ENERGY LABORATORY

MASSACHUSETTS INSTITUTE
OF TECHNOLOGY

DEVELOPMENT OF MODELS FOR THE TWO-DIMENSIONAL, TWO-FLUID CODE FOR SODIUM BOILING NATOF-2D

by

R. G. Zielinski and M. S. Kazimi

Energy Laboratory Report No. MIT-EL 81-030

September 1981

Energy Laboratory

and

Department of Nuclear Engineering

Massachusetts Institute of Technology
Cambridge, Mass. 02139

DEVELOPMENT OF MODELS FOR THE TWO-DIMENSIONAL, TWO-FLUID CODE FOR SODIUM BOILING NATOF-2D

by

R. G. Zielinski and M. S. Kazimi

September 1981

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 80-030

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

A. Topical Reports (For availability check Energy Laboratory Headquarters, 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 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 No. MIT-EL-78-026, September 1978.

M. S. Kazimi and M. Massoud, "A Condensed Review of Nuclear Reactor Thermal-Hydraulic Computer Codes for Two-Phase Flow Analysis," MIT Energy Laboratory Report No. 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 No. MIT-EL-79-046.

J.N. Loomis and W.D. Hinkle, "Reactor Core Thermal-Hydraulic Analysis--Improvement and Application of the Code COBRA-IIIC/MIT," MIT Energy Laboratory Report No. MIT-EL-80-027, September 1980.

D.P. Griggs, A.F. Henry and M.S. Kazimi, "Development of a Three-Dimensional Two-Fluid Code with Transient Neutronic Feedback for LWR Applications," MIT Energy Laboratory No. MIT-EL-81-013, April 1981.

J.E. Kelly, S.P. Kao and M.S. Kazimi, "THERMIT-2: A Two-Fluid Model for Light Water Reactor Subchannel Transient Analysis," MIT Energy Laboratory Report No. MIT-EL-81-014, April 1981.

H.C. No and M.S. Kazimi, "The Effect of Virtual Mass on the Characteristics and the Numerical Stability in Two-Phase Flows," MIT Energy Laboratory Report No. MIT-EL-81-023, April 1981.

J.W. Jackson and N.E. Todreas, "COBRA IIIC/MIT-2: A Digital Computer Program for Steady State and Transient Thermal-Hydraulic Analysis of Rod Bundle Nuclear Fuel Elements," MIT-EL-81-018, June 1981.

J.E. Kelly, S.P. Kao, and M.S. Kazimi, "User's Guide for THERMIT-2: A Version of THERMIT for both Core-Wide and Subchannel Analysis of Light Water Reactors," MIT Energy Laboratory Report No. MIT-EL 81-029, August 1981.

A.2 P. Moreno, C. Chiu, R. Bowring, E. Khan, J. Liu, and N. Todreas, "Methods for Steady-State Thermal/Hydraulic Analysis of PWR Cores," MIT Energy Laboratory Report No. MIT-EL 76-006, Rev. 1, July 1977, (Orig. 3/77).

J. Liu and N. Todreas, "Transient Thermal Analysis of PWR's by a Single Pass Procedure Using a Simplified Model Layout," MIT Energy Laboratory Report No. MIT-EL 77-008, Final, February 1979 (Draft, June 1977).

J. Liu and N. Todreas, "The Comparison of Available Data on PWR Assembly Thermal Behavior with Analytic Predictions," MIT Energy Laboratory Report No. MIT-EL 77-009, Final February 1979, (Draft, June 1977).

A.3 L. Guillebaud, A. Levin, W. Boyd, A. Faya, and 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 Energy Laboratory No. 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 Report," 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.D. Hinkle, M.S. 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 MTI 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. S. 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.

R.G. Zielinski and M.S. Kazimi, "Development of Models for the Two-Dimensional, Two-Fluid Code for Sodium Boiling NATOF-2D," MIT Energy Laboratory Report No. MIT-EL-81-030, September 1981.

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," Heat Transfer-Orlando 1980, AIChE Symposium Series 199, Vol. 76, August 1980.

J.E. Kelly and M.S. Kazimi, "THERMIT, A Three-Dimensional, Two-Fluid Code for LWR Transient Analysis," Transactions of American Nuclear Society, 34, p. 893, 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.

S.P. Kao and M.S. Kazimi, "CHF Predictions In Rod Bundles," Trans. ANS, 35, 766 June 1981.

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.

DEVELOPMENT OF MODELS FOR THE TWO-DIMENSIONAL, TWO-FLUID CODE FOR SODIUM BOILING NATOF-2D

ABSTRACT

Several features were incorporated into NATOF-2D, a two-dimensional, two fluid code developed at M.I.T. for the purpose of analysis of sodium boiling transients under LMFBR conditions. They include improved interfacial mass, momentum and energy exchange rate models, and a cell-to-cell radial heat conduction mechanism which was calibrated by simulation of Westinghouse Blanket Heat Transfer Test Program Runs 544 and 545. Finally, a direct method of pressure field solution was implemented into NATOF-2D, replacing the iterative technique previously available, and resulted in substantially reduced computational costs.

The models incorporated into NATOF-2D were tested by running the code to simulate the results of the THORS Bundle 6A Experiments performed at Oak Ridge National Laboratory, and four tests from the W-1 SLSF Experiment performed by the Hanford Engineering Development Laboratory. The results demonstrate the increased accuracy provided by the inclusion of these effects.

NOTICE

ACKNOWLEDGEMENTS

Funding for this project was provided by the United States Department of Energy. This support was deeply appreciated.

A very special thanks is due to Andrei L. Schor, whose enthusiasm for Sodium Boiling provided a constant source of information and inspiration.

The work described in this report was performed primarily by the principal author, Robert G. Zielinski, who submitted this work in partial fulfillment for the M.S. degree in Nuclear Engineering at M.I.T.

Table of Contents

List of Figures

List of Figures (continued)

List of Figures (continued)

List of Figures (continued)

List of Tables

Nomenclature

Letter	Definition	Units(SI)
a	Fuel pellet radius	m
A	Area	m^2
$\underline{\underline{A}}$	Matrix of pressure coefficients	-
Ar	Volumetrically averaged Radial area between cells	m^2
Ar*	Radial area constant	-
b	Fuel rod radius	m
CF	Condensation factor	-
d	Wire wrap diameter	m
D	Diameter	m
D_c	Conductive diameter	m
D_e	Equivalent diameter	m
f	friction factor	-
g	Gravity constant	m/s^2
h_{fg}	Enthalpy of vaporization	J/kg
h	Enthalpy	J/kg
H	Heat transfer coefficient	$w/m^2-{}^oK$
j	Mass flux	kg/m^2-s
K	Momentum exchange coefficient	kg/m^3-s
$\underline{\underline{L}}$	Lower triangular matrix	-
ℓ_{ij}	Effective conduction mixing length	-
ℓ^*_{ij}	Centroid-to-centroid distance of adjacent cells	m
M	Momentum exchange rate	kg/m^2-s^2

Nomenclature (continued)

Letter		Units (SI)
M	Molecular weight of particles	kg/mole
n	Row number	-
Nu	Nusselt Number	-
P	Pressure	N/m^2
P	Fuel pitch	m
P	Perimeter	m
q	Heat flux	W/m^2
q'''	Power density	W/m^3
q"	Heat flux	W/m^2
Q	Heat generation rate	W/m^3
r	Radial spacial coordinate	m
R	Universal gas constant	J/mole-^oK
S_{ij}	Length of common cell boundary	m
SBT	Stable boiling timestep	s
t	Time	s
T	Temperature	^oK
U	Velocity	m/s
$\underline{\underline{U}}$	Upper triangular matrix	-
V	Volume	m
w	Bandwidth of matrix	-
W	Total inlet mass flow rate	kg/s
z	Axial spacial coordinate	m

Nomenclature (continued)

Greek	Definition	Units(SI)
α	Void fraction	-
α	Thermal diffusivity	m^2/s
δ	Increment	-
Δ	Increment, spacing	-
Γ	Mass exchange rate	kg/m^3-s
η	Interfacial velocity weighting factor	-
ρ	Density	kg/m^3
σ	Mass exchange coefficient	-

Subscripts

r	Radial position
z	Axial position
v	Vapor
ℓ	Liquid
i	Interface
e	Evaporation
c	Condensation
s	Saturation
T	Total
int	Interface
c	Conduction
w	Wall

Nomenclature (continued)

Superscripts

n+1	New time step
n	Old time step

Chapter 1
INTRODUCTION

1.1 Description of the Code

The computer code NATOF-2D was developed at the Massachusetts Institute of Technology for the simulation of both steady state and transient conditions in Liquid Metal Fast Breeder Reactors / 1 /. The code uses the two fluid model of conservation equations, and a two-dimensional r-z geometry which takes advantage of the symmetry found in LMFBR bundles. The two dimensional nature of the calculation allows the multidimensional effects of sodium boiling to be observed, without the corresponding high computational costs of a three dimensional code.

The model treats the liquid and vapor phases separately, coupled by only the exchange coefficients. No assumption is made about the relationship between the properties of the two phases, which allows greater generality. The method thus requires the solution of the mass, momentum and energy conservation equations for each phase.

For calculational purposes, the fuel assembly is divided into a finite number of axial and radial mess cells. There is no constraint as to the positioning or number of axial levels other than at each level the mesh spacing

remains constant. However, the boundaries between cells in the radial direction must fall between the fuel pin centerlines, and so the number of radial cells is limited to the number of fuel pin rows. Figure 1.1 shows a typical arrangement of cells used by NATOF-2D.

The fluid properties of a cell are treated as the volumetric average of the properties in that cell, which necessitates the use of sufficiently small cells in order to obtain detailed information. The fluid velocities are evaluated at the faces of the cell, and are assumed to be uniform across each cell face. The unknowns of the calculation are P, α, T_v, T_ℓ, U_{vz}, U_{vr}, $U_{\ell z}$, and $U_{\ell r}$.

NATOF-2D uses a partially implicit method to solve the fluid dynamics equations. The terms involving sonic velocity and interfacial exchange are treated implicitly. However, for the convective terms, only the velocities are treated implicitly, while all other factors are evaluated at the previous timestep. This method imposes a timestep limitation such that

$$\Delta t \leq \frac{\Delta z}{U_z}$$

In most cases, this is not a detrimental constraint, since this timestep is usually the same order of magnitude as the time at which information is required.

The equations are solved by reduction to a Newton Iteration problem, in which the unknowns become linearized.

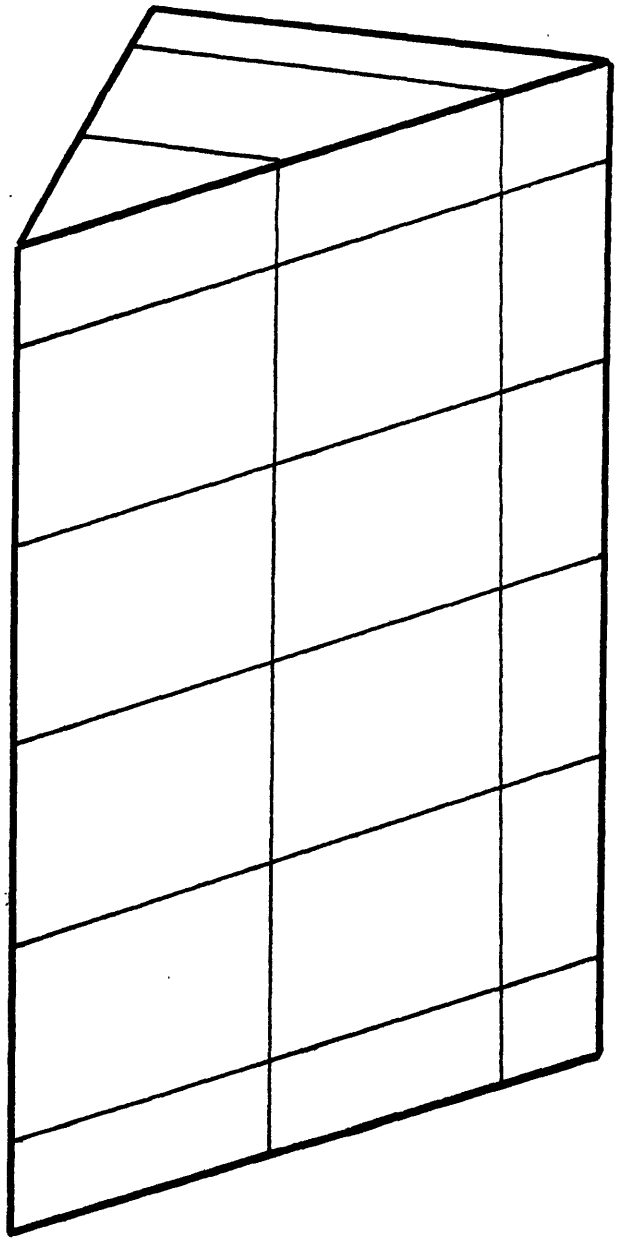


Figure 1.1 Typical Arrangement of Cells Used in NATOF-2D

These equations are further reduced to a set of linear equations involving only the pressures of a cell. The pressure field is then solved for by either an iterative or a direct technique, and all variables are then updated. The advantage of using a Newton Iteration technique is that a solution can always be attained by taking a sufficiently small timestep. The heat conduction equations are solved implicitly and coupled to the fluid dydnamics equations.

The code has the capability to operate with pressure, velocity or flow boundary conditions at the inlet, and a pressure boundary condition at the outlet. The velocity and flow inlet boundary conditions are new features incorporated into NATOF-2D, and are described in Appendix A.

NATOF-2D is able to handle the most severe sodium boiling conditions, including flow reversal. The work covered in this thesis addresses some of the major difficulties encountered in past sodium boiling simulations.

1.2 Scope of Work

1.2.1 Interfacial Mass, Energy and Momentum Exchange Models

The constituative equations used for the calculation of the interfacial mass, energy and momentum exchange rates have been improved to more physically account for the observed phenomena. The terms have a pronounced effect on the ability of the two-fluid model to simulate sodium boiling transients, since one of the major assumptions of this work is that for void fractions below 0.957 the vapor phase does not come in contact with the wall. Thus these terms often represent the only source of mass, momentum and energy for the vapor phase.

The mass exchange rate, which has the strongest effect of any constituative relation on the running of the code, has been implemented in a more basic form than before, using the kinetic theory of condensation. It is treated in a fully implicit manner so that all dependencies on the independent variables are accounted for. The momentum exchange rate has been modified to take into account the effects of mass exchange. Finally, the energy exchange rate has been modelled to prevent the appearance of highly subcooled vapor or superheated liquid in two phase flow transients.

1.2.2 Fluid Conduction Model

The high conductivity of liquid sodium coupled with the turbulence found in LMFBR bundles usually results in small radial temperature gradients across the core. Previously, the only mechanism available in NATOF-2D for the modelling of this phenomenon was energy exchange between cells due solely to mass transfer. However, the small radial velocities allowed large temperature differences to exist between internal channels and the edge channel.

Therefore, a radial heat conduction model has been incorporated into NATOF-2D. The model is applied only when single phase liquid is present in adjacent cells since the conductivity of the vapor phase is very low. Presently, only radial conduction has been employed in the code, since axial convection effects tend to dominate any axial conduction effects.

Calibration of the model is accomplished by simulation of two Westinghouse Blandket Heat Transfer Test Program experiments / 2 /. The model developed is also compared to analytic results based on conduction mixing length theory / 3 /.

1.2.3 Direct Solution of the Pressure Field

The computer time usage of NATOF-2D is strongly dependent on the solution technique used for the calculation of the pressure field. A more efficient method has been

implemented into NATOF-2D which uses a direct method to solve the pressure field matrix, rather than the iterative technique previously employed. The advantages of this are substantially reduced running time, and the capability of using smaller axial mesh cell spacings.

1.2.4 Comparison to Experiments on Boiling Behavior

The major experiences encountered while running NATOF-2D are documented in Chapter 5 to serve as a foundation for future work, and also provide an explanation for any changes deemed necessary to the previously derived models, especially the mass exchange rate. Also, some of the difficulties with sodium boiling codes in general are discussed.

Chapter 6 discusses the results obtained for five transients performed by NATOF-2D. One test was a simulation of the Thors Bundle 6A experiments conducted at Oak Ridge National Laboratory / 4 /, while the other four are from the SLSF W-1 experiments done at the Hanford Engineering Development Laboratory / 5 /.

Finally Chapter 7 summarizes the findings of this thesis, and makes recommendations for future development of NATOF-2D.

Chapter 2
INTERFACIAL MASS, ENERGY, AND MOMENTUM EXCHANGE MODELS

2.1 Introduction

In the two fluid model NATOF-2D, each phase in the flow field is described by a set of mass, energy, and momentum equations. Each of these equations takes into account the interactions which occur between the phases. This is accomplished by the use of empirical correlations or simple physical models, that describe the mass, energy and momentum exchange rates at the liquid/vapor interface.

One of the requirements of two phase flow modelling is that no mass, energy, or momentum be gained or lost at the interface. This is the so called "jump condition" at the interface. This requirement is met if the conservation equations of each phase can be summed together, and the interface exchange terms cancel each other.

For the sodium boiling transients which NATOF-2D was designed to simulate, these exchange rates take on a special significance. One of the basic assumptions of this work is that only the liquid phase is in contact with the wall for values of void fraction up to 0.957. Thus, for many applications, the vapor phase is entirely dependent on the liquid phase as a mass, energy or momentum source, and

thereby dependent on the accuracy of the exchange models incorporated into this code.

This chapter will cover the models developed for interfacial transport exchange, and compare the results with those previously used in NATOF-2D / 1 /.

2.2 Conservation Equations Used in NATOF-2D

Since this chapter deals with the modelling of the interfacial mass, enegy and momentum exchange rates, the conservation equations in the form used by NATOF-2D are summarized in this section. Since NATOF-2D is a two-phase, two-dimensional R-Z code, for each phase there will be one mass and one energy conservation equation, and two momentum equations (one for each direction) at each node. Given below are the eight conservation equations written in control volume form.

Mass Conservation

liquid phase:

$$\frac{\partial}{\partial t}\int_V (1-\alpha)\rho_\ell dV + \int_{A_{z+}} - \int_{A_{z-}} (1-\alpha)\rho_\ell U_{\ell z} dA + \int_{Ar+} - \int_{Ar-} (1-\alpha)\rho_\ell U_{\ell r} dA = -\int_V \Gamma \, dV \qquad (2.1)$$

vapor phase:

$$\frac{\partial}{\partial t}\int_V \alpha\rho_v dV + \int_{A_{z+}} - \int_{A_{z-}} \alpha\rho_v U_{vz} dA - \int_{A_{r+}} - \int_{A_{r-}} \alpha\rho_v U_{vr} dA = \int_V \Gamma \, dV \qquad (2.2)$$

Energy Conservation

liquid phase:

$$\frac{\partial}{\partial t}\int_V (1-\alpha)\rho_\ell (e_\ell + U_\ell^2/2)dV + \int_{A_{z+}} - \int_{A_{z-}} (1-\alpha)\rho_\ell U_\ell (e_\ell + U_\ell^2/2)dA +$$

$$\int_{A_{r+}} - \int_{A_{r-}} (1-\alpha)\rho_\ell U_{\ell r}(e_\ell + U_\ell^2/2)dA = \int_V Q_\ell dV - \int_V (1-\alpha)\rho_\ell g U_{\ell z}dV -$$

$$\int_{A_W} U_\ell \cdot f_\ell dA + \int_{A_\ell} P\cdot n\cdot U_\ell dA + \int_V P\,\frac{\partial \alpha}{\partial t}\,dV - \int_{A_i} q_{\ell i}dA \qquad (2.3)$$

vapor phase

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

$$\int_{A_{r+}} - \int_{A_{r-}} \alpha\rho_v U_{vr}(e_v + U_v^2/2)dA = \int_V Q_v dV - \int_V \alpha\rho_v g U_{vz}dV -$$

$$\int_{A_W} U_v \cdot f_v dA + \int_{A_V} P\cdot n\cdot U_v dA - \int_V P\,\frac{\partial \alpha}{\partial t}\,dV + \int_{A_i} q_{vi}dA \qquad (2.4)$$

Momentum Conservation--Axial Direction

liquid phase

$$\frac{\partial}{\partial t}\int_V (1-\alpha)\rho_\ell U_{\ell z}dV + \int_{A_{z+}} - \int_{A_{z-}} (1-\alpha)\rho_\ell U_{\ell z}^2 dA +$$

$$\int_{A_{r+}} - \int_{A_{r-}} (1-\alpha)\rho_\ell U_{\ell z}U_{\ell r}dA - \oint_{A_\ell} P\cdot\hat{k}\cdot\hat{n}\,dA = -\int_{A_W} f_{\ell z}dA -$$

$$\int_V (1-\alpha)\rho_\ell g\,dV - \int_V M_{\ell z}dV \qquad (2.5)$$

vapor phase

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

$$\oint_{A_v} P\cdot\hat{k}\cdot\hat{n}\,dA = -\int_{A_w} f_{vz}\,dA - \int_V \alpha\rho_v g\,dV + \int_V M_{vz}\,dV \qquad (2.6)$$

Momentum Conservation--Radial Direction

liquid phase

$$\frac{\partial}{\partial t}\int_V (1-\alpha)\rho_\ell U_{\ell r}\,dV + \int_{A_{z+}} - \int_{A_{z-}} (1-\alpha)\rho_\ell U_{\ell r}U_{\ell z}\,dA +$$

$$\int_{A_{r+}} - \int_{A_{r-}} (1-\alpha)\rho_\ell U_{\ell r}^2\,dA - \oint_{A_\ell} P\cdot\hat{r}\cdot\hat{n}\,dA =$$

$$-\int_{A_w} f_{\ell r}\,dA + \int_V M_{\ell r}\,dV \qquad (2.7)$$

vapor phase

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

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

(2.8)

2.3 Interfacial Mass Exchange

In the mass conservation equations for the liquid and vapor phase (equations 2.1 and 2.2) Γ represents the mass exchange rate between phases, and will be defined as positive for evaporation. Γ has units of kg/m^3-s. At the present time, the accepted model for the mass exchange rate is based on the kinetic theory of condensation. This model views the interaction simply as the difference between a flux of particles arriving at the interface, and a flux of particles departing from the interface. The particles are assumed to be arriving from the vapor phase, and departing from the liquid phase. When the arrival rate exceeds the departure rate, condensation is occurring. In the reverse situation, evaporation takes place and when the net flux is zero, an equilibrium condition exists. The derivation of the mass exchange rate is essentially due to Schrage / 6 /.

Using a Maxwell-Boltzmann distribution, it is possible to show that in a stationary container the mass flux of particles passing in either direction through the interface is given by:

$$j_i = \left(\frac{M}{2\pi R}\right)^{\frac{1}{2}} \frac{P}{T^{\frac{1}{2}}} \tag{2.9}$$

where

j_i = mass flux of phase i (kg/m^2-s)

M = molecular weight of particles

R = universal gas constant

P = pressure exerted by the particles

T = temperature of the particles

If there exists a progress velocity on the vapor side towards the interface such that $j_v = \rho_v V_p$ then

$$j_v = \left(\frac{M}{2\pi R}\right)^{\frac{1}{2}} \frac{P}{T^{\frac{1}{2}}} \Psi \quad . \tag{2.10}$$

where

$$\Psi = e^{-\phi^2} + \phi\pi(1 + \text{erf}\phi) \tag{2.11}$$

$$\phi = \frac{V_p}{(2RT/M)^{\frac{1}{2}}} = \frac{j}{\rho_v(2RT/M)^{\frac{1}{2}}} \tag{2.12}$$

V_p = progress velocity

At the liquid-vapor interface not all the molecules striking the surface will condense. Therefore, σ_c is defined as the fraction of molecules striking the surface which actually do condense. In a similar manner, σ_e represents the ratio of the flux of molecules actually leaving the interface to the flux given by equation 2.9.

At the condensing surface, molecules are arriving at a progress flow rate $\rho_v V_p$, and molecules are departing the surface at a rate equivalent to that of molecules in a

stationary container. Thus the net flux towards the surface is given by:

$$j = j_v - j_\ell$$

$$j = \left(\frac{M}{2\pi R}\right)^{\frac{1}{2}} \left(\sigma_c \Psi \frac{P_v}{T_v^{\frac{1}{2}}} - \sigma_e \frac{P_\ell}{T_\ell^{\frac{1}{2}}}\right) \tag{2.13}$$

If it is assume that $\phi << 1$, or in other words that the condensation rate is low, Ψ can be approximated by the following expression:

$$\Psi = \sqrt{\pi} \left(\frac{j}{\rho_v (2RT_v/M)^{\frac{1}{2}}} \right) + 1 \tag{2.14}$$

Substituting this into equation 2.13 yields

$$j = \frac{2}{2 - \sigma_c} \left(\frac{M}{2\pi R}\right)^{\frac{1}{2}} \left(\sigma_c \frac{P_v}{T_v^{\frac{1}{2}}} - \sigma_e \frac{P_\ell}{T_\ell^{\frac{1}{2}}}\right) \tag{2.15}$$

When the two phases are in equilibrium, the net flux, j, is equal to zero, and $\sigma_c = \sigma_e$. Since the values of the individual coefficients in non-equilibrium systems have not been determined, it is justified to set $\sigma_c = \sigma_e = \sigma$. Using this approximation the net flux becomes:

$$j = \frac{2\sigma}{2 - \sigma} \left(\frac{M}{2\pi R}\right)^{\frac{1}{2}} \left(\frac{P_v}{T_v^{\frac{1}{2}}} - \frac{P_\ell}{T_\ell^{\frac{1}{2}}}\right) \tag{2.16}$$

and the mass exchange rate is thus

$$\Gamma = -jA = A \frac{2\sigma}{2 - \sigma} \left(\frac{M}{2\pi R}\right)^{\frac{1}{2}} \left(\frac{P_\ell}{T_\ell^{\frac{1}{2}}} - \frac{P_v}{T_v^{\frac{1}{2}}}\right) \tag{2.17}$$

where

A = the interfacial area per unit volume

The literature shows a wide variation in the value of σ for sodium, ranging from $\sigma = 1.0$ at low pressures to $\sigma = 0.001$ at atmospheric pressures (See figure 2.1). Rohsenow / 7 /, however, attributes this variation to the presence of non-condensible gases which tend to congregate at the interface. These gases add an additional resistance to condensation. Tests conducted on nearly gas-free systems where the flow was high show that any gases present are swept away from the interface, and $\sigma = 1.0$ for all pressures.

