE(6,201) NTS,NIT,DT
      WRITE(6,202) FMI,FHI,FME,FHE,QT
      DO 10 J = 1,NJM1
C
      WRITE(6,203) J
      WRITE(6,204)
C
      DO 10 I = 1,NI
      KI = NI - I +1
      KO = (J-1)*NI + KI
      KP = KO - 2*J + 1
```

```
      IF(KI.EQ.1) KP = KP + 1
      IF(KI.EQ.NI)KP = KP - 1
C
      PP = ORBI(M(1)+KO-1)/1.D+05
      TVP = ORBI(M(3)+KO-1) - 273.14
      TLP = ORBI(M(5)+KO-1) - 273.14
      TSP = ORBI(M(67)+KO-1) - 273.14
      TWP = ORBI(M(68)+KP-1) - 273.14
      AP = ORBI(M(7)+KO-1)
      UVZ = ORBI(M(23)+KO-1)
      ULZ = ORBI(M(24)+KO-1)
      UVR = ORBI(M(25)+KO-1)
      ULR = ORBI(M(26)+KO-1)
C
      WRITE(6,205) KI,PP,AP,TVP,TLP,TSP,TWP,
     *             UVZ,ULZ,UVR,ULR
   10 CONTINUE
C
      J = NJ
C
      WRITE(6,203) J
      WRITE(6,206)
C
      DO 11 I = 1,NI
      KI = NI - I +1
      KO = (J-1)*NI + KI
      KP = KO - 2*J + 1
      IF(KI.EQ.1) KP = KP + 1
      IF(KI.EQ.NI)KP = KP - 1
C
      PP = ORBI(M(1)+KO-1)/1.D+05
      TVP = ORBI(M(3)+KO-1) - 273.14
      TLP = ORBI(M(5)+KO-1) - 273.14
      TSP = ORBI(M(67)+KO-1) - 273.14
      TWP = ORBI(M(68)+KP-1) - 273.14
      TCP = ORBI(M(91)+KI-1) - 273.14
      AP = ORBI(M(7)+KO-1)
      UVZ = ORBI(M(23)+KO-1)
      ULZ = ORBI(M(24)+KO-1)
C
      WRITE(6,207) KI,PP,AP,TVP,TLP,TSP,TWP,TCP,
     *             UVZ,ULZ
   11 CONTINUE
C
      IF(IERR.NE.0) GO TO 6
```

```
      GO TO 2
206 FORMAT(1X,'IZ',5X,'P',10X,'VOID',7X,'TV',8X,'TL',7X,'TSAT',
     *       5X,'TWALL',7X,'TCAN',6X,'UVZ',9X,'ULZ'/
     *       6X,'(BAR)',19X,'--------------(DEGREE CELSIUS)',
     *       '--------------',2(5X,'(M/SEC)')/)
207 FORMAT(1X,I2,2X,F9.4,2X,F8.6,5(2X,F8.3),2(2X,F10.5))
200 FORMAT(1H1,10X,'FLOW MAP AT TIME = ',F10.4,' SEC.'/)
201 FORMAT(1X,'NUMBER OF TIME STEPS = ',I10/
     *       1X,'NUMBER OF ITERATIONS = ',I10/
     *       1X,'TIME STEP SIZE = 'D10.4,' SEC.'/)
202 FORMAT(1X,'INLET MASS FLOW RATE  = ',D12.6,' KG/SEC',
     *       6X,'INLET ENTHALPY FLOW  = ',D12.6,' WATT'/
     *       1X,'OUTLET MASS FLOW RATE = ',D12.6,' KG/SEC',
     *       6X,'OUTLET ENTHALPY FLOW = ',D12.6,' WATT'/
     *       1X,'TOTAL HEAT TRANSFERED = ',D12.6,' WATT'/)
203 FORMAT(1H0,40X,'CHANNEL NUMBER ',I5/)
204 FORMAT(1X,'IZ',5X,'P',10X,'VOID',7X,'TV',8X,'TL',7X,'TSAT',
     *       5X,'TWALL',7X,'UVZ',9X,'ULZ',9X,'UVR',9X,'ULR'/
     *       6X,'(BAR)',19X,'---------(DEGREE CELSIUS)--------',
     *       4(5X,'(M/SEC)')/)
205 FORMAT(1X,I2,2X,F9.4,2X,F8.6,4(2X,F8.3),4(2X,F10.5))
C
      END
```

Subroutine read1

```
      SUBROUTINE READ1
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /RHEA/ TSET(40),TSHSET(40),DTMAX,DTM1
      COMMON /REA/ NN,NP,NB,NW,NTR,NPIN,NPM1,NSET(40),NSHSET(40)
      COMMON /DIM/ DZ(40),DZ1(40),DRO(40),DR1(40),DR2(40),DR3(40),
     *             DR4(40),NI,NJ,NIM1,NIM2,NJM1,NNI,NNJ,NNJJ
      COMMON /GRVTY/ GZ,GR
      COMMON /CNTRL/ EPS1,EPS2,RES,IT1,IT2,IT3,ITM1,ITM2,IGAUSS
      COMMON /GAUSS/ NZ,NR,NZM1
      COMMON /TEMPO/ TIME,DT,DTO,DTLS,NDT
      COMMON /ICONST/ NCF,NCC,NG
C
      READ(5,118) NI,NJ,NCF,NCLD
      WRITE(7,118)NI,NJ,NCF,NCLD
      NN = NI*NJ
      NIM1 = NI - 1
      NIM2 = NI - 2
      NJM1 = NJ - 1
      NP = (NI - 2)*NJ
      NB = 21*NN
      NW = 24*NN
      NNJ = NN - NI
      NNJJ = NNJ - NI
      NNI = NN - NJ
      NR = NJ
      NZ = NI - 2
      NZM1 = NZ - 1
      NG = NCF + 1
      NCC = NG + 1
      NPIN = NCC + NCLD
      NPM1 = NPIN - 1
      NTR = NPIN*NP
C
C
      L = 1
    1 CONTINUE
      READ(5,121) NSET(L),TSET(L)
      WRITE(7,121)NSET(L),TSET(L)
      L = L + 1
      IF(L.GT.50) GO TO 2
      IF(NSET(L-1)) 2,2,1
    2 CONTINUE
C
```

```
      GZ = 9.80665
      GR = 0.D0
C
C
      READ(5,119) ITM1,IGAUSS,DTMAX,EPS1,EPS2
      WRITE(7,119)ITM1,IGAUSS,DTMAX,EPS1,EPS2
      DT = DTMAX
C
      ITM2 = ITM1
  118 FORMAT(4I5)
  119 FORMAT(2I10,3D15.9)
  120 FORMAT(2D15.9)
  121 FORMAT(I5,D15.9)
      RETURN
      END
```

```
      SUBROUTINE READ2(P,TV,TL,ALFA,UVZ,ULZ,UVR,ULR,DH,DV,
     *                 QSI,TR,DTR,TINIT,TW,SPPD,TCAN,
     *                 NP,NTR,NPIN,NPM1,NN,NCAN)
      IMPLICIT REAL*8 (A-H,O-Z)
      LOGICAL LP,LDATA,LSS
      COMMON /NUMBER/ ZERO,ONE,BIG,SMALL
      COMMON /BCOND/ TB(51),PNB1(51),PNB2(51),PNB3(51),OMP(51),
     *               PNT1(51),PNT2(51),PNT3(51),OMT(51),ALB1(51),
     *               ALB2(51),ALB3(51),OMA(51),TVB1(51),TVB2(51),
     *               TVB3(51),OMV(51),TLB1(51),TLB2(51),TLB3(51),
     *               OML(51),HNW1(51),HNW2(51),HNW3(51),OMH(51),
     *               LMAX,LP(51)
      COMMON /PSHAPE/ SHAPE(100)
      COMMON /DIM/ DZ(40),DZ1(40),DRO(40),DR1(40),DR2(40),DR3(40),
     *             DR4(40),NI,NJ,NIM1,NIM2,NJM1,NNI,NNJ,NNJJ
      COMMON /PINO/ RODR(20),VP(20),VM(20),RADR,PPP(20)
      COMMON /GCONST/ DIL,RADFU,RADCL
      COMMON /CCONST/ CA0,CA1,CA2,CA3,CB0,CB1,CB2,CB3
      COMMON /FCONST/ FA0,FA1,FA2,FA3,FB0,FB1,FB2,AD,APU,LPLNM(40)
      COMMON /ICONST/ NCF,NCC,NG
      COMMON /PD/ D4,POD2
      COMMON /POVERD/ R
      COMMON /HXCN/ ACOV
      COMMON /STST/ TAFP,LSS
      DIMENSION P(NN),TV(NN),TL(NN),ALFA(NN),UVZ(NN),ULZ(NN),
     *          UVR(NN),ULR(NN),DH(NN),DV(NN),QSI(NN),TR(NTR),
     *          DTR(NTR),TW(NP),SPPD(NN),TCAN(NCAN)
      DIMENSION RAD(20),XIN(5),N(20)
C
      FA0 = 1.81D+06
      FA1 = 3.72D+03
      FA2 = -2.51D0
      FA3 = 6.59D-04
      FB0 = 10.8D0
      FB1 = -8.84D-03
      FB2 = 2.25D-06
C
      CA0 = 4.28D+06
      CA1 = 3.75D+02
      CA2 = -7.45D-03
      CA3 = ZERO
      CB0 = 16.27
      CB1 = ZERO
```

```
      CB2 = ZERO
      CB3 = ZERO
C
C
C
      TB(1) = ZERO
      L = 2
    2 CONTINUE
      READ(5,1001) LP(L),TB(L)
      WRITE(7,1001)LP(L),TB(L)
      IF(TB(L).LE.TB(L-1)) GO TO 3
      READ(5,1002) PNB1(L),PNB2(L),PNB3(L),OMP(L)
      READ(5,1002) PNT1(L),PNT2(L),PNT3(L),OMT(L)
      READ(5,1002) ALB1(L),ALB2(L),ALB3(L),OMA(L)
      READ(5,1002) TVB1(L),TVB2(L),TVB3(L),OMV(L)
      READ(5,1002) TLB1(L),TLB2(L),TLB3(L),OML(L)
      READ(5,1002) HNW1(L),HNW2(L),HNW3(L),OMH(L)
C
      WRITE(7,1002)PNB1(L),PNB2(L),PNB3(L),OMP(L)
      WRITE(7,1002)PNT1(L),PNT2(L),PNT3(L),OMT(L)
      WRITE(7,1002)ALB1(L),ALB2(L),ALB3(L),OMA(L)
      WRITE(7,1002)TVB1(L),TVB2(L),TVB3(L),OMV(L)
      WRITE(7,1002)TLB1(L),TLB2(L),TLB3(L),OML(L)
      WRITE(7,1002)HNW1(L),HNW2(L),HNW3(L),OMH(L)
C
      L = L + 1
      IF(L.GT.51) GO TO 3
      GO TO 2
    3 CONTINUE
      LMAX = L
      DO 4 KO = 1,NN
      QSI(KO) = (4.*D/(PITCH - D))**2
    4 CONTINUE
C
      READ(5,1003) NROW,PITCH,D,E
      WRITE(7,1003)NROW,PITCH,D,E
C
      POVD = PITCH/D
      POD2 = POVD*POVD
      D4 = 4./D
      R = -16.15 + 24.96*POVD - 8.55*POVD*POVD
C
      READ(5,1004) (N(J),J=1,19)
      WRITE(7,1004)(N(J),J=1,19)
      KRES = 0
```

```
      5 CONTINUE
        READ(5,1005) LDATA,(XIN(K),K=1,5)
        WRITE(7,1005)LDATA,(XIN(K),K=1,5)
        IF(.NOT.LDATA) GO TO 205
        DO 105 I = 1,5
        KO = KRES + I
        IF(KO.GT.NI) GO TO 5
        DZ(KO) = XIN(I)
  105 CONTINUE
        KRES = KRES + 5
        GO TO 5
  205 CONTINUE
C
        KRES = 3*NI
  305 CONTINUE
        READ(5,1005) LDATA,(XIN(K),K=1,5)
        WRITE(7,1005)LDATA,(XIN(K),K=1,5)
        IF(.NOT.LDATA) GO TO 505
        DO 405 I = 1,5
        KO = KRES + I
        IF(KO.GT.NCAN) GO TO 305
        TCAN(KO) = XIN(I)
  405 CONTINUE
        KRES = KRES + 5
        GO TO 305
  505 CONTINUE
        KRES = 0
      6 CONTINUE
        READ(5,1005) LDATA,(XIN(K),K=1,5)
        WRITE(7,1005)LDATA,(XIN(K),K=1,5)
        IF(.NOT.LDATA) GO TO 206
        DO 106 I = 1,5
        KO = KRES + I
        IF(KO.GT.NN) GO TO 6
        SHAPE(KO) = XIN(I)
  106 CONTINUE
        KRES = KRES + 5
        GO TO 6
  206 CONTINUE
        KRES = 0
  306 CONTINUE
        READ(5,1005) LDATA,(XIN(K),K=1,5)
        WRITE(7,1005)LDATA,(XIN(K),K=1,5)
        IF(.NOT.LDATA) GO TO 506
        DO 406 I = 1,5
```

```
      KO = KRES + I
      IF(KO.GT.NN) GO TO 306
      SPPD(KO) = XIN(I)
  406 CONTINUE
      KRES = KRES + 5
      GO TO 306
  506 CONTINUE
C
      DZ1(1) = DZ(1)
      DO 7 I = 2,NI
      DZ1(I) = (DZ(I) + DZ(I-1))/2.D0
    7 CONTINUE
C
      A1 = DSQRT(3.D0)/2.D0
      A2 = 3.1415927/4.D0
      W = PITCH - D
C
      X = (PITCH*PITCH*A1 - (D*D + W*W)*A2)/A2/D
      XI = 4.D0/X
C
      DO 8 J = 1,NJM1
      DO 8 I = 1,NI
      KO = (J-1)*NI + I
      DH(KO) = X
      DV(KO) = XI
    8 CONTINUE
C
      DO 9 J = 2,NJM1
C
      N41 = N(J) - 1
      N42 = N(J-1) - 1
      DN4 = N41*N41 - N42*N42
      DR4(J) = DN4*X*A2*D*3.D0
C
      NX = N(J) - N(J-1)
      NX1 = 2*N41
      NX2 = (2*N42 + NX)*NX
      DNX1 = NX1
      DR1(J) = DNX1/NX2/PITCH/A1
      DR2(J) = 2.D0*N42/NX2/PITCH/A1
      DRO(J) = PITCH*A1*NX
    9 CONTINUE
C
      DN4 = (N(1) - 1)*(N(1) - 1)
      DR4(1) = DN4*X*A2*D*3.D0
```

```
C
      DR1(1) = 2.D0/PITCH/A1/(N(1)-1)
      DR2(1) = 0.D0
      DRO(1) = PITCH*A1*(N(1)-1)
C
      B1 = (N(NJM1) + NROW - 2)
      B2 = (NROW - N(NJM1))
      B3 = (NROW - 1)
C
      XX = B1*B2/2.D0 + B3/2.D0 + 1.D0/6.D0
      PT = B3*PITCH + (D/2.D0 + E)/A1 + A2*D*XX*4.D0
      AC = (B1*PITCH + (D/2.D0 + E)/A1)*(B2*PITCH*A1 + D/2.D0 + E)*
     *      0.5D0 - A2*(D*D + E*E)*XX
      Y = 4.D0*AC/PT
      PP = A2*D*XX*4.D0
      YY = PP/AC
      ARM = (ONE - A2/A1*(D*D + W*W)/(PITCH*PITCH))*
     *      (N(NJM1) - 1)*PITCH
C
      DR1(NJ) = ZERO
      DR2(NJ) = ARM/AC
      DRO(NJ) = B2*PITCH + D/2.D0 + E
      DR4(NJ) = AC*6.D0
      ACOV = (B3*PITCH + (D/2.D0 + E)/A1)/AC
C
      DO 10 I = 1,NI
      KO = NJM1*NI + I
      DH(KO) = Y
      DV(KO) = YY
   10 CONTINUE
C
      DR3(NJ) = DRO(NJ)
      DO 11 J = 1,NJM1
      DR3(J) = (DRO(J) + DRO(J+1))/2.D0
   11 CONTINUE
      KRES = 0
   12 CONTINUE
      READ(5,1005) LDATA,(XIN(K),K=1,5)
      WRITE(7,1005)LDATA,(XIN(K),K=1,5)
      IF(.NOT.LDATA) GO TO 212
      DO 112 I = 1,5
      KO = KRES + I
      IF(KO.GT.NPIN) GO TO 12
      PPP(KO) = XIN(I)
  112 CONTINUE
```