In the models developed for NATOF-2D, it is assumed that only the liquid phase is in contact with the wall for values of void fraction up to α_{dryout}. Below this value, all heat gains to the vapor phase are solely from the liquid phase. When the liquid phase is evaporating, the vapor phase is entering the system at the saturation temperature. Similarly, condensation occurs when the liquid phases loses heat to the wall, and becomes subcooled. The vapor phase again condenses at the saturation temperature. Thus, for $\alpha < \alpha_{dryout}$, it is justified to set T_V to T_S and P_V to P_S in equation 2.17.

For values of $\alpha > \alpha_{dryout}$, the liquid becomes entrained in the vapor phase, and then it is the vapor which experiences the heat losses and gains. Thus for this case, $T_\ell = T_S$, and $P_\ell = P_S$ in equation 2.17. In order to obtain

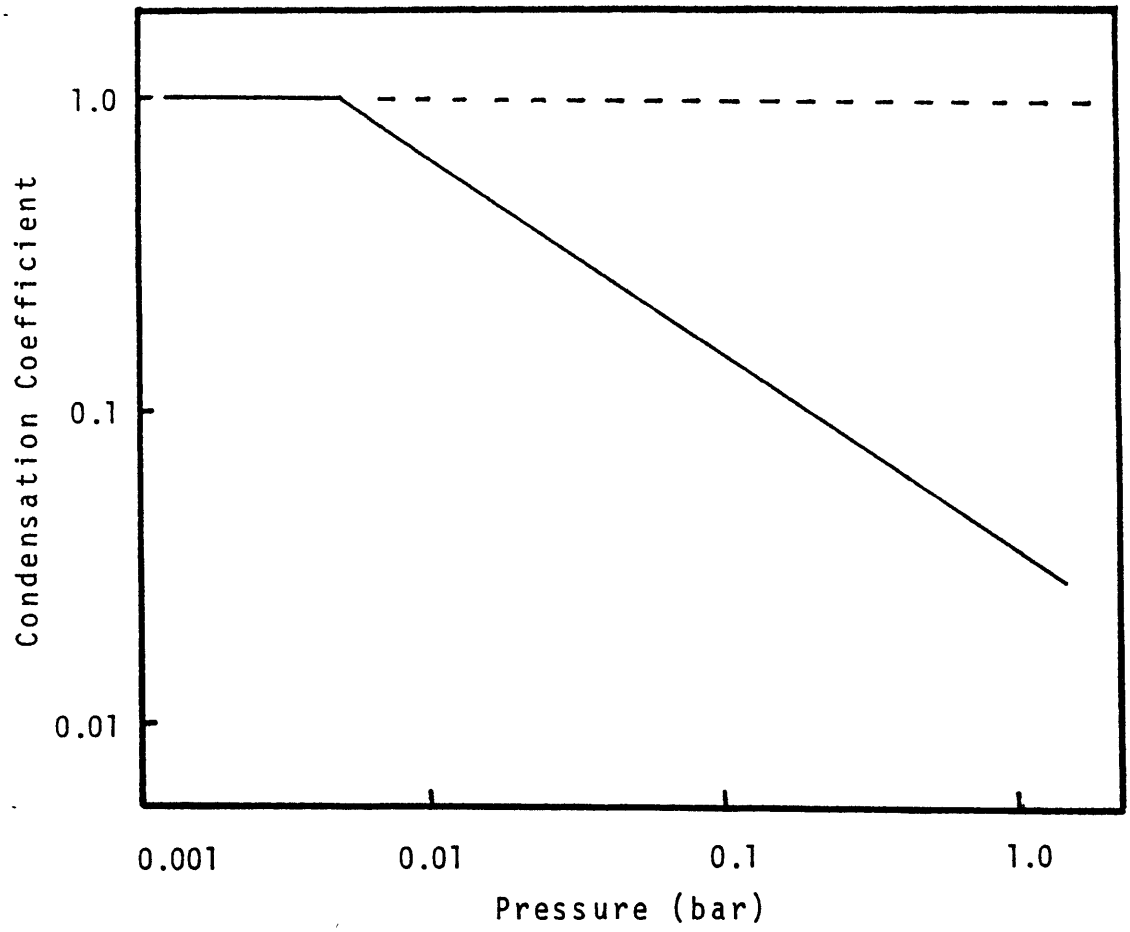


Figure 2.1 Condensation Coefficient as a Function of Pressure

the correct behavior of this relation, it is necessary to reverse the sign of equation 2.17 so that in a superheated vapor environment, the entrained liquid evaporates, instead of condensing. Equation 2.30 of the next section confirms this behavior.

For the range of temperatures in which sodium boiling and condensation occurs, $\ell_\ell^{\frac{1}{2}} \simeq T_S^{\frac{1}{2}}$ and $T_V^{\frac{1}{2}} \simeq T_S^{\frac{1}{2}}$. With this approximation, the final form of the mass exchange rate is arrived at.

$\alpha < \alpha_{dryout}$

$$\Gamma = A^{n+1} \frac{2\sigma}{2 - \sigma} \left(\frac{M}{2\pi R}\right)^{\frac{1}{2}} \left(\frac{P_\ell - P_S}{T_S^{\frac{1}{2}}}\right)^{n+1} \qquad (2.18)$$

$\alpha > \alpha_{dryout}$

$$\Gamma = A^{n+1} \frac{2\sigma}{2 - \sigma} \left(\frac{M}{2\pi R}\right)^{\frac{1}{2}} \left(\frac{P_V - P_S}{T_S^{\frac{1}{2}}}\right)^{n+1} \qquad (2.19)$$

where

P_V = pressure corresponding to a saturation temperature of T_V

P_ℓ = pressure corresponding to a saturation temperature of T_ℓ

P_S = system pressure

T_S = saturation temperature

A = interfacial area calculated implicitly

The formulations previously used for the mass exchange rate in NATOF-2D were:

For evaporation

$$\Gamma = \sigma A^{n} \alpha^{n} (1-\alpha)^{n+1} \sqrt{\frac{R}{2\pi M}} \left(\frac{\rho_v{}^2 h_{fg}}{P_s}\right)^{n} \left(\frac{T_\ell - T_s}{T_s^{\frac{1}{2}}}\right)^{n+1} \tag{2.20}$$

For condensation

$$\Gamma = \sigma A^{n} \alpha^{n+1} (1-\alpha)^{n} \sqrt{\frac{R}{2\pi M}} \left(\frac{\rho_v{}^2 h_{fg}}{P_s}\right)^{n} \left(\frac{T_v - T_s}{T_s^{\frac{1}{2}}}\right)^{n+1} \tag{2.21}$$

These relations were based on the assumption that $\Delta T/T_s << 1$, where $\Delta T = T_\ell - T_s$. Simulations by NATOF-2D, however, show that ΔT can be quite large. These results will be discussed in further detail in Chapter 5. The interfacial areas in equations 2.20 and 2.21 were calculated explicitly, and the term $\alpha(1 - \alpha)$ was added to force Γ to go to zero for single phase flow. The present formulation treats all terms implicitly, including the areas, and has eliminated the $\alpha(1 - \alpha)$ term. The areas are from Wilson / 8 /, and depend on the flow regime. The following is a summary of the equations used.

$\alpha < \alpha_m$

$$A_1 = \frac{3\alpha}{r_m} \qquad \alpha_m = \frac{2}{3}\left(\frac{r_m{}^2}{D}\right) \frac{\pi}{\sqrt{3}\,(P/D)^2 - \pi/2} \tag{2.22}$$

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

$\alpha_m < \alpha < 0.55$

$$A_2 = \frac{4}{D}\sqrt{\frac{3\pi\alpha}{2\sqrt{3}\,(P/D)^2 - \pi}} \tag{2.23}$$

$0.55 < \alpha < 0.65$

$$A_3 = a + b\cdot\bar{\alpha} + c\cdot\bar{\alpha}^2 + d\cdot\bar{\alpha}^3$$

where

$$\bar{\alpha} = \frac{\alpha - 0.55}{0.65 - 0.55} \qquad a = A_2 \qquad b = \frac{\partial A_2}{\partial \alpha}$$

$$c = 3(A_4 - A_3) - \frac{\partial A_4}{\partial \alpha} - 2\cdot\frac{\partial A_2}{\partial \alpha} \tag{2.24}$$

$$d = \frac{\partial A_4}{\partial \alpha} + \frac{\partial A_2}{\partial \alpha} + 2(A_2 - A_4)$$

$0.65 < \alpha < 0.957$

$$A_4 = \frac{4}{D}\sqrt{\frac{2\sqrt{3}\pi(P/D)^2}{(2\sqrt{3}(P/D)^2 - \pi)^2} - \frac{\pi\alpha}{2\sqrt{3}(P/D)^2 - \pi}} \tag{2.25}$$

$0.957 < \alpha < 1.0$

$$A_5 = A_4 \cdot \sqrt{\frac{1 - \alpha}{1 - 0.957}} \tag{2.26}$$

A transition regime area, A_3, which is a polynomial fit between A_2 and A_4, has been added in order to keep the areas and their derivatives with respect to α continuous. A comparison was made between the previous and present mass exchange rate formulations. The system pressure used for this comparison was 2 bars, and the results are shown in figure 2.2 and 2.3 for liquid superheats of $2^{\circ}C$ and $20^{\circ}C$ respectively. The results show that the new formulation predicts a more rapid vapor production especially in high void regions. Even discounting the effects of the $\alpha(1 - \alpha)$ term, the present mass exchange rate still is 2 to 4 times greater than the one previously implemented. Thus more vigorous and sustained boiling for the same superheats is expected.

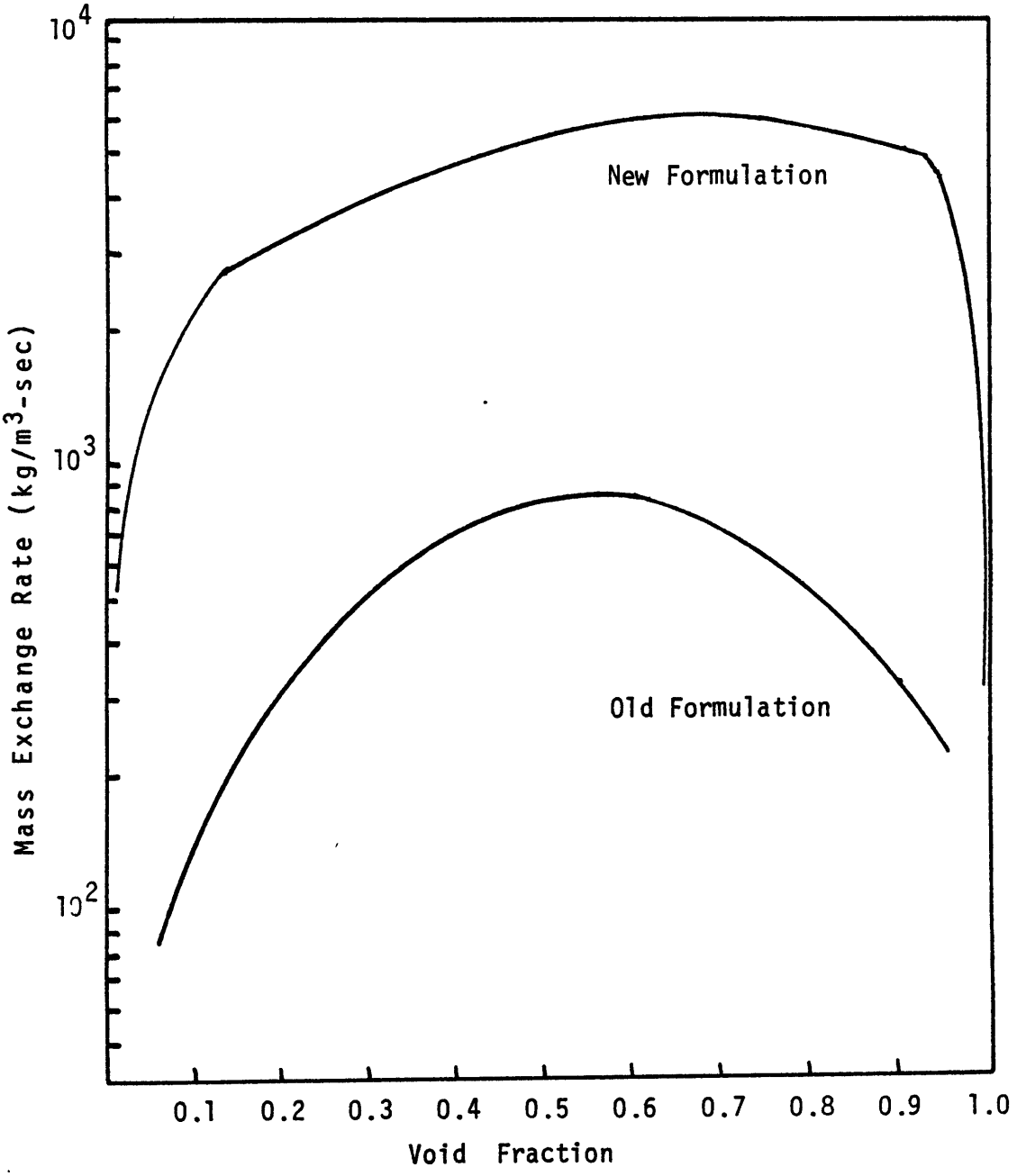


Figure 2.2 A Comparison of New and Old Mass Exchange Rates for a Superheat of 2^oC

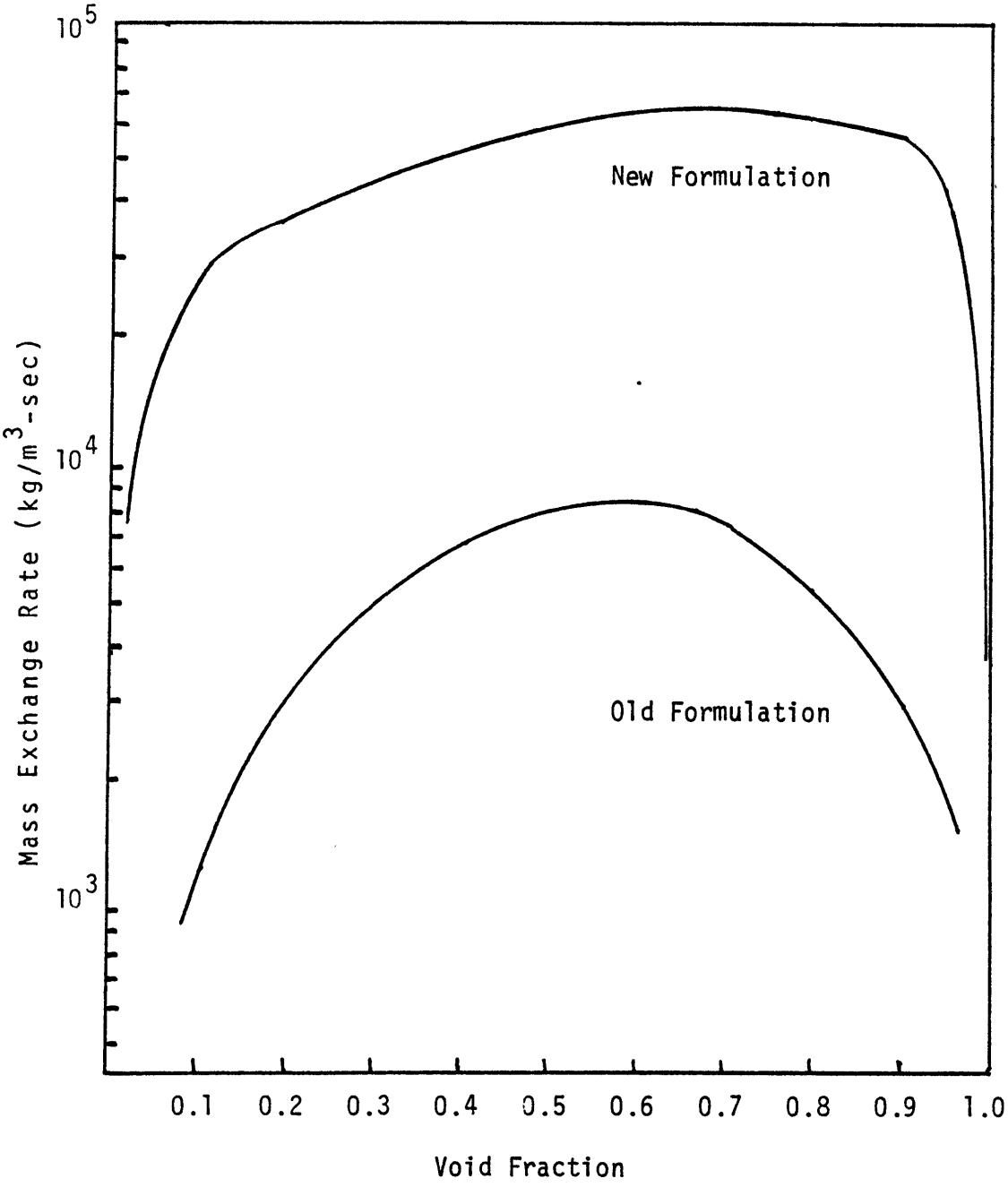


Figure 2.3 A Comparison of New and Old Mass Exchange Rates for a Superheat of 20°C

2.4 Energy Exchange Rate

Reliable constituative relations for interphase heat transfer are not available at the present time. This is due in part to the insufficient attention which this phenomenon has recieved until only recently, and also to the extreme difficulty in gathering useful data on the subject.

Starting with the two phase energy conservation equations, equations 2.3 and 2.4, one can define an energy exchange due to the difference in temperature between the phase and the interface, and an energy exchange associated with the heat transferred by virtue of mass exchange. With this premise, the energy exchange from the liquid/vapor interface to the vapor become

$$q_{iv} = \Gamma \cdot h_{vs} + A_i H_{iv}(T_i - T_v) \tag{2.27}$$

Similarly, the energy exchange from the liquid to the liquid/vapor interface is:

$$q_{\ell i} = \Gamma \cdot h_{\ell s} + A_i H_{\ell i}(T_\ell - T_i) \tag{2.28}$$

where

Γ = mass exhange rate

h_{vs} = enthalpy of the vapor at the saturation temperature

$h_{\ell s}$ = enthalpy of the liquid at the saturation temperature

A_i = interfacial area

H_{iv} = interface to vapor phase heat transfer coefficient

$H_{\ell i}$ = liquid phase to interface heat transfer coefficient

Since the "jump condition" at the interface requires that

$$q_{iv} = q_{\ell i}$$

we have

$$\Gamma \cdot h_{vs} + A_i H_{iv}(T_i - T_v) = \Gamma \cdot h_{\ell s} + A_i H_{\ell i}(T_\ell - T_s) \qquad (2.29)$$

Equation 2.29 can be used to solve for the mass exchange rate to yield

$$\Gamma = \frac{H_{iv}A_i(T_v - T_i) + H_{\ell i}A_i(T_\ell - T_i)}{h_{fg}} \qquad (2.30)$$

The above relationship shows that if H_{iv} and $H_{\ell i}$ were known, and if T_i was defined, the mass exchange rate would be determined. Unfortunately, there is a lack of data on the interface heat transfer coefficients at the present time. Therefore, an alternative is to use either equation 2.27 or 2.28 and the formulation given in section 2.3 for the interfacial energy exchange rate. One cannot use equation 2.27 for the vapor energy equation and equation 2.28 for the liquid energy equation simultaneously since there would be no guarantee that the jump condition was being satisfied.

Attempts to define an interface temperature with a

value somewhere in the range between the liquid and vapor temperatures have proven fruitless. For an interface temperature based on two infinite bodies in contact, T_i is given by the relation

$$\frac{T_v - T_i}{T_i - T_\ell} = \frac{\sqrt{(k\rho c_p)_\ell}}{\sqrt{(k\rho c_p)_v}} \qquad (2.31)$$

Since the conductivity and density of the liquid phase is so much greater than that of the vapor phase, solution of equation 2.31 yields $T_i \simeq T_\ell$. This result would be acceptable is T_ℓ stayed near the saturation temperature when both phases are present, but difficulties experienced in attaining a high sodium vapor condensation rate have resulted in vapor coexisting with liquid which is subcooled by as much as 100°C.

Therefore, the decision was made to set the interfacial temperature to the saturation temperature. The saturation temperature was chosen since it is the equilibrium temperature for a two-phase mixture. As previously stated, for values of $\alpha < \alpha_{dryout}$, the vapor gains heat solely from the liquid. In an evaporating state the assumption that $T_i = T_s$ implies that all the liquid superheat is utilized as latent heat for evaporation. And in a condensing state where $T_\ell < T_s$, the vapor is kept at the saturation temperature, and all heat losses from the vapor are by virtue of mass transfer to the liquid phase. For

$\alpha > \alpha_{dryout}$, the roles of each phase will be reversed. With this understanding, the final form of the interfacial energy exchange rate becomes:

$\alpha < \alpha_{dryout}$

$$q_i = \Gamma^{n+1} h_{vs} + A_i^{n+1} H_{iv} (T_s^{n+1} - T_v^{n+1}) \tag{2.32}$$

$\alpha > \alpha_{dryout}$

$$q_i = \Gamma^{n+1} h_{\ell s} + A^{n+1} H_{\ell i} (T_\ell^{n+1} - T_s^{n+1}) \tag{2.33}$$

where

$$H_{iv} = Nu \frac{k_\ell}{D_e}$$

$$H_{\ell i} = Nu \frac{k_v}{D_e}$$

The previous formulation of the interfacial heat exchange rate was

$$q_i = \Gamma_e h_{vs} + \Gamma_c h_{\ell s} + A_i H_i (T_\ell - T_v) \tag{2.34}$$

This formulation effectively kept T_v equal to T_ℓ, and led to situations of the vapor phase being subcooled by as much as 100 °C. The present formulation has eliminated this problem as is shown in figure 2.5.

The nusselt number chosen for the interfacial heat transfer coefficients has a pronounced effect on the temperature of the phases. To illustrate this, three

simulations of a sodium boiling transient were run in which only the interfacial nusselt number was varied. In these cases there was no switch in correlations at $\alpha=\alpha_{dryout}$. The temperatures given correspond to those found at the top of the heated section of the fuel bundle. As can be seen in figure 2.4, where the vapor and liquid temperatures have been plotted versus time, a small nusselt number (Nu = 0.006) leads to quite a variation between the vapor temperature and the saturation temperature. At approximately 0.55 seconds after boiling inception, at a void fraction corresponding to α_{dryout}, the vapor phase began to superheat to high levels. The liquid temperature stayed very close to the saturation temperature.

When Nu = 6, $T_v \simeq T_s$ and $T_\ell \simeq T_s$ as figure 2.5 indicates. This test case also showed that the saturation temperature was more stable with time, and less prone to wild fluctuations. For Nu = 6000, the results were about the same.

Based on these simulations, a value of Nu = 10 is recommended. A value in this range will keep the vapor at the saturation temperature, but not make the sensible heat contribution term the dominating one in equations 2.32 and 2.33.

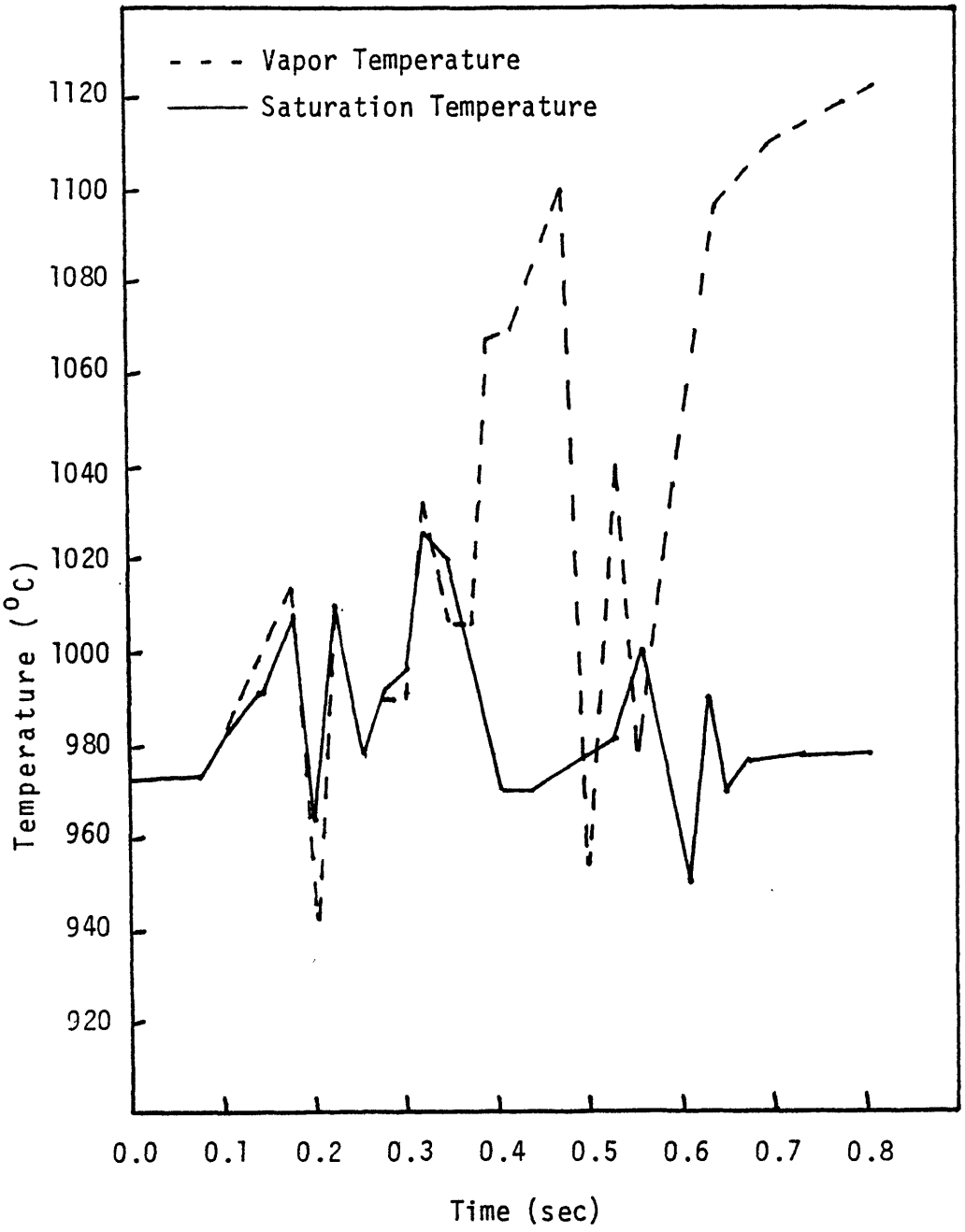


Figure 2.4 Variations Between Vapor and Saturation Temperatures for an Interfacial Heat Transfer Nusselt Number of 0.006

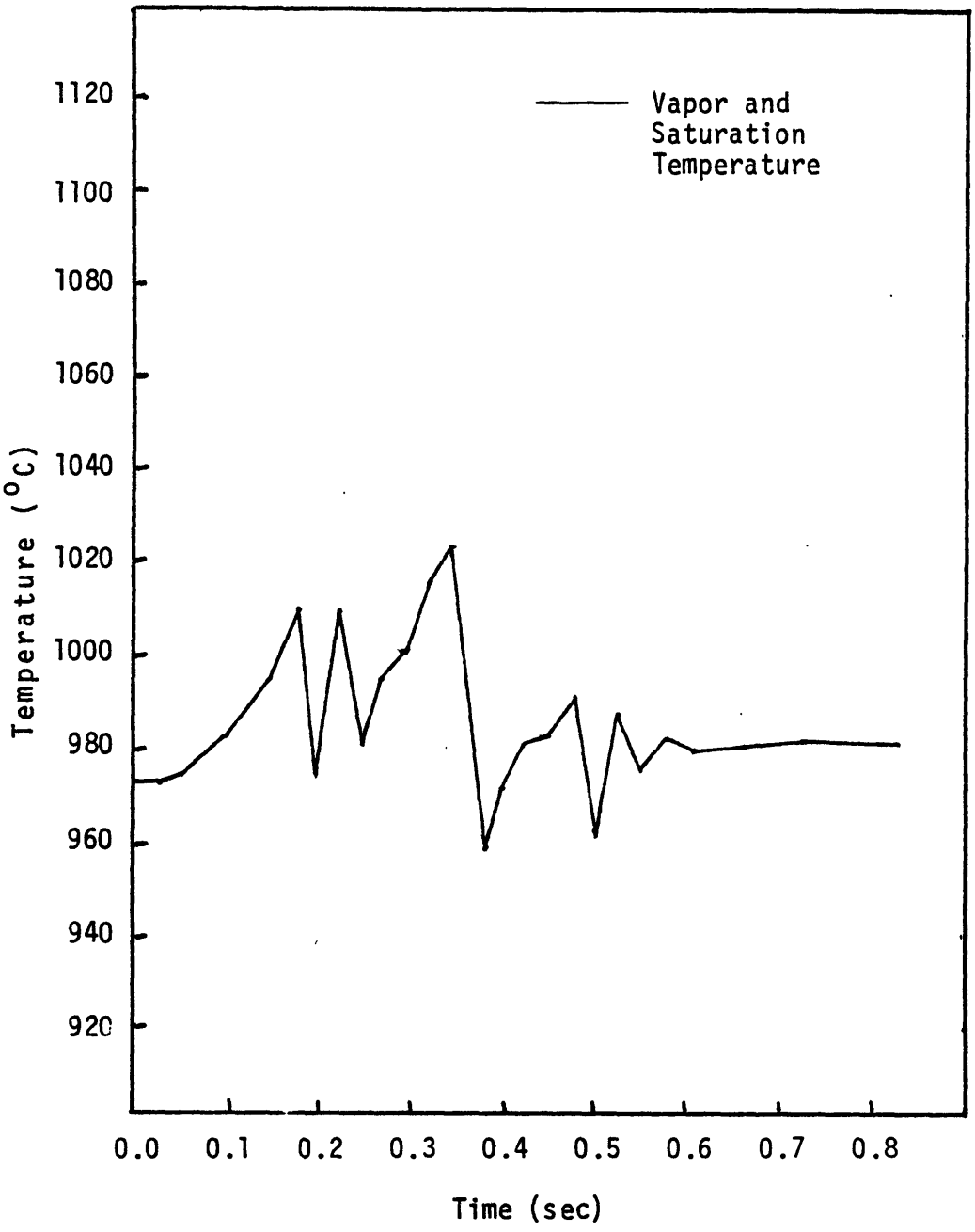


Figure 2.5 Vapor and Saturation Temperatures for an Interfacial Heat Transfer Nusselt Number of 6.0

2.5 Interfacial Momentum Transfer

The interfacial momentum exchange rate, similar to the interfacial energy exchange rate, is composed of two terms. The first term takes into account the momentum gain due to mass exchange across the interface, and the second term accounts for the effect of shear stresses at the interface. This section will show how these terms can be combined into a single term which contains both of these effects.

The momentum conservation equation for the vapor phase in the z-direction written in differential form is

$$\frac{\partial}{\partial t}(\alpha\rho_v U_{vz}) + \frac{\partial}{\partial z}(\alpha\rho_v U_{vz}^2) + \frac{\partial}{\partial r}(\alpha\rho_v U_{vr}U_{vz}) + \alpha\frac{\partial P}{\partial z} = -f_{wz} - \alpha\rho_v g - M_{iz} - U_{iz}\Gamma \qquad (2.35)$$

where

M_{iz} = shear stress contribution

$U_{iz}\Gamma$ = contribution due to mass exchange which is traveling at an interfacial velocity $u_i = \eta u_\ell + (1 - \eta)u_v$, where η is a weighting factor ($0 \leq \eta \leq 1$)

In order to facilitate the implimentation of the finite difference scheme utilized by NATOF-2D, it is necessary to cast equation 2.35 into non-conservative form. This is accomplished by applying the product rule of differentiation to the following terms:

$$\frac{\partial}{\partial t}(\alpha\rho_v U_{vz}) = \alpha\rho_v \frac{\partial U_{vz}}{\partial t} + U_{vz}\frac{\partial}{\partial t}(\alpha\rho_v) \tag{2.36}$$

$$\frac{\partial}{\partial z}(\alpha\rho_v U_{vz}^2) = \alpha\rho_v U_{vz}\frac{\partial U_{vz}}{\partial z} + U_{vz}\frac{\partial}{\partial z}(\alpha\rho_v U_{vz}) \tag{2.37}$$

$$\frac{\partial}{\partial r}(\alpha\rho_v U_{vz}U_{vr}) = \alpha\rho_v U_{vr}\frac{\partial U_{vz}}{\partial r} + U_{vz}\frac{\partial}{\partial r}(\alpha\rho U_{vr}) \tag{2.38}$$

Substituting these values into equation 2.35, the vapor momentum equation becomes:

$$\alpha\rho_v \frac{\partial U_{vz}}{\partial t} + U_{vz}\frac{\partial}{\partial t}(\alpha\rho_v) + \alpha\rho_v U_{vz}\frac{\partial U_{vz}}{\partial t} + U_{vz}\frac{\partial}{\partial z}(\alpha\rho U_{vz}) +$$
$$\alpha\rho_v U_{vr}\frac{\partial U_{vz}}{\partial r} + U_{vz}\frac{\partial}{\partial r}(\alpha\rho U_{vr}) + \alpha\frac{\partial P}{\partial z} =$$
$$-f_{wz} - \alpha\rho_v g - M_{iz} + U_{iz}\Gamma \tag{2.39}$$

The vapor mass conservation equation is given by:

$$\frac{\partial}{\partial t}(\alpha\rho_v) + \frac{\partial}{\partial z}(\alpha\rho_v U_{vz}) + \frac{\partial}{\partial r}(\alpha\rho_v U_{vr}) = \Gamma \tag{2.40}$$

and this can be substituted into equation 2.39 to yield the non-conservative form

$$\alpha\rho_v \frac{\partial U_{vz}}{\partial t} + \alpha\rho_v U_{vz}\frac{\partial U_{vz}}{\partial z} + \alpha\rho_v U_{vr}\frac{\partial U_{vz}}{\partial r} + \alpha\frac{\partial P}{\partial z} =$$

$$-f_{wz} - \alpha\rho_v g - M_{iz} + U_{iz}\Gamma - U_{vz}\Gamma \tag{2.41}$$

Next M'_{vz} is defined such that

$$M'_{vz} = -M_{iz} + U_{iz}\Gamma - U_{vz}\Gamma \tag{2.42}$$

where

$$M_{iz} = K\cdot(U_{vz} - U_{\ell z}) \qquad (2.43)$$

K = interfacial momentum exchange coefficient

M'_{vz} can be rearranged by the following procedure:

$$\begin{aligned} M'_{vz} &= -K(U_{vz} - U_{\ell z}) + [\ \eta U_{\ell z} + (1-\eta)U_{vz} - U_{vz}]\cdot\Gamma \\ &= -K(U_{vz} - U_{\ell z}) + [\ \eta(U_{\ell z} - U_{vz}) + U_{vz} - U_{vz}]\cdot\Gamma \\ &= -(K + \eta\Gamma)\cdot(U_{vz} - U_{\ell z}) \end{aligned} \qquad (2.44)$$

One can follow the same procedure for the liquid phase momentum equation to obtain the non-conservative form, which is:

$$(1-\alpha)\rho_\ell \frac{\partial U_{\ell z}}{\partial t} + (1-\alpha)\rho_\ell U_{\ell z} \frac{\partial U_{\ell z}}{\partial z} + (1-\alpha)\rho_\ell U_{\ell r} \frac{\partial U_{\ell z}}{\partial r} +$$

$$(1-\alpha)\frac{\partial P}{\partial r} = -f_{wz} - (1-\alpha)\rho_\ell g + M_{iz} - U_{iz}\Gamma + U_{\ell z}\Gamma \qquad (2.45)$$

Defining $M'_{\ell z} = M_{iz} - U_{iz}\Gamma + U_{\ell z}\Gamma$

where $M_{iz} = K\cdot(U_{vz} - U_{\ell z})$

one can simplify $M'_{\ell z}$ to obtain

$$M'_{\ell z} = (K - (1-\eta)\Gamma)\cdot(U_{vz} - U_{\ell z}) \qquad (2.46)$$

In order to better interpret these results, consider a

situation where $\eta = 0.5$ so that $U_i = (U_{\ell z} + U_{vz})/2$, and where $U_v > U_\ell$. For an evaporating condition ($\Gamma > 0$), the terms M'_{vz} and $M'_{\ell z}$ both decrease. The vapor phase bulk momentum decreases by picking up slower particles ($U_i < U_v$) and the liquid phase bulk momentum decreases by losing particles traveling at $U_i > U_\ell$.

In a condensing condition, both M'_{vz} and $M'_{\ell z}$ increase. The vapor phase bulk momentum increases by losing its slow particles and the liquid phase gains momentum by receiving fast particles.

A comparison of K and $\eta\Gamma$ verses void fraction was made in order to access the importance of this phenomenon. As can be seen in figure 2.6, for values of $\alpha > 0.88$, the term is the dominating one. This is a desired result, since as the liquid becomes entrained in the vapor phase, the slip ratio should decrease as the liquid particles become borne in the vapor phase. Parameters used for this comparison are given in Table 2.1.

To determine what effect this modification actually has on NATOF-2D simulations, a sodium boiling transient was run with the new correlation (with $\eta = 0.5$), and compared to the same transient without it. The results showed an insignificant difference for the full range of void fractions.

Simulations were also run in which η was varied in the range from 0.0 to 1.0. The only noticeable difference was

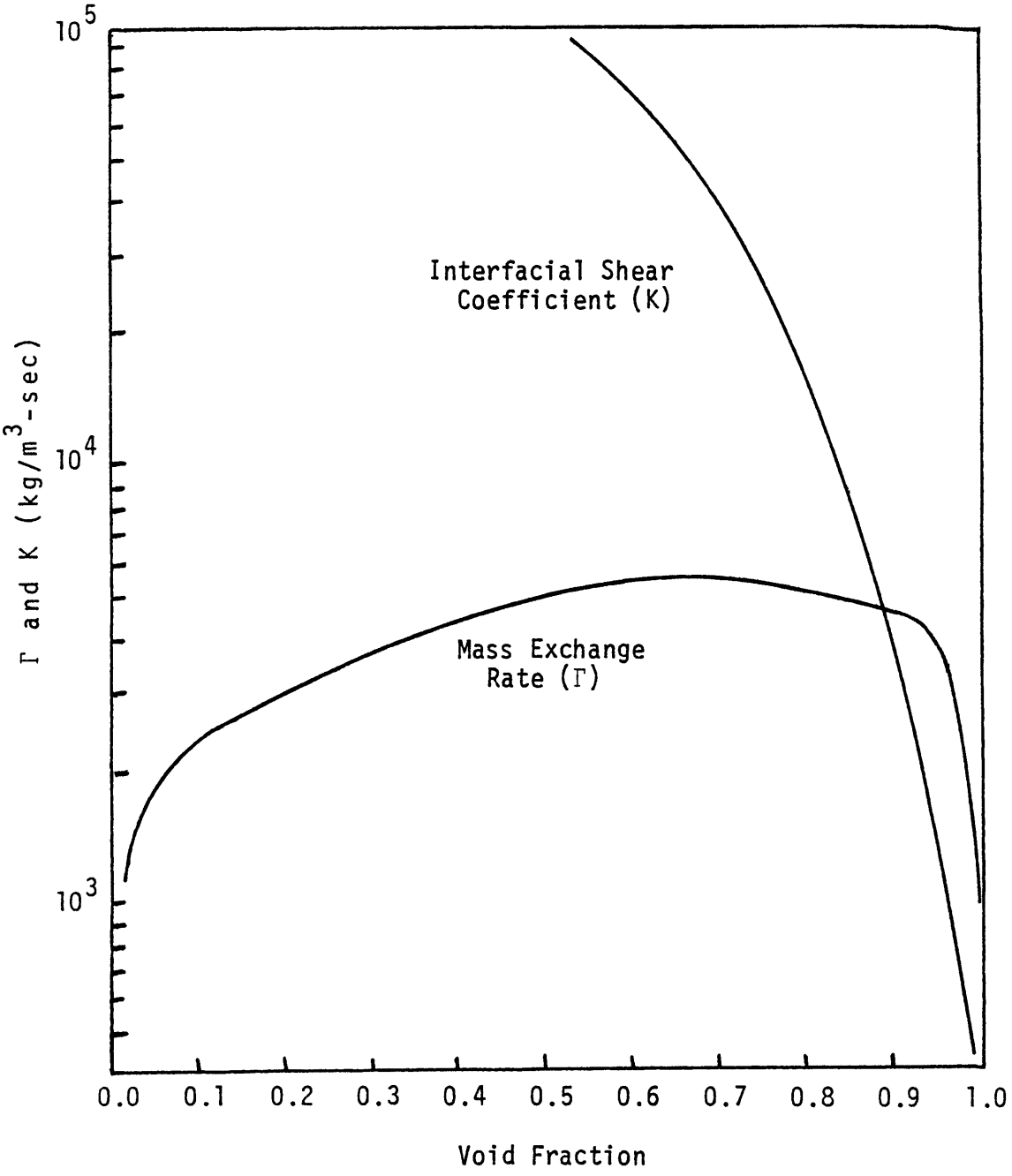


Figure 2.6 Values of Γ and K as a Function of Void Fraction for Typical LMFBR Transient Conditions

Table 2.1

Parameters Used in K versus $\eta\Gamma$ Comparison

Pressure (N/m^2)	$2. \times 10^5$
Saturation Temperature (^oK)	1235.59
Fuel Pin Diameter (m)	5.842×10^{-3}
Hydraulic Diameter (m)	4.223×10^{-3}
Pitch/Diameter	1.25
Vapor Density (kg/m^2)	0.53
Vapor Velocity (m/s)	25.0
Liquid Velocity (m/s)	5.9
$T_\ell - T_{sat}$ (^oK)	2.0

that for $\eta = 0.0$ the vapor velocity was lower than for $\eta = 1.0$, and for $\eta = 0.0$ the liquid velocity was higher than for $\eta = 1.0$. Since these results are for a region where condensation is occurring ($\Gamma < 0$), this was expected. Refering to equation 2.44, the term $(K + \eta\Gamma)$ is smallest when $\eta = 1$. Thus the vapor phase isn't slowed down by the liquid phase as much. Refering to equation 2.46, the term $(K - (1-\eta)\Gamma)$ is smallest for $\eta = 1$, and so the liquid phase is not dragged as much by the vapor phase. Hence, the lower velocity.

2.6 Programming Information

Both the new mass exchange rate and energy exchange rate were incorporated into subroutine NONEQ.

The momentum exchange rate was incorporated into subroutine WS. Since Γ is required in this formulation, and since it must be evaluated at the previous time step, the value of the mass exchange rate is stored in subroutine ONESTP for use in the following time step.

Chapter 3

FLUID CONDUCTION MODEL

3.1 Introduction

Some of the previous sodium boiling transients simulated with NATOF-2D have shown a large difference in the fluid termperature between the central channels and the edge channel. A small variation is expected since the edge channel experiences heat losses to the hexcan container, and since there is usually a lower power to flow ratio in the ouside channel. However, whereas in the W-1 SLSF experiments a radial temperature variation of 10°C was reported for steady state operation / 9 /, NATOF-2D predicted a difference of 60°C / 1 /.

In LMFBR bundles, the fuel rods are helically wound with spacer wires. These wires act as a spacing agent between fuel rods, and tend to sweep the coolant transversely around the bundle. This results in turbulence and good mixing of the coolant. NATOF-2D, as originally developed, is unable to simulate this phenomenom. The radial velocities found in NATOF-2D are due solely to the radial pressure gradient, which in most cases is rather small in magnitude. Since mass transfer between cells was the only mechanism available for energy exchange, the large temperature gradients persisted. When boiling occurs, the

previously mentioned sweeping effects become negligible compared with the expansion of the vapor phase.

Therefore to account for the observed temperature profile, radial heat conduction has been incorporated into the code. The heat transfer between cells has been modelled in terms of "effective" conduction between the fluid in adjacent cells. Besides modelling the pure conduction effects, the formulation will also be used to account for mixing and diffusive effects in the fuel bundle. Axial heat conduction has been neglected since the the high axial velocities allow the effects of convection to dominate any conductive effects. Also, the low conductivity of the vapor phase makes any vapor-liquid or vapor-vapor radial heat transfer effects negligible. This chapter will present the methodology for calculating radial heat conduction, and offer typical values for the effective nusselt number for conduction.

3.2 Formulation

For the arrangment of cells shown in figure 3.1, the total heat transfer rate to cell i can be expressed as the sum of the heat transfer rates through each of its two faces. In this formulation, the heat flux is given by an effective heat transfer coefficient times the difference in the temperature of the adjacent cells. Written explicitly, this becomes:

$$q_{iT} = q_{i-1,i} + q_{i+1,i} \tag{3.1}$$

where

$$q_{i-1,i} = A_{i-1,i} h_{i-1,i} (T_{i-1} - T_i) \tag{3.2}$$

$$q_{i+1,i} = A_{i+1,i} h_{i+1,i} (T_{i+1} - T_i) \tag{3.3}$$

and

q_{iT} = total heat transfer rate to cell i

$q_{i-1,i}$ = heat from cell i-1 to cell i

$q_{i+1,i}$ = heat from cell i+1 to cell i

$h_{i-1,i}$ = effective heat tranxfer coefficient between cell i-1 and cell i

T_i = temperature of cell i

$A_{i+1,i}$ = intercell area

On either side of the interface seperating two adjacent cells, a heat transfer coefficient has been defined with the form:

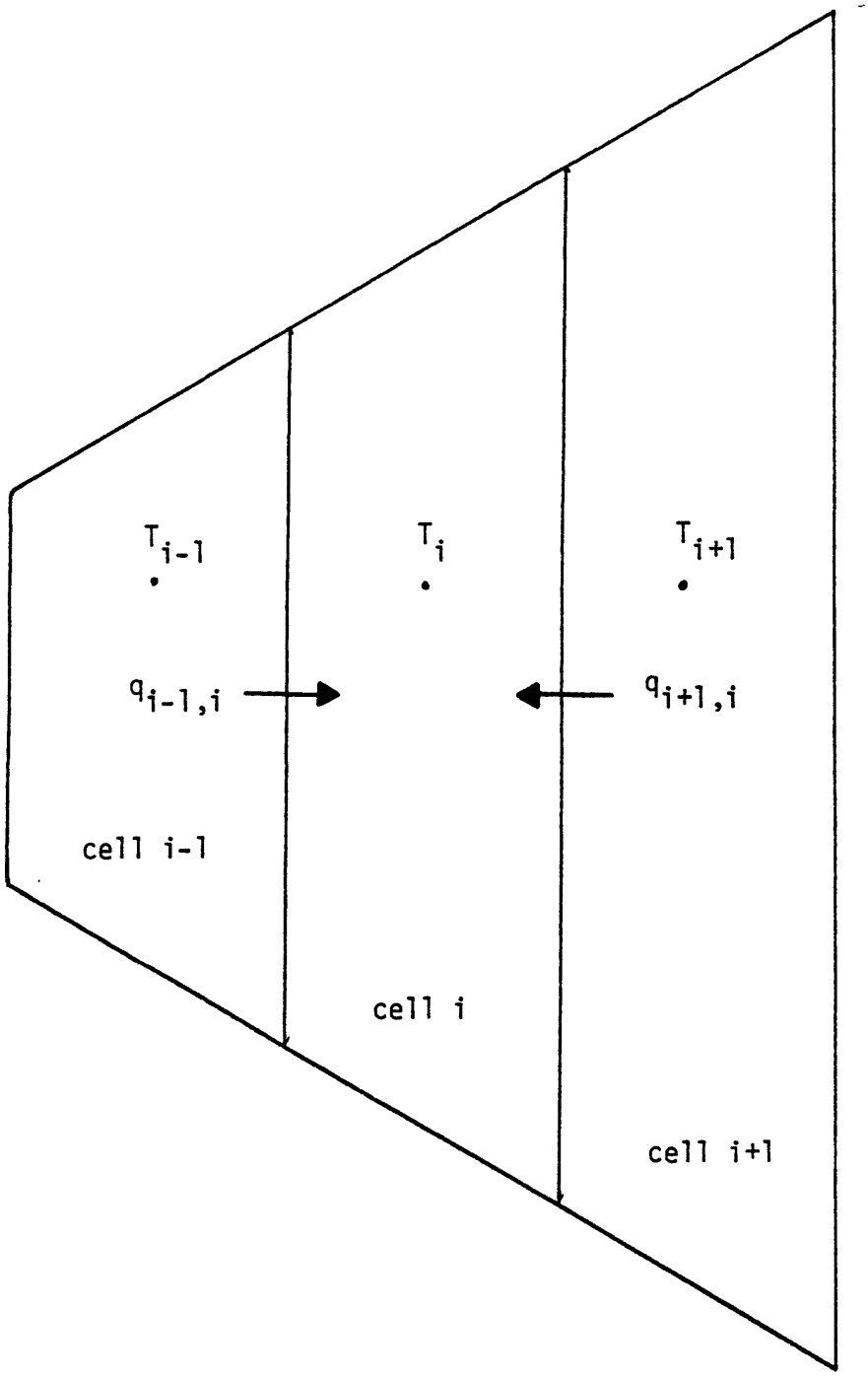


Figure 3.1 Top View of Fluid Channels Showing the Radial Heat Transfer Between Them

$$h_i = Nu \cdot \frac{K_\ell}{1/2\ D_c} \tag{3.4}$$

where

Nu = effective nusselt number

K_ℓ = conductivity of the liquid in cell i

D_c = conductive diameter of cell i

$$= \frac{4 \cdot A_{flow}}{P_c}$$

P_c = perimeter of fluid-fluid conduction

Conservation of energy requires that the heat flux from cell i to the interface of cells i and i+1 be equal and opposite to the heat flux from cell i+1 to the interface, and so an interface temperature, T_{int}, can be defined such that

$$h_i(T_{int} - T_i) = -h_{i+1}(T_{int} - T_{i+1}) \tag{3.5}$$

Solving for the interface temperature yields

$$T_{int} = \frac{h_i T_i + h_{i+1} T_{i+1}}{h_i + h_{i+1}} \tag{3.6}$$

Since the heat flux to the interface from cell i is the same as the heat flux between cells i and i+1, the right hand side of equation 3.5 can be equated to equation 3.3 to yield:

$$h_i(T_{int} - T_i) = h_{i+1,i}(T_{i+1,i} - T_i) \tag{3.7}$$

Substituting in equation 3.6 for T_{int}, $h_{i+1,i}$ can now be solved for. The result is:

$$h_{i+1,i} = \frac{h_{i+1} \cdot h_i}{h_{i+1} + h_i} \qquad (3.8)$$

Considering the case where $h_i = h_{i+1}$, equation 3.8 reduces to

$$h_{i+1,i} = 1/2 \cdot h_{i+1}$$

$$= Nu \cdot \frac{K_\ell}{D_c}$$

as one would expect.