```
      KRES = KRES + 5
      GO TO 12
  212 CONTINUE
C
      READ(5,1006) AD,APU,DIL
      READ(5,1007) (LPLNM(K),K = 1,39)
      READ(5,1008) RADR,THC,THG
C
      WRITE(7,1006)AD,APU,DIL
      WRITE(7,1007)(LPLNM(K),K = 1,39)
      WRITE(7,1008)RADR,THC,THG
C
      RADFU = RADR - THG - THC
      RADCL = RADFU + THG
      NCLD = NPIN - NCC
      DRF = RADFU/NCF
      DRC = THC/NCLD
      TAFP = RADFU*RADFU/D
C
      RAD(1) = ZERO
      DO 14 K = 1,NCF
      RAD(K+1) = RAD(K) + DRF
   14 CONTINUE
      RAD(NG+1) = RAD(NG) + THG
      DO 15 K = NCC,NPM1
      RAD(K+1) = RAD(K) + DRC
   15 CONTINUE
      DO 16 K = 1,NPM1
      IF(K.EQ.NG) RODR(K) = (RAD(K+1) + RAD(K))/2.D0
      IF(K.NE.NG) RODR(K) = (RAD(K+1)+RAD(K))/(RAD(K+1)-RAD(K))/2.D0
   16 CONTINUE
C
      VM(1) = ZERO
      VP(1) = DRF*DRF/8.D0
      RM = (RADR + RAD(NPM1))/2.D0
      VM(NPIN) = (RADR*RADR + W*W/4.D0- RM*RM)/2.D0
      VP(NPIN) = ZERO
      DO 17 K = 2,NPM1
      RP = (RAD(K+1) + RAD(K))/2.D0
      RM = (RAD(K) + RAD(K-1))/2.D0
      VP(K) = (RP*RP -RAD(K)*RAD(K))/2.D0
      VM(K) = (RAD(K)*RAD(K) - RM*RM)/2.D0
   17 CONTINUE
C
      READ(5,1009) LSS,TINIT
```

```
      TB(1) = ZERO
      IF (LSS) GO TO 19
      DO 1 KO = 1,NN
      READ(5,1000) KCHECK,TV(KO),TL(KO),P(KO),ALFA(KO)
      READ(5,1000) KCHECK,UVZ(KO),ULZ(KO),UVR(KO),ULR(KO)
      IF(KCHECK.EQ.KO)GO TO 1
      IERR = 4
      RETURN
    1 CONTINUE
      KRES = 0
   13 CONTINUE
      READ(5,1005) LDATA,(XIN(K),K=1,5)
      IF(.NOT.LDATA) GO TO 213
      DO 113 I = 1,5
      KO = KRES + I
      IF(KO.GT.NTR) GO TO 13
      TR(KO) = XIN(I)
  113 CONTINUE
      KRES = KRES + 5
      GO TO 13
  213 CONTINUE
C
      KRES = 0
  313 CONTINUE
      READ(5,1005) LDATA,(XIN(K),K=1,5)
      IF(.NOT.LDATA) GO TO 513
      DO 413 I = 1,5
      KO = KRES + I
      K3 = KO + 2*NI
      IF(KO.GT.NI) GO TO 313
      TCAN(KO) = XIN(I)
      TCAN(K3) = XIN(I)
  413 CONTINUE
      KRES = KRES + 5
      GO TO 313
  513 CONTINUE
C
      DO 18 I = 1,NIM2
      DO 18 J = 1,NJ
      KP = (J-1)*NIM2 + I
      KT = KP*NPIN
      TW(KP) = TR(KT)
   18 CONTINUE
      RETURN
   19 CONTINUE
```

```
      READ(5,1010) PIN,POUT,TIN,TAV
      QPP = HNW2(2)*RADFU*RADFU/RADR/2.D0
      CALL SS (PIN,POUT,TIN,TAV,QPP,P,TV,TL,UVZ,ULZ,UVR,ULR,ALFA,
     *         TW,TR,DTR,DH,DV,NN,NP,NTR,NPIN,NPM1)
      RETURN
C
 1000 FORMAT(I5,4D15.9)
 1001 FORMAT(L1,F15.5)
 1002 FORMAT(4D15.9)
 1003 FORMAT(I5,3D15.9)
 1004 FORMAT(19I4)
 1005 FORMAT(L1,5D15.9)
 1006 FORMAT(3D15.9)
 1007 FORMAT(39I2)
 1008 FORMAT(3D15.9)
 1009 FORMAT(L1,D15.9)
 1010 FORMAT(4D15.9)
      END
```

```
      SUBROUTINE SS(PIN,POUT,TIN,TAV,Q,P,TV,TL,UVZ,ULZ,UVR,ULR,ALFA,
     *              TW,TR,DTR,DH,DV,NN,NP,NTR,NPIN,NPM1)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /NUMBER/ ZERO,ONE,BIG,SMALL
      COMMON /DIM/ DZ(40),DZ1(40),DRO(40),DR1(40),DR2(40),DR3(40),
     *             DR4(40),NI,NJ,NIM1,NIM2,NJM1,NNI,NNJ,NNJJ
      COMMON /PSHAPE/ SHAPE(100)
      COMMON /GRVTY/ GZ,GR
      DIMENSION PROP(3,4)
      DIMENSION P(NN),TV(NN),TL(NN),UVZ(NN),ULZ(NN),UVR(NN),ULR(NN),
     *          ALFA(NN),TW(NP),TR(NTR),DTR(NTR),DH(NN),DV(NN)
C
C             SUBROUTINE SS PUTS AN INITIAL GUESS IN THE VARIABLES
C             TV,TL,P,UVZ,ULZ,UVR,ULR,ALFA AND TR, IN ORDER TO
C             ACCELERATE THE CONVERGENCE TO THE STEADY STATE PROBLEM.
C
      H = ZERO
      DO 1 I = 2,NI
      H = H + DZ1(I)
    1 CONTINUE
      DP = (PIN - POUT)/H
C
      CALL STATE (TAV,TAV,PIN,PROP,0)
      RHO = PROP(1,2)
      DPG = DP - RHO*GZ
C
      A = (RHO*DH(2)/VISCL(TAV))**.2*DH(2)/RHO/.100
      X = ONE/1.800
      V = (A*DPG)**X
C
      DO 2 J = 1,NJM1
      DO 2 I = 1,NI
      KO = (J-1)*NI + I
      ULZ(KO) = V
      UVZ(KO) = V
      ULR(KO) = ZERO
      UVR(KO) = ZERO
      ALFA(KO) = ZERO
    2 CONTINUE
C
      A = (RHO*DH(NNJ+2)/VISCL(TAV))**.2*DH(NNJ+2)/RHO/.100
      V = (A*DPG)**X
C
```

```
      DO 3 I = 1,NI
      KO = NNJ + I
      ULZ(KO) = V
      UVZ(KO) = V
      ULR(KO) = ZERO
      UVR(KO) = ZERO
      ALFA(KO) = ZERO
    3 CONTINUE
C
      TL(1) = TIN
      TV(1) = TIN
      P(1) = PIN
      DO 4 J = 1,NJ
      KO = J*NI - NIM1
      TL(KO) = TIN
      TV(KO) = TIN
      P(KO) = PIN
C
      DO 4 I = 2,NI
      KO = (J-1)*NI + I
      P(KO) = P(KO-1) - DP*DZ1(I)
      UXX = ULZ(KO)
      IF(UXX.EQ.ZERO) UXX = ONE
      TL(KO) = TL(KO-1) + Q*SHAPE(KO)*DV(KO)*DZ1(I)/RHO/UXX/
     /          CPL(TL(KO-1))
      TV(KO) = TL(KO)
    4 CONTINUE
C
      DT = .100
C
      DO 7 J = 1,NJ
      DO 7 I = 2,NIM1
      KO = (J-1)*NI + I
      KP = KO + 1 - J*2
      KT = (KP-1)*NPIN + 1
      KR = KP*NPIN
C
      TW(KP) = TL(KO)
      TS = SAT(P(KO))
      CALL HTCF (P(KO),TV(KO),TL(KO),ALFA(KO),PROP(1,1),
     *           PROP(1,2),PROP(1,3),PROP(1,4),DH(KO),TS,TW(KP),
     *           HCONV,HCONL,HNB,UVZ(KO),ULZ(KO))
C
      DO 5 K = 1,NPIN
      KTR = (KP-1)*NPIN + K
```

```
      TR(KTR) = TW(KP)
    5 CONTINUE
    6 CONTINUE
      TTR = TR(KT)
      CALL FPROP(TR(KT),NPIN,NPM1,I)
      CALL FPIN (TV(KO),TL(KO),TS,TW(KP),DTW,HCONV,HCONL,HNB,
     *           TR(KT),DTR(KT),DT,NPIN,NPM1,KO)
C
      TR(KR) = TW(KP)
      DO 16 KK = 1,NPM1
      KS = KR - KK
      TR(KS) = TR(KS) - DTR(KS)*TR(KS+1)
   16 CONTINUE
      TTR = DABS(TTR - TR(KT))/DT
      IF(TTR.GT.ONE) GO TO 6
    7 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE TMSTEP(O,NO,
     *                  NN,NP,NB,NW,NTR,NPIN,NPM1,NCAN)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /ERROR/ IERR
      COMMON /NUMBER/ ZERO,ONE,BIG,SMALL
      COMMON /RHEA/ TSET(40),TSHSET(40),DTMAX,DTM1
      COMMON /CNTRL/ EPS1,EPS2,RES,IT1,IT2,IT3,ITM1,ITM2,IGAUSS
      COMMON /DIM/ DZ(40),DZ1(40),DRO(40),DR1(40),DR2(40),DR3(40),
     *             DR4(40),NI,NJ,NIM1,NIM2,NJM1,NNI,NNJ,NNJJ
      COMMON /TEMPO/ TIME,DT,DTO,DTLS,NDT
      COMMON /PNTR1/K(100),M(100)
C
C
      DIMENSION O(NO)
C
C
C
      DTLS = DT
      TMS = ZERO
      IERR = 0
      DO 100 J = 1,NJ
      DO 100 I = 2,NI
      KO = (J-1)*NI + I - 1
      K23 = KO + M(23)
      K24 = KO + M(24)
      K25 = KO + M(25)
      JO = J
C
      TSVZ = DABS(O(K23)/DZ1(I))
      TSLZ = DABS(O(K24)/DZ1(I))
      TSVR = DABS(O(K25)/DR3(J))
C
      TMS = DMAX1(TSVZ,TSLZ,TSVR,TMS)
  100 CONTINUE
C
      IF(TMS) 101,101,102
  101 DT = DTMAX
      GO TO 103
  102 DT = 0.95D0/TMS
      DT = DMIN1(DTMAX,DT,2.0*DTLS)
  103 CONTINUE
      IT2 = 0
      NDT = 0
```

```
      TIME = TIME+DT
C
      DO 104 KO = 1,NN
      KL = KO - 1
      K68 = M(68) + KL
      K82 = M(82) + KL
      O(K82) = O(K68)
  104 CONTINUE
C
      CALL DONOR(O(M( 1)),O(M( 2)),O(M( 3)),O(M( 4)),O(M( 5)),
     *           O(M( 6)),O(M( 7)),O(M( 8)),O(M( 9)),O(M(10)),
     *           O(M(11)),O(M(12)),O(M(13)),O(M(14)),O(M(15)),
     *           O(M(16)),O(M(17)),O(M(18)),O(M(19)),O(M(20)),
     *           O(M(21)),O(M(22)),O(M(23)),O(M(24)),O(M(25)),
     *           O(M(26)),O(M(27)),O(M(28)),O(M(29)),O(M(30)),
     *           O(M(31)),O(M(32)),O(M(33)),O(M(34)),O(M(41)),
     *           O(M(42)),O(M(47)),O(M(48)),O(M(52)),O(M(53)),
     *           O(M(58)),O(M(59)),O(M(73)),O(M(86)),O(M(87)),
     *           O(M(88)),O(M(89)),NN,NP)
C
    1 CONTINUE
      CALL BC(O(M( 1)),O(M( 3)),O(M( 5)),O(M( 7)),TIME,
     *        O(M(24)),NN,NI,NIM1)
      CALL WS(O(M( 2)),O(M( 4)),O(M( 6)),O(M( 8)),O(M( 9)),
     *        O(M(10)),O(M(11)),O(M(12)),O(M(13)),O(M(14)),
     *        O(M(15)),O(M(16)),O(M(17)),O(M(18)),O(M(27)),
     *        O(M(28)),O(M(29)),O(M(30)),O(M(39)),O(M(40)),
     *        O(M(43)),O(M(44)),O(M(45)),O(M(46)),O(M(47)),
     *        O(M(48)),O(M(49)),O(M(50)),O(M(51)),
     *        O(M(54)),O(M(55)),O(M(56)),O(M(57)),O(M(58)),
     *        O(M(59)),O(M(60)),O(M(61)),O(M(62)),O(M(63)),
     *        O(M(65)),O(M(66)),O(M(90)),NN)
C
      DO 1001 I = 2,NIM1
      DO 1001 J = 1,NJ
      KO = (J-1)*NI + I - 1
      KP = KO + 1 -J*2
      KT = KP*NPIN
      K01 = M( 1) + KO
      K03 = M( 3) + KO
      K05 = M( 5) + KO
      K07 = M( 7) + KO
      K11 = M(11) + KO
      K12 = M(12) + KO
      K17 = M(17) + KO
```

```
      K18 = M(18) + KO
      K23 = M(23) + KO
      K24 = M(24) + KO
      K63 = M(63) + KO
      K67 = M(67) + KO
      K68 = M(68) + KP
      K69 = M(69) + KP
      K70 = M(70) + KP
      K71 = M(71) + KP
      K72 = M(72) + KP
      K80 = M(80) + KT
      K81 = M(81) + KT
      KF = KO + 1
C
      UV = (O(K23) + O(K23 + 1))/2.DO
      UL = (O(K24) + O(K24 + 1))/2.DO
C
      CALL HTCF (O(K01),O(K03),O(K05),O(K07),O(K11),O(K12),
     *           O(K17),O(K18),O(K63),O(K67),O(K68),O(K70),
     *           O(K71),O(K72),UV,UL)
      CALL FPROP(O(K80),NPIN,NPM1,I)
      CALL FPIN (O(K03),O(K05),O(K67),O(K68),O(K69),O(K70),
     *           O(K71),O(K72),O(K80),O(K81),DT,NPIN,NPM1,KF)
 1001 CONTINUE
C
      CALL THXCN(O(M(3)),O(M(5)),O(M(70)),O(M(71)),O(M(91)),
     *           DT,NN,NI,NJ,NCAN,NIM1,NIM2)
C
      IF(IERR.NE.0) RETURN
      IT2 = 0
    2 CONTINUE
      IT2 = IT2+1
      CALL ONESTP(O(M( 1)),O(M( 2)),O(M( 3)),O(M( 5)),O(M( 7)),
     *            O(M( 8)),O(M( 9)),O(M(10)),O(M(11)),O(M(12)),
     *            O(M(17)),O(M(18)),O(M(19)),O(M(20)),O(M(21)),
     *            O(M(22)),O(M(23)),O(M(24)),O(M(25)),O(M(26)),
     *            O(M(35)),O(M(36)),O(M(37)),O(M(38)),O(M(39)),
     *            O(M(65)),O(M(67)),O(M(68)),O(M(69)),O(M(70)),
     *            O(M(71)),O(M(72)),O(M(73)),O(M(74)),O(M(75)),
     *            O(M(76)),O(M(77)),O(M(78)),O(M(79)),O(M(83)),
     *            O(M(84)),O(M(86)),O(M(87)),O(M(88)),O(M(89)),
     *            O(M(91)),DT,NN,NB,NP,NW,NCAN)
      IF(IERR.NE.0) GO TO 5
      IF(RES.GT.EPS1) GO TO 4
      IT3 = IT3 + IT2
```

```
      CALL FTP(O(M( 3)),O(M( 5)),O(M(67)),O(M(68)),O(M(70)),
     *         O(M(71)),O(M(72)),O(M(80)),O(M(81)),O(M(85)),
     *         NI,NJ,NN,NP,NTR,NPM1,NIM2,NPIN)
      CALL THXCNO(O(M(91)),NCAN,NI)
      RETURN
    4 IF(IT2.LT.ITM2) GO TO 2
    5 CONTINUE
      NDT = NDT+1
      IT3 =IT3+IT2
      IT2 = 0
      TIME = TIME-DT
      DTO = DT
      DT = DT*0.1
      IF(DT.LT.1.D-07) IERR = 21
      TIME = TIME+DT
      DO 6 KO = 1,NN
      KL = KO - 1
      K01 = M( 1) + KL
      K02 = M( 2) + KL
      K03 = M( 3) + KL
      K04 = M( 4) + KL
      K05 = M( 5) + KL
      K06 = M( 6) + KL
      K07 = M( 7) + KL
      K08 = M( 8) + KL
      K23 = M(23) + KL
      K24 = M(24) + KL
      K25 = M(25) + KL
      K26 = M(26) + KL
      K27 = M(27) + KL
      K28 = M(28) + KL
      K29 = M(29) + KL
      K30 = M(30) + KL
      K73 = M(73) + KL
      K68 = M(68) + KL
      K82 = M(82) + KL
C
      O(K03) = O(K04)
      O(K05) = O(K06)
      O(K07) = O(K08)
      O(K23) = O(K27)
      O(K24) = O(K28)
      O(K25) = O(K29)
      O(K26) = O(K30)
      O(K01) = O(K02)
```