In summary, the methodology of this approach is to calculate h as given by equation 3.4 for each cell, and then use these values to solve for $h_{i+1,i}$. Once this is accomplished, equation 3.1 can be evaluated for each cell.

3.3 Intercell Areas

NATOF-2D is structured in such a way that the boundary between cells lies at the plane connecting the fuel pin centerlines as one travels radially outward. An illustration of this, along with the numbering of the boundaries is given in figure 3.2.

Treating the bundle as a porous body, the radial heat transfer area becomes dependent only on the radial distance r_n. Thus

$$Ar = Ar^* \cdot r_n \tag{3.9}$$

where

- Ar = volumetrically averaged radial area between cell boundaries
- Ar^* = radial area constant
- r_n = $\sqrt{3}/2 \cdot n \cdot p$
- n = row number (1,2,3,.....)
- p = fuel rod pitch (m)

Considering for the moment the unit cell shown in figure 3.3, Ar^* can be solved for by requiring that

$$\begin{aligned} V_{cell} &= \int_0^{r_1} Ar \cdot dr \\ &= \int_0^{r_1} Ar^* \cdot r_n dr \\ &= Ar^* \cdot r_1^2/2 \end{aligned} \tag{3.10}$$

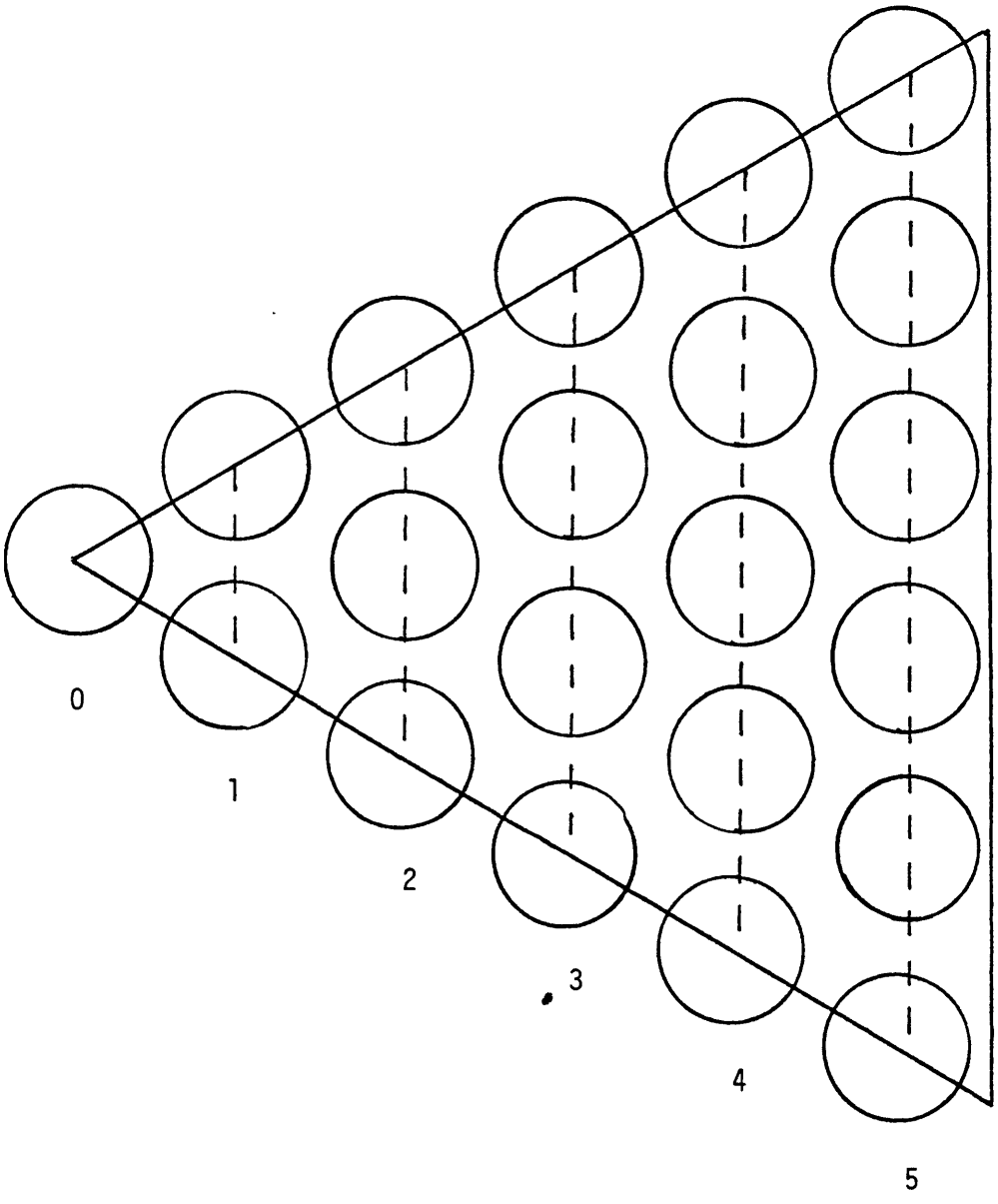


Figure 3.2 Top View of Fluid Channels Showing Radial Cell Boundary Numbering Scheme

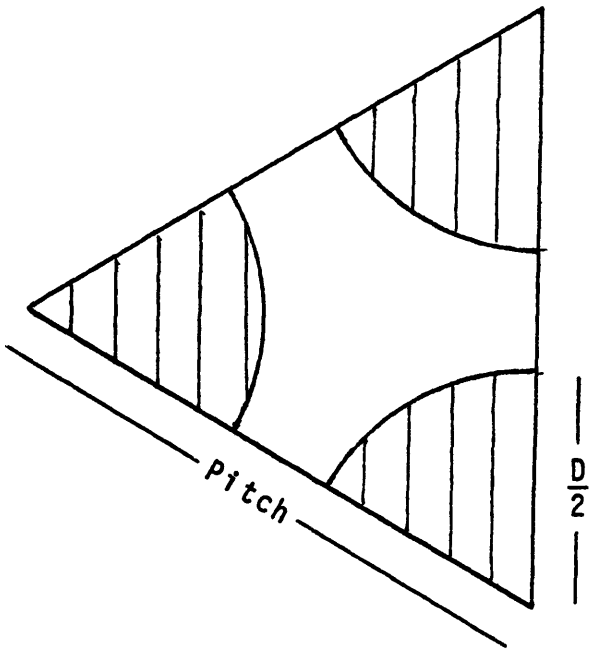


Figure 3.3 Unit Cell Used in NATOF-2D

The volume of the unit cell is given by

$$V_{cell} = \frac{1}{2}\left(P^2 \cdot \frac{\sqrt{3}}{2} - A_{pin} \right)\Delta z \qquad (3.11)$$

where

A_{pin} = area of the pin and wirewrap
$= \frac{\pi}{4}(D^2 + d^2)$

D = fuel pin diameter

d = wire wrap diameter

Δz = axial height of cell

Using the relation in equation 3.10 yields:

$$Ar^* = \frac{4}{3}\left(\frac{\sqrt{3}}{2} - \frac{A_{pin}}{P^2}\right)\Delta z \qquad (3.12)$$

The final form of the radial heat transfer area is then

$$Ar = \left(1 - \frac{2}{\sqrt{3}} \cdot \frac{A_{pin}}{P^2} \right) nP\Delta z \qquad (n = 1,2,\ldots) \qquad (3.13)$$

When this formulation is implimented in NATOF-2D, it is necessary to divide the total heat transferred by the volume of the cell so that the term will appear as a heat source term in the energy conservation equation. For a cell whose boundaries lie at n_{j-1} and n_j, excluding the edge cell, the total volume is given by

$$V = \left(P^2 \cdot \frac{\sqrt{3}}{2} - A_{pin} \right) \cdot \frac{n_j^2 - n_{j-1}^2}{2} \qquad (3.14)$$

3.4 Implementation Form

Up until this point, no mention has been made of the time step discretinization used in the code for this formulation. In this section, the options available in NATOF-2D and the advantages and limitations of each are covered.

The first option is to treat the calculation in a fully explicit manner such that

$$q_{iT}^{n+1} = A_{i-1,i}\,h_{i-1,i}^{n}(T_{i-1}^{n}-T_i^{n}) + A_{i+1,i}\,h_{i+1,i}^{n}(T_{i+1,i}^{n}-T_i^{n}) \tag{3.15}$$

where the superscript (n+1) refers to the present time step, and (n) refers to the previous time step. Since all terms on the RHS of equation 3.15 are known values, this option requires that the calculation be performed only once per time step. Thus the cpu costs for the explicit calculation are low. It also ensures strict energy conservation since

$$q_{i-1,i} = -q_{i,i-1}.$$

A fourier stability analysis performed on equation 3.15 shows that this formulation limits the time step size to

$$\Delta t \leq \frac{\Delta r^2}{2\cdot\alpha\cdot Nu} \tag{3.16}$$

where

$$\alpha = \frac{k}{\rho c_p}$$

In most cases, the convective time limit ($\Delta t \leq \Delta z/U_z$) is more restrictive than the conductive limit. However, a

feature has been implimented into the code which calculates the time step limitation when the explicit calculation is utilized, and maintains a time step value below the conductive limit.

The second option available is to treat the radial heat transfer calculation semi-implicitly. The form of the calculation is:

$$q_{iT}^{n+1} = A_{i-1,i}\, h_{i-1,i}^{n}\,(T_{i-1}^{n} - T_i^{n+1}) + A_{i+1,i}\, h_{i+1,i}^{n}\,(T_{i+1}^{n} - T^{n+1}) \tag{3.17}$$

A stability analysis applied to equation 3.17 shows that the scheme is unconditionally stable, and therefore poses no constraint to the time step size. However, it does have two limitations. The first is that it fails to conserve energy since the relation

$$h_{i-1,i}^{n}\,(T_{i-1}^{n} - T_i^{n+1}) = -h_{i,i-1}^{n}\,(T_i^{n} - T_{i-1}^{n+1}) \tag{3.18}$$

will not be satisfied in general. The second limitation is that this calculation needs to be performed once per newton iteration, instead of once per time step. Thus, the cpu usage will be greater than the explicit method.

A fully implicit calculation of the form

$$q_{iT}^{n+1} = A_{i-1,i}\, h_{i-1,i}^{n}\,(T_{i-1}^{n+1} - T_i^{n+1}) + A_{i+1,i}\, h_{i+1,i}^{n}\,(T_{i+1}^{n+1} - T_i^{n+1}) \tag{3.19}$$

cannot be utilized by NATOF-2D since the solution scheme of

the code requires that the only linkage between cells be by the pressures of the cells. A formulation such of this would also link the cell temperatures.

As can be seen in figure 3.2, all interior cells are similar, and therefore we are somewhat justified in using the same effective nusselt number Nu_1. The edge channel, however, has a quite different shape, and so a second nusselt number, Nu_2, is used to take into account the effects of any differences.

3.5 Experimental Calibration

In order to obtain a practical value for the effective nusselt number for radial heat conduction, a steady-state, single phase sodium experiment was chosen from the Westinghouse Blanket Heat Transfer Test Program / 2 /. The heat transfer test section was a mockup of an LMFBR blanket assembly. The test section consisted of 61 rods contained in a hexagonal duct. Each rod delivered an axial heat output approximating a chopped cosine distribution with a 1.4 maximum-to-average ratio over a 114.3 cm. length. In test No. 544, the total bundle power was 440 kw, and the radial power distribution was uniform. Test parameters are given in Table 3.1, and the input for the NATOF-2D simulation is given in appendix C.

The test procedure was to adjust the test loop operating parameters until the desired sodium flow and inlet temperature was achieved. At this point, power to the bundle was gradually increased until the test section power gradient and temperature rise attained operating conditions. The test section was then allowed to achieve a steady state configuration, at which point data was collected. For test No. 544, the temperature profile across the bundle at three different axial levels was recorded. These levels corresponded to the heated zone midplane, the outlet of the heated zone, and 25 inches downstream of the heated zone.

In the NATOF-2D simulation, the proper flow and total

Table 3.1

Westinghouse Blanket Heat Transfer Test Program

Rod Bundle Test Section Design / 2 /

PARAMETER	
Number of Rods	61
Rod Diameter (cm)	1.32
Length of Heated Zone (cm)	114.3
Total Bundle Length (cm)	265.
Wire Wrap Spacer Diameter (cm)	.094
Triangular Rod Pitch	1.43
Wire Wrap Pitch (cm)	10.16
Pitch to Diameter Ratio	1.082
Duct Inside Diameter (cm)	11.4
Axial Power Distribution, Cosine	
Max/Avg	1.40
Sodium Inlet Temperature (^{o}C)	316.
Sodium Flow Rate (m^2/hr)	
Run 544	13.5
Run 545	12.0
Test Bundle Power (kw)	440

enthalpy rise through the core was established. Then the nusselt numbers were varied until the temperature profile obtained matched as closely as possible the experimental results. As the effective nusselt number was increased, the radial temperature profile at the end of the heated section became flatter, as shown in figure 3.4. A comparison between the experimental results and the NATOF-2D simulation for different elevations is given in figures 3.5, 3.6 and 3.7.

From this experiment, the recommended values for the effective nusselt number are

$$Nu_1 = 22$$

$$Nu_2 = 28$$

A second experiment from the same series of tests was simulated by NATOF-2D in order to verify the generality of the previous results. This was Run No. 545. In this test, the same total power was used as before, but the radial power distribution was varied to give a power skew which peaked at the edge pins and was at a minimum at the center. The normalized heat input per rod is shown in figure 3.8. The results for three different elevations are shown in figures 3.9, 3.10, and 3.11.

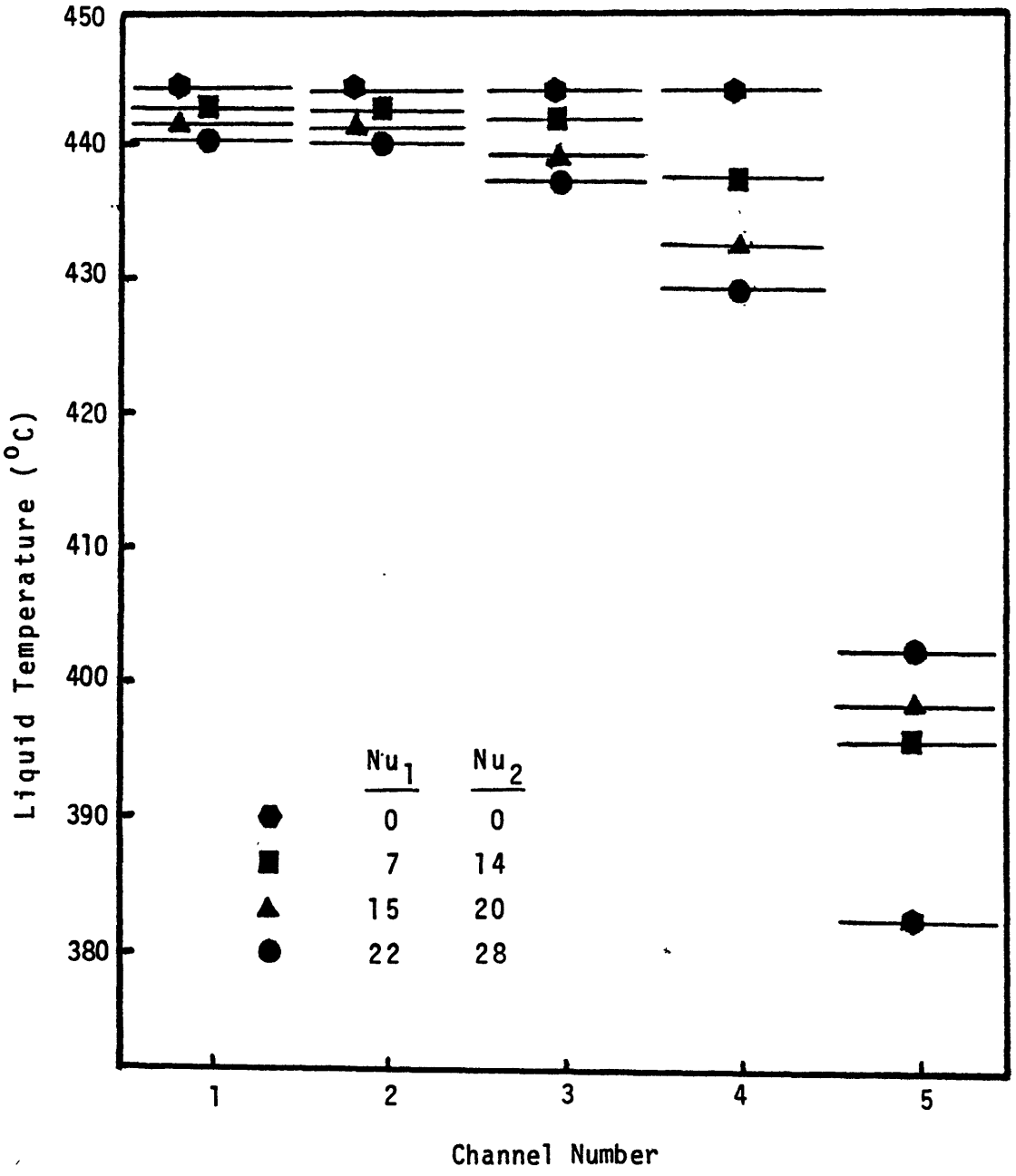


Figure 3.4 Radial Temperature Profiles at the End of the Heated Section for Various Effective Nusselt Numbers

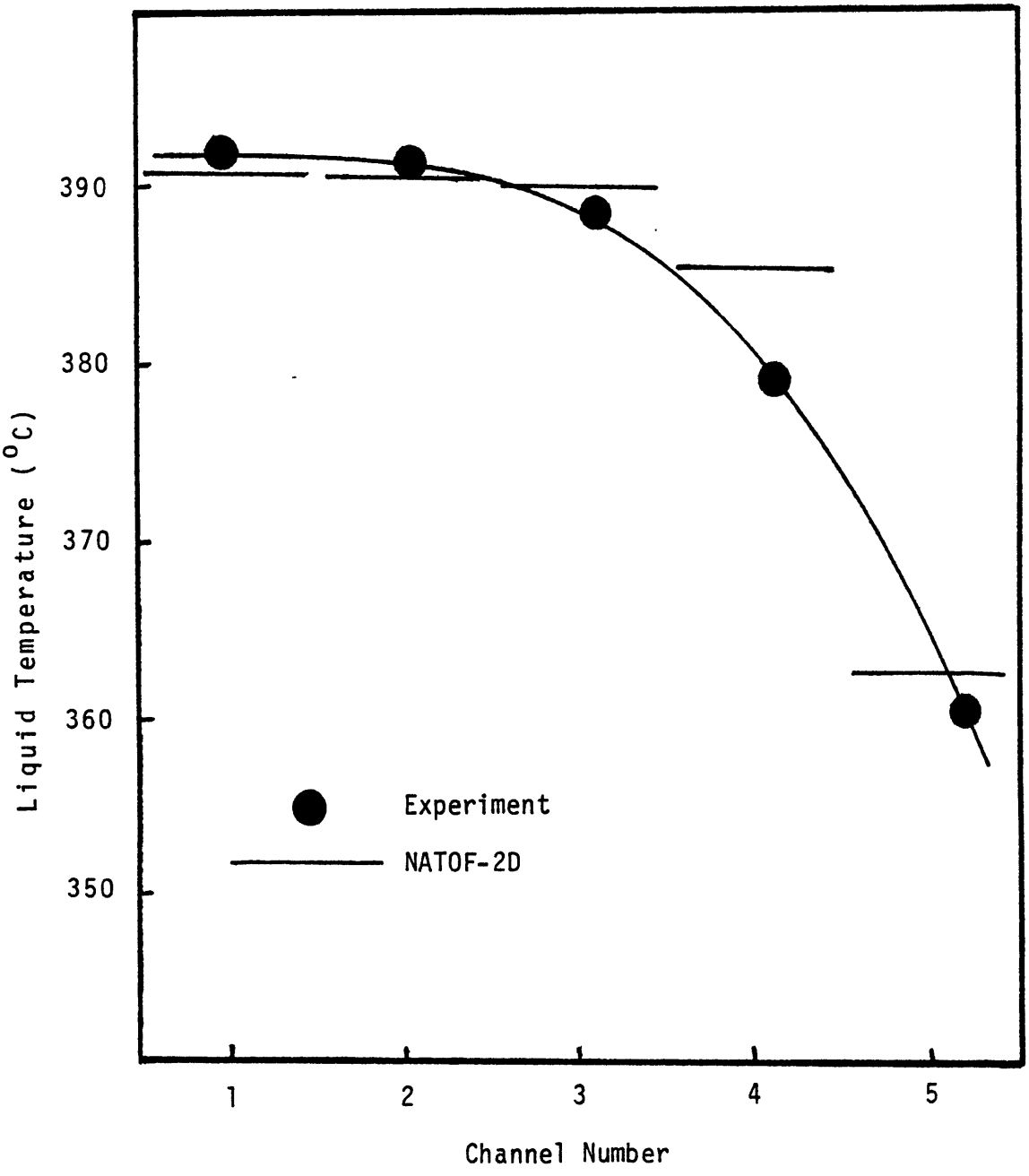


Figure 3.5 A Comparison Between Westinghouse Run 544 and NATOF-2D Radial Temperature Profiles at the Heated Zone Midplane for Nu_1 = 22 and Nu_2 = 28 / 2 /

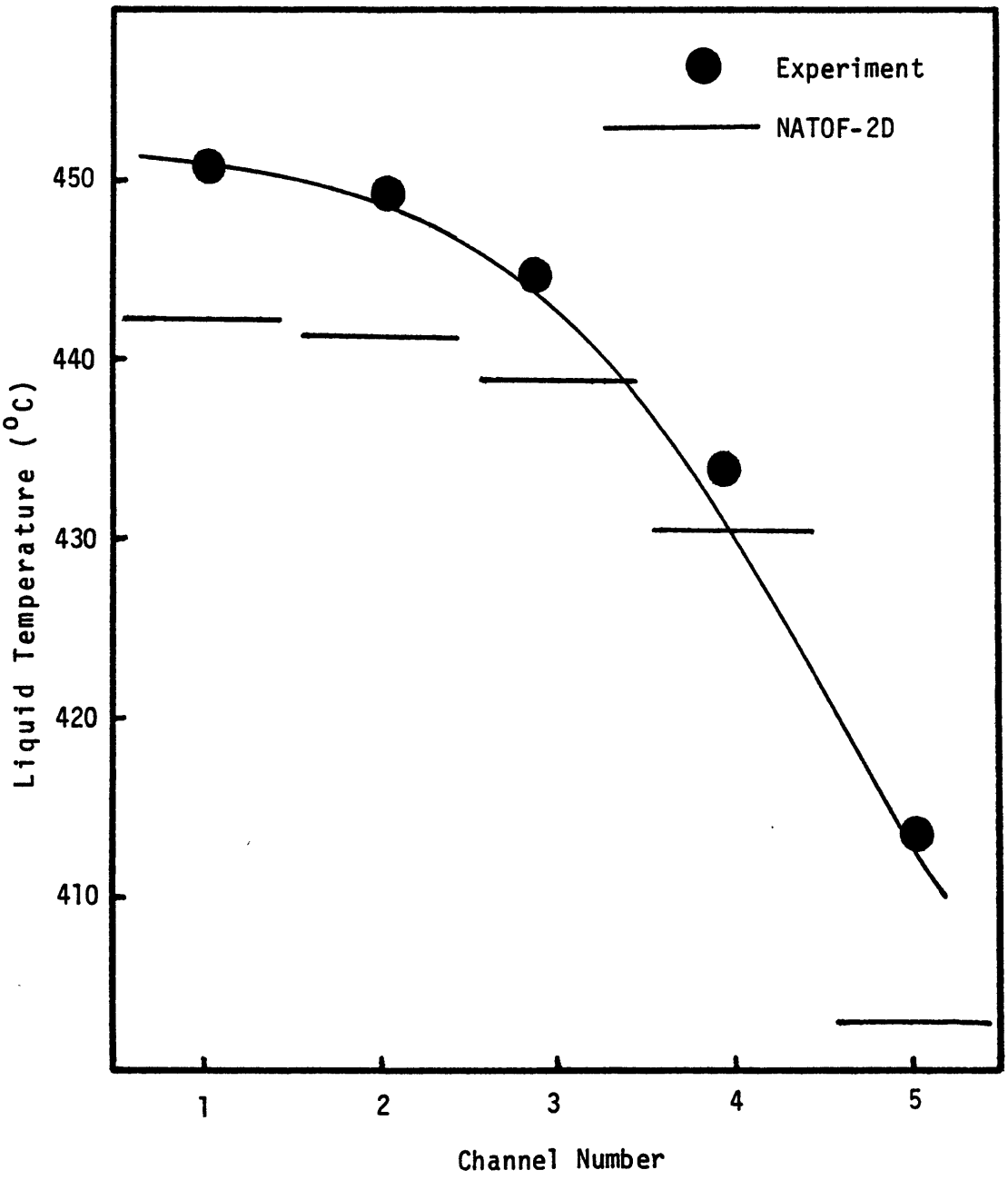


Figure 3.6 A comparison Between Westinghouse Run 544 and NATOF-2D Radial Temperature Profiles at the End of the Heated Zone for Nu_1 = 22 and Nu_2 = 28 / 2 /