```
      O(K73) = ZERO
      O(K68) = O(K82)
    6 CONTINUE
      CALL FTP(O(M( 3)),O(M( 5)),O(M(67)),O(M(68)),O(M(70)),
     *         O(M(71)),O(M(72)),O(M(80)),O(M(81)),O(M(85)),
     *         NI,NJ,NN,NP,NTR,NPM1,NIM2,NPIN)
      IF(IERR.GT.20) RETURN
      IF(NDT.GT.3) RETURN
      IERR = 0
      GO TO 1
      END
```

```
      SUBROUTINE DONOR(P,PO,TV,TVO,TL,TLO,ALFAN,ALFAO,ALFAZ,ALFAR,
     *                 RHOV,RHOL,RHOVZ,RHOLZ,RHOVR,RHOLR,
     *                 HV,HL,HVZ,HLZ,HVR,HLR,UVZN,
     *                 ULZN,UVRN,ULRN,UVZO,ULZO,UVRO,ULRO,
     *                 UVRZ,ULRZ,UVZR,ULZR,WZ1,WZ2,
     *                 WZ7,WZ8,WR1,WR2,WR7,WR8,DPN,AVZD,ALZD,
     *                 AVRD,ALRD,NN,NP)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /NUMBER/ ZERO,ONE,BIG,SMALL
      COMMON / GRVTY/ GZ,GR
      COMMON /DIM/ DZ(40),DZ1(40),DRO(40),DR1(40),DR2(40),DR3(40),
     *             DR4(40),NI,NJ,NIM1,NIM2,NJM1,NNI,NNJ,NNJJ
      COMMON /TEMPO/ TIME,DT,DTO,DTLS,NDT
      DIMENSION P(NN),PO(NN),TV(NN),TVO(NN),TL(NN),TLO(NN),ALFAN(NN),
     *          ALFAO(NN),ALFAZ(NN),ALFAR(NN),RHOV(NN),RHOL(NN),
     *          RHOVZ(NN),RHOLZ(NN),RHOVR(NN),RHOLR(NN),
     *          HV(NN),HL(NN),HVZ(NN),HLZ(NN),HVR(NN),HLR(NN),
     *          UVZN(NN),ULZN(NN),UVRN(NN),ULRN(NN),UVZO(NN),
     *          ULZO(NN),UVRO(NN),ULRO(NN),UVRZ(NN),ULRZ(NN),
     *          UVZR(NN),ULZR(NN),WZ1(NN),WZ2(NN),WZ7(NN),WZ8(NN),
     *          WR1(NN),WR2(NN),WR7(NN),WR8(NN),DPN(NN),
     *          AVZD(NN),ALZD(NN),AVRD(NN),ALRD(NN)
      DIMENSION PROP(3,4),S(5,2)
C
      IFLAG = 0
      DTR = DT/DTLS
      DO 101 KO = 1,NN
      CALL STATE (TV(KO),TL(KO),P(KO),PROP,IFLAG)
C
      IF(ALFAN(KO).GT.1.D-08) GO TO 100
      ALFAN(KO) = ZERO
      TV(KO) = TL(KO)
  100 CONTINUE
      TVO(KO) = TV(KO)
      TLO(KO) = TL(KO)
      PO(KO) = P(KO)
      RHOV(KO)   = PROP(1,1)
      RHOL(KO)   = PROP(1,2)
      HV(KO)     = PROP(1,3)
      HL(KO)     = PROP(1,4)
      ALFAO(KO)  = ALFAN(KO)
C
      IF(DABS(UVRN(KO)).LT.1.D-10) UVRN(KO) = ZERO
```

```
      IF(DABS(ULRN(KO)).LT.1.D-10) ULRN(KO) = ZERO
      UVZO(KO) = UVZN(KO)
      UVRO(KO) = UVRN(KO)
      ULZO(KO) = ULZN(KO)
      ULRO(KO) = ULRN(KO)
  101 CONTINUE
      DO 1101 J = 1,NJ
      DO 1101 I = 2,NIM1
      KO = (J-1)*NI + I
      KP = (J-1)*NIM1 - J + I
 1101 DPN(KP) = P(KO) - PO(KO)
      DO 2101 I = 2,NI
      II = I - 1
      DO 2101 J = 1,NJM1
      JJ = J + 1
      KO = (J - 1)*NI + I
      KI = KO - 1
      KJ = KO + NI
C
      DZM = DZ(I) + DZ(II)
      DRM = DRO(J) + DRO(JJ)
C
      ALFAZ(KO) = (ALFAO(KO)*DZ(I) + ALFAO(KI)*DZ(II))/DZM
      RHOVZ(KO) = (RHOV(KO)*DZ(I) + RHOV(KI)*DZ(II))/DZM
      RHOLZ(KO) = (RHOL(KO)*DZ(I) + RHOL(KI)*DZ(II))/DZM
C
      ALFAR(KO) = (ALFAO(KO)*DRO(J) + ALFAO(KJ)*DRO(JJ))/DRM
      RHOVR(KO) = (RHOV(KO)*DRO(J) + RHOV(KJ)*DRO(JJ))/DRM
      RHOLR(KO) = (RHOL(KO)*DRO(J) + RHOL(KO)*DRO(JJ))/DRM
 2101 CONTINUE
C
      DO 3101 J = 1,NJ
      KO = (J - 1)*NI + 1
      ALFAZ(KO) = ALFAO(KO)
      RHOVZ(KO) = RHOV(KO)
      RHOLZ(KO) = RHOL(KO)
 3101 CONTINUE
C
      DO 4101 I = 2,NI
      KO = NNJ + I
      II = I - 1
      KI = KO -1
      DZM = DZ(I) + DZ(II)
C
      ALFAZ(KO) = (ALFAO(KO)*DZ(I) + ALFAO(KI)*DZ(II))/DZM
```

```
      RHOVZ(KO) = (RHOV(KO)*DZ(I) + RHOV(KI)*DZ(II))/DZM
      RHOLZ(KO) = (RHOL(KO)*DZ(I) + RHOL(KI)*DZ(II))/DZM
 4101 CONTINUE
      DO 102 J = 2,NJM1
      DO 102 I = 2,NIM1
      KO = (J-1)*NI+I
C
      UVRZ(KO) = (UVRO(KO)+UVRO(KO-1)+UVRO(KO-NI)+UVRO(KO-1-NI))/4.
      ULRZ(KO) = (ULRO(KO)+ULRO(KO-1)+ULRO(KO-NI)+ULRO(KO-1-NI))/4.
      UVZR(KO) = (UVZO(KO)+UVZO(KO+1)+UVZO(KO+NI)+UVZO(KO+1+NI))/4.
      ULZR(KO) = (ULZO(KO)+ULZO(KO+1)+ULZO(KO+NI)+ULZO(KO+1+NI))/4.
C
      KD = 0
      IF(UVZO(KO).GE.ZERO) KD = -1
      KN = KO+KD
      IO = I +KD
C
      HVZ(KO) = HV(KN)
      AVZD(KO) = ALFAO(KN)
      WZ1(KO) = ALFAO(KN)*RHOV(KN)
      WZ7(KO) = (UVZO(KN+1)-UVZO(KN))/DZ(IO)*UVZO(KO)
C
      KD = 0
      IF(ULZO(KO).GE.ZERO) KD = -1
      IO = I+KD
      KN = KO+KD
      HLZ(KO) = HL(KN)
      ALZD(KO) = ONE - ALFAO(KN)
      WZ2(KO) = (ONE-ALFAO(KN))*RHOL(KN)
      WZ8(KO) = (ULZO(KN+1)-ULZO(KN))/DZ(IO)*ULZO(KO)
C
C
      KD = NI
      IF(UVRO(KO).GE.ZERO) KD = 0
      JO = J + KD/NI
      KN = KO + KD
C
      HVR(KO) = HV(KN)
      AVRD(KO) = ALFAO(KN)
      WR1(KO) = ALFAO(KN)*RHOV(KN)
      WR7(KO) = (UVRO(KN)-UVRO(KN-NI))/DRO(JO)*UVRO(KO)
      IF(J.EQ.NJM1) WR7(KO) = -UVRO(KN-NI)*UVRO(KO)/DRO(JO)
C
C
      KD = NI
```

```
      IF(ULRO(KO).GE.ZERO) KD = 0
      JO = J + KD/NI
      KN = KO+KD
C
      HLR(KO) = HL(KN)
      ALRD(KO) = ONE - ALFAO(KN)
      WR2(KO) = (ONE-ALFAO(KN))*RHOL(KN)
      WR8(KO) = (ULRO(KN)-ULRO(KN-NI))/DRO(JO)*ULRO(KO)
      IF(J.EQ.NJM1) WR8(KO) = -ULRO(KN-NI)*ULRO(KO)/DRO(JO)
C
C
      KD = NI
      IF(UVRZ(KO).GE.ZERO) KD = 0
      KN = KO + KD
      JO = J - 1 + KD/NI
C
      WZ7(KO) = ((UVZO(KN)-UVZO(KN-NI))*UVRZ(KO)/DR3(JO) +
     +          WZ7(KO) + GZ)*ALFAZ(KO)*RHOVZ(KO)
C
      KD = NI
      IF(ULRZ(KO).GE.ZERO) KD = 0
      KN = KO + KD
      JO = J - 1 + KD/NI
C
      WZ8(KO) = ((ULZO(KN)-ULZO(KN-NI))*ULRZ(KO)/DR3(JO) +
     +          WZ8(KO) + GZ)*(ONE-ALFAZ(KO))*RHOLZ(KO)
C
      KD = 0
      IF(UVZR(KO).GE.ZERO) KD = -1
      KN = KO + KD
      IO = I + KD + 1
C
      WR7(KO) = ((UVRO(KN+1)-UVRO(KN))*UVZR(KO)/DZ1(IO) +
     +          WR7(KO) + GR)*ALFAR(KO)*RHOVR(KO)
C
      KD = 0
      IF(ULZR(KO).GE.ZERO) KD = -1
      KN = KO + KD
      IO = I + KD + 1
C
      WR8(KO) = ((ULRO(KN+1)-ULRO(KN))*ULZR(KO)/DZ1(IO) +
     +          WR8(KO) + GR)*(ONE-ALFAR(KO))*RHOLR(KO)
  102 CONTINUE
C
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)
```

```
      WZ7(KO) = (UVZO(KN+1)-UVZO(KN))/DZ(IO)*UVZO(KO)
C
      KD = 0
      IF(ULZO(KO).GE.ZERO) KD = -1
      IO = KO + KD
      KN = KO+KD
      HLZ(KO) = HL(KN)
      ALZD(KO) = ONE - ALFAO(KN)
      WZ2(KO) = (ONE-ALFAO(KN))*RHOL(KN)
      WZ8(KO) = (ULZO(KN+1)-ULZO(KN))/DZ(IO)*ULZO(KO)
C
C
      KD = 0
      IF(UVZR(KO).GE.ZERO) KD = -1
      KN = KO + KD
      IO = KO + KD + 1
      WR7(KO) = (UVRO(KN+1)-UVRO(KN))*UVZR(KO)/DZ1(IO)
C
      KD = 0
      IF(ULZR(KO).GE.ZERO) KD = -1
      KN = KO + KD
      IO = KO + KD + 1
      WR8(KO) = (ULRO(KN+1)-ULRO(KN))*ULZR(KO)/DZ1(IO)
C
      IF(UVRO(KO))104,105,105
  104 KN = KO+NI
      JO = 2
C
      HVR(KO) = HV(KN)
      AVRD(KO) = ALFAO(KN)
      WR1(KO) = ALFAO(KN)*RHOV(KN)
      WR7(KO) = ((UVRO(KN) - UVRO(KO))/DRO(JO)*UVRO(KO)+
     +            WR7(KO)+GR)*ALFAR(KO)*RHOVR(KO)
C
C
      GO TO 106
  105 HVR(KO) = HV(KO)
      AVRD(KO) = ALFAO(KO)
      WR1(KO) = ALFAO(KO)*RHOV(KO)
      WR7(KO) = (UVRO(KO)/DRO(1)*UVRO(KO)+WR7(KO)+GR)*
     *          ALFAR(KO)*RHOVR(KO)
C
C
  106 CONTINUE
      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))/DRO(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)/DRO(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.
```

```
      ULRZ(KO) = (ULRO(KO-NI)+ULRO(KO-1-NI))/4.
C
      KD = 0
      IF(UVZO(KO).GE.ZERO) KD = -1
      KN = KO+KD
      IO = I +KD
C
      HVZ(KO) = HV(KN)
      AVZD(KO) = ALFAO(KN)
      WZ1(KO) = ALFAO(KN)*RHOV(KN)
      WZ7(KO) = (UVZO(KN+1)-UVZO(KN))/DZ(IO)*UVZO(KO)
C
      KD = 0
      IF(ULZO(KO).GE.ZERO) KD = -1
      IO = I+KD
      KN = KO+KD
      HLZ(KO) = HL(KN)
      ALZD(KO) = ONE - ALFAO(KN)
      WZ2(KO) = (ONE-ALFAO(KN))*RHOL(KN)
      WZ8(KO) = (ULZO(KN+1)-ULZO(KN))/DZ(IO)*ULZO(KO)
C
C
      IF(UVRZ(KO)) 1110,2110,2110
 1110 WZ7(KO) = (WZ7(KO) - UVZO(KO)*UVRZ(KO)/DR3(3) +
     +            GZ)*ALFAZ(KO)*RHOVZ(KO)
      GO TO 3110
 2110 WZ7(KO) = (WZ7(KO) + (UVZO(KO)-UVZO(KO-NI))*UVRZ(KO)/DR3(2) +
     +            GZ)*ALFAZ(KO)*RHOVZ(KO)
C
 3110 CONTINUE
      IF(ULRZ(KO)) 4110,5110,5110
 4110 WZ8(KO) = (WZ8(KO) - ULZO(KO)*ULRZ(KO)/DR3(3) +
     +            GZ)*(ONE-ALFAZ(KO))*RHOLZ(KO)
      GO TO 6110
 5110 WZ8(KO) = (WZ8(KO) +(ULZO(KO)-ULZO(KO-NI))*ULRZ(KO)/DR3(3) +
     +            GZ)*(ONE-ALFAZ(KO))*RHOLZ(KO)
 6110 CONTINUE
C
  111 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE WS(PO,TVO,TLO,ALFAO,ALFAZ,ALFAR,RHOV,
     *              RHOL,RHOVZ,RHOLZ,RHOVR,RHOLR,HV,HL,
     *              UVZO,ULZO,UVRO,ULRO,
     *              WEV,WEL,WZ3,WZ4,WZ5,WZ6,WZ7,WZ8,WZ9,
     *              WZ10,WZ11,WR3,WR4,WR5,WR6,WR7,WR8,WR9,
     *              WR10,WR11,DH,DV,QSI,SPPD,NN)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /DIM/ DZ(40),DZ1(40),DRO(40),DR1(40),DR2(40),DR3(40),
     *             DR4(40),NI,NJ,NIM1,NIM2,NJM1,NNI,NNJ,NNJJ
      COMMON /TEMPO/ TIME,DT,DTO,DTLS,NDT
      COMMON /NUMBER/ ZERO,ONE,BIG,SMALL
      DIMENSION PO(NN),TVO(NN),TLO(NN),ALFAO(NN),ALFAZ(NN),
     *          ALFAR(NN),RHOV(NN),RHOL(NN),RHOVZ(NN),RHOLZ(NN),
     *          RHOVR(NN),RHOLR(NN),HV(NN),HL(NN),UVZO(NN),
     *          ULZO(NN),UVRO(NN),ULRO(NN),WEV(NN),WEL(NN),
     *          WZ3(NN),WZ4(NN),WZ5(NN),WZ6(NN),WZ7(NN),WZ8(NN),
     *          WZ9(NN),WZ10(NN),WZ11(NN),WR3(NN),WR4(NN),
     *          WR5(NN),WR6(NN),WR7(NN),WR8(NN),WR9(NN),WR10(NN),
     *          WR11(NN),DH(NN),DV(NN),QSI(NN),SPPD(NN)
C
C
C
C           SUBROUTINE WS COMPLETE THE EVALUATION OF THE
C           EXPLICIT TERMS INVOLVED IN THE SOLUTION OF
C           THE PROBLEM STATED WITH SUBROUTINE DONOR.
C           HERE ARE SET THE TERMS CONTAINING THE TIME
C           INCREMENT DT.IT IS WRITTEN SEPARATELY FROM
C           SUBROUTINE DONOR IN ORDER TO ALLOW A CHANGE
C           IN THE VALUE OF DT WHEN THE PROBLRM DOES NOT
C           CONVERGE WITH THE PREVIOUS DT.
C           (SEE NEXT COMENT IN THIS SUBROUTINE.)
C
C
C
      DO 5 JO = 1,NJ
      DO 5 IO = 2,NI
      KO = (JO-1)*NI+IO
C
      WWZ1 = ALFAZ(KO)*RHOVZ(KO)
      WWZ2 = (ONE - ALFAZ(KO))*RHOLZ(KO)
      WWR1 = ALFAR(KO)*RHOVR(KO)
      WWR2 = (ONE - ALFAR(KO))*RHOLR(KO)
C
```

```
      CALL COEFF(TVO(KO),TLO(KO),UVZO(KO),UVRO(KO),ULZO(KO),ULRO(KO),
     *           ALFAZ(KO),ALFAR(KO),RHOVZ(KO),RHOVR(KO),
     *           RHOLZ(KO),RHOLR(KO),DH(KO),DV(KO),QSI(KO),
     *           SPPD(KO),WWZ1,WWZ2,WWR1,WWR2,
     *           FVZ,FLZ,FVR,FLR,C1Z,C1R)
C
      WEV(KO) = -(RHOV(KO)*HV(KO)+PO(KO))*ALFAO(KO)/DT
      WEL(KO) = -(RHOL(KO)*HL(KO)+PO(KO))*(ONE-ALFAO(KO))/DT
C
      IF(NDT.NE.0) GO TO 1
C
C
C          SINCE THE PROGRAM ALLOWS A CHANGE IN THE VALUE
C          OF THE TIME INCREMENT DT,EVEN IF THE TIME STEP
C          IS NOT COMPLITED,WE PUT A CHECK HERE TO KNOW
C          IF SUCH A CHANGE DID OCCUR (IN THIS CASE NDT
C          WOULD BE DIFFERENT THAN ZERO) IN CASE THE TEST
C          BE TRUE,WE SUBTRACT THE TERMS WHICH HAVE THE
C          OLD DT AND ADD THEM BACK WITH THE NEW VALUE
C          OF DT.
C
      WZ4(KO) = C1Z
      WZ6(KO) = C1Z
      WR4(KO) = C1R
      WR6(KO) = C1R
C
      WZ3(KO) = WZ4(KO) + ALFAZ(KO)*RHOVZ(KO)/DT + FVZ
      WZ5(KO) = WZ6(KO) + (ONE-ALFAZ(KO))*RHOLZ(KO)/DT + FLZ
      WR3(KO) = WR4(KO) + ALFAR(KO)*RHOVR(KO)/DT + FVR
      WR5(KO) = WR6(KO) + (ONE-ALFAR(KO))*RHOLR(KO)/DT + FLR
C
C
      WZ7(KO) = WZ7(KO) - UVZO(KO)/DT*ALFAZ(KO)*RHOVZ(KO)
      WZ8(KO) = WZ8(KO) - ULZO(KO)/DT*(ONE-ALFAZ(KO))*RHOLZ(KO)
      WR7(KO) = WR7(KO) - UVRO(KO)/DT*ALFAR(KO)*RHOVR(KO)
      WR8(KO) = WR8(KO) - ULRO(KO)/DT*(ONE-ALFAR(KO))*RHOLR(KO)
      GO TO 2
C
    1 DTC = ONE/DTO - ONE/DT
C
      WZ7(KO) = UVZO(KO)*ALFAZ(KO)*RHOVZ(KO)*DTC + WZ7(KO)
      WZ8(KO) = ULZO(KO)*(ONE-ALFAZ(KO))*RHOLZ(KO)*DTC + WZ8(KO)
      WR7(KO) = UVRO(KO)*ALFAR(KO)*RHOVR(KO)*DTC + WR7(KO)
      WR8(KO) = ULRO(KO)*(ONE-ALFAR(KO))*RHOLR(KO)*DTC + WR8(KO)
      WZ3(KO) = WZ3(KO) - ALFAZ(KO)*RHOVZ(KO)*DTC
```

```
      CALL COEFF(TVO(KO),TLO(KO),UVZO(KO),UVRO(KO),ULZO(KO),ULRO(KO),
     *           ALFAZ(KO),ALFAR(KO),RHOVZ(KO),RHOVR(KO),
     *           RHOLZ(KO),RHOLR(KO),DH(KO),DV(KO),QSI(KO),
     *           SPPD(KO),WWZ1,WWZ2,WWR1,WWR2,
     *           FVZ,FLZ,FVR,FLR,C1Z,C1R)
C
      WEV(KO) = -(RHOV(KO)*HV(KO)+PO(KO))*ALFAO(KO)/DT
      WEL(KO) = -(RHOL(KO)*HL(KO)+PO(KO))*(ONE-ALFAO(KO))/DT
C
      IF(NDT.NE.0) GO TO 1
C
C
C          SINCE THE PROGRAM ALLOWS A CHANGE IN THE VALUE
C          OF THE TIME INCREMENT DT,EVEN IF THE TIME STEP
C          IS NOT COMPLITED,WE PUT A CHECK HERE TO KNOW
C          IF SUCH A CHANGE DID OCCUR (IN THIS CASE NDT
C          WOULD BE DIFFERENT THAN ZERO) IN CASE THE TEST
C          BE TRUE,WE SUBTRACT THE TERMS WHICH HAVE THE
C          OLD DT AND ADD THEM BACK WITH THE NEW VALUE
C          OF DT.
C
      WZ4(KO) = C1Z
      WZ6(KO) = C1Z
      WR4(KO) = C1R
      WR6(KO) = C1R
C
      WZ3(KO) = WZ4(KO) + ALFAZ(KO)*RHOVZ(KO)/DT + FVZ
      WZ5(KO) = WZ6(KO) + (ONE-ALFAZ(KO))*RHOLZ(KO)/DT + FLZ
      WR3(KO) = WR4(KO) + ALFAR(KO)*RHOVR(KO)/DT + FVR
      WR5(KO) = WR6(KO) + (ONE-ALFAR(KO))*RHOLR(KO)/DT + FLR
C
C
      WZ7(KO) = WZ7(KO) - UVZO(KO)/DT*ALFAZ(KO)*RHOVZ(KO)
      WZ8(KO) = WZ8(KO) - ULZO(KO)/DT*(ONE-ALFAZ(KO))*RHOLZ(KO)
      WR7(KO) = WR7(KO) - UVRO(KO)/DT*ALFAR(KO)*RHOVR(KO)
      WR8(KO) = WR8(KO) - ULRO(KO)/DT*(ONE-ALFAR(KO))*RHOLR(KO)
      GO TO 2
C
    1 DTC = ONE/DTO - ONE/DT
C
      WZ7(KO) = UVZO(KO)*ALFAZ(KO)*RHOVZ(KO)*DTC + WZ7(KO)
      WZ8(KO) = ULZO(KO)*(ONE-ALFAZ(KO))*RHOLZ(KO)*DTC + WZ8(KO)
      WR7(KO) = UVRO(KO)*ALFAR(KO)*RHOVR(KO)*DTC + WR7(KO)
      WR8(KO) = ULRO(KO)*(ONE-ALFAR(KO))*RHOLR(KO)*DTC + WR8(KO)
      WZ3(KO) = WZ3(KO) - ALFAZ(KO)*RHOVZ(KO)*DTC
```

```
      WZ5(KO) = WZ5(KO) - (ONE-ALFAZ(KO))*RHOLZ(KO)*DTC
      WR3(KO) = WR3(KO) - ALFAR(KO)*RHOVR(KO)*DTC
      WR5(KO) = WR5(KO) - (ONE-ALFAR(KO))*RHOLR(KO)*DTC
C
    2 IF(WZ3(KO).GT.SMALL) GO TO 3
C
C
C           THIS TEST IS DONE TO CHECK THE PRESENCE OF
C           VAPOR IN THE CELL AT THE PRESENT TIME STEP.
C           IN CASE THERE IS NO VAPOR NOR EVAPORATION
C           (WZ3 = ZERO),THE VAPOR MOMENTUM EQUATION
C           BECOMES TRIVIAL AND THE LIQUID EQUATION
C           STANDS ALONE.
C
      WZ11(KO) = ZERO
      WZ9(KO)  = ZERO
      WZ10(KO) = -(ONE-ALFAZ(KO))/DZ1(IO)/WZ5(KO)
      GO TO 5
C
    3 IF(WZ5(KO).GT.SMALL) GO TO 4
C
C
C           THIS TEST IS DONE TO CHECK THE PRESENCE OF
C           LIQUID IN THE CELL AT THE PRESENT TIME STEP.
C           IN CASE THERE IS NO LIQUID NOR CONDENSATION
C           (WZ5 = ZERO),THE LIQUID MOMENTUM EQUATION
C           BECOMES TRIVIAL AND THE VAPOR EQUATION
C           STANDS ALONE.
C
C
      WZ11(KO) = ZERO
      WZ10(KO) = ZERO
      WZ9(KO)  = -ALFAZ(KO)/DZ1(IO)/WZ3(KO)
      GO TO 5
C
    4 WZ11(KO) = WZ3(KO)*WZ5(KO)-WZ4(KO)*WZ6(KO)
      WZ10(KO) = -(ALFAZ(KO)*WZ6(KO)+(ONE-ALFAZ(KO))*WZ3(KO))/
     /             DZ1(IO)/WZ11(KO)
      WZ9(KO)  = -(ALFAZ(KO)*WZ5(KO)+(ONE-ALFAZ(KO))*WZ4(KO))/
     /             DZ1(IO)/WZ11(KO)
C
    5 CONTINUE
C
C
C           THE SAME TEST WHICH WAS DONE FOR THE
```

```
C          Z-DIRECTION (SEE COMENTS ABOVE) IS
C          DONE HERE FOR THE R-DIRECTION.NOTE
C          THAT SINCE THE MOMENTUM EQUATIONS ARE
C          EVALUATED AT DIFFERENT LOCATIONS FOR
C          EACH DIRECTION,IT IS POSSIBLE THAT
C          ONE PHASE IS ABSENT IN ONE DIRECTION
C          EQUATIONS AND PRESENT IN THE OTHER
C          DIRECTION EQUATIONS.
C
C
      DO 8 JO = 1,NJM1
      DO 8 IO = 2,NIM1
      KO = (JO-1)*NI + IO
C
      IF(WR3(KO).GT.SMALL) GO TO 6
      WR11(KO) = ZERO
      WR9(KO)  = ZERO
      WR10(KO) = -(ONE-ALFAR(KO))/DR3(JO)/WR5(KO)
      GO TO 8
C
    6 IF(WR5(KO).GT.SMALL) GO TO 7
      WR11(KO) = ZERO
      WR10(KO) = ZERO
      WR9(KO)  = -ALFAR(KO)/DR3(JO)/WR3(KO)
      GO TO 8
C
    7 WR11(KO) = WR3(KO)*WR5(KO) - WR4(KO)*WR6(KO)
      WR10(KO) = -(ALFAR(KO)*WR6(KO)+(ONE-ALFAR(KO))*WR3(KO))/
     /             DR3(JO)/WR11(KO)
      WR9(KO)  = -(ALFAR(KO)*WR5(KO)+(ONE-ALFAR(KO))*WR4(KO))/
     /             DR3(JO)/WR11(KO)
    8 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE ONESTP(PN,PO,TVN,TLN,ALFAN,ALFAO,ALFAZ,ALFAR,
     *                  RHOV,RHOL,HV,HL,HVZ,HLZ,HVR,HLR,
     *                  UVZN,ULZN,UVRN,ULRN,
     *                  FUVZN,FULZN,FUVRN,FULRN,W,DV,TS,
     *                  TW,DTW,HCONV,HCONL,HNB,DPN,A1,A2,A3,
     *                  A4,YP,B,BETA,GAMMA,AVZD,ALZD,AVRD,
     *                  ALRD,TCAN,DT,NN,NB,NP,NW,NCAN)
C
      IMPLICIT REAL*8 (A-H,O-Z)
C
C
      COMMON /NUMBER/ ZERO,ONE,BIG,SMALL
      COMMON /ERROR/ IERR
      COMMON /DIM/ DZ(40),DZ1(40),DRO(40),DR1(40),DR2(40),DR3(40),
     *             DR4(40),NI,NJ,NIM1,NIM2,NJM1,NNI,NNJ,NNJJ
C
      DIMENSION EPSLON(9),RES(9)
      DIMENSION PN(NN),PO(NN),TVN(NN),TLN(NN),ALFAN(NN),ALFAO(NN),
     *          ALFAZ(NN),ALFAR(NN),RHOV(NN),RHOL(NN),HV(NN),HL(NN),
     *          HVZ(NN),HLZ(NN),HVR(NN),HLR(NN),
     *          UVZN(NN),ULZN(NN),UVRN(NN),ULRN(NN),
     *          FUVZN(NN),FULZN(NN),FUVRN(NN),FULRN(NN),
     *          W(NW),DV(NN),TS(NN),TW(NN),DTW(NN),
     *          HCONV(NN),HCONL(NN),HNB(NN),DPN(NN),
     *          A1(NN),A2(NN),A3(NN),A4(NN),YP(NN),B(NB),
     *          BETA(NN),GAMMA(NN),AVZD(NN),ALZD(NN),AVRD(NN),
     *          ALRD(NN),TCAN(NCAN)
C
      DIMENSION A(65),F(9),PROP(3,4),S(5,2),Q(4,2),K(30),M(30)
C
      IFLAG = 1
C
C                THE MOMENTUM EQUATIONS (Z-DIRECTION) AT THE BOTTON
C
      MM = NNJ + 2
      DO 4 KO = 2,MM,NI
      DO 1 L = 1,27
    1 K(L) = (L-1)*NN + KO
C
      IF(W(K(5)).GT.SMALL) GO TO 2
C
C
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
```

```
      FULZN(KO)=(F(5)*V06+F(6)*V03)/V07
    4 CONTINUE
C
C           THE CENTRAL CELLS
C
      A(4) = ZERO
      A(12)= ZERO
      A(20)= ZERO
      A(28)= ZERO
      DO 122 KO = 2,NIM1
      DO 5 L=1,27
    5 K(L) = (L-1)*NN+KO
      KM=KO+1
      KP = KO - 1
      CALL STATE (TVN(KO),TLN(KO),PN(KO),PROP,IFLAG)
      CALL NONEQ(ALFAO(KO),ALFAN(KO),TVN(KO),TLN(KO),PN(KO),
     *           RHOV(KO),RHOL(KO),TS(KO),S,IFLAG)
      CALL CONDT (TVN(KO),TLN(KO),PN(KO),ALFAO(KO),TS(KO),TW(KP),
     *            DTW(KP),HCONV(KP),HCONL(KP),HNB(KP),DV(KO),Q,KO)
      CALL IPHTC (HIF,ALFAN(KO))
C
      V01=ALFAO(KO)/DT
      V02=(ONE-ALFAO(KO))/DT
      V03=ALFAN(KO)/DT
      V04=(ONE-ALFAN(KO))/DT
      V05=S(1,1)
      V06=S(2,1)
      V07=S(3,1)
      V08=S(4,1)
      V09=S(5,1)
      V10=W(K(3)+1)/DZ(KO)
      V11=W(K(3))/DZ(KO)
      V12=W(K(14))*DR1(1)
C
      V14=W(K(4)+1)/DZ(KO)
      V15=W(K(4))/DZ(KO)
      V16=W(K(15))*DR1(1)
C
      V18=HVZ(KM)*V10 + PO(KO)*AVZD(KM)/DZ(KO)
      V19=HVZ(KO)*V11 + PO(KO)/DZ(KO)*AVZD(KO)
      V20=HVR(KO)*V12 + PO(KO)*DR1(1)*AVRD(KO)
C
      V22=HLZ(KM)*V14 + PO(KO)*ALZD(KM)/DZ(KO)
      V23=HLZ(KO)*V15 + PO(KO)*ALZD(KO)/DZ(KO)
      V24=HLR(KO)*V16 + PO(KO)*ALRD(KO)*DR1(1)
```