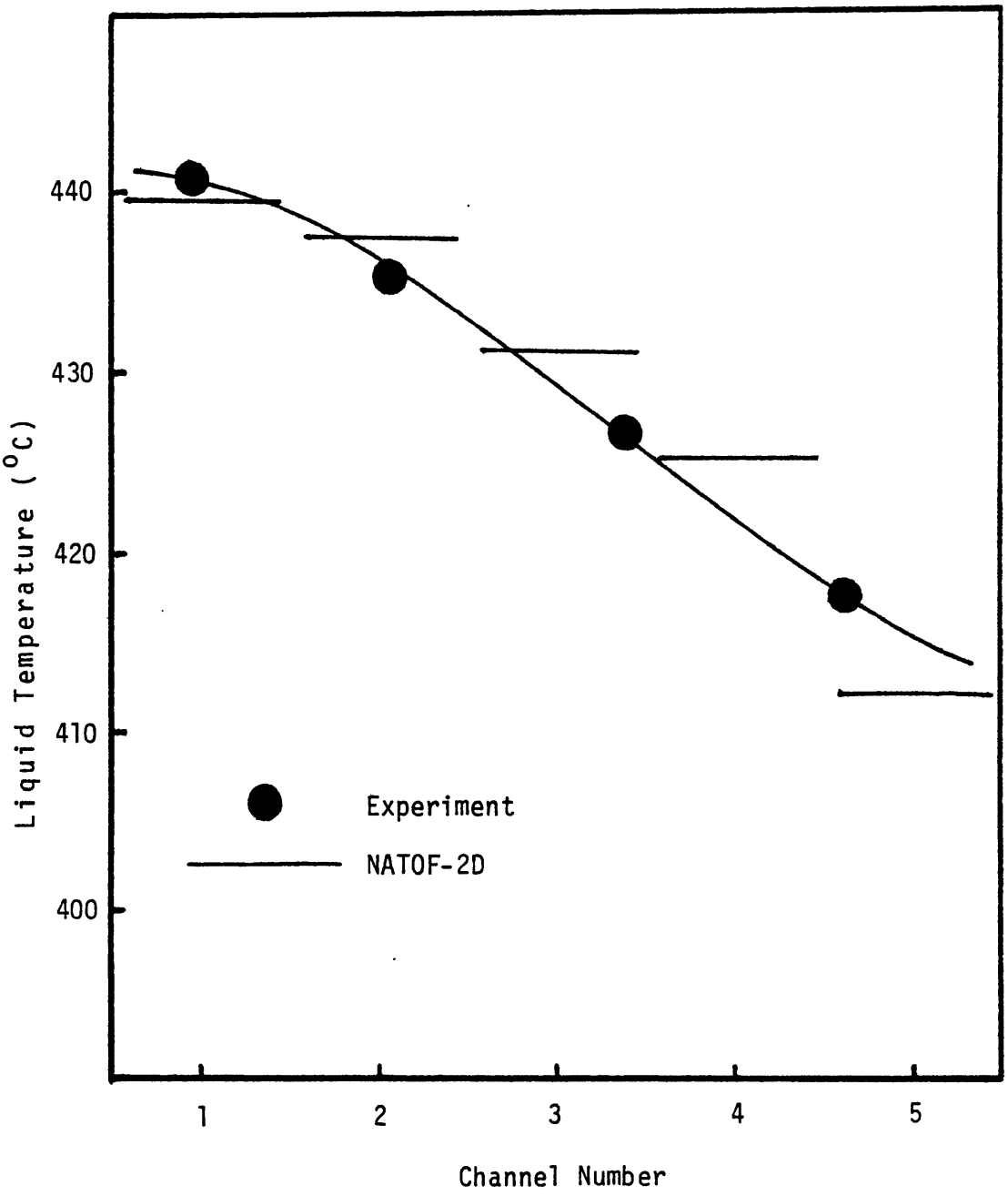


Figure 3.7 A Comparison Between Westinghouse Run 544 and NATOF-2D Radial Temperature Profiles 25 Inches Downstream of Heated Zone for Nu_1 = 22 and Nu_2 = 28 / 2 /

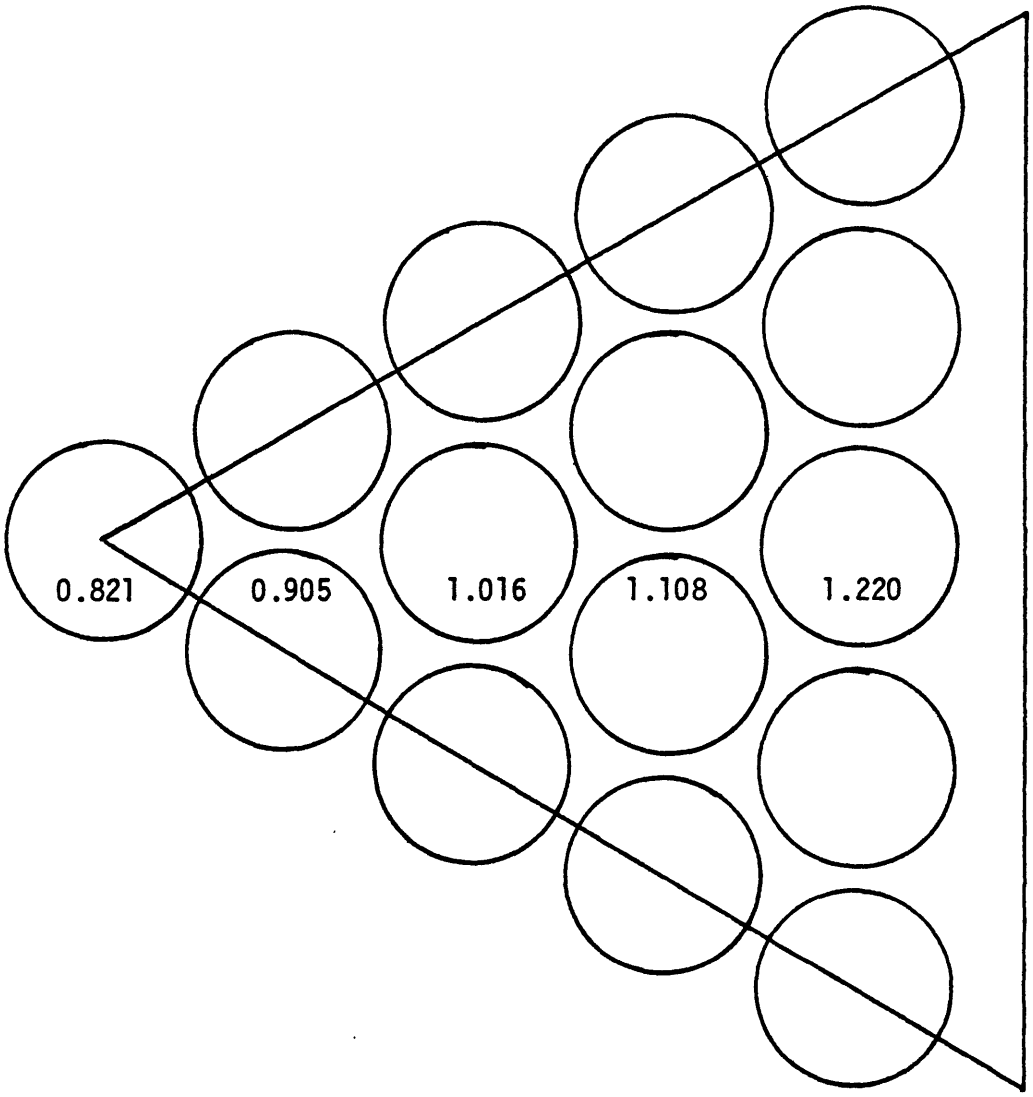


Figure 3.8 Normalized Heat Input per Rod for Westinghouse Blanket Heat Transfer Test Program Run 545 / 2 /

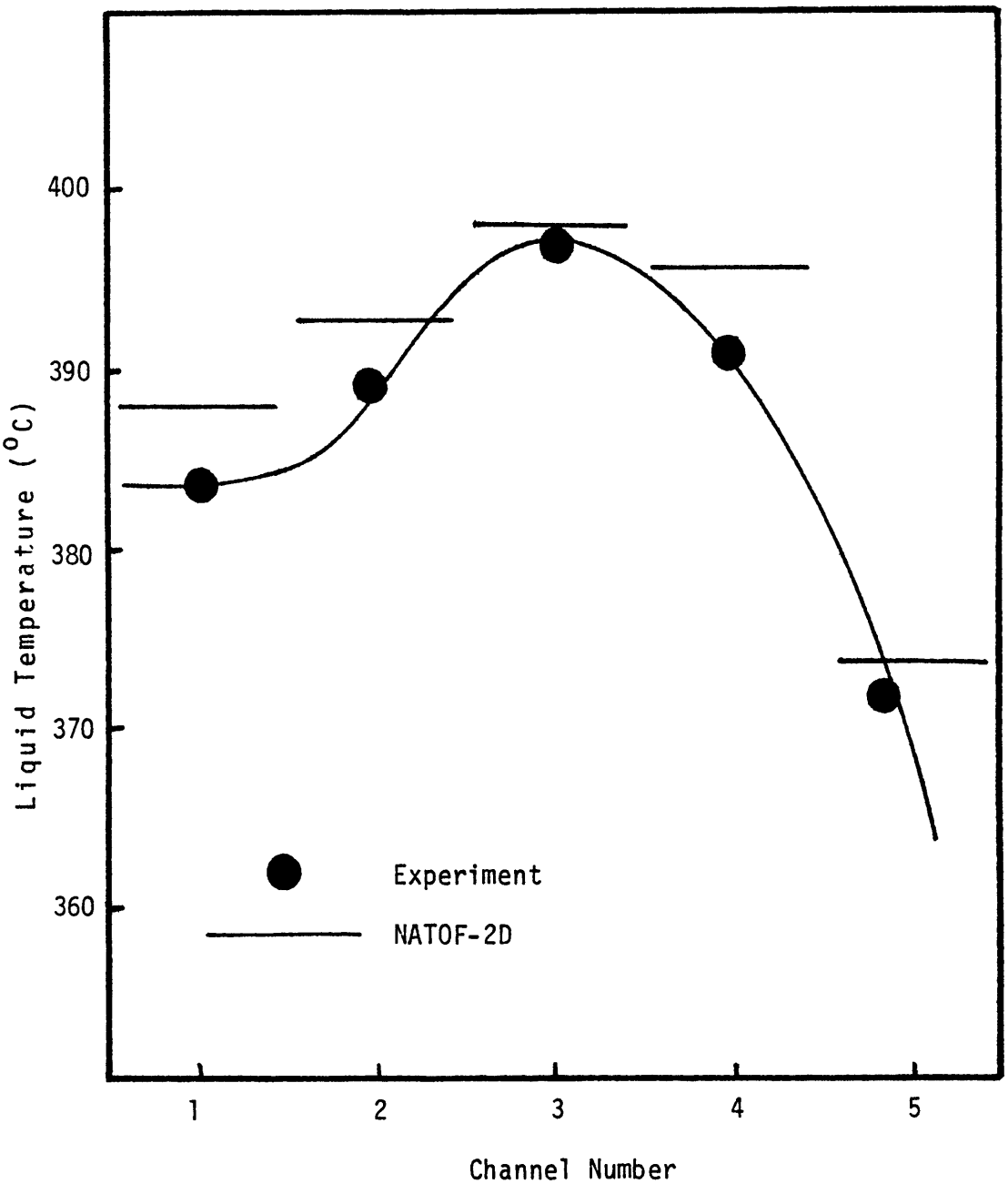


Figure 3.9 A Comparison Between Westinghouse Run 545 nad NATOF-2D Radial Temperature Profiles at the Heated Zone Midplane for Nu_1 = 22 and Nu_2 = 28 / 2 /

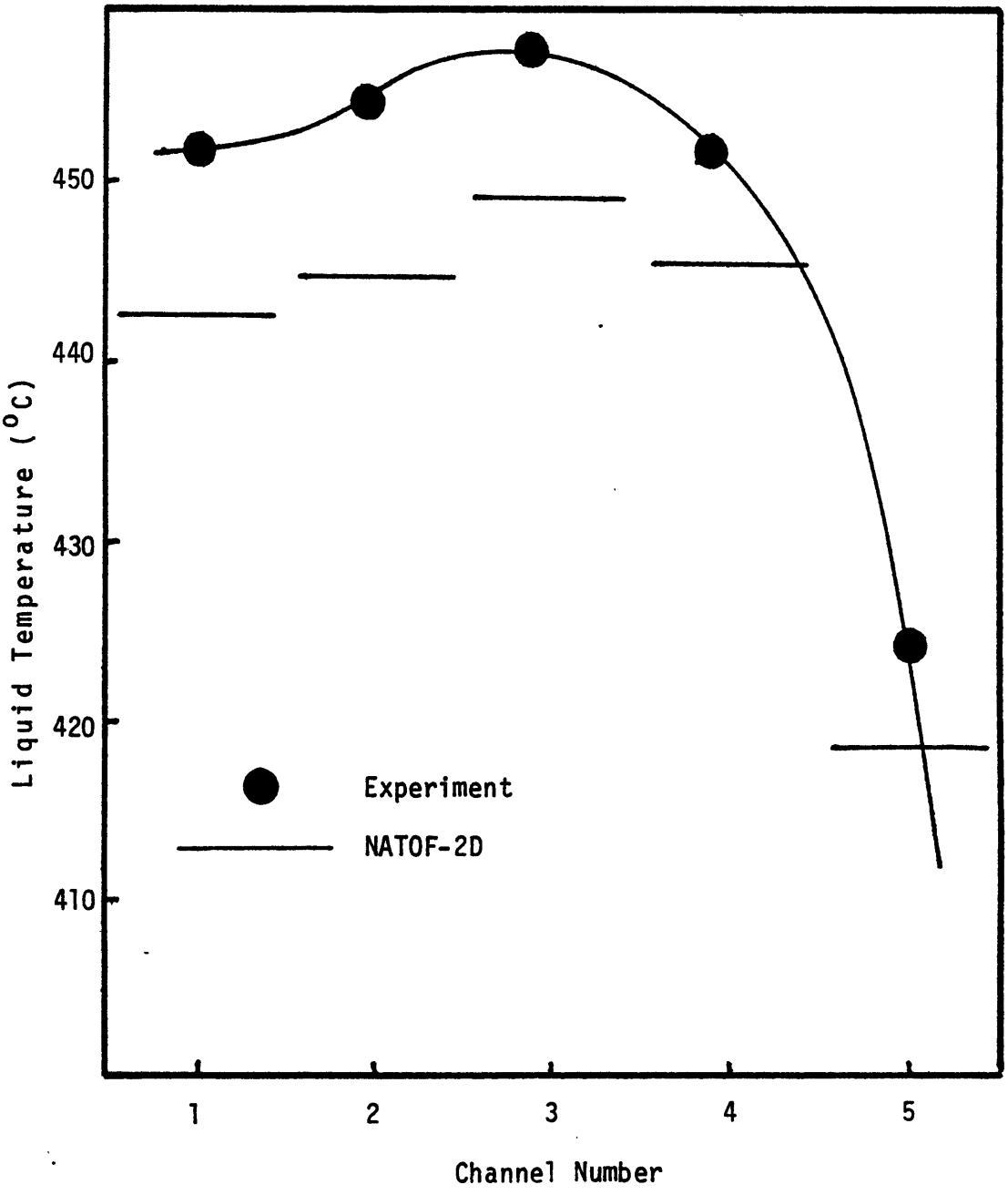


Figure 3.10 A Comparison Between Westinghouse Run 545 and NATOF-2D Radial Temperature Profiles at the End of the Heated Zone for Nu_1 = 22 and Nu_2 = 28 / 2 /

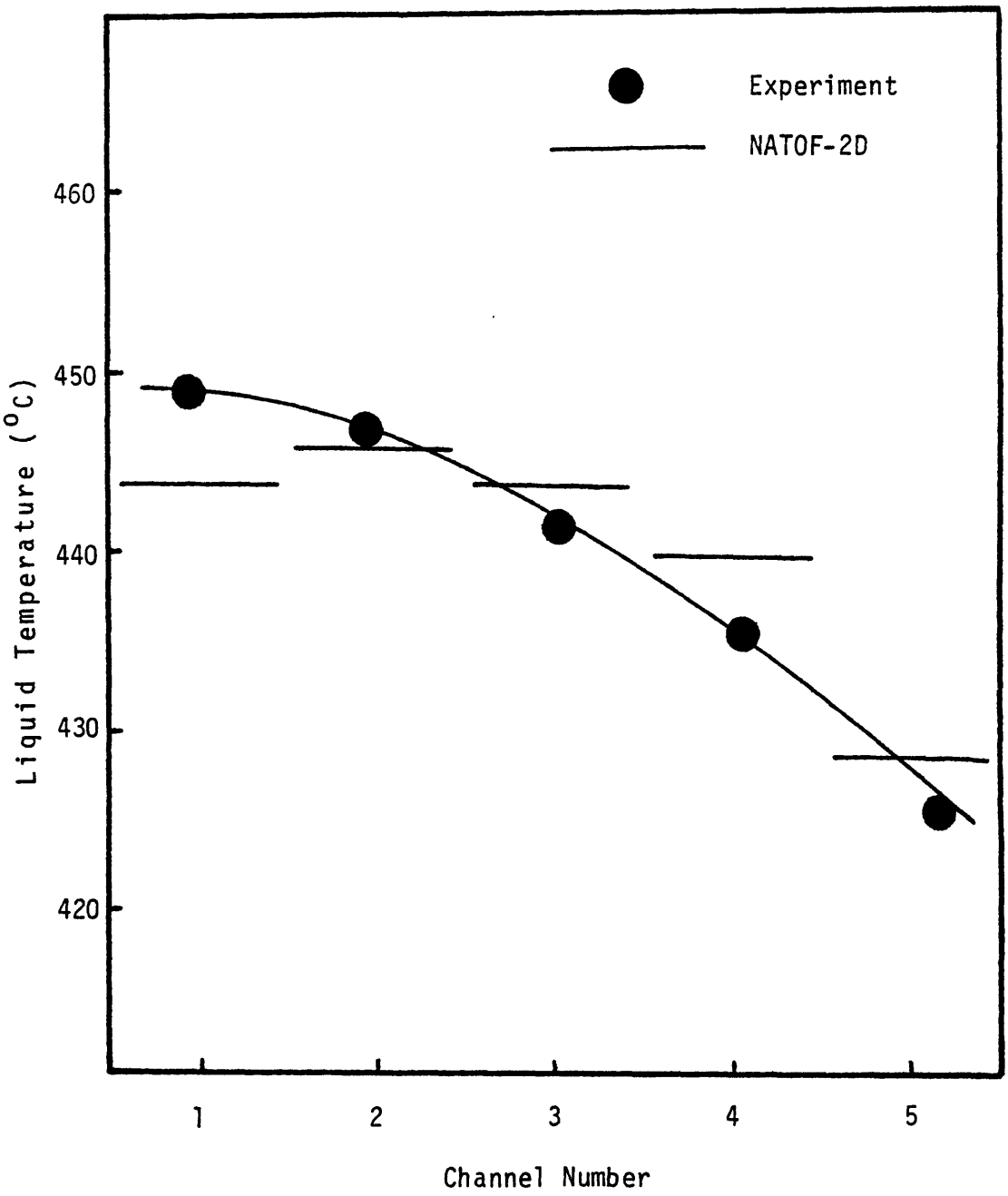


Figure 3.11 A Comparison Between Westinghouse Run 545 and NATOF-2D Radial Temperature Profiles 25 Inches Downstream of the Heated Zone for Nu_1 = 22 and Nu_2 = 28 / 2 /

3.6 A Comparison with Effective Conduction Mixing Lengths

In this section, a comparison is made between the fluid-to-fluid conduction model implemented in NATOF-2D, and an analytic model developed to determine the effective mixing lengths for energy transport by conduction in subcahnnel codes. The model used for the evaluation of mixing lengths is from M. R. Yeung / 3 /.

Before making the comparison, a brief outline is given of the method of M. R. Yeung to calculate the effective conduction mixing lengths. In this model, the heat transfer rate due to conduction between subcahnnels i and j is given by the relation:

$$Q_{ij} = K_\ell \frac{S_{ij}(\bar{T}_i - \bar{T}_j)}{\ell_{ij}} = K_\ell \frac{S_{ij}}{\ell^*_{ij}} (\bar{T}_i - \bar{T}_j)\frac{1}{L_{ij}} \qquad (3.20)$$

where

S_{ij} = the length of the common boundary

ℓ_{ij} = the effective conduction mixing length

ℓ^*_{ij} = the centroid-to-centroid distance of adjacent coolant channels

K_ℓ = conductivity of the liquid phase

L_{ij} = ratio of the effective conductive mixing length to the centroid-to-centroid distance

$= \ell_{ij}/\ell^*_{ij}$

The effective conduction mixing length takes into account

the fact that the actual heat flux due to conduction,

$$q'' = -K_\ell \cdot \frac{\partial T}{\partial x}\Big)_{interface} \tag{3.21}$$

may be quite different than that given by equation 3.20 since subchannel codes deal with bulk temperatures while attempting to model a localized effect. As an illustration of this, possible temperature distributions which yield the same bulk temperature are shown in figure 3.12. As can be seen, $\partial T/\partial x)_{int}$ can vary widely.

Equation 3.20 can be rearranged to give:

$$L_{ij} = K_\ell \cdot \frac{S_{ij}}{\ell^*_{ij}} \cdot \frac{(\bar{T}_i - \bar{T}_j)}{Q_{ij}} \tag{3.22}$$

Since the total heat transfer can be expressed as

$$Q_{ij} = \int_{S_{ij}} q''_s \, ds \tag{3.23}$$

equation 3.23 can be substituted into equation 3.20, and also the dimensionless group $q'''a^2/2k$ and the rod radius b can multiply and divide 3.20 to yield a form which can be analytically determined by evaluating each quantity. This form is:

$$L_{ij} = \frac{S_{ij}}{\ell^*_{ij}} \cdot \frac{\dfrac{(\bar{T}_i - \bar{T}_j)}{q'''a^2/2K_\ell}}{\displaystyle\int_s \frac{q''_s}{q'''a^2/2b} \, d\left(\frac{s}{b}\right)} \tag{3.24}$$

where

a = fuel pellet radius

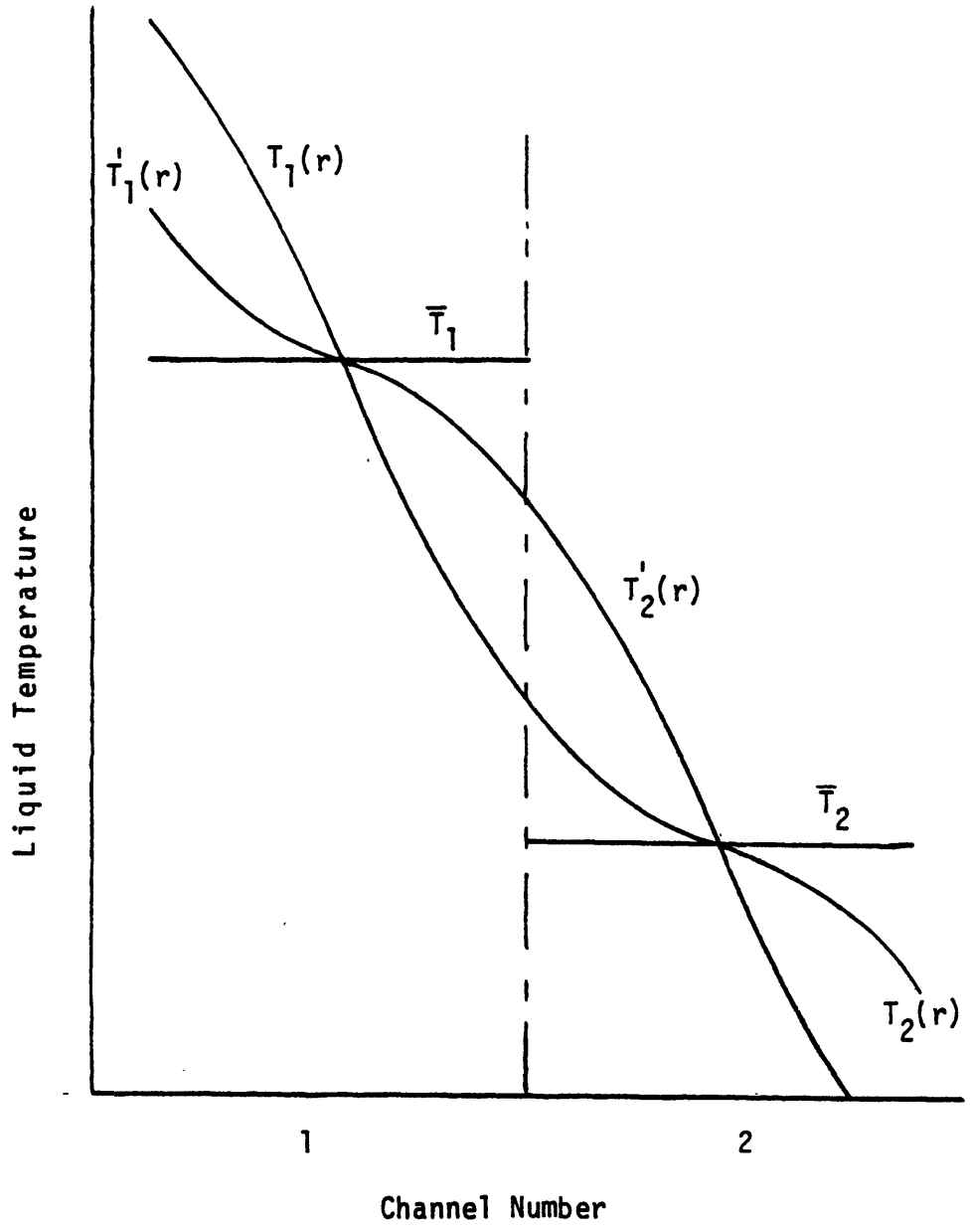


Figure 3.12 Possible Temperature Distributions which Yield the Same Cell Averaged Temperature

b = fuel rod radius

q''_s = heat flux at common boundary

q''' = power density of the fuel

The method of evaluating each of these terms from the local temperature and power distributions and the geometry is given in reference / 3 /. For the purposes of this comparison, it will suffice to give the results for a 19-pin hexagonal bundle. The geometry used for calculating the effective mixing lengths is shown in figure 3.13, where the dashed lines denote cell boundaries. The results of the calculation are give in Table 3.2.