```
C
      V26=(TVN(KO)-TLN(KO))*HIF
      V27=V03*PROP(1,1)
      V28=V04*PROP(1,2)
      V29 = V01*RHOV(KO)
      V30 = V02*RHOL(KO)
      V31 = HV(KO)*V29
      V32 = HL(KO)*V30
C
C                THE RESIDUALS OF CONSERVATION EQUATIONS
C
      F(1) = V27 -V29 +UVZN(KM)*V10 -UVZN(KO)*V11 + UVRN(KO)*
     *       V12 -V05
      F(2) = PROP(1,3)*V27 - V31 +UVZN(KM)*V18 -
     *       UVZN(KO)*V19 + UVRN(KO)*V20 - S(1,2) - Q(1,1) +
     +       PO(KO)*(V03 - V01)
      F(3) = V28 - V30 + ULZN(KM)*V14 - ULZN(KO)*V15 +
     +       ULRN(KO)*V16 +V05
      F(4) = PROP(1,4)*V28 - V32 + ULZN(KM)*V22 -
     *       ULZN(KO)*V23 + ULRN(KO)*V24 + S(1,2) - Q(1,2) +
     +       PO(KO)*(V04 - V02)
      F(5) = W(K(5)+1)*UVZN(KM) -W(K(6)+1)*ULZN(KM) + (PN(KM)-PN(KO))*
     *       ALFAZ(KM)/DZ1(KM) + W(K(9)+1)
      F(6) = W(K(7)+1)*ULZN(KM) -W(K(8)+1)*UVZN(KM) + (PN(KM)-PN(KO))*
     *       (ONE-ALFAZ(KM))/DZ1(KM) + W(K(10)+1)
      F(7) = W(K(16))*UVRN(KO) - W(K(17))*ULRN(KO) + (PN(KO+NI)-PN(KO))*
     *       ALFAR(KO)/DR3(1) + W(K(20))
      F(8) = W(K(18))*ULRN(KO) - W(K(19))*UVRN(KO) + (PN(KO+NI)-PN(KO))*
     *       (ONE-ALFAR(KO))/DR3(1) + W(K(21))
C
C
      A(1) =  PROP(1,1)/DT - V09
      A(9) =  (PROP(1,3)*PROP(1,1) + PO(KO))/DT - S(5,2)
      A(17) = -PROP(1,2)/DT + V09
      A(25) = -(PROP(1,4)*PROP(1,2) + PO(KO))/DT + S(5,2)
C
      A(2) =  PROP(2,1)*V03-V06
      A(10)=(PROP(1,1)*PROP(2,3)+PROP(1,3)*PROP(2,1))*V03 -
     -      Q(2,1) - S(2,2)
      A(18)=V06
      A(26) = S(2,2) - Q(2,2)
C
      A(3)  = -V07
      A(11) = - S(3,2)
      A(19)=PROP(2,2)*V04+V07
```

```
1706        A(27)=(PROP(1,2)*PROP(2,4)+PROP(1,4)*PROP(2,2))*V04 -
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
1739        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))
```

```
      GO TO 11
    9 CONTINUE
      IF(W(K(18)).GT.SMALL) GO TO 10
      FUVRN(KO) = -F(7)/W(K(16))
      FULRN(KO) = ZERO
      GO TO 11
   10 CONTINUE
      FUVRN(KO)=-(W(K(18))*F(7)+W(K(17))*F(8))/W(K(24))
      FULRN(KO)=-(W(K(19))*F(7)+W(K(16))*F(8))/W(K(24))
   11 CONTINUE
C
      F(1)=-F(1)-FUVZN(KM)*V10+FUVZN(KO)*V11-FUVRN(KO)*V12
      F(2)=-F(2)-FUVZN(KM)*V18+FUVZN(KO)*V19-FUVRN(KO)*V20
      F(3)=-F(3)-FULZN(KM)*V14+FULZN(KO)*V15-FULRN(KO)*V16
      F(4)=-F(4)-FULZN(KM)*V22+FULZN(KO)*V23-FULRN(KO)*V24
C
C
C
      DO 111 L = 1,27
  111 K(L) = L*NN + KO
      IX2 = 1
      DO 12 IX1 = 8,24,8
      AUX = A(IX1+1)/A(1)
      IX2 = IX2 + 1
      F(IX2) = F(IX2) - F(1)*AUX
      DO 12 IX3 = 2,8
      IX4 = IX1 + IX3
   12 A(IX4) = A(IX4) - A(IX3)*AUX
      DO 13 L = 1,7
   13 B(K(L)) = -A(L+1)/A(1)
      B(KO) = F(1)/A(1)
C
      IF(DABS(A(10)).GT.SMALL) GO TO 16
C
C           ONLY LIQUID IN THE CELL
C
      B(K(8)) = ZERO
      B(K(9)) = ONE
      DO 14 L = 10,14
   14 B(K(L)) = ZERO
      B(K(15)) = F(4)/A(27)
      DO 15 L = 16,20
   15 B(K(L)) = -A(L+12)/A(27)
C
      AUX = A(19)/A(27)
```

```
      AUP = A(22) - A(30)*AUX
      A1(KP) = (A(20) - A(28)*AUX)/AUP
      A2(KP) = (A(21) - A(29)*AUX)/AUP
      A3(KP) = (A(23) - A(31)*AUX)/AUP
      A4(KP) = (A(24) - A(32)*AUX)/AUP
      YP(KP) = (F(3) - F(4)*AUX)/AUP
      GO TO 22
C
   16 CONTINUE
      IF(DABS(A(27)).GT.SMALL) GO TO 18
C
C           ONLY VAPOR IN THE CELL
C
      B(K(8)) = F(2)/A(10)
      B(K(15)) = B(K(8))
      B(K(9)) = ZERO
      DO 17 L = 10,14
      B(K(L)) = -A(L+2)/A(10)
      LL = L + 6
   17 B(K(LL)) = B(K(L))
C
      AUX = A(18)/A(10)
      AUP = A(22) - A(14)*AUX
      A1(KP) = (A(20) - A(12)*AUX)/AUP
      A2(KP) = (A(21) - A(13)*AUX)/AUP
      A3(KP) = (A(23) - A(15)*AUX)/AUP
      A4(KP) = (A(24) - A(16)*AUX)/AUP
      YP(KP) = (F(3) - F(2)*AUX)/AUP
      GO TO 22
C
C             BOTH PHASES PRESENT
C
   18 CONTINUE
      B(K(8)) = F(2)/A(10)
      DO 19 L = 9,14
   19 B(K(L)) = -A(L+2)/A(10)
C
      IX2 = 2
      DO 20 IX1 = 18,26,8
      AUX = A(IX1)/A(10)
      IX2 = IX2 + 1
      F(IX2) = F(IX2) - F(2)*AUX
      DO 20 IX3 = 1,5
      IX4 = IX1 + IX3
      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
```

```
      V05 = S(1,1)
      V06 = S(2,1)
      V07 = S(3,1)
      V08 = S(4,1)
      V09 = S(5,1)
      V10 = W(K(3)+1)/DZ(I)
      V11 = W(K(3))/DZ(I)
      V12 = W(K(14))*DR1(JO)
      V13 = W(K(14)-NI)*DR2(JO)
      V14 = W(K(4)+1)/DZ(I)
      V15 = W(K(4))/DZ(I)
      V16 = W(K(15))*DR1(JO)
      V17 = W(K(15)-NI)*DR2(JO)
      V18 = HVZ(KM)*V10 + PO(KO)/DZ(I)*AVZD(KM)
      V19 = HVZ(KO)*V11 + PO(KO)/DZ(I)*AVZD(KO)
      V20 = HVR(KO)*V12 + PO(KO)*DR1(JO)*AVRD(KO)
      V21 = HVR(KO-NI)*V13 + PO(KO)*DR2(JO)*AVRD(KO-NI)
      V22 = HLZ(KM)*V14 + PO(KO)*ALZD(KM)/DZ(I)
      V23 = HLZ(KO)*V15 + PO(KO)*ALZD(KO)/DZ(I)
      V24 = HLR(KO)*V16 + PO(KO)*ALRD(KO)*DR1(JO)
      V25 = HLR(KO-NI)*V17 + PO(KO)*ALRD(KO-NI)*DR2(JO)
      V26 = (TVN(KO)-TLN(KO))*HIF
      V27 = V03*PROP(1,1)
      V28 = V04*PROP(1,2)
      V29 = V01*RHOV(KO)
      V30 = V02*RHOL(KO)
      V31 = HV(KO)*V29
      V32 = HL(KO)*V30
C
C
C
       F(1) = V27 - V29 + UVZN(KM)*V10 - UVZN(KO)*V11 +
     +        UVRN(KO)*V12 - UVRN(KO-NI)*V13 - V05
       F(2) = PROP(1,3)*V27 - V31 +UVZN(KM)*V18 -
     -        UVZN(KO)*V19 + UVRN(KO)*V20 - UVRN(KO-NI)*V21 -
     -        S(1,2) - Q(1,1) + PO(KO)*(V03 - V01)
       F(3) = V28 - V30 + ULZN(KM)*V14 - ULZN(KO)*V15 +
     +        ULRN(KO)*V16 - ULRN(KO-NI)*V17 + V05
       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)
       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)
       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)
```

```
C
      IF(J.LT.NNJ) GO TO 24
      CALL HEXCAN(TCAN(I),TCAN(NI+I),TVN(KO),TLN(KO),HCONV(KP),
     *            HCONL(KP),QVC,QLC,DQCDTV,DQCDTL)
      F(2) = F(2) + QVC
      F(4) = F(4) + QLC
      F(7) = ZERO
      F(8) = ZERO
      GO TO 25
   24 CONTINUE
      F(7) = W(K(16))*UVRN(KO) -W(K(17))*ULRN(KO) + (PN(KO+NI)-PN(KO))*
     *       ALFAR(KO)/DR3(JO) + W(K(20))
      F(8) = W(K(18))*ULRN(KO) - W(K(19))*UVRN(KO) + (PN(KO+NI)-PN(KO))*
     *       (ONE-ALFAR(KO))/DR3(JO) + W(K(21))
   25 CONTINUE
C
C
C
      A(1)=PROP(1,1)/DT - V09
      A(2)=PROP(2,1)*V03-V06
      A(3)=-V07
      A(4)=W(K(22)-NI)*V13
C
      A(9)=(PROP(1,3)*PROP(1,1) + PO(KO))/DT - S(5,2)
      A(10)=(PROP(1,1)*PROP(2,3)+PROP(1,3)*PROP(2,1))*V03 -
     -       Q(2,1) - S(2,2)
      A(11)= - S(3,2)
      A(12)=W(K(22)-NI)*V21
C
      A(17)=-PROP(1,2)/DT + V09
      A(18)=V06
      A(19)=PROP(2,2)*V04+V07
      A(20)=W(K(23)-NI)*V17
C
      A(25) = -(PROP(1,4)*PROP(1,2) + PO(KO))/DT + S(5,2)
      A(26)= S(2,2)
      A(27)=(PROP(1,2)*PROP(2,4)+PROP(1,4)*PROP(2,2))*V04 -
     -      Q(3,2) + S(3,2)
      A(28)=W(K(23)-NI)*V25
C
      A(5)  = W(K(11))*V11
      A(13) = W(K(11))*V19
      A(21) = W(K(12))*V15
      A(29) = W(K(12))*V23
C
```

```
      A(7)   = W(K(11)+1)*V10
      A(15)  = W(K(11)+1)*V18
      A(23)  = W(K(12)+1)*V14
      A(31)  = W(K(12)+1)*V22
C
C
      IF(J.GE.NNJ) GO TO 125
      A(8)   = W(K(22))*V12
      A(16)  = W(K(22))*V20
      A(24)  = W(K(23))*V16
      A(32)  = W(K(23))*V24
C
      GO TO 225
C
  125 A(8)  = ZERO
      A(16) = ZERO
      A(24) = ZERO
      A(32) = ZERO
      A(10) = A(10) + DQCDTV
      A(27) = A(27) + DQCDTL
  225 CONTINUE
C
C
C
      A(6) = PROP(3,1)*V03 - V08 - A(4) - A(5) - A(7) - A(8)
      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)
      A(22)=PROP(3,2)*V04+V08-A(20)-A(21)-A(23)-A(24)
      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)
C
C
C
      IF(W(K(5)+1).GT.SMALL) GO TO 26
      FUVZN(KM) = ZERO
      FULZN(KM) = -F(6)/W(K(7)+1)
      GO TO 28
   26 IF(W(K(7)+1).GT.SMALL) GO TO 27
      FUVZN(KM) = -F(5)/W(K(5)+1)
      FULZN(KM) = ZERO
      GO TO 28
   27 CONTINUE
      FUVZN(KM) = -(W(K(7)+1)*F(5)+W(K(6)+1)*F(6))/W(K(13)+1)
      FULZN(KM) = -(W(K(8)+1)*F(5)+W(K(5)+1)*F(6))/W(K(13)+1)
   28 CONTINUE
```

```
      IF(JO.EQ.NJ) GO TO 31
      IF(W(K(16)).GT.SMALL) GO TO 29
      FUVRN(KO) = ZERO
      FULRN(KO) = -F(8)/W(K(18))
      GO TO 31
   29 CONTINUE
      IF(W(K(18)).GT.SMALL) GO TO 30
      FUVRN(KO) = -F(7)/W(K(16))
      FULRN(KO) = ZERO
      GO TO 31
   30 CONTINUE
      FUVRN(KO)=-(W(K(18))*F(7)+W(K(17))*F(8))/W(K(24))
      FULRN(KO)=-(W(K(19))*F(7)+W(K(16))*F(8))/W(K(24))
   31 CONTINUE
C
      F(1)=-F(1)-FUVZN(KM)*V10+FUVZN(KO)*V11-FUVRN(KO)*V12
     +     + FUVRN(KO-NI)*V13
      F(2)=-F(2)-FUVZN(KM)*V18+FUVZN(KO)*V19-FUVRN(KO)*V20
     +     + FUVRN(KO-NI)*V21
      F(3)=-F(3)-FULZN(KM)*V14+FULZN(KO)*V15-FULRN(KO)*V16
     +     + FULRN(KO-NI)*V17
      F(4)=-F(4)-FULZN(KM)*V22+FULZN(KO)*V23-FULRN(KO)*V24
     +     + FULRN(KO-NI)*V25
C
C
C
      DO 32 L = 1,27
   32 K(L) = L*NN + KO
      IX2 = 1
      DO 33 IX1 = 8,24,8
      AUX = A(IX1+1)/A(1)
      IX2 = IX2 + 1
      F(IX2) = F(IX2) - F(1)*AUX
      DO 33 IX3 = 2,8
      IX4 = IX1 + IX3
   33 A(IX4) = A(IX4) - A(IX3)*AUX
      DO 34 L = 1,7
   34 B(K(L)) = -A(L+1)/A(1)
      B(KO) = F(1)/A(1)
C
      IF(DABS(A(10)).GT.SMALL) GO TO 37
C
C           ONLY LIQUID IN THE CELL
C
      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
```

```
      IX2 = 2
      DO 41 IX1 = 18,26,8
      AUX = A(IX1)/A(10)
      IX2 = IX2 + 1
      F(IX2) = F(IX2) - F(2)*AUX
      DO 41 IX3 = 1,6
      IX4 = IX1 + IX3
      IX5 = IX3 + 10
   41 A(IX4) = A(IX4) - A(IX5)*AUX
C
C
      B(K(15)) = F(3)/A(19)
      DO 42 L = 16,20
   42 B(K(L)) = -A(L+4)/A(19)
C
      AUX = A(27)/A(19)
      AUP = A(30) - A(22)*AUX
      A1(KP) = (A(28) - A(20)*AUX)/AUP
      A2(KP) = (A(29) - A(21)*AUX)/AUP
      A3(KP) = (A(31) - A(23)*AUX)/AUP
      A4(KP) = (A(32) - A(24)*AUX)/AUP
      YP(KP) = (F(4) - F(3)*AUX)/AUP
C
   43 CONTINUE
C
      DDT = DABS(A1(KP)) + DABS(A2(KP)) + DABS(A3(KP)) +DABS(A4(KP))
      IF(DDT.GT.ONE) GO TO 58
C
   46 CONTINUE
C
      CALL GAUSIE(A1,A2,A3,A4,YP,DPN,BETA,GAMMA,NN)
C
C               CELL (2,1)
C
      KO = 2
      KP = KO - 1
      KQ = KP + NIM2
      DO 47 L = 1,27
      M(L) = (L-1)*NN + KO
   47 K(L) = L*NN+KO
C
C
      DTL = B(K(15)) +
     +      B(K(18))*DPN(KP) + B(K(19))*DPN(KO) + B(K(20))*DPN(KQ)
      DTV = B(K(8)) + B(K(9))*DTL +
```

```
     +       B(K(12))*DPN(KP) + B(K(13))*DPN(KO) +
     +       B(K(14))*DPN(KQ)
      DAL = B(KO) + B(K(1))*DTV + B(K(2))*DTL +
     +       B(K(5))*DPN(KP) + B(K(6))*DPN(KO) +
     +       B(K(7))*DPN(KQ)
      PN(KO)  = PN(KO) + DPN(KP)
      IF(PN(KO).LT.1.D+04) GO TO 59
      IF(PN(KO).GT.4.D+07) GO TO 60
      TLN(KO) = TLN(KO) + DTL
      TVN(KO) = TVN(KO) + DTV
      ALFAN(KO) = ALFAN(KO) + DAL
      TX = SAT(PN(KO))
      DTS = TX - TS(KO)
      TS(KO) = TX
      TW(KP) = TW(KP) + (HCONV(KP)*DTV + HCONL(KP)*DTL +
     +         HNB(KP)*DTS)*DTW(KP)
C
C
      UVZN(KO) = W(M(11))*DPN(KP)  + FUVZN(KO) + UVZN(KO)
      ULZN(KO) = W(M(12))*DPN(KP)  + FULZN(KO) + ULZN(KO)
      UVRN(KO) = W(M(22))*(DPN(KQ)-DPN(KP)) + FUVRN(KO) + UVRN(KO)
      ULRN(KO) = W(M(23))*(DPN(KQ)-DPN(KP)) + FULRN(KO) + ULRN(KO)
      UVZN(1)  = UVZN(KO)
      ULZN(1)  = ULZN(KO)
C
C                  CELLS (I,1) , I=3,NI-2
C
      DO 49 I = 3,NIM2
      KO = I
      KP = KO - 1
      KM = KO
      KQ = KP + NIM2
      KR = KP - 1
      DO 48 L = 1,27
      M(L) = (L-1)*NN + KO
   48 K(L) = L*NN+KO
C
C
      DTL = B(K(15)) + B(K(17))*DPN(KR) +
     +      B(K(18))*DPN(KP) + B(K(19))*DPN(KM) + B(K(20))*DPN(KQ)
      DTV = B(K(8)) + B(K(9))*DTL +
     +      B(K(11))*DPN(KR) + B(K(12))*DPN(KP) + B(K(13))*DPN(KM) +
     +      B(K(14))*DPN(KQ)
      DAL = B(KO) + B(K(1))*DTV + B(K(2))*DTL +
     +      B(K(4))*DPN(KR) + B(K(5))*DPN(KP) + B(K(6))*DPN(KM) +
```

```
     +      B(K(7))*DPN(KQ)
      PN(KO)  = PN(KO) + DPN(KP)
      IF(PN(KO).LT.1.D+04) GO TO 59
      IF(PN(KO).GT.4.D+07) GO TO 60
      TLN(KO) = TLN(KO) + DTL
      TVN(KO) = TVN(KO) + DTV
      ALFAN(KO) = ALFAN(KO) + DAL
      TX = SAT(PN(KO))
      DTS = TX - TS(KO)
      TS(KO) = TX
      TW(KP) = TW(KP) + (HCONV(KP)*DTV + HCONL(KP)*DTL +
     +         HNB(KP)*DTS)*DTW(KP)
C
C
      UVZN(KO) = W(M(11))*(DPN(KP)-DPN(KR))  + FUVZN(KO) + UVZN(KO)
      ULZN(KO) = W(M(12))*(DPN(KP)-DPN(KR))  + FULZN(KO) + ULZN(KO)
      UVRN(KO) = W(M(22))*(DPN(KQ)-DPN(KP)) + FUVRN(KO) + UVRN(KO)
      ULRN(KO) = W(M(23))*(DPN(KQ)-DPN(KP)) + FULRN(KO) + ULRN(KO)
   49 CONTINUE
C
C                  CELL (NIM1,1)
C
      KO = NIM1
      KP = KO - 1
      KQ = KP + NIM2
      KR = KP - 1
      DO 148 L = 1,27
      M(L) = (L-1)*NN + KO
  148 K(L) = L*NN+KO
C
C
      DTL = B(K(15)) + B(K(17))*DPN(KR) +
     +      B(K(18))*DPN(KP) + B(K(20))*DPN(KQ)
      DTV = B(K(8)) + B(K(9))*DTL +
     +      B(K(11))*DPN(KR) + B(K(12))*DPN(KP) +
     +      B(K(14))*DPN(KQ)
      DAL = B(KO) + B(K(1))*DTV + B(K(2))*DTL +
     +      B(K(4))*DPN(KR) + B(K(5))*DPN(KP) +
     +      B(K(7))*DPN(KQ)
      PN(KO)  = PN(KO) + DPN(KP)
      IF(PN(KO).LT.1.D+04) GO TO 59
      IF(PN(KO).GT.4.D+07) GO TO 60
      TLN(KO) = TLN(KO) + DTL
      TVN(KO) = TVN(KO) + DTV
      ALFAN(KO) = ALFAN(KO) + DAL
```

```
      TX = SAT(PN(KO))
      DTS = TX - TS(KO)
      TS(KO) = TX
      TW(KP) = TW(KP) + (HCONV(KP)*DTV + HCONL(KP)*DTL +
     +          HNB(KP)*DTS)*DTW(KP)
C
C
      UVZN(KO) = W(M(11))*(DPN(KP)-DPN(KR))  + FUVZN(KO) + UVZN(KO)
      ULZN(KO) = W(M(12))*(DPN(KP)-DPN(KR))  + FULZN(KO) + ULZN(KO)
      UVRN(KO) = W(M(22))*(DPN(KQ)-DPN(KP)) + FUVRN(KO) + UVRN(KO)
      ULRN(KO) = W(M(23))*(DPN(KQ)-DPN(KP)) + FULRN(KO) + ULRN(KO)
C
C                  CELLS (2,J) , J = 2,NJ-1
C
C
      DO 51 J = NI,NNJJ,NI
      KO = J+2
      KP = KO - 1 - 2*J/NI
      KM = KP + 1
      KQ = KP + NIM2
      KR = KP - 1
      KS = KP - NIM2
      DO 50 L = 1,27
      M(L) = (L-1)*NN + KO
   50 K(L) = L*NN+KO
C
C
      DTL = B(K(15)) + B(K(16))*DPN(KS) +
     +      B(K(18))*DPN(KP) + B(K(19))*DPN(KM) + B(K(20))*DPN(KQ)
      DTV = B(K(8)) + B(K(9))*DTL + B(K(10))*DPN(KS) +
     +      B(K(12))*DPN(KP) + B(K(13))*DPN(KM) +
     +      B(K(14))*DPN(KQ)
      DAL = B(KO) + B(K(1))*DTV + B(K(2))*DTL + B(K(3))*DPN(KS) +
     +      B(K(5))*DPN(KP) + B(K(6))*DPN(KM) +
     +      B(K(7))*DPN(KQ)
      PN(KO)  = PN(KO) + DPN(KP)
      IF(PN(KO).LT.1.D+04) GO TO 59
      IF(PN(KO).GT.4.D+07) GO TO 60
      TLN(KO) = TLN(KO) + DTL
      TVN(KO) = TVN(KO) + DTV
      ALFAN(KO) = ALFAN(KO) + DAL
      TX = SAT(PN(KO))
      DTS = TX - TS(KO)
      TS(KO) = TX
      TW(KP) = TW(KP) + (HCONV(KP)*DTV + HCONL(KP)*DTL +
```

```
     +          HNB(KP)*DTS)*DTW(KP)
C
C
      UVZN(KO) = W(M(11))*DPN(KP)  + FUVZN(KO) + UVZN(KO)
      ULZN(KO)  = W(M(12))*DPN(KP)  + FULZN(KO) + ULZN(KO)
      UVRN(KO) = W(M(22))*(DPN(KQ)-DPN(KP)) + FUVRN(KO) + UVRN(KO)
      ULRN(KO) = W(M(23))*(DPN(KQ)-DPN(KP)) + FULRN(KO) + ULRN(KO)
      UVZN(KO-1) = UVZN(KO)
      ULZN(KO-1) = ULZN(KO)
   51 CONTINUE
C
C         CELLS (I,J) , I=3,NI-2 , J=2,NJ-1
C
      DO 53 J = NI,NNJJ,NI
      DO 53 I = 3,NIM2
      KO = I+J
      KP = KO - 1 - 2*J/NI
      KM = KP + 1
      KQ = KP +NIM2
      KR = KP - 1
      KS = KP - NIM2
      DO 52 L = 1,27
      M(L) = (L-1)*NN + KO
   52 K(L) = L*NN+KO
C
C
      DTL = B(K(15)) + B(K(16))*DPN(KS) + B(K(17))*DPN(KR) +
     +      B(K(18))*DPN(KP) + B(K(19))*DPN(KM) + B(K(20))*DPN(KQ)
      DTV = B(K(8)) + B(K(9))*DTL + B(K(10))*DPN(KS) +
     +      B(K(11))*DPN(KR) + B(K(12))*DPN(KP) + B(K(13))*DPN(KM) +
     +      B(K(14))*DPN(KQ)
      DAL = B(KO) + B(K(1))*DTV + B(K(2))*DTL + B(K(3))*DPN(KS) +
     +      B(K(4))*DPN(KR) + B(K(5))*DPN(KP) + B(K(6))*DPN(KM) +
     +      B(K(7))*DPN(KQ)
      PN(KO)  = PN(KO) + DPN(KP)
      IF(PN(KO).LT.1.D+04) GO TO 59
      IF(PN(KO).GT.4.D+07) GO TO 60
      TLN(KO) = TLN(KO) + DTL
      TVN(KO) = TVN(KO) + DTV
      ALFAN(KO) = ALFAN(KO) + DAL
      TX = SAT(PN(KO))
      DTS = TX - TS(KO)
      TS(KO) = TX
      TW(KP) = TW(KP) + (HCONV(KP)*DTV + HCONL(KP)*DTL +
     +         HNB(KP)*DTS)*DTW(KP)
```

```
C
C
      UVZN(KO) = W(M(11))*(DPN(KP)-DPN(KR))  + FUVZN(KO) + UVZN(KO)
      ULZN(KO) = W(M(12))*(DPN(KP)-DPN(KR))  + FULZN(KO) + ULZN(KO)
      UVRN(KO) = W(M(22))*(DPN(KQ)-DPN(KP)) + FUVRN(KO) + UVRN(KO)
      ULRN(KO) = W(M(23))*(DPN(KQ)-DPN(KP)) + FULRN(KO) + ULRN(KO)
   53 CONTINUE
C
C         CELLS (NIM1,J) ,  J=2,NJ-1
C
      DO 153 J = NI,NNJJ,NI
      I = NIM1
      KO = I+J
      KP = KO - 1 - 2*J/NI
      KQ = KP +NIM2
      KR = KP - 1
      KS = KP - NIM2
      DO 152 L = 1,27
      M(L) = (L-1)*NN + KO
  152 K(L) = L*NN+KO
C
C
      DTL = B(K(15)) + B(K(16))*DPN(KS) + B(K(17))*DPN(KR) +
     +      B(K(18))*DPN(KP)  + B(K(20))*DPN(KQ)
      DTV = B(K(8)) + B(K(9))*DTL + B(K(10))*DPN(KS) +
     +      B(K(11))*DPN(KR) + B(K(12))*DPN(KP) +
     +      B(K(14))*DPN(KQ)
      DAL = B(KO) + B(K(1))*DTV + B(K(2))*DTL + B(K(3))*DPN(KS) +
     +      B(K(4))*DPN(KR) + B(K(5))*DPN(KP) +
     +      B(K(7))*DPN(KQ)
      PN(KO)  = PN(KO) + DPN(KP)
      IF(PN(KO).LT.1.D+04) GO TO 59
      IF(PN(KO).GT.4.D+07) GO TO 60
      TLN(KO) = TLN(KO) + DTL
      TVN(KO) = TVN(KO) + DTV
      ALFAN(KO) = ALFAN(KO) + DAL
      TX = SAT(PN(KO))
      DTS = TX - TS(KO)
      TS(KO) = TX
      TW(KP) = TW(KP) + (HCONV(KP)*DTV + HCONL(KP)*DTL +
     +         HNB(KP)*DTS)*DTW(KP)
C
C
      UVZN(KO) = W(M(11))*(DPN(KP)-DPN(KR))  + FUVZN(KO) + UVZN(KO)
      ULZN(KO) = W(M(12))*(DPN(KP)-DPN(KR))  + FULZN(KO) + ULZN(KO)
```

```
      UVRN(KO) = W(M(22))*(DPN(KQ)-DPN(KP)) + FUVRN(KO) + UVRN(KO)
      ULRN(KO) = W(M(23))*(DPN(KQ)-DPN(KP)) + FULRN(KO) + ULRN(KO)
  153 CONTINUE
C
C           CELLS (I,NJ) , I=3,NI-1
C
      DO 55 I = 3,NIM1
      KO = I + NNJ
      KP = KO + 1 - 2*NJ
      KM = KP + 1
      KQ = KP + NIM2
      KR = KP -1
      KS = KP - NIM2
      DO 54 L = 1,27
      M(L) = (L-1)*NN + KO
   54 K(L) = L*NN+KO
C
C
      DTL = B(K(15)) + B(K(16))*DPN(KS) + B(K(17))*DPN(KR) +
     +      B(K(18))*DPN(KP) + B(K(19))*DPN(KM)
      DTV = B(K(8)) + B(K(9))*DTL + B(K(10))*DPN(KS) +
     +      B(K(11))*DPN(KR) + B(K(12))*DPN(KP) + B(K(13))*DPN(KM)
      DAL = B(KO) + B(K(1))*DTV + B(K(2))*DTL + B(K(3))*DPN(KS) +
     +      B(K(4))*DPN(KR) + B(K(5))*DPN(KP) + B(K(6))*DPN(KM)
      PN(KO)  = PN(KO) + DPN(KP)
      IF(PN(KO).LT.1.D+04) GO TO 59
      IF(PN(KO).GT.4.D+07) GO TO 60
      TLN(KO) = TLN(KO) + DTL
      TVN(KO) = TVN(KO) + DTV
      ALFAN(KO) = ALFAN(KO) + DAL
      TX = SAT(PN(KO))
      DTS = TX - TS(KO)
      TS(KO) = TX
      TW(KP) = TW(KP) + (HCONV(KP)*DTV + HCONL(KP)*DTL +
     +         HNB(KP)*DTS)*DTW(KP)
      TCAN(I) = TCAN(I) + TCAN(NI + I)*(HCONV(KP)*DTV +
     +          HCONL(KP)*DTL)
C
C
      UVZN(KO) = W(M(11))*(DPN(KP)-DPN(KR))  + FUVZN(KO) + UVZN(KO)
      ULZN(KO) = W(M(12))*(DPN(KP)-DPN(KR))  + FULZN(KO) + ULZN(KO)
      UVRN(KO) = ZERO
      ULRN(KO) = ZERO
   55 CONTINUE
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) +
2454       +      B(K(18))*DPN(KP) + B(K(19))*DPN(KM)
2455        DTV = B(K(8)) + B(K(9))*DTL + B(K(10))*DPN(KS) +
2456       +      B(K(13))*DPN(KM) + B(K(12))*DPN(KP)
2457        DAL = B(KO) + B(K(1))*DTV + B(K(2))*DTL + B(K(3))*DPN(KS) +
2458       +      B(K(6))*DPN(KM) + B(K(5))*DPN(KP)
2459        PN(KO)  = PN(KO) + DPN(KP)
2460        IF(PN(KO).LT.1.D+04) GO TO 59
2461        IF(PN(KO).GT.4.D+07) GO TO 60
2462        TLN(KO) = TLN(KO) + DTL
2463        TVN(KO) = TVN(KO) + DTV
2464        ALFAN(KO) = ALFAN(KO) + DAL
2465        TX = SAT(PN(KO))
2466        DTS = TX - TS(KO)
2467        TS(KO) = TX
2468        TW(KP) = TW(KP) + (HCONV(KP)*DTV + HCONL(KP)*DTL +
2469       +          HNB(KP)*DTS)*DTW(KP)
2470        TCAN(2) = TCAN(2) + TCAN(NI + 2)*(HCONV(KP)*DTV +
```

```
     +          HCONL(KP)*DTL)
C
C
      UVZN(KO)    = W(M(11))*DPN(KP)  + FUVZN(KO) + UVZN(KO)
      ULZN(KO) = W(M(12))*DPN(KP)  + FULZN(KO) + ULZN(KO)
      UVRN(KO) = ZERO
      ULRN(KO) = ZERO
      UVZN(KO-1) = UVZN(KO)
      ULZN(KO-1) = ULZN(KO)
C
      DO 357 KO = 1,NN
      IF(ALFAN(KO).GE.ZERO) GO TO 257
      IF(ALFAN(KO).LT.-1.D-05) IERR = 3
      ALFAN(KO) = ZERO
  257 CONTINUE
      IF(ALFAN(KO).LE.ONE) GO TO 2257
      IF(ALFAN(KO).GT.1.00001) IERR = 3
      ALFAN(KO) = ONE
 2257 CONTINUE
      IF(TVN(KO).LT.4.D+02) IERR = 14
      IF(TVN(KO).GT.3.D+03) IERR = 15
      IF(TLN(KO).LT.4.D+02) IERR = 16
      IF(TLN(KO).GT.3.D+03) IERR = 17
  357 CONTINUE
      RETURN
   58 IERR = 2
      RETURN
   59 IERR = 12
      RETURN
   60 IERR = 13
      RETURN
      END
```