To compare the NATOF-2D formulation with the conductive mixing length results, the heat transfer of both formulations are equated such that

$$Ar \cdot h_{ij}(T_i - T_j) = K_\ell \cdot \frac{2 \cdot S_{ij} \Delta z}{\ell^*_{ij}} (\overline{T}_i - \overline{T}_j) \frac{1}{L_{ij}} \qquad (3.25)$$

where the factor 2 has been added to the RHS of equation 3.25 to take into account the fact that NATOF-2D divides the core into 6 symmetrical volumes, while Yeung's work divides the core into 12. Thus the area used in NATOF-2D is twice as large.

Assuming that k_ℓ is the same for both formulations, h_{ij} is then given by

$$h_{ij} = \frac{2 \cdot Nu \cdot K_\ell}{D_{c1} + D_{c2}} \qquad (3.26)$$

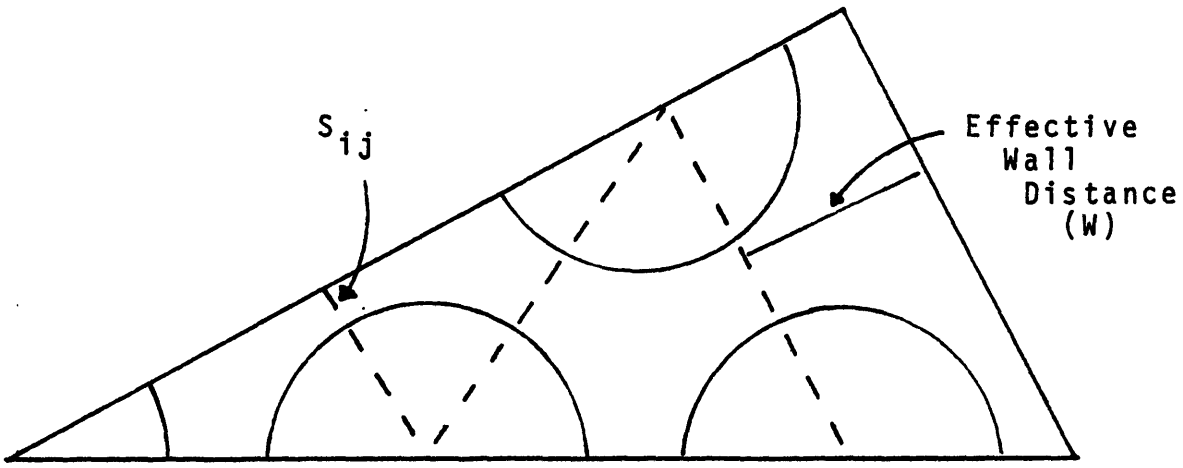


Figure 3.13 19-pin Cell Geometry Used for Calculation of Effective Mixing Lengths

Table 3.2

$$S_{12}/\ell^*_{12} = \frac{1}{2}\left(\frac{P}{D} - 1\right) \frac{1}{\sqrt{3}}\frac{P}{D}$$

$$S_{23}/\ell^*_{23} = \left(\frac{P}{D} - 1\right) \frac{1}{\sqrt{3}}\frac{P}{D}$$

$$S_{34}/\ell^*_{34} = \frac{(P/D - 1)}{\frac{1}{2\sqrt{3}}\frac{P}{D} + \frac{1}{8}\left(\frac{2W}{D}\right)^2 \cdot \frac{P}{D} + \frac{1}{12\sqrt{3}}\left(\frac{2W}{D}\right)^3 - \frac{1}{16}}{\frac{1}{2}\cdot\frac{P}{D}\cdot\frac{2W}{D} + \frac{1}{4\sqrt{3}}\cdot\frac{2W}{D} - \frac{\pi}{12}}$$

Effective Mixing Lengths

2W/D	L_{12}	L_{23}	L_{34}
Fuel Bundle			
1.10	.85	.86	1.12
1.20	.75	.27	.57
1.30	.79	.77	.79
1.40	.791	.79	.81
Blanket Bundle			
1.04	.69	.69	.645
1.08	.69	.69	.68
1.10	.69	.69	.69
1.20	.69	.69	.70

Rearranging equation 3.25 in dimensionless form, and substituting in 3.26 yields:

$$Nu = \left(\frac{D_{c1} + D_{c2}}{\ell^*_{ij}}\right)\left(\frac{S_{ij}}{Ar}\right)\frac{1}{L_{ij}} \qquad (3.27)$$

As can be seen, both formulations are of the same form, and differ only by a constant multiple.

First, considering cells 1-2, each term of equation of 3.27 can be evaluated to get:

$$S_{12}/\ell^*_{12} = \frac{1}{2}\left(\frac{P}{D} - 1\right) \Big/ \frac{1}{\sqrt{3}}\frac{P}{D}$$

$$Ar = \left(1 - \frac{2}{\sqrt{3}}\frac{\pi}{4}\left(\frac{D}{P}\right)^2\right)\cdot P\cdot\Delta z$$

$$D_{c1} + D_{c2} = 4\left(1 - \frac{2}{\sqrt{3}}\frac{\pi}{4}\left(\frac{D}{P}\right)^2\right)\frac{\sqrt{3}}{2}P^2/(P - D)$$

Substituting these values into equation 3.27 yields

$$Nu = \frac{3}{L_{12}} \qquad (3.28)$$

From Table 3.2, L_{12} has a value equal to 0.69. For a blanket bundle of the type simulated, this relation shows that the Nusselt number, due to conduction only, should be:

$$Nu = 4.348$$

For the edge cell, the complex geometry requires that each term be numerically evaluated. For this comparison, typical dimensions of a blanket assembly were used:

$$D = 1.320 \times 10^{-2} \text{ meters}$$

$$P = 1.426 \times 10^{-2} \text{ meters}$$

$$P/D = 1.08$$

$$W = P - D/2 = 7.66 \times 10^{-3} \text{ meters}$$

The result is that

$$Nu = 4.6322 \frac{1}{L_{34}}$$

$$= 6.617$$

The results show that the effective nusselt number calculated from mixing length theory is much smaller than that required for the experiment calibration. This is to be expected for two reasons. The first is that the mixing length theory only takes into account fluid-to-fluid conduction effects, while the effective nusselt number is also accounting for turbulence and mixing. The second reason is that the mixing length results are specifically for 19 pin bundles, while the simulations were conducted for 61 pin bundles. It is expected that for smaller bundle sizes the effective nusselt numbers will also decrease.

3.7 Programming Information

Two additional subroutines have been added to NATOF-2D. The first is subroutine QCOND which calculates the heat transferred per unit volume, and its derivative (when the implicit formulation is required). The second is subroutine HTRAN, which calculates the effective heat transfer coefficient.

The user specifies the type of calculation to be performed by specifying the sign of the nusselt numbers, which are a user input. A negative nusselt number refers to a semi-implicit calculation, while a positive nusselt number refers to a fully explicit calculation.

The nusselt numbers given in this chapter should be used as a gauge for the ones actually used, which can best be determined by calibration to steady state results of the experiment being simulated.

Chapter 4

DIRECT SOLUTION OF THE PRESSURE FIELD

4.1 Introduction

In the solution scheme employed by NATOF-2D, the eight conservation equations, the equation of state, and the equations governing the exchange terms are reduced to a single equation for each cell which involves only the pressure of a cell and its (up to four) neighbors. The form of the equation is:

$$a_{ij}P_{ij-1} + b_{ij}P_{i-1j} + c_{ij}P_{ij} + d_{ij}P_{i+1j} + e_{ij}P_{ij+1} = f_{ij} \qquad (4.1)$$

As can be seen, the pressure of a cell is influenced only by the pressure of the cells directly in contact with it. When written out in matrix form, this large system of equations is a five-stripe band matrix, i.e. a matrix whose non-zero components are near the diagonal and contained in five bands. For example, the resulting matrix for the solution of a problem with four axial levels and three radial nodes (figure 4.1) is shown in figure 4.2.

Previously, NATOF-2D used an iterative solution technique known as block-tri-diagonal, which is an extension of the Gauss-Siedel iterative technique. Like all iterative methods, this scheme started from an initial approximation and proceeded to calculate a sequence of further

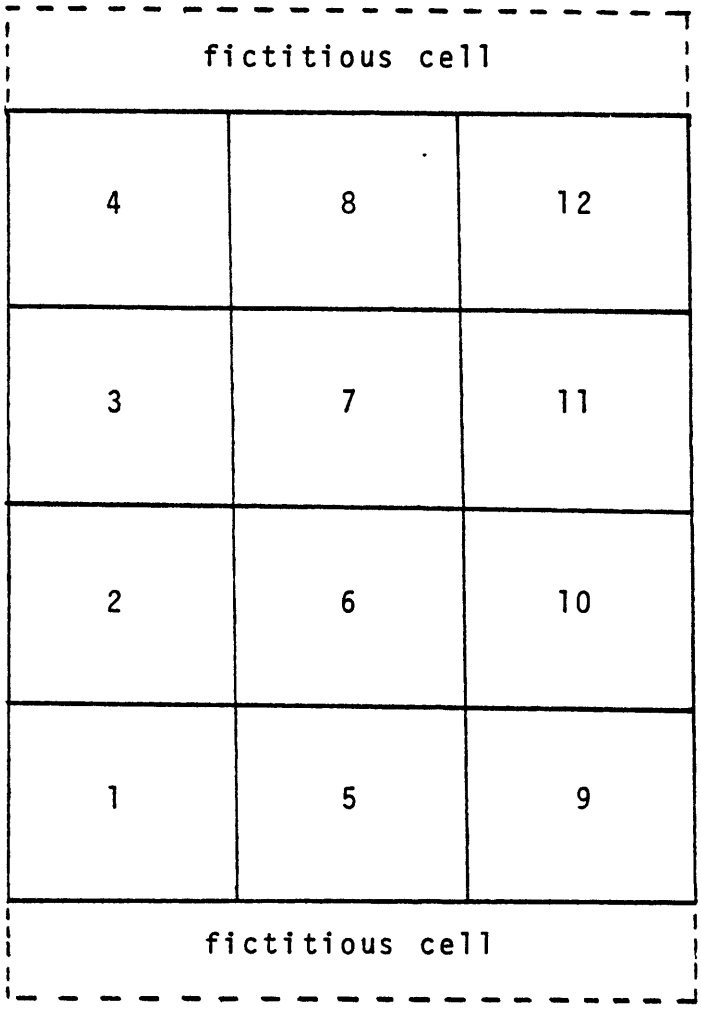


Figure 4.1 Arrangement of Cells for Pressure Field Matrix Shown in Figure 4.2

$$
\begin{vmatrix}
a_{1,1} & a_{1,2} & 0 & 0 & a_{1,5} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
a_{2,1} & a_{2,2} & a_{2,3} & 0 & 0 & a_{2,6} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & a_{3,2} & a_{3,3} & a_{3,4} & 0 & 0 & a_{3,7} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & a_{4,3} & a_{4,4} & 0 & 0 & 0 & a_{4,8} & 0 & 0 & 0 & 0 \\
a_{5,1} & 0 & 0 & 0 & a_{5,5} & a_{5,6} & 0 & 0 & a_{5,9} & 0 & 0 & 0 \\
0 & a_{6,2} & 0 & 0 & a_{6,5} & a_{6,6} & a_{6,7} & 0 & 0 & a_{6,10} & 0 & 0 \\
0 & 0 & a_{7,3} & 0 & 0 & a_{7,6} & a_{7,7} & a_{7,8} & 0 & 0 & a_{7,11} & 0 \\
0 & 0 & 0 & a_{8,4} & 0 & 0 & a_{8,7} & a_{8,8} & 0 & 0 & 0 & a_{8,12} \\
0 & 0 & 0 & 0 & a_{9,5} & 0 & 0 & 0 & a_{9,9} & a_{9,10} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & a_{10,6} & 0 & 0 & a_{10,9} & a_{10,10} & a_{10,11} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & a_{11,7} & 0 & 0 & a_{11,10} & a_{11,11} & a_{11,12} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & a_{12,8} & 0 & 0 & a_{12,11} & a_{12,12}
\end{vmatrix}
$$

Figure 4.2 Pressure Field Matrix

approximations which eventually gave the required solution within a user defined convergence.

The iteration method depended on the matrix being diagonally dominant, i.e. the terms b_{ij} and d_{ij} of equation 4.1 being much smaller than the terms a_{ij} and e_{ij}. This situation allowed the pressure field to be solved for directly in the radial direction, and then iterations were performed in the axial direction until the solution converged. Diagonal dominance could only be maintained by having an axial mesh spacing which was much greater than the radial spacing. This limited the number of cells which could be used in simulations, and thus prevented high resolution.

Since the method employed for the pressure field solution has a strong influence on the running time of the code, a more efficient technique would result in drastically reduced costs. The method now employed is a direct method, i.e. a method which calculates the required solution without any intermediate approximations. The following section will give a background on direct methods and the solution technique employed in NATOF-2D.

4.2 Direct Method Solution Techniques

If a matrix is of the form shown in figure 4.3, the solution is easily accomplished by what is called "back substitution". The n-th equation gives x_n directly (b_n/u_n), and then the (n-1)th equation can be solved for x_{n-1} since x_n is known. This procedure can be continued until x_1 is determined. In matrix terms, the system of equations $\underline{\underline{A}}\underline{x} = \underline{b}$ is easy to solve when $\underline{\underline{A}}$ is an upper triangular matrix. A similar situation occurs when $\underline{\underline{A}}$ is a lower triangular matrix (figure 4.4) and "forward substitution" is employed. The triangular form of the matrix can be obtained by Gaussian Elimination, for example, which uses row interchanges and addition and subtraction of multiples of rows to eliminate all terms below the diagonal. This technique can be used for small matrices, but for the large systems occurring in most NATOF-2D calculations, the computational costs become prohibitive.

In order to avoid the number of row interchanges required by Gaussian elimination, triangularization can be performed on the matrix. Triangulazization refers to factoring the matrix into a lower and a upper triangular such that

$$\underline{\underline{A}} = \underline{\underline{L}}\,\underline{\underline{U}} \tag{4.2}$$

where

$\underline{\underline{L}}$ = lower triangular matrix

$\underline{\underline{U}}$ = upper triangular matrix

$$\begin{vmatrix} u_{11} & u_{12} & u_{13} & u_{14} & u_{15} & u_{16} \\ 0 & u_{22} & u_{23} & u_{24} & u_{25} & u_{26} \\ 0 & 0 & u_{33} & u_{34} & u_{35} & u_{36} \\ 0 & 0 & 0 & u_{44} & u_{45} & u_{46} \\ 0 & 0 & 0 & 0 & u_{55} & u_{56} \\ 0 & 0 & 0 & 0 & 0 & u_{66} \end{vmatrix} \times \begin{vmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{vmatrix} = \begin{vmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \end{vmatrix}$$

Figure 4.3 Upper Triangular Matrix

$$\begin{vmatrix} l_{11} & 0 & 0 & 0 & 0 & 0 \\ l_{21} & l_{22} & 0 & 0 & 0 & 0 \\ l_{31} & l_{32} & l_{33} & 0 & 0 & 0 \\ l_{41} & l_{42} & l_{43} & l_{44} & 0 & 0 \\ l_{51} & l_{52} & l_{53} & l_{54} & l_{55} & 0 \\ l_{61} & l_{62} & l_{63} & l_{64} & l_{65} & l_{66} \end{vmatrix} \times \begin{vmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{vmatrix} = \begin{vmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \end{vmatrix}$$

Figure 4.4 Lower Triangular Matrix

Thus the system of equations

$$\underline{\underline{A}}\underline{x} = \underline{b}$$

becomes

$$\underline{\underline{L}}\underline{\underline{U}}\underline{x} = \underline{b} \tag{4.3}$$

The factorization, when possible, is unique. / 10 / Defining $\underline{y} = \underline{\underline{U}}\underline{x}$, the system $\underline{\underline{L}}\underline{y} = \underline{b}$ can be solved for $\underline{y}$ by forward substitution. Then, $\underline{\underline{U}}\underline{x} = \underline{y}$ can be solved for $\underline{x}$ by backward substitution. This is the basic technique used for the direct solution of the pressure field employed by NATOF-2D.

Since the pressure field matrix is in band form, the number of operations required for the LU factorization is reduced due to the large number of zeros. This is especially true if the bandwidth is much less than the dimension of the matrix. The bandwidth of a matrix $\underline{\underline{A}}$ has a value w if $a_{ij} = 0$ whenever $|i-j| \geq w$. For example, the matrix in figure 4.2 has a bandwidth of 5.

Taking into account the presence of the zeros, the terms of the upper triangular matrix are given by / 11 /:

$$u_{ij} = a_{ij} - \sum_{k = \max(1,j-w+1)}^{i-1} \ell_{ik} u_{kj} \tag{4.4}$$

where

$j = i, \ldots \min(i+w-1, n)$

n = dimension of the matrix $\underline{\underline{A}}$

The terms of the lower triangular matrix are given by the relation:

$$\ell_{ij} = u_{jj}^{-1} \left(a_{ij} - \sum_{k = \max(1,i-w+1)}^{j-1} u_{kj} \ell_{ik} \right) \qquad (4.5)$$

where

$$i = j+1, \ldots \min(j+1-w, n)$$

A comparison between the count of operations using Gaussian elimination and using a LU factorization of a band matrix is given by Franklin / 11 /. For Gaussian elimination there are:

$C_1 = n^2 + (n-1)n(n+1)/3$ multiplications or divisions

$C_2 = n(n-1) + (n-1)n(2n+1)/6$ additions or subtractions

For the LU factorization and solution there are

$C_1 = w(w-1)(3n-2w+1)/3 + (2w-1)n - w(w-1)$ multiplications or divisions

$C_2 = w(w-1)(3n-2w+1)/3$ additions or subtractions

For the matrix given in figure 4.2 the count would be

	C_1	C_2
Gaussian Elimination	1616	1480
LU Factorization and Solution	384	260

As can be seen, the saving is substantial.

The above result shows that the number of operations required for the LU factorization and solution has a strong dependence on the bandwidth of the matrix. The bandwidth can be drastically reduced by reordering the numbering of the cells. In NATOF-2D, the numbering scheme is to count from the bottom to the top for each cell in a channel. Thus a problem with twelve axial levels and three radial nodes has a bandwidth of thirteen. However, by rearranging the numbering so that the cells are numbered across for each axial level, the bandwidth is reduced to four.

Since there will be a number of divisions by the diagonal elements, a partial pivoting strategy is also employed to reduce cumulative rounding error. This is achieved by reordering the rows of the matrix such that the largest elements appear on the diagonal. / 12 /

4.3 A Comparison of Direct and Iterative Methods in NATOF-2D

In this section a comparison is made between the Central Processing Unit (CPU) time usage of the direct and iterative solution methods for both steady state and transient calculations. The test case used for the comparison had 12 axial levels and 5 radial nodes, which gave a matrix of typical size for most NATOF-2D calculations. The results are plotted as CPU time versus time into the simulation.

For the steady state calculation, shown in figure 4.5, the CPU usage for the direct solution is about 100 seconds less than for the iterative solution. The major difference in CPU usage occurs at the start of the calculation, when the system is settling down. As the time into the calculation increases, the CPU usage per Newton iteration decreases for the iterative solution. This is to be expected, since the change in pressure per time step is converging to zero, and therefore fewer iterations are required to meet the convergence criterion. For steady state calculations of a longer duration, the iterative solution may in fact be quicker, since the direct method takes a fixed amount of time to solve the pressure field matrix regardless of the pressure change increment.

For the single phase transient case, shown in figure 4.6, the direct solution used only half the CPU time of the

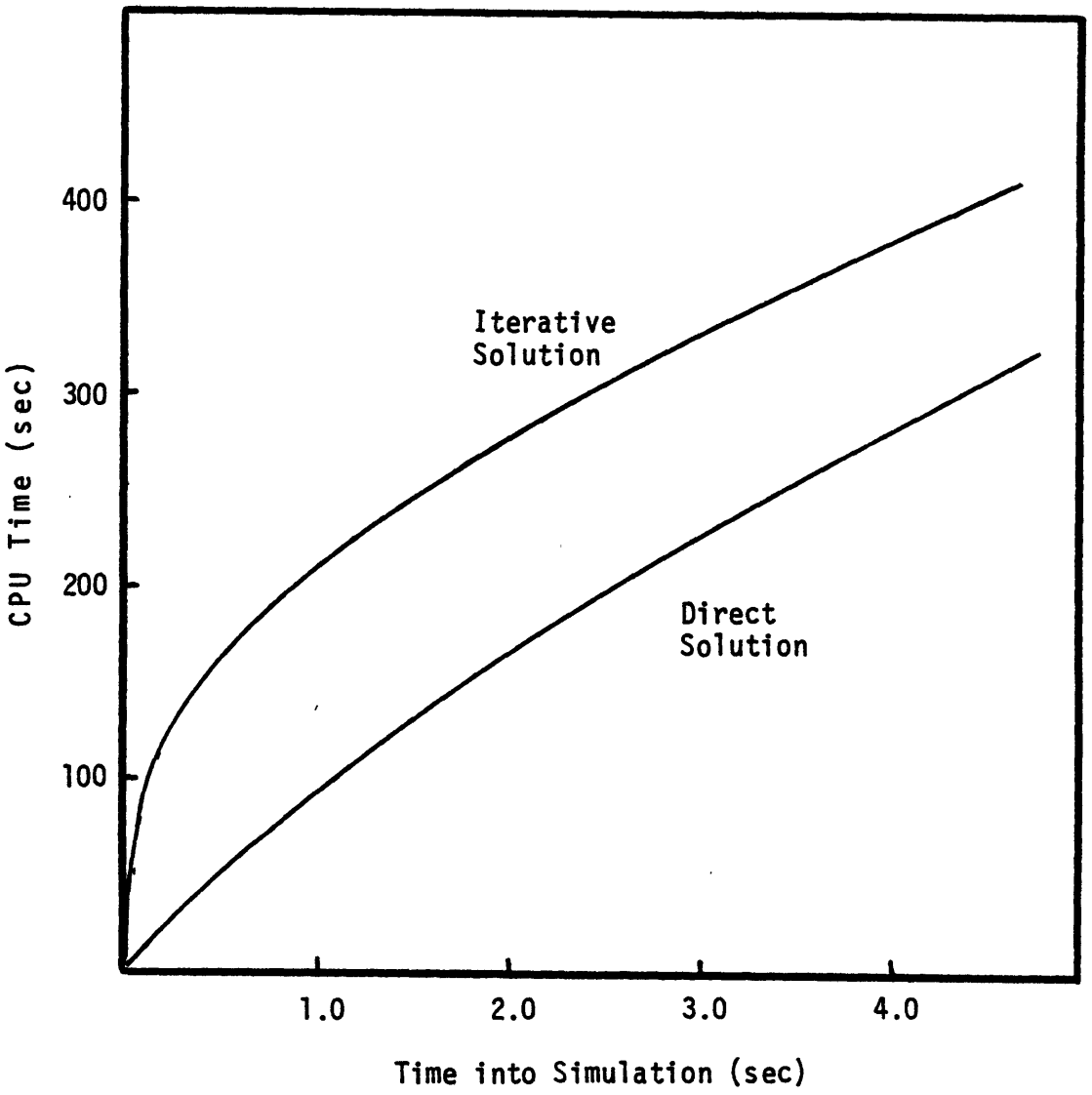


Figure 4.5 A Comparison of Steady State CPU Usage Between the Direct and Iterative Techniques (10 Axial Levels, 5 Radial Nodes)

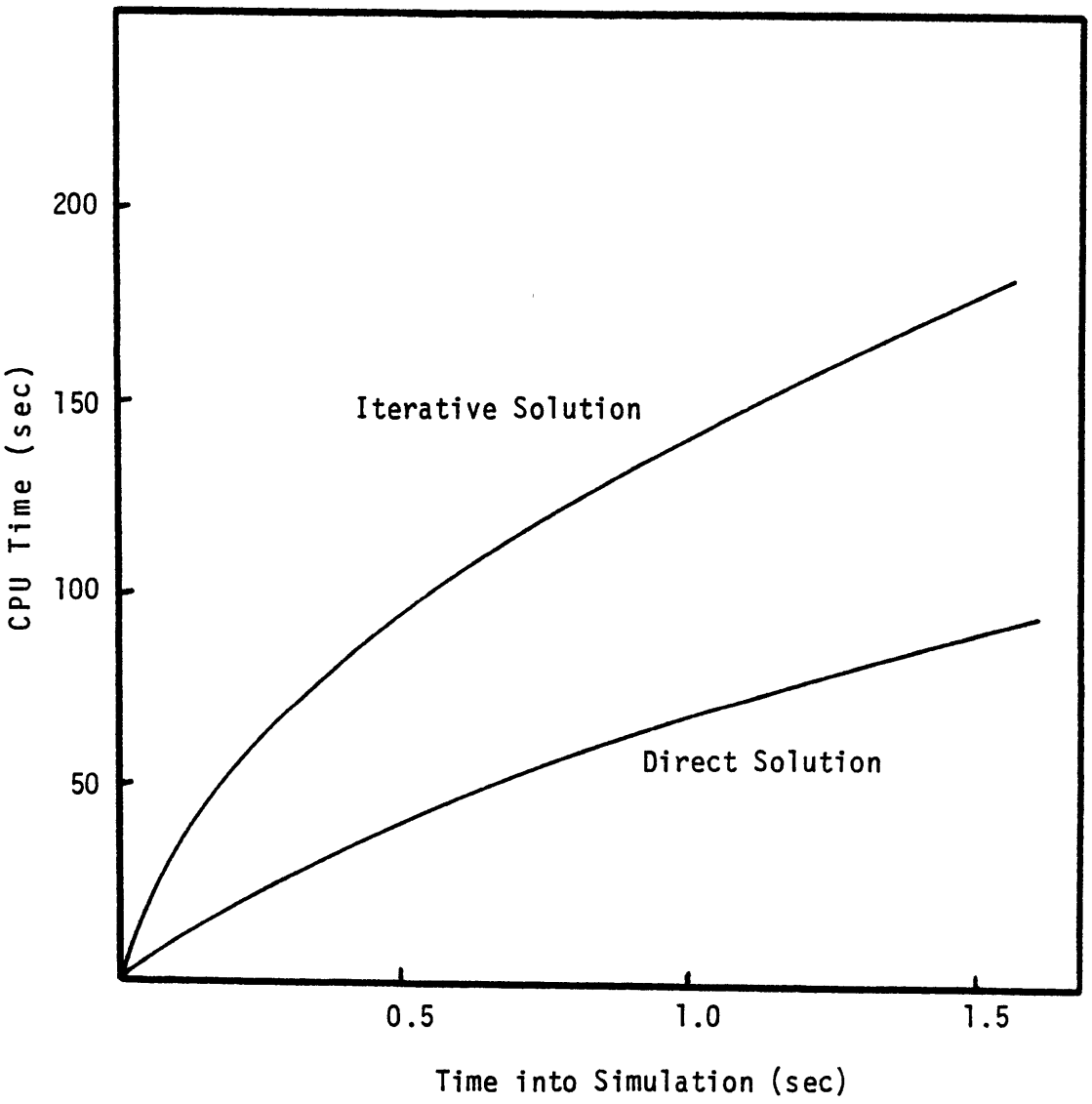


Figure 4.6 A Comparison of Transient CPU Usage Between the Direct and Iterative Techniques (10 Axial Levels, 5 Radial Nodes)

iterative solution. This is a considerable savings when the CPU requirements are large. Experience has shown that during boiling transients, the large pressure changes cause the iterative solution to have an even greater CPU time usage relative to the direct method.

A transient was also run with a decreased mesh spacing, so that there were now 42 cells in the axial direction, covering approximately the same length as before. The iterative technique took four times as much CPU time as the direct solution, thus demonstrating the important role diagonal dominance plays in the iterative technique.

4.4 Programming Information

Two subroutines have been incorporated into NATOF-2D to perform the direct solution. The first one is subroutine DIRECT which rearranges the pressure field matrix in order to minimize the bandwidth. The second is subroutine LEQT1B which performs the direct solution. Subroutine LEQT1B is a commercial subroutine available on the MULTICS computer system at M.I.T. which performs Lower-Upper Factorization and the solution of Band Matrices. The basic algorithum of this subroutine can be found in / 10 /. Due to copyright laws, this subroutine cannot be disseminated outside of the Massachusetts Institute of Technology. However, a subroutine of similar form should be available on most computer systems.

The user has the option to specify either a direct or iterative solution technique by setting the input parameter indgs to either 0 or 1 respectively.

Chapter 5

EXPERIENCES WITH NATOF-2D

5.1 Introduction

The original development of NATOF-2D wasn't completed until June of 1980, and therefore prior experience with running the code was of a limited nature. However, the extremely low computational cost available at M.I.T. over the past six months have allowed numerous testing simulations to be made which yielded information in the area of code capability and constraints. In this chapter some of the experiences encountered will be documented in order to provide a foundation for future work, and also provide an explanation for any changes made to the constituative relations.

5.2 Double versus Single Precision

In an effort to reduce both memory storage and computational costs, NATOF-2D was converted to a single precision code. Single precision refers to the number of significant digits in which a variable is stored and to which a variable is computed. On MULTICS there are 9 significant digits in single precision, and 18 in double precision. At the time, it was felt that carrying out calculations to the 18th place was being excessive.

A comparison was made between the results of the same single phase transient computed in both single and double precision. The results showed exact agreement in the value of variables up to the eigth significant digit. This was encouraging since the single precision computational costs were 25% less.

However, in two phase boiling transients where the timestep size was considerably smaller, problems were encountered in obtaining convergence of the Newton Iterations for the single precision version of NATOF-2D. To explore this problem, the code was modified so that the user could impose a series of decreasing time step sizes on a calculation. The procedure was to allow a single phase calculation to reach a steady state solution, and then gradually reduce the timestep size. Since the code was already at a steady state solution, convergence should always be attained. This is particularly true for the

iteration scheme used, since as $\Delta t \to 0$, $\Delta P \to 0$. However, the results showed that for small timestep sizes, the code actually diverged, and had difficulty in reaching the convergence criteria (in NATOF-2D, convergence is assumed if δP_{max} < user input). If the convergence criteria was relaxed, the code could run to slightly smaller timestep sizes, but it still wasn't converging on zero as a steady state solution implies. Table 5.1 lists the smallest timestep size for various convergence criterias.

Table 5.1

Newton Iteration Convergence (N/m^2)	Minimum Timestep Size (seconds)
0.01	10^{-2}
0.1	10^{-3}
1.0	10^{-4}
10.0	10^{-6}

The problem experienced was traced to the energy conservation equation. The solution scheme employed requires that the quantity

$$\frac{\left[(1 - \alpha)\rho_\ell e_\ell \right]^{n+1} - \left[(1 - \alpha)\rho_\ell e_\ell \right]^{n}}{\Delta t} \qquad (5.1)$$

be evaluated at each timestep. For small timesteps sizes and/or in a steady state configuration, the term $\rho_\ell e_\ell$ has a very small variation. However, the magnitude of this term is typically of the order of 10^9, which is the same as the machine precision. In any single precision computation, it is reasonable to assume that the 1st 8 digits are valid, but that the 9th digit is subject to "noise" fluctuations and computational roundoff errors. Yet for small timesteps, the calculation was relying on this term to solve the equation. Any error would then be inversely proportional to the timestep size.

When the code was returned to double precision, timesteps of 10^{-10} seconds could be attained with no stability problems. From this experience it has been concluded that double precision is a necessity for NATOF-2D.

5.3 On the Modelling of Sodium Reactors

Numerous problems have been encountered in the phenomenological modelling of sodium boiling transients which were not encountered in PWR and BWR modelling. The reason for this difficulty can be traced to the characteristics of LMFBRs and the properties of sodium. A comparison between typical PWR and LMFBR characteristics, and water and sodium properties is given in Table 5.2 and Table 5.3 respectively.

As shown in Table 5.2, the most striking difference in core properties is the temperature rise of the coolant per unit length. For a PWR this is 9.7°C/m, while for an LMFBR it is 125.4°C/m. In the numerical scheme employed in NATOF-2D, the core is divided into a finite number of axial levels. Unless the number is large, there will always exist a substantial temperature difference from cell to cell. In some of the loss-of-flow transients simulated, this can be as much as 150°C. Combined with the higher power density, a model for LMFBR transient analysis experiences rapid temperature changes throughout its length not experienced by PWR codes.

The second major difference is the density ratio, ρ_ℓ/ρ_v, which is approximately 6 for water at 2200 psi, and 1000 for sodium at 44 psi. At atmospheric pressure, the ratio for sodium increases to 3000. The large density ratio for sodium leads to a rapid voiding of the

Table 5.2

A Comparison of PWR and LMFBR properties / 13 /

	PWR	Proposed LMFBR
Core Thermal Power (MWth)	3,411	3,800
Core Diameter (m)	3.4	3.11
Core Height (m)	3.7	1.22
Core Power Density (kw/liter)	98	395.7
Reactor Inlet Temperature (^{o}C)	289	385
Reactor Outlet Temperature (^{o}C)	325	538
System Flow Rate (total 10^{6}lb/hr)	136	136.8

Table 5.3

A comparison of Water and Sodium Properties / 14 /

	Water	Sodium
Pressure (psi)	2250	44
Saturation Temperature (^{o}C)	346.	1016.
Liquid Density (kg/m^{3})	593.4	707.5
Vapor Density (kg/m^{3})	102.	0.748
Liquid Specific Heat (J/kg-^{o}K)	9211.	1324.
Liquid Conductivity (W/m-^{o}K)	0.4074	46.03
Vapor Specific Heat (J/kg-^{o}K)	7709.4	281.8
Vapor Conductivity (W/m-^{o}K)	0.1061	.073

core for extremely small superheats. Accompanying this phenomenon is the expulsion of the liquid phase, and a mass depletion of the core often resulting in flow reversal. The numerical difficulty experienced during the initial stages of boiling is attributable to the fact that the density of the cell varies by a factor of 1000 in a very short timespan.

The boiling transients which the code is required to simulate are thus of an extremely harsh nature. For these reasons, the mass exchange rate plays a critical role in the calculations.

5.4 The Mass Exchange Rate

Of all the constituative relations used in NATOF-2D, the mass exchange rate is probably the most important. The basic physical requirements of any mass exchange model is that the vapor production rate should not exceed the limit established by equilibrium, and also that it should prevent a two-phase situation with highly superheated liquid or subcooled vapor. In essence, it should tend toward equilibrium.

The mass exchange rate determines the rate of vapor evaporation and condensation. The high power density of the core and the high density ratio leads to void fractions of 0.9 in as little as 1/10th of a second. As the void travels into subcooled liquid regions, it is required to condense quickly, since it contains a negligible amount of energy. However, unlike cells where evaporation is occurring and thus the mixture density remains low throughout the transient, cells where condensation is occurring are required to experience rapid density changes throughout the transient.

Since at even void fractions of 0.95, the vapor phase represents only 2% of the mass in the cell, rapid condensation requires that either the pressure of the cell decrease (in order to lower the saturation temperature) or else that the mass flux into the cell be extremely large. The requirements of a large condensation rate have often

lead to code failure on a negative cell pressure error. In either case, the change in pressure for the timestep is large.

The pressure of a cell is the key variable in the numerical scheme of NATOF-2D. Large variations in pressure thus effect the stability of the code, since it is the change in pressure, δP, of a Newton iteration, which determines if convergence has occurred. Defining the convergence criteria in relative terms such that

$$\text{convergence} = \frac{\delta P}{P}$$

a convergence of 10^{-10} can easily be attained in single phase calculations. However, in boiling transients, the convergence must be relaxed to 10^{-3} or 10^{-4}. This has proven a necessity if timesteps are to be taken which are within computational time limitations (i.e. 10^{-3} or 10^{-4} sec)

Small timestep sizes, well below the convective limit, are necessary, since the effect of reducing the timestep size is to reduce the magnitude of Γ, which has units of $kg/s\text{-}m^3$. One noticeable phenomenon during sodium boiling siumlations, is the appearance of a stable boiling timestep, or SBT. The SBT refers to the timestep size at which the code can attain a reasonable convergence (10^{-3} - 10^{-4}) in a single Newton Iteration, but above which convergence cannot be obtained regardless of the number of Newton Iterations.

The SBT appears to be a function of the convergence criteria, the mesh cell spacing distance (whose effects are covered in the next section) and the condensation rate.

The condensation rate, as previously mentioned, has a profound effect, since it requires a large density change corresponding to a small energy change. By numerically reducing the rate (i.e. multiplying it by a small number, which will be designated CF), the calculation proceeds more quickly. What essentially occurs is that the code is allowed to operate in a highly nonequilibrium, two-phase low density mode which prevents the need to make the large density change. The effect is so pronounced that setting the condensation rate to zero allows the code to run at a timestep not limited by any boiling effects (except during the short period of boiling inception) but by the convective limit (10^{-2} sec due to the high vapor velocities), while setting the multiplicative factor to one necessitates timestep sizes of 10^{-7} seconds. In the past, this manipulation has been justified by reference to experiments which showed the condensation rate to be slightly lower than the evaporation rate. However, the fact that analytic results done with NATOF-2D show vapor coexisting with liquid subcooled by hundreds of degrees negates this premise. However, in the experiments simulated in this work, and covered in chapter 6, this manipulation has been necessary in order to obtain results of a time spanning any reasonable

duration.

In the previous formulation used for the mass exchange rate, the term $\alpha(1 - \alpha)$ was added. This term had the effect of inhibiting condensation at small void fractions, and thus allowed the code to run smoother than the formulation presently implimented, which requires higher condensation.

5.5 Varying Mesh Spacings

In order to minimize the effects of the large temperature gradients in the axial direction, the number of axial cells was increased. The two advantags of this are that by decreasing ΔT from cell to cell, the mass exchange rate decreases, and also the results are of a more detailed nature. Smaller mesh cells reduce the inaccuracies caused by volume averaging of the fluid properties. As mentioned in Chapter 4, NATOF-2D could not use small mesh spacing due to the necessity of maintaining diagonal dominance in order to obtain the pressure field solution. However, the direct solution technique does not have this constraint, and so small mesh spacing becomes possible.

There is a practical limit to the number of axial levels. For example, if the number of levels is increased by a factor of 10, the number of computational steps per Newton Iteration will be roughly ten times as much. However, the convective time limit, $\Delta z/v$, would decrease the time step size also by a factor of ten, so that the computation would need about 100 times more cpu time to compute the same time length in single phase. It is doubtful that a time step size two orders of magnitude greater than before could be taken during the boiling transient.

For testing purposes, the number of axial levels was increased by a factor of 4, from 10 levels to 40 levels, to

see whether any increased timesteps could be taken during boiling. The decreased mesh spacing allowed timesteps to be taken which varied from 2×10^{-3} seconds to 10^{-4} seconds, with a condensation factor of = 0.01. The cpu usage was roughly three times greater than for the 10 axial level case, which had a CF = 0.002. Increasing CF to 0.1, for the same 40 axial level calculation resulted in a cpu usage 6 times greater, and a SBT of 2×10^{-4} seconds.

The conclusion drawn from this is that higher accuracy results can be attained without the corresponding cpu costs by decreasing the mesh spacing. For the experimental test simulations covered in Chapter 6, 40 axial levels were used.

Chapter 6
VERIFICATION OF MODELS

6.1 Introduction

In order to test the capability of NATOF-2D and the validity of the models described in the previous chapters, a total of five sodium boiling transients were simulated. The first one was THORS Bundle 6A Run 101 conducted at Oak Ridge National Laboratory / 4 /. The other four were from the W-1 SLSF Experiments, done by the Hanford Engineering Development Laboratory, and include two Loss of Piping Integrity transients and two Boiling Window Tests / 5 /.

The tests cover a wide range of conditions under which the code will be required to operate. In contrast to previous simulations / 1 /, the decision has been made to drastically increase the number of mesh cells. This improved the quality of the results, but also constrained the length of the calculation due to the large CPU usage.

In the next sections, a description of the runs will be given, and the results will be presented.

6.2 THORS Bundle 6A Experiment, Test 71H Run 101

6.2.1 Description of the THORS Bundle 6A Experiments

The purpose of the THORS Bundle 6A Experiments / 4 / was to investigate the extent of dynamic boiling stability at low flow conditions. The tests were conducted in the THORS Facility, an engineering-scale high-temperature sodium facility for the thermo-hydraulic testing of LMFBR subassemblies.

The test section used was a full-length simulated LMFBR fuel subassembly. It consisted of 19 electrically heated fuel pin simulator units spaced by helical wire-wrap spacers. The heated length of Bundle 6A was 0.9 meters, and had variable pitch heater windings to produce a 1.3 axial peak-to-mean chopped cosine power distribution.

Appreciable effort was expended in designing and fabricating a low thermal inertia bundle housing. However, a posttest analysis revealed that sodium had penetrated the entire housing region, and thus the housing had a high heat capacity.

The selected run from this series was Test 71H, Run 101. The bundle power level was 127 kw. This was the lowest power run that exhibited dryout. At 3.2 seconds into the transient, the initial flow of 0.39 l/s was decreased over a period of 3.6 seconds to a flow of 0.12 l/s. Boiling inception occurred at 13.7 seconds. Flow oscillations

occurred at a frequency of 1.1 to 1.5 Hz. A geometry description of the bundle is given in Table 6.1. In Figure 6.1 the locations of the axial levels used in the NATOF-2D simulation are shown.

The simulation was carried out under inlet velocity, outlet pressure boundary conditions during the single phase part of the transient. This allowed the exact flow rate to be maintained, while the large number of axial cells permitted the proper flow splits to be established in the first three or four cells. At the point of boiling inception, the boundary conditions were switched to pressure/pressure, and the transient was continued.

For this simulation, CF was 0.01, and the fluid-to-fluid radial heat conduction nusselt numbers were $Nu_1 = 13$ and $Nu_2 = 13$. The input parameters for the simulation are given in Appendix C.3.

6.2.2 Simulation Results

The time of boiling inception during the NATOF-2D simulation was approximately 2.7 seconds sooner than the THORS experiment. This can be attributed to underestimating the heat capacity of the hexcan. In NATOF-2D, the hexcan is modelled as a heat capacitance. During transients the hexcan is heated up and cooled off by the coolant, with no losses to the environment. In principal, by adjusting the heat capacity of the hexcan, the boiling inception time can

Table 6.1

Description of THORS Bundle 6A

Number of Pins	19
Pin Diameter (m)	5.84×10^{-3}
Pitch/Diameter Ratio	
Wire Wrap Diameter (m)	1.42×10^{-3}
Distance from fuel pin to the wall (m)	0.71×10^{-3}
Heated Length (m)	0.9
Axial Power Distribution, Cosine peak/mean	1.3
Radial Power Distribution, peak/mean	1.0
Total Bundle Power (kw)	127
Pin Power (kw/pin)	6.7
Inlet Flow, Steady State (kg/s)	0.3344
Inlet Flow, Boiling Inception (kg/s)	0.1029
Inlet Liquid Temperature (^{o}K)	660.91

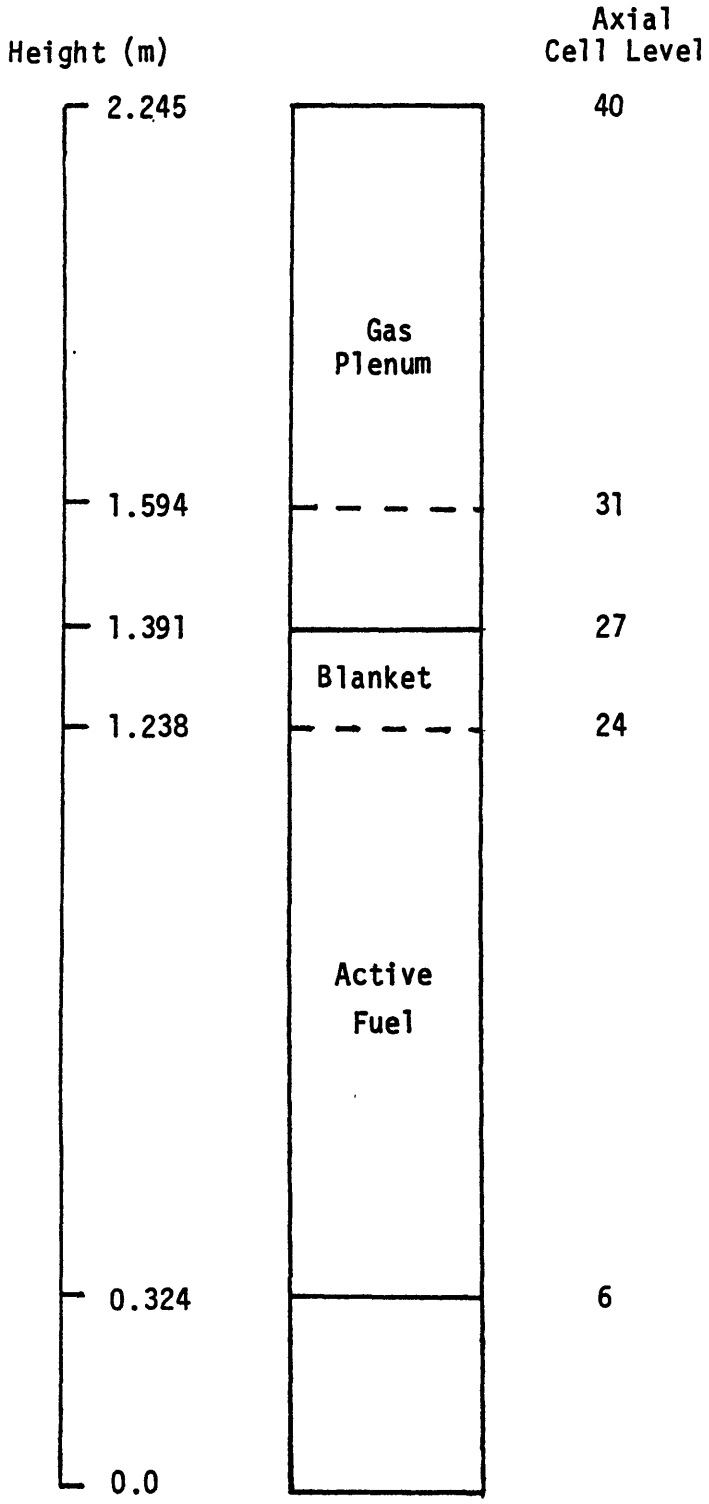


Figure 6.1 Location of Cells used in the NATOF-2D Simulation of THORS Bundle 6A Experiment

assume any desired value. Prior to the start of a calculation, the value is estimated based on the properties of the can. The results of this test indicate the need for a more sophisticated model. Appendix D discusses the sensitivity of the boiling inception time to the hexcan heat capacity.

At the time of boiling inception, flow reversal followed almost immediately. There are two reasons for this. The first is that the small rate of condensation allows the channel to void rapidly, which increases the fluid pressure and forms a flow blockage. The second reason is that information about pressure drops at the inlet due to valve adjustments was not documented, so this effect could not be accurately simulated. As with the hexcan heat capacity, it is possible to adjust the spacer pressure drop feature of NATOF-2D to obtain the experiment's flow reversal time. Appendix E gives typical values of the spacer pressure drop, and its effect on flow reversal.