```
      SUBROUTINE COEFF(TV,TL,UVZ,UVR,ULZ,ULR,ALFAZ,ALFAR,
     *                 RHOVZ,RHOVR,RHOLZ,RHOLR,DH,DV,QSI,
     *                 SPPD,WZ1,WZ2,WR1,WR2,FRVZ,FRLZ,FRVR,
     *                 FRLR,C1Z,C1R)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /NUMBER/ ZERO,ONE,BIG,SMALL
      DATA TWO,PTWO,ADRY,CADRY/2.D0,.2D0,.957D0,0.043D0/
C
C           SUBROUTINE COEFF CALCULATES THE MOMENTUM EXCHANGE
C           COEFFICIENTS.
C           C1. ARE THE INTERPHASE MOMENTUM EXCHANGE COEFFICIENTS
C           FOR THE TWO DIRECTIONS.
C           FR.. ARE THE WALL FRICTION COEFFICIENTS FOR BOTH PHASES
C           AND DIRECTIONS.
C
      VV = VISCV (TV)
      VL = VISCL (TL)
C
      AUVZ = DABS (UVZ)
      AUVR = DABS (UVR)
      AULZ = DABS (ULZ)
      AULR = DABS (ULR)
C
      REVZ = WZ1*AUVZ*DH/VV +SMALL
      RELZ = RHOLZ*AULZ*DH/VL + SMALL
      REVR = WR1*AUVR*QSI*DH/VV + SMALL
      RELR = WR2*AULR*QSI*DH/VL + SMALL
C
      FVZ = 0.180D0/REVZ**PTWO + SPPD*DH
      FLZ = 0.180D0/RELZ**PTWO + SPPD*DH
      FVR = PTWO/REVR**PTWO
      FLR = PTWO/RELR**PTWO
C
      FRVZ = (ALFAZ - ADRY)/CADRY*RHOVZ*AUVZ*FVZ/TWO/DH
      FRVR = (ALFAR - ADRY)/CADRY*180.*VV/(DH*DH)*QSI
      FRLZ =       RHOLZ*AULZ*FLZ/TWO/DH
      FRLR =180.*VL/(DH*DH)*QSI
      XZ = (ONE - ALFAZ)/CADRY
      XR = (ONE - ALFAR)/CADRY
C
      IF(ALFAZ.GT.ADRY) GO TO 1
      FRVZ = ZERO
      XZ = ONE
```

```
      1 CONTINUE
        IF(ALFAR.GT.ADRY) GO TO 2
        FRVR = ZERO
        XR = ONE
      2 CONTINUE
C
        FRLZ = FRLZ*XZ
        FRLR = FRLR*XR
C
        X = (ONE + (ONE-ALFAZ)*75.D0)**.95*4.31
C
        C1Z = ((ONE - ALFAZ)*DABS(UVZ - ULZ)*RHOVZ/TWO +
     +        VL/DH)*X/DH
        C1R = ((ONE - ALFAR)*DABS(UVR - ULR)*RHOVR/TWO +
     +        VL/DH)*X*QSI*QSI/DH
C
        RETURN
        END
```

```
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(T)
      IMPLICIT REAL*8 (A-H,O-Z)
C
C           FUNCTION VISCL RETURNS THE SODIUM LIQUID VISCOSITY
C           IN (KG/M/SEC),AS A FUNCTION OF THE TEMPERATURE
C           IN DEGREE CELSIUS
C
      TK = T
      VISCL = DEXP(508.07/TK - 5.7316 - .4925*DLOG(TK))
      RETURN
      END
```

```
      FUNCTION VISCV(T)
      IMPLICIT REAL*8 (A-H,O-Z)
C
C           FUNCTION VISCV RETURNS THE SODIUM VAPOR VISCOSITY
C           IN (KG/M/SEC),AS A FUNCTION OF THE TEMPERATURE
C           IN DEGREE CELSIUS
C
      TK = T
      VISCV = 6.085D-09*TK + 1.261D-05
      RETURN
      END
```

```
      FUNCTION SURTEN (T)
      IMPLICIT REAL*8 (A-H,O-Z)
C
C           FUNCTION SURTEN RETURNS THE SURFACE TENSION OF LIQUID
C           SODIUM IN NEWTON/METER
C           CORRELATION FROM GOLDEN AND TOKAR,
C
      TC = T - 273.14
      SURTEN = 2.067D-01 - 1.0D-04*TC
      IF(SURTEN.LT.0.D0) SURTEN = 0.D0
      RETURN
      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
```

```
2652        FUNCTION DTSDP(P)
2653        IMPLICIT REAL*8 (A-H,O-Z)
2654 C
2655 C             CALCULATES THE DERIVATIVE OF THE SATURATION
2656 C             TEMPERTURE WITH RESPECT TO THE PRESSURE
2657 C
2658        X = 21.9358 - DLOG(P)
2659        DTSDP = 12020./(X*X*P)
2660        RETURN
2661        END
```

Function condl

```
      FUNCTION CONDL(T)
      IMPLICIT REAL*8 (A-H,O-Z)
      DATA A1,A2,A3,X1,X2,X3 /54.306,-1.878D-02,2.0914D-06,1.8D0,
     *                        459.67D0,1.7307D0/
C
      TF = X1*T - X2
      T2 = TF*TF
      C = A1 + A2*TF + A3*T2
      CONDL = C*X3
      RETURN
      END
```

```
2673        FUNCTION CONDV(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        CONDV = X3*C
2682        RETURN
2683        END
```

Function cpl

```
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(T)
      IMPLICIT REAL*8 (A-H,O-Z)
C
      TX = T - 844.1
      PRV = .7596D0 + .810D-06*TX*TX
      RETURN
      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

```
      FUNCTION HFG(P)
      IMPLICIT REAL*8 (A-H,O-Z)
C
      T = SAT(P)
      HFG = 5.089D+06 - 1.043D+03*T
      RETURN
      END
```

```
      SUBROUTINE HTCF (P,TV,TL,ALFA,RHOV,RHOL,HV,HL,DH,TS,TW,
     *                 HCONV,HCONL,HNB,UV,UL)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /NUMBER/ ZERO,ONE,BIG,SMALL
      COMMON /POVERD/ R
C
      HCONV = ZERO
      HCONL = ZERO
      HNB   = ZERO
C
      VV = VISCV(TV)
      VL = VISCL(TL)
      PV = PRV(TV)
      PL = PRL(TL)
      CV = CONDV(TV)
      CL = CONDL(TL)
      AUV = DABS(UV)
      AUL = DABS(UL)
      SIG = SURTEN(TL)
C
C               COMPUTE QUALITY
C
      GV = ALFA*RHOV*AUV
      GL = (ONE-ALFA)*RHOL*AUL
      G  = GV + GL
      IF((UV-UL)*UL.LE.ZERO) GO TO 1
      X = GV/G
      GO TO 2
    1 CONTINUE
      X = ALFA*RHOV/(ALFA*RHOV + (ONE-ALFA)*RHOL)
    2 CONTINUE
C
C               SINGLE PHASE : DITTUS-BOELTER CORRELATION (VAPOR)
C
      IF(ALFA.LE.0.96) GO TO 3
      REV = RHOV*AUV*DH/VV
      HCONV = 0.023*REV**0.8*PV**0.4*CV/DH
      RETURN
    3 CONTINUE
C
C               SINGLE PHASE : SCHAD CORRELATION (LIQUID)
C
      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        HNB = ZERO
2805        RETURN
2806        END
```

```
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 (TV,TL,P,PROP,IFLAG)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /ERROR/ IERR
      COMMON /NUMBER/ ZERO,ONE,BIG,SMALL
      DIMENSION PROP(3,4)
      DATA RV0,RV1,RV2,RV22 /1.605D-02,2.51D-06,-3.23D-13,-6.46D-13/
      DATA RL0,RL1,RL2,RL3,RLP,RL22,RL33 /1.0116D+03,-0.2205,
     1     -1.9224D-05,5.6377D-09,2.26D-07,-3.8448D-05,
     2     1.69131D-08/
      DATA EV0,EV1,EV2,EV3,EV22,EV33 /5.0215D+06,5.8714D+02,
     1     -.41672,1.54272D-04,-.83344,4.62816D-04/
      DATA EL0,EL1,EL2,EL3,EL22,EL33 /-6.75075D+04,1.63014D+03,
     1     -.41672,1.54272D-04,-.83344,4.62816D-04/
C
C           ALL PROPERTIES IN SI UNITS
C           PROPERTIES BASED IN
C           GOLDEN,G.H. AND TOKAR,J.V.,
C           THERMOPHYSICAL PROPERTIES OF SODIUM, ANL-7323
C           WITH THE ADDITION OF PRESSURE DEPENDENCE IN THE
C           LIQUID DENSITY.
C                THIS ADDITION WAS MADE BECAUSE THE NUMERICAL
C           STABILITY OF THE MODEL REQUIRES A NON ZERO,
C           POSITIVE VALUE OF THE PRESSURE DERIVATIVE OF
C           THE DENSITY .
C
C                ALSO A REQUIREMENT FOR THE NUMERICAL CONVERGENCE
C           IS THE DERIVATIVES OF PROPERTIES WITH RESPECT TO
C           TEMPERATURE AND PRESSURE BEING THE MATHEMATICAL
C           DERIVATIVES OF THE EXPRESSIONS FOR THE PROPERTIES
C
C
      TS = SAT(P)
      X1 = (RV2*P + RV1)*P + RV0
      PROP(1,1) = X1*TS/TV
      PROP(1,2) = ((RL3*TL + RL2)*TL + RL1)*TL + RL0 + RLP*P
      PROP(1,3) = ((EV3*TV + EV2)*TV + EV1)*TV + EV0 - P/PROP(1,1)
      PROP(1,4) = ((EL3*TL + EL2)*TL + EL1)*TL + EL0
C
      PROP(2,1) = -PROP(1,1)/TV
      PROP(2,2) = (RL33*TL + RL22)*TL + RL1
      PROP(2,3) = (EV33*TV + EV22)*TV + EV1
      PROP(2,4) = (EL33*TL + EL22)*TL + EL1
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(ALFAO,ALFA,TV,TL,P,RHOV,RHOL,TS,S,IFLAG)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /ERROR/ IERR
      COMMON /NUMBER/ ZERO,ONE,BIG,SMALL
      COMMON /PD/ D4,POD2
      DIMENSION S(5,2)
      DATA AN,RGAS /1.3333333D+07,.14469D+03/,HALF /0.5D0/
      DATA PI,SR3,CADRY,ADRY /3.141592654,3.464101616,0.043,0.957/
      DATA H0,H1 /5.089D+06,-.1043D+04/
      DATA RNU /6.D+03/
      DATA HL0,HL1,HL2,HL3 /-6.75075D+04,1.63014D+03,
     *     -.41672D0,1.54272D-04/
C
C           SUBROUTINE NONEQ CALCULATES THE MASS AND ENERGY EXCHANGE RATES
C           AND ITS DERIVATIVES.
C           AN = 4/3*N, N = 1.0D+07 BUBLES/CUBIC METER
C           RGAS = SQUARE ROOT OF GAS CONSTANT FOR SODIUM OVER 2*PI
C           POD2 = PITCH TO DIAMETER RATIO SQUARED
C
C           S(1, ) =  EXCHANGE RATE          S( ,1) = MASS
C           S(2, ) = D/DTV                   S( ,2) = ENERGY
C           S(3, ) = D/DTL
C           S(4, ) = D/DP
C           S(5, ) = D/DALFA
C
      AX = ALFAO
      IF(ALFAO.LT.1.D-04) AX = 1.D-04
      IF(ALFAO.GT.0.9999) AX = 0.9999
C
      TS = SAT(P)
      HLG = H1*TS + H0
      X = ONE/(SR3*POD2 - PI)
C
      AM = 1.2D-07*PI*X*D4*D4
      IF(ALFAO.GT.0.6) GO TO 10
C
      XX = 3.*PI*AX*X
      GO TO 20
   10 CONTINUE
      Y = ONE
      IF(AX.GT.ADRY) Y = (ONE - AX)/CADRY
      XK = 1.8/(SR3*POD2*X - 0.6)
      XX = (SR3*POD2*X - AX)*X*Y*PI*XK
```

```
   20 CONTINUE
      A = DSQRT(XX)*D4
C
   30 CONTINUE
      CE = A*RGAS*RHOV*RHOV
      CC = CE*(ONE - AX)
      CE = CE*AX
C
      EL = ZERO
      CL = ZERO
      IF(TL.GT.TS) EL = 1.0D0
      IF(TS.GT.TV) CL = 5.D-03
C
      CE = CE*EL
      CC = CC*CL
C
      DDP = DTSDP(P)
      SRTS = DSQRT(TS)
      DTL = (TL - TS)/SRTS
      DTV = (TS - TV)/SRTS
C
C             MASS EXCHANGE RATE
C
      SE = DTL*CE*(ONE - ALFA)
      SC = DTV*CC*ALFA
      S(1,1) = SE - SC
C
C             DERIVATIVES
C
      S(2,1) = CC*ALFA/SRTS
      S(3,1) = CE*(ONE-ALFA)/SRTS
      DSEVAP = CE*(ALFA-ONE)*(TS+TL)/TS/SRTS*HALF*DDP
      DSCOND = CC*       ALFA*(TS+TV)/TS/SRTS*HALF*DDP
      S(4,1) = DSEVAP - DSCOND
      S(5,1) = -CE*DTL - CC*DTV
C
C                ENERGY EXCHANGE RATE
C
      U = A*CONDL(TV)*RNU*D4
      HL = ((HL3*TS + HL2)*TS + HL1)*TS + HL0
      HV = HL + HLG
      DHLDP = ((3.*HL3*TS + 2.*HL2)*TS + HL1)*DDP
      DHVDP = DHLDP + H1*DDP
C
      S(1,2) = SE*HV - SC*HL + U*(TL - TV)
```

```
C
C                DERIVATIVES
C
      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(TV,TL,P,ALFA,TS,TW,DTW,
     *                 HCONV,HCONL,HNB,DV,Q,KO)
      IMPLICIT REAL*8 (A-H,O-Z)
      LOGICAL LSS
      COMMON /STST/ TAFP,LSS
      COMMON /ERROR/ IERR
      COMMON /NUMBER/ ZERO,ONE,BIG,SMALL
      DIMENSION Q(4,2)
C
      Q(1,1) = (TW - TV)*HCONV*DV
      Q(1,2) = ((TW - TL)*HCONL + (TW - TS)*HNB)*DV
      Q(2,1) = (DTW*HCONV - 1)*HCONV*DV
      Q(2,2) = ZERO
      Q(3,1) = ZERO
      Q(3,2) = ((HCONL + HNB)*DTW - 1)*HCONL*DV
      Q(4,1) = ZERO
      Q(4,2) = ((HCONL + HNB)*DTW - 1)*HNB*DV*DTSOP(P)
      RETURN
      END
```

```
      SUBROUTINE HEXCAN(TCAN,DTC,TV,TL,HCONV,HCONL,QV,QL,
     *                  DQDTV,DQDTL)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /NUMBER/ ZERO,ONE,BIG,SMALL
      COMMON /HXCN/ ACOV
C
C          SUBROUTINE HEXCAN CALCULATES THE HEAT TRANSFERED TO
C          THE HEXCAN AND ITS DERIVATIVES.
C
      QV = ACOV*HCONV*(TV - TCAN)
      QL = ACOV*HCONL*(TL - TCAN)
      DQDTV = ACOV*HCONV*(ONE - DTC*HCONV)
      DQDTL = ACOV*HCONL*(ONE - DTC*HCONL)
      RETURN
      END
```

```
      SUBROUTINE FPROP(TRN,NPIN,NPM1,I)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /NUMBER/ ZERO,ONE,BIG,SMALL
      COMMON /PIN1/ CPIN(20),ROCP(20)
      COMMON /ICONST/ NCF,NCC,NG
      DIMENSION TRN(NPIN)
C
C           FUEL PROPERTIES
C
      DO 1 K = 1,NCF
      T = (TRN(K+1) + TRN(K))/2.D0
      CALL FUEL (T,K,I)
    1 CONTINUE
C
C           CLAD PROPERTIES
C
      DO 2 K = NCC,NPM1
      T = (TRN(K+1) + TRN(K))/2.D0
      CALL CLAD (T,K)
    2 CONTINUE
C
C           GAP CONDUCTIVITY
C
      T = (TRN(NG+1) + TRN(NG))/2.D0
      CALL GAP (T,TRN(NG),TRN(NG+1),NG)
      RETURN
      END
```

Subroutine fuel

```
      SUBROUTINE FUEL (T,K,I)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /NUMBER/ ZERO,ONE,BIG,SMALL
      COMMON /PIN1/ CPIN(20),ROCP(20)
      COMMON /FCONST/ A0,A1,A2,A3,
     *                B0,B1,B2,AD,APU,LPLNM(40)
C
      T2 = T*T
      T3 = T*T2
      X = 2.74D0 - 5.8D-04*T
C
      CPIN(K) = (B0 + B1*T + B2*T2)*(ONE - (ONE - AD)*X)
      ROCP(K) = (A0 + A1*T + A2*T2 + A3*T3)*AD*(ONE + 0.045*APU)
      IF(LPLNM(I).EQ.0) ROCP(K) = 1.D+04
      RETURN
      END
```

```
3037        SUBROUTINE CLAD (T,K)
3038        IMPLICIT REAL*8 (A-H,O-Z)
3039        COMMON /NUMBER/ ZERO,ONE,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 (T,TF,TC,NG)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /NUMBER/ ZERO,ONE,BIG,SMALL
      COMMON /PIN1/ CPIN(20),ROCP(20)
      COMMON /GCONST/ DIL,RADFU,RADCL
C
      DATA ESB,HMIN /1.7D-08,3.705D+03/
      DATA C1,C2 /2.D0,1.5D+01/
      DATA G1,G2,G3 /1.32D-04,0.61D-04,1.8D+03/
C
C             CONDUCTION HEAT TRANSFER
C
      DGAP = RADCL - RADFU
      CG = C2**DIL*C1
      HG = ONE/((DGAP + G1)/CG + G2) + G3
C
C             RADIATION HEAT TRANSFER
C
      HR = (TF*TF + TC*TC)*(TF + TC)*ESB
      HGAP = HG + HR
      IF(HGAP.LT.HMIN) HGAP = HMIN
C
      ROCP(NG) = ZERO
      CPIN(NG) = HGAP
      RETURN
      END
```

```
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 /PIN0/ RODR(20),VP(20),VM(20),RADR,PPP(20)
3082        COMMON /PIN1/ 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) = RODR(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) = -RODR(KM1)*CPIN(KM1)
3098        A2(K) = -A1(K) + RODR(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) = -RODR(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          DO 3 K = 1,NPM1
3127          TR(K) = B1(K)
3128          DTR(K) = A3(K)
3129        3 CONTINUE
3130          RETURN
3131          END
```

```
      SUBROUTINE FTP(TV,TL,TS,TW,HCONV,HCONL,HNB,TR,DTR,QPP,
     *               NI,NJ,NN,NP,NTR,NPM1,NIM2,NPIN)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /NUMBER/ ZERO,ONE,BIG,SMALL
      DIMENSION TR(NTR),DTR(NTR),TW(NP),TS(NN),TV(NN),TL(NN),
     *          HCONV(NP),HCONL(NP),HNB(NP),QPP(NN)
C
      TWMAX = ZERO
      TRMAX = ZERO
C
      DO 3 I = 1,NIM2
      DO 3 J = 1,NJ
      KO = (J-1)*NI + I + 1
      KP = (J-1)*NIM2 + I
      KR = KP*NPIN
C
      TR(KR) = TW(KP)
C
      DO 1 KK = 1,NPM1
      KTR = KR - KK
C
      TR(KTR) = TR(KTR) - DTR(KTR)*TR(KTR+1)
      IF(TRMAX.GT.TR(KTR)) GO TO 1
      TRMAX = TR(KTR)
      KTRMAX = KTR
    1 CONTINUE
C
      IF(TWMAX.GT.TW(KP)) GO TO 2
      TWMAX = TW(KP)
      KTWMAX = KO
    2 CONTINUE
C
      QPP(KP) = HCONV(KP)*(TW(KP) - TV(KO)) + HCONL(KP)*
     *          (TW(KP)-TL(KO)) + HNB(KP)*(TW(KP)-TS(KO))
    3 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE THXCN(TV,TL,HCONV,HCONL,TCAN,DT,NN,NI,NJ,NCAN,
     *                 NIM1,NIM2)
      IMPLICIT REAL*8 (A-H,O-Z)
      LOGICAL LSS
      COMMON /NUMBER/ ZERO,ONE,BIG,SMALL
      COMMON /STST/ TAFP,LSS
      DIMENSION TV(NN),TL(NN),HCONV(NN),HCONL(NN),TCAN(NCAN)
C
C
C         SUBROUTINE THXCN PERFORMS THE FIRT CALCULATION OF THE
C         HEXCAN TEMPERATURE.
C
      DTI = ONE/DT
      IF(LSS) DTI = ZERO
C
      DO 10 I = 2,NIM1
      KO = (NJ-1)*NI + I
      KP = (NJ-1)*NIM2 + I - 1
      K2 = NI + I
      K3 = K2 + NI
      K4 = K3 + NI
C
      TCAN(K2) = ONE/(TCAN(K4)*DTI + HCONV(KP) + HCONL(KP))
      TCAN(I) = (TCAN(K4)*TCAN(K3)*DTI + HCONV(KP)*TV(KO) +
     +           HCONL(KP)*TL(KO))*TCAN(K2)
   10 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE THXCN0(TCAN,NCAN,NI)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION TCAN(NCAN)
C
C         SUBROUTINE THXCN0 TRANSFERS THE NEW VALUE OF THE HEXCAN
C         TEMPERATURE TO THE OLD HEXCAN TEMPERATURE ARRAY.
C
      TCAN(1) = TCAN(2)
      TCAN(NI) = TCAN(NI-1)
      DO 10 I = 1,NI
      K3 = 2*NI + I
      TCAN(K3) = TCAN(I)
   10 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE POWER (HEAT,KO)
      IMPLICIT REAL*8 (A-H,O-Z)
      LOGICAL LP
      COMMON /ERROR/ IERR
      COMMON /NUMBER/ ZERO,ONE,BIG,SMALL
      COMMON /PSHAPE/ SHAPE(100)
      COMMON /TEMPO/ TIME,DT,DTO,DTLS,NDT
      COMMON /BCOND/ TB(51),PNB1(51),PNB2(51),PNB3(51),OMP(51),
     *               PNT1(51),PNT2(51),PNT3(51),OMT(51),ALB1(51),
     *               ALB2(51),ALB3(51),OMA(51),TVB1(51),TVB2(51),
     *               TVB3(51),OMV(51),TLB1(51),TLB2(51),TLB3(51),
     *               OML(51),HNW1(51),HNW2(51),HNW3(51),OMH(51),
     *               LMAX,LP(51)
C
C
      L = 2
    1 CONTINUE
      IF(TIME.LE.TB(L)) GO TO 2
      L = L + 1
      IF(L.GT.LMAX) RETURN
      GO TO 1
    2 CONTINUE
      DTIME = TIME - TB(L-1)
      HEAT = HNW1(L)*DTIME + HNW2(L)
      IF(LP(L)) GO TO 3
      HEAT = DCOS(OMH(L)*DTIME)*HEAT + HNW3(L)
    3 CONTINUE
      HEAT = SHAPE(KO)*HEAT
      RETURN
      END
```

```
      SUBROUTINE GAUSIE (A1,A2,A3,A4,F,X,BETA,GAMMA,NC)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /NUMBER/ ZERO,ONE,BIG,SMALL
      COMMON /GAUSS/ NZ,NR,NZM1
      COMMON /ERROR/ IERR
      COMMON /CNTRL/ EPS1,EPS2,RES,IT1,IT2,IT3,ITM1,ITM2,ITRMAX
      DIMENSION A1(NC),A2(NC),A3(NC),A4(NC),F(NC),X(NC),
     *          BETA(NC),GAMMA(NC)
C
      ITR = 0
    1 CONTINUE
C
C          NEW SOLUTION AT THE BOTTON
C
      I = 1
      GAMMA(1) = F(I) - A3(I)*X(I+1)
      BETA(1)  = ONE
C
      DO 2 J = 2,NR
      K = (J-1)*NZ + I
      K1 = K - NZ
C
      BETA(J) = ONE - A1(K)*A4(K1)/BETA(J-1)
      GAMMA(J) = (F(K)-A3(K)*X(K+1) - A1(K)*GAMMA(J-1))/BETA(J)
    2 CONTINUE
C
      K = (NR-1)*NZ + I
      CONV = DABS(X(K) - GAMMA(NR))
      X(K) = GAMMA(NR)
      DO 3 J = 2,NR
      K = NR - J + 1
      KX = (K-1)*NZ + I
      XA = GAMMA(K) - A4(KX)*X(KX+NZ)/BETA(K)
      DX = DABS(X(KX) - XA)
      IF(DX.GT.CONV) CONV = DX
      X(KX) = XA
    3 CONTINUE
C
C          NEW SOLUTION OUT OF THE BOUNDARIES
C
      DO 6 I = 2,NZM1
C
      GAMMA(1) = F(I) - A2(I)*X(I-1) - A3(I)*X(I+1)
```

```
      DO 4 J = 2,NR
      K = (J-1)*NZ + I
      K1 = K - NZ
C
      BETA(J) = ONE - A1(K)*A4(K1)/BETA(J-1)
      GAMMA(J)= (F(K) - A2(K)*X(K-1) - A3(K)*X(K+1) -
     -           A1(K)*GAMMA(J-1))/BETA(J)
    4 CONTINUE
C
      K = (NR-1)*NZ + I
      DX = DABS(X(K) - GAMMA(NR))
      IF(DX.GT.CONV) CONV = DX
      X(K) = GAMMA(NR)
C
      DO 5 J = 2,NR
      K = NR - J + 1
      KX = (K-1)*NZ + I
      XA = GAMMA(K) - A4(KX)*X(KX+NZ)/BETA(K)
      DX = DABS(X(KX) - XA)
      IF(DX.GT.CONV) CONV = DX
      X(KX) = XA
    5 CONTINUE
    6 CONTINUE
C
C         NEW SOLUTION AT THE TOP
C
      I = NZ
      GAMMA(1) = F(I) - A2(I)*X(I-1)
      DO 7 J = 2,NR
      K = (J-1)*NZ + I
      K1 = K - NZ
C
      BETA(J) = ONE - A1(K)*A4(K1)/BETA(J-1)
      GAMMA(J) = (F(K) - A2(K)*X(K-1) - A1(K)*GAMMA(J-1))/
     /                  BETA(J)
    7 CONTINUE
C
      K = (NR-1)*NZ + I
      DX = DABS(X(K) - GAMMA(NR))
      IF(DX.GT.CONV) CONV = DX
      X(K) = GAMMA(NR)
C
      DO 8 J = 2,NR
      K = NR - J + 1
      KX = (K-1)*NZ + I
```

```
      XA = GAMMA(K) - A4(KX)*X(KX+NZ)/BETA(K)
      DX = DABS(X(KX) - XA)
      IF(DX.GT.CONV) CONV = DX
      X(KX) = XA
    8 CONTINUE
C
C        CONVERGENCE TEST
C
      IF(CONV - EPS2) 11,11,9
    9 IF(ITR - ITRMAX) 1,10,10
   10 IERR = 1
   11 CONTINUE
      RES = ZERO
      DO 12 L = 1,NC
      XX = DABS(X(L))
      IF(XX.GT.RES) RES = XX
   12 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE ERRMES(TIME)
C
C               SUBROUTINE ERRMES PRINTS THE ERROR MESSAGES
C               WHENEVER THE EXECUTION OF THE PROGRAM HAS
C               BEEN TERMINATED DUE TO NUMERICAL ERRORS SUCH
C               AS INSTABILITY,VARIABLES OUT OF RANGE ETC.
C
C
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /ERROR/ IERR
C
      WRITE(6,1100) TIME
C
      IF(IERR - 2) 1,2,100
  100 IF(IERR - 4) 3,4,101
  101 IF(IERR - 22) 21,22,102
  102 IF(IERR - 24) 23,24,103
  103 IF(IERR - 26) 25,26,104
  104 IF(IERR - 28) 27,50,50
C
    1 WRITE(6,1001)
      GO TO 200
    2 WRITE(6,1002)
      GO TO 200
    3 WRITE(6,1003)
      GO TO 200
    4 WRITE(6,1004)
      GO TO 200
   21 WRITE(6,1021)
      GO TO 200
   22 WRITE(6,1022)
      GO TO 200
   23 WRITE(6,1023)
      GO TO 200
   24 WRITE(6,1024)
      GO TO 200
   25 WRITE(6,1025)
      GO TO 200
   26 WRITE(6,1026)
      GO TO 200
   27 WRITE(6,1027)
      GO TO 200
   50 WRITE(6,1050)
```

```
C
  200 CONTINUE
      WRITE(6,1101)
 1100 FORMAT(1H1,35(' *')//10X,'EXECUTION TERMINATED ON ERROR',
     *' CONDITION AT TIME ',F10.4//)
 1001 FORMAT(1X,'THE PRESSURE MATRIX INVERSION DOES NOT CONVERGE'/
     *       1X,'IN THE MAXIMUM NUMBER OF ITERATIONS ALLOWED'//
     *       1X,'ERROR CONDITION NUMBER = 1'/)
 1002 FORMAT(1X,'THE PRESSURE MATRIX IS NOT DIAGONAL DOMINANT'//
     *       1X,'ERROR CONDITION NUMBER = 2'/)
 1003 FORMAT(1X,'THE VOID FRACTION TOOK A VALUE EITHER LOWER THAN'/
     *       1X,'ZERO OR GREATER THAN ONE'//
     *       1X,'ERROR CONDITION NUMBER = 3'/)
 1004 FORMAT(1X,'THE INITIAL CONDITIONS INPUT DATA IS NOT IN THE'/
     *       1X,'PROPER ORDER'//
     *       1X,'ERROR CONDITION NUMBER = 4'/)
 1021 FORMAT(1X,'THE TIME STEP SIZE TOOK A VALUE TOO SMALL'/
     *       1X,'ERROR CONDITION NUMBER = 21'/)
 1022 FORMAT(1X,'THE PRESSURE TOOK A VALUE TOO SMALL'/
     *       1X,'ERROR CONDITION NUMBER = 22'/)
 1023 FORMAT(1X,'THE PRESSURE TOOK A VALUE TOO HIGH'/
     *        1X,'ERROR CONDITION NUMBER = 23'/)
 1024 FORMAT(1X,'THE VAPOR TEMPERATURE TOOK A VALUE TOO SMALL'/
     *       1X,'ERROR CONDITION NUMBER = 24'/)
 1025 FORMAT(1X,'THE VAPOR TEMPERATURE TOOK A VALUE TOO HIGH'/
     *       1X,'ERROR CONDITION NUMBER = 25'/)
 1026 FORMAT(1X,'THE LIQUID TEMPERATURE TOOK A VALUE TOO SMALL'/
     *        1X,'ERROR CONDITION NUMBER = 26'/)
 1027 FORMAT(1X,'THE LIQUID TEMPERATURE TOOK A VALUE TOO HIGH'/
     *       1X,'ERROR CONDITION NUMBER = 27'/)
 1050 FORMAT(1X,'A QUIT SIGNAL WAS ISSUED BY THE TERMINAL OPERATOR'/
     *       1X,'ERROR CONDITION NUMBER = 50'/)
 1101 FORMAT(1X,35(' *'))
      RETURN
      END
```

Subroutine saver

```
      SUBROUTINE SAVER(P,TV,TL,ALFA,UVZ,ULZ,UVR,ULR,TR,TCAN,
     *                 TIME,NTR,NN,NCAN,NI)
      IMPLICIT REAL*8 (A-H,O-Z)
      LOGICAL LDATA
      DIMENSION P(NN),TV(NN),TL(NN),ALFA(NN),UVZ(NN),ULZ(NN),
     *          UVR(NN),TCAN(NCAN),TR(NTR),ULR(NN)
      DIMENSION XOUT(5)
      LDATA = .FALSE.
      WRITE(7,103) LDATA,TIME
      DO 1 KO = 1,NN
      WRITE(7,100) KO,TV(KO),TL(KO),P(KO),ALFA(KO)
      WRITE(7,100) KO,UVZ(KO),ULZ(KO),UVR(KO),ULR(KO)
    1 CONTINUE
      LDATA = .TRUE.
      KRES = 0
    2 CONTINUE
      DO 3 K = 1,5
      KM = KRES + K
      IF(KM.GT.NTR) GO TO 4
      XOUT(K) = TR(KM)
    3 CONTINUE
      WRITE(7,101) LDATA,(XOUT(KL),KL=1,5)
      KRES = KRES + 5
      GO TO 2
    4 CONTINUE
      WRITE(7,101) LDATA,(XOUT(KL),KL=1,5)
      LDATA = .FALSE.
      WRITE(7,102) LDATA
      LDATA = .TRUE.
      KRES = 2*NI
      K3 = 3*NI
    5 CONTINUE
      DO 6 K = 1,5
      KM = KRES + K
      IF(KM.GT.K3) GO TO 7
      XOUT(K) = TCAN(KM)
    6 CONTINUE
      WRITE(7,101) LDATA,(XOUT(KL),KL=1,5)
      KRES = KRES + 5
      GO TO 5
    7 CONTINUE
      WRITE(7,101) LDATA,(XOUT(KL),KL=1,5)
      LDATA = .FALSE.
```

```
      WRITE(7,102) LDATA
  100 FORMAT(I5,4D15.9)
  101 FORMAT(L1,5D15.9)
  102 FORMAT(L1)
  103 FORMAT(L1,D15.9)
      RETURN
      END
```