Figure 6.2 shows the inlet mass flow rate during the transient. The flow oscillations were quite severe and had a frequency of about 3 Hz. This was about twice the frequency of oscillations found in the experiment. In Figure 6.3, the temperature profile of the central channel at various points of time during the transient is plotted. Figure 6.4 shows the temperature history at the end of the heated section for both the central and edge channels. This

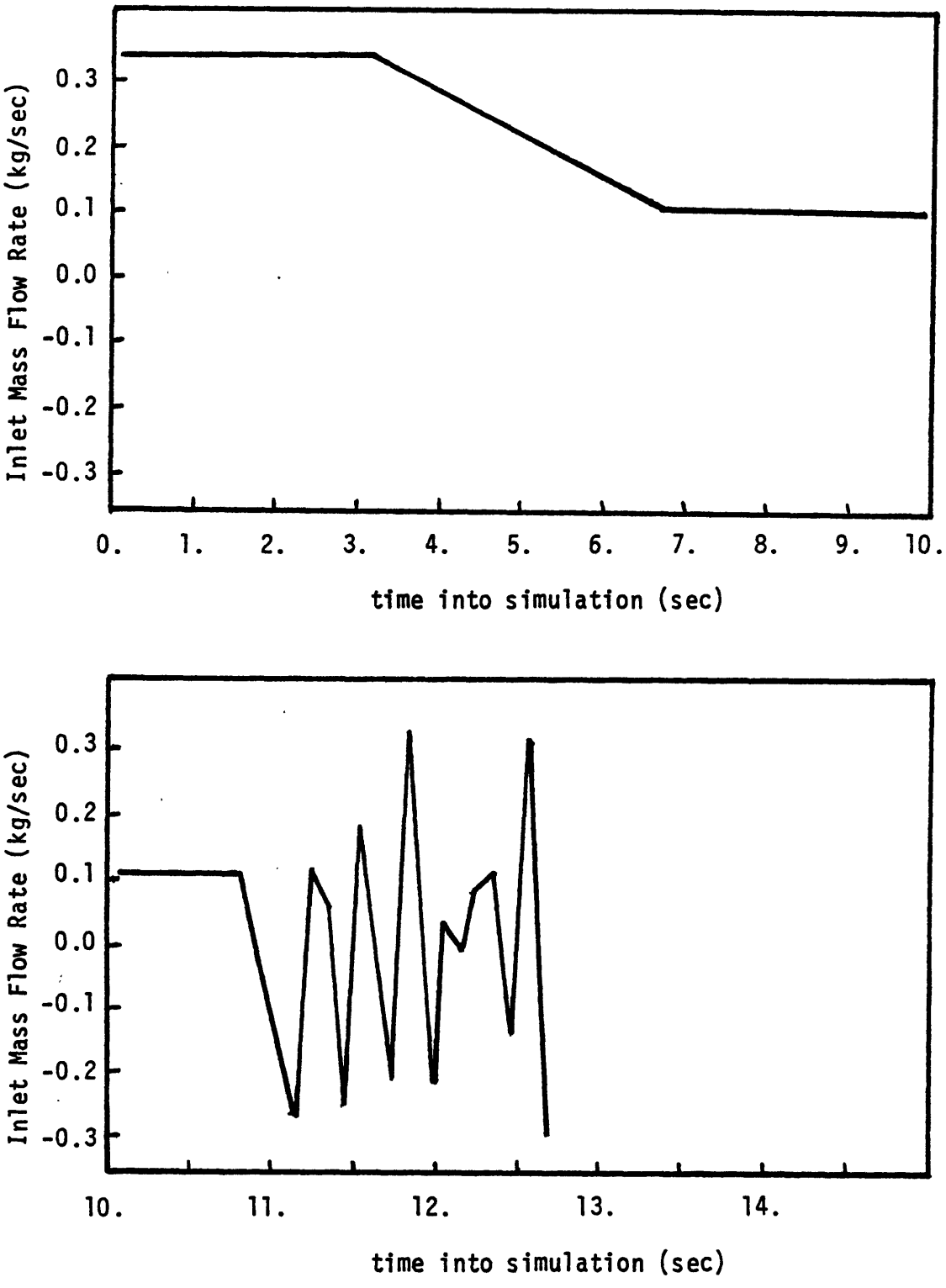


Figure 6.2 NATOF-2D Predicted Inlet Mass Flow Rate of THORS Bundle 6A Test 71H, Run 101

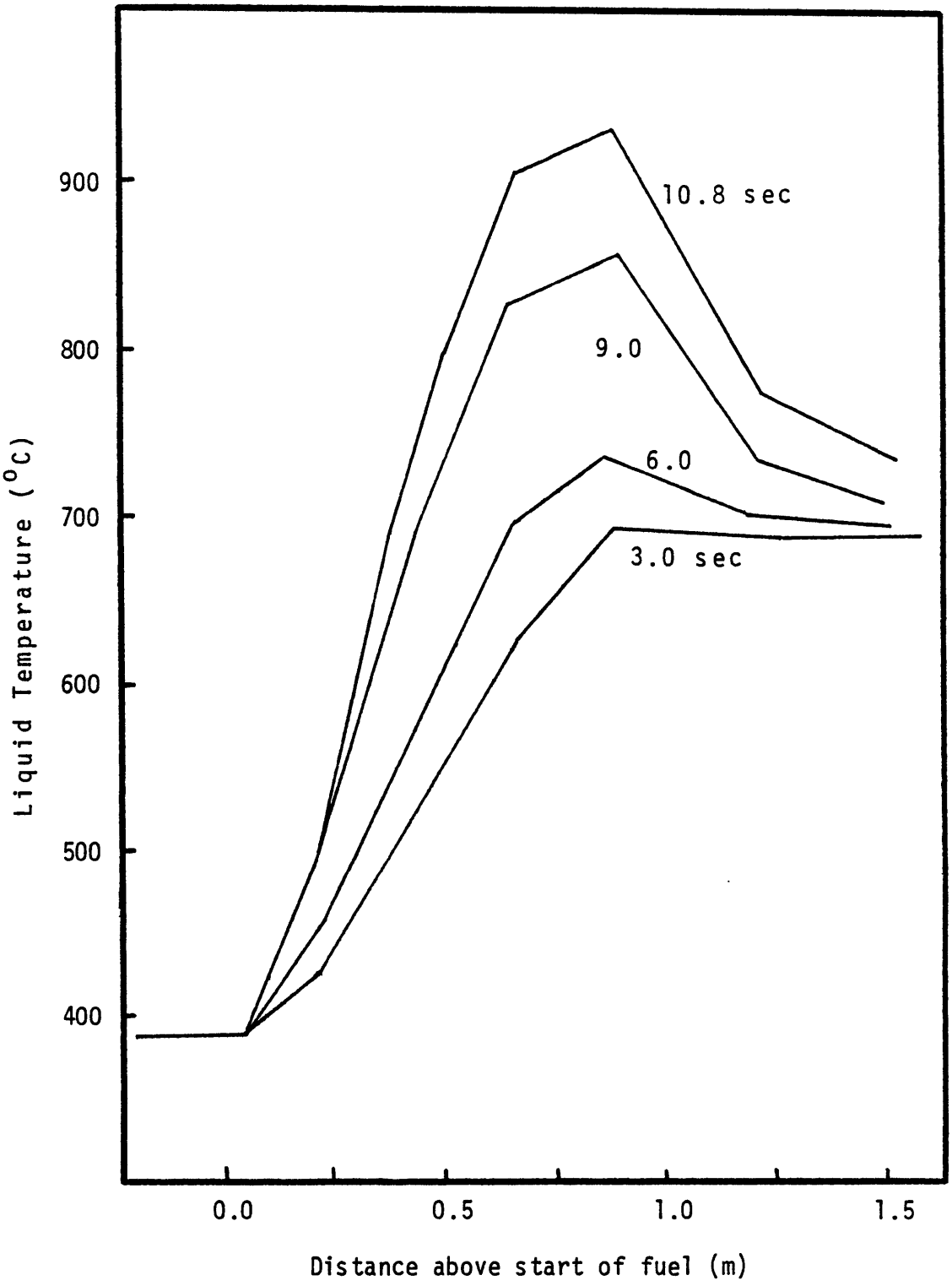


Figure 6.3 NATOF-2D Temperature Profiles of Central Channel at Various Points in Time (THORS Bundle 6A Test 71H, Run 101)

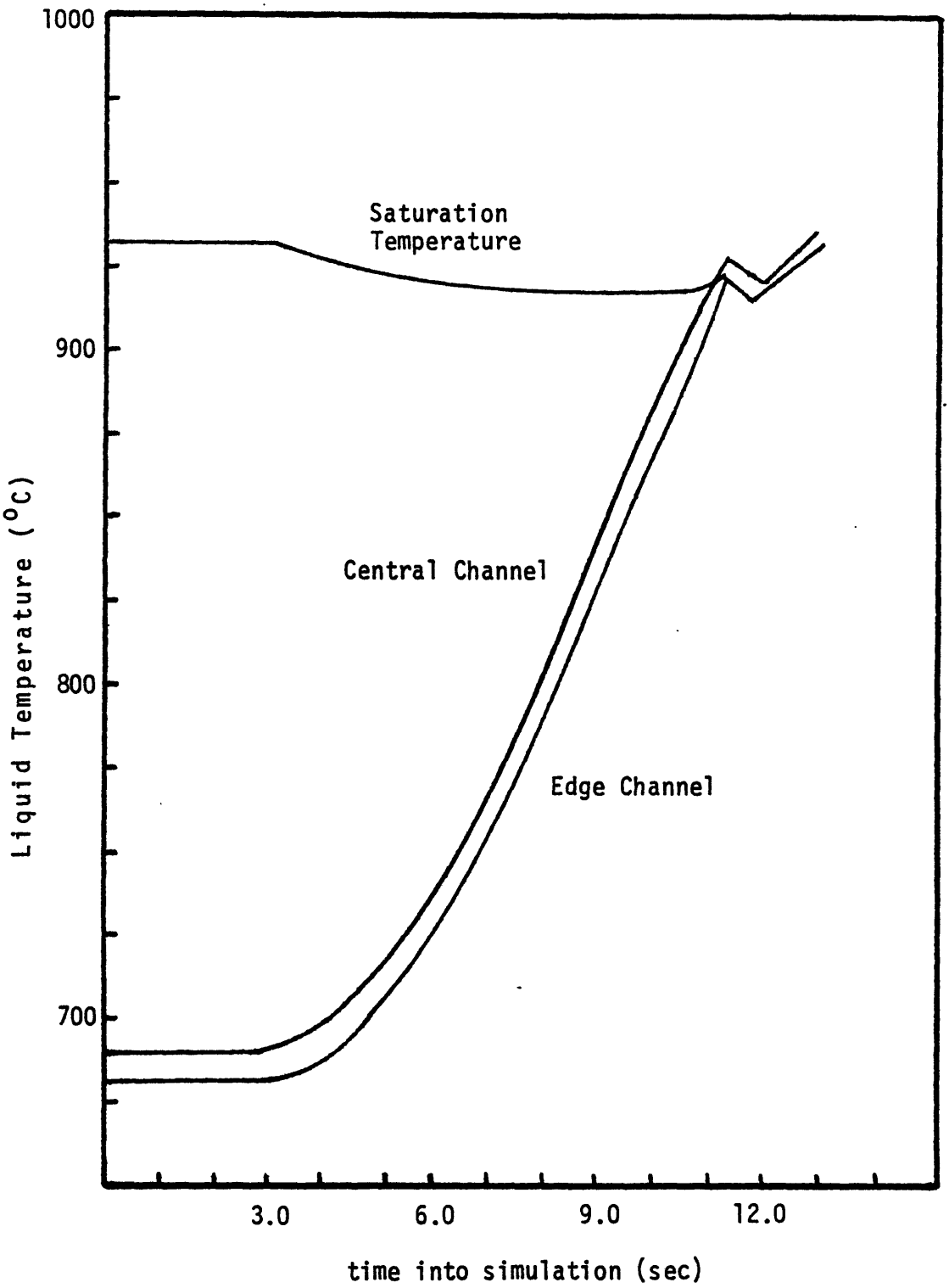
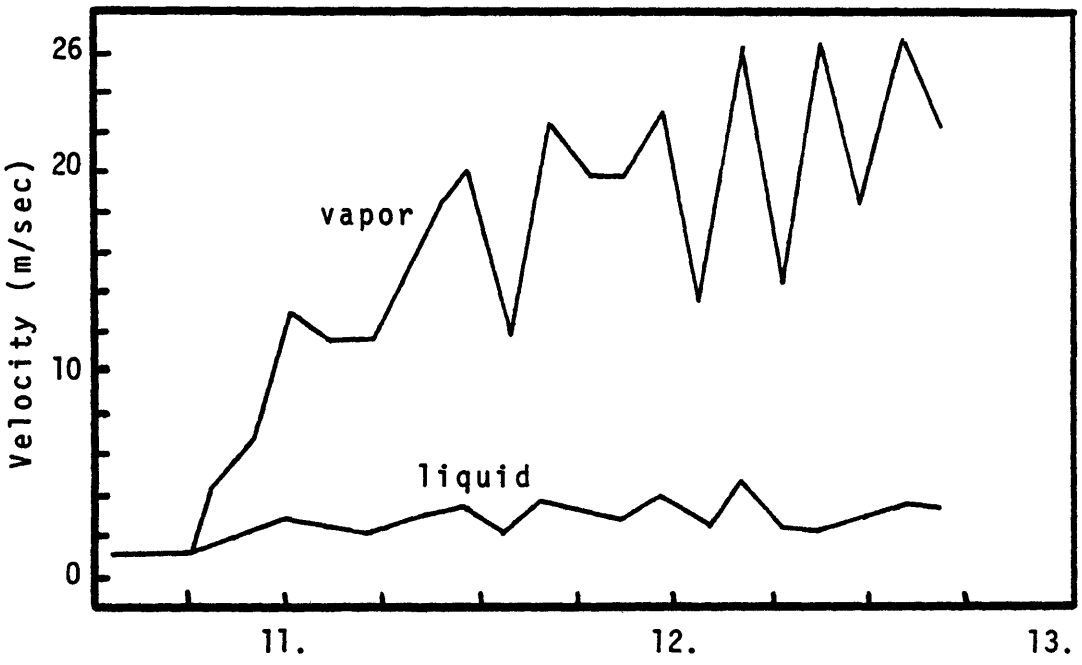


Figure 6.4 NATOF-2D Predicted Temperature Histories at End of the Heated Zone for the Central and Edge Channels (THORS Bundle 6A Test 71H, Run 101)

figure demonstrates the effects of the radial heat conduction model in reducing radial temperature variations to levels comparable with the test results.

Figure 6.5 compares the vapor and liquid velocities. The large difference in velocities indicates the need for a separated flow model.



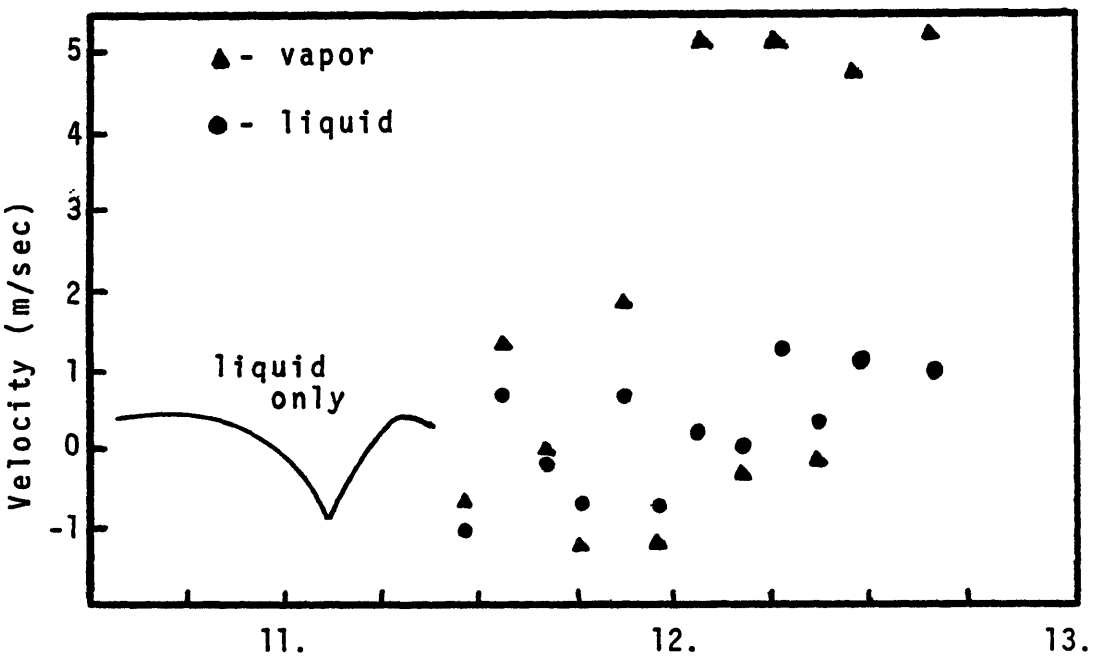


Figure 6.5 A Comparison of NATOF-2D Predicted Liquid and Vapor Velocities
(THORS Bundle 6A Test 71H, Run 101)

6.3 The W-1 SLSF Experiments

6.3.1 Test Objective

The W-1 SLSF Experiment / 5 / was designed to provide experimental data on sodium boiling and boiling "stability" in a fuel pin bundle during flow transients under LMFBR accident conditions. The test was divided into two groups. The first was the Loss-of-Piping Integrity (LOPI) accident simulation. The objective of this series was to determine the heat transfer characteristics from fuel pins to sodium coolant during a transient simulating a double ended pipe break at the primary vessel inlet nozzle. The difference in heat transfer characteristics as a result of fuel conditions were studied.

The second group of tests was the Boiling Window Tests. The objective of this series was to establish the family of flow/heat-flux combinations that will produce incipient boiling in the bundle for a given inlet temperature. The test runs were designed to determine whether or not there is a regime of boiling beyond the onset that persists and does not immediately lead to dryout for low flow and intermediate to high heat fluxes.

The SLSF W-1 boiling window tests were conducted in three operating phases: approach to boiling, incipient boiling and dryout with fuel pin failure. The incipient boiling tests were designed to determine low heat flux

combinations for which the onset of sodium boiling is achieved. One dryout test was performed to identify the far end of the "boiling window" at the highest heat flux tested.

6.3.2 Test Apparatus and Procedure

The tests were carried out on the Sodium Loop Safety Facility (SLSF) at the Engineering Test Reactor under the direction of the Hanford Engineering Development Laboratory.

Each LOPI was initiated from steady state full power full-flow conditions. Over the first 0.5 seconds of the transient the inlet flow was dramatically reduced. At 0.65 seconds into the transient, the reactor was scrammed. The test section was returned to full flow after approximately 3 seconds from time zero.

The Boiling Window Tests were initiated at a steady-state flow of 1.95 kg/sec. The flow was linearly reduced to its "low flow" value in 0.5 seconds, where it was held for a specific time, and then linearly returned to its initial state in 0.5 seconds.

The geometric parameters of the fuel bundle and the characteristics of the tests are given in table 6.2 and 6.3 respectively. Figure 6.6 shows the axial cell locations used in the NATOF-2D simulations.

6.3.3 Tests Chosen For Simulation

From the available results, a total of four test were

Table 6.2

Geometric Parameters of the W-1 SLSF Bundle

Number of Pins	19
Pin Diameter (m)	5.842×10^{-3}
Fuel Pellet Diameter (m)	4.94×10^{-3}
Wire Wrap Diameter (m)	
inner pins	1.422×10^{-3}
outer pins	7.11×10^{-4}
Fuel Pitch (m)	7.264×10^{-3}
Pitch/Diameter Ratio	1.25
Flat-to-Flat (m)	3.26×10^{-2}
Length of Active Fuel	0.9144
Axial Power Distribution, Cosine	
peak/mean	1.4
Radial Power Distribution, Cosine	
peak/mean	1.0
Fuel	Uranium-Plutonium mixed oxide, Pu 25% of total mass
Fill Gas	10% Helium-Neon
Inlet Liquid Temperature (^{o}C)	388.

Table 6.3

Power and Flow Rates of the W-1 SLSF Tests

LOPI 2A	
Power (kw)	661.8
Steady State Flow Rate (kg/s)	1.95
Low Flow Rate (kg/s)	.65
LOPI 4	
Power (kw)	705.3
Steady State Flow Rate (kg/s)	1.95
Low Flow Rate (kg/s)	.65
BWT 2'	
Power (kw)	348.3
Steady State Flow Rate (kg/s)	1.95
Low Flow Rate (kg/s)	.47
BWT 7B'	
Power (kw)	661.8
Steady State Flow Rate (kg/s)	1.98
Low Flow Rate (kg/s)	.74

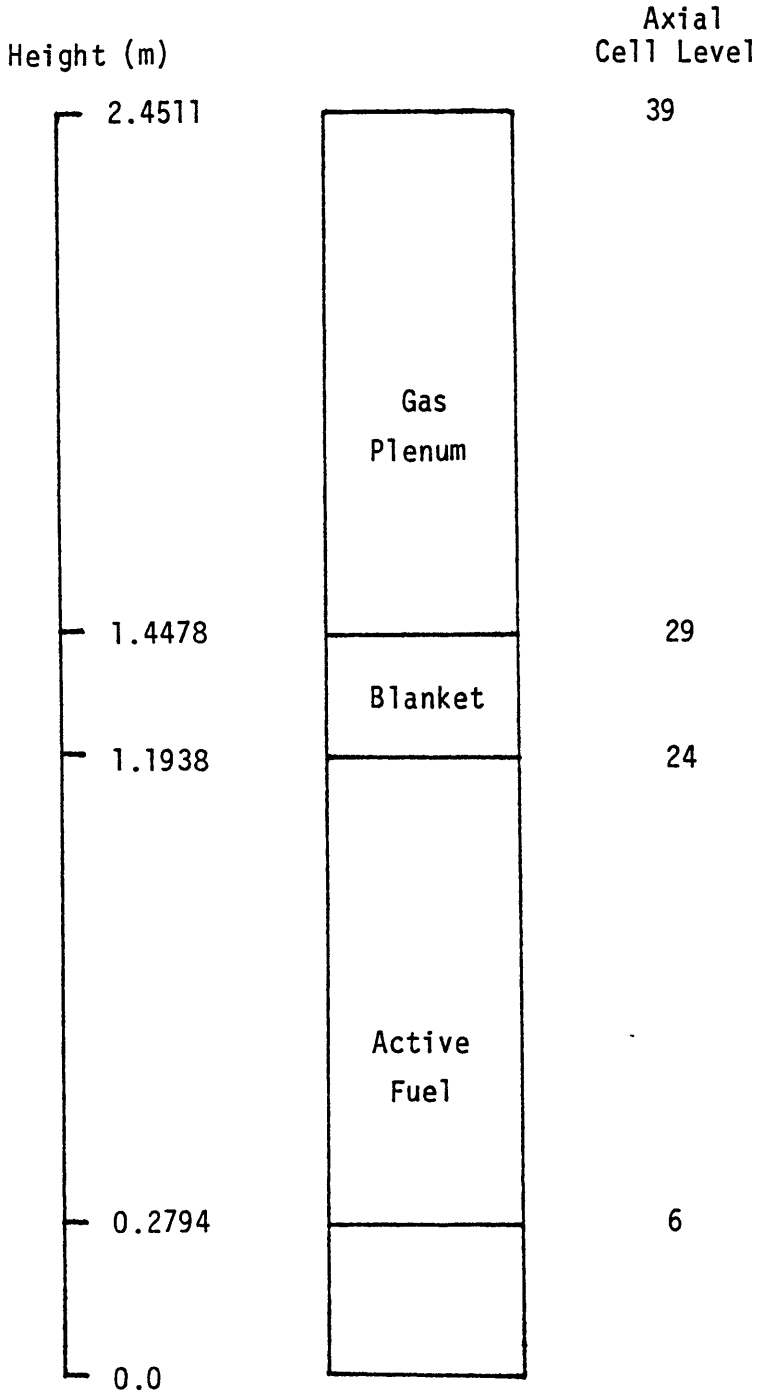


Figure 6.6 Location of Cells used in the NATOF-2D Simulation of W-1 SLSF Experiment

chosen for simulation with NATOF-2D. Two tests were selected from the LOPI tests, and two from the BWT tests. These tests were selectively chosen to provide a full range of transients: from single phase calculation to a full dryout simulation.

The first LOPI simulation is LOPI 2A. In this test, the maximum coolant temperature reached 941°C and showed no indication of boiling. This test will provide some calibration information for the future test of the series.

The second LOPI simulation is LOPI 4. The maximum coolant temperature reached was 956°C, and the data showed about 0.5 seconds of boiling. Failure of one of the thermocouples used for the test section power calculations resulted in LOPI 4 being run at approximately 5% overpower (15.12 kw/pin). This transient will test the sensitivity of the code to such an occurrence.

The first BWT simulation is BWT 2'. In this test approximately 0.8 seconds of boiling was observed. The coolant temperature reached 953°C at a pin power of 7.5 kw/ft. The test section was held at 24% of full flow for a duration of 4 seconds. This test offers the opportunity of simulating a low-power/low-flow sodium boiling transient.

The second BWT to be simulated is BWT7B'. This is the most interesting and demanding test to be run by the code. Approximately 2.0 seconds of boiling occurred before clad dryout at a pin power of 14.4 kw/ft. The test section had

38% of full flow for a period of 3.0 seconds. Inlet flow oscillations, flow reversal and dryout is the worst case hypothesized for sodium boiling codes. Ability to model this sequence of events will severly test the limitations of NATOF-2D.

The input files used in the NATOF-2D simulation of the W-1 SLSF Tests are given in Appendix C.

6.4 W-1 SLSF Simulation Results

6.4.1 LOPI 2A

The LOPI 2A simulation was carried out under velocity/pressure boundary conditions in order to accurately duplicate the experiment's inlet flow rate (shown in Figure 6.7). A comparison between the NATOF-2D predicted liquid temperature and the experiment's result is shown in Figure 6.8. The shape of the two curves are fairly close, with NATOF-2D predicting slightly higher temperatures. Differences are to be expected however, since average cell temperatures are being compared with temperatures taken at a point. This simulation shows that the fuel pin properties used in the code are quite good, and accurately model the response of real fuel pins during a reactor scram.

Figure 6.9 compares the temperatures at the end of the heated zone for the central and edge channels. This figure shows how the edge channel takes longer to respond to the transient due to the effects of heat losses to the can.

6.4.2 LOPI 4

The LOPI 4 simulation was also run under velocity/pressure boundary conditions. The flow rate is shown in 6.10. As in the experiment, there was approximately 0.5 seconds of boiling. The NATOF-2D predicted temperature at the end of the heated zone closely matched the experiment for the central channel (Figure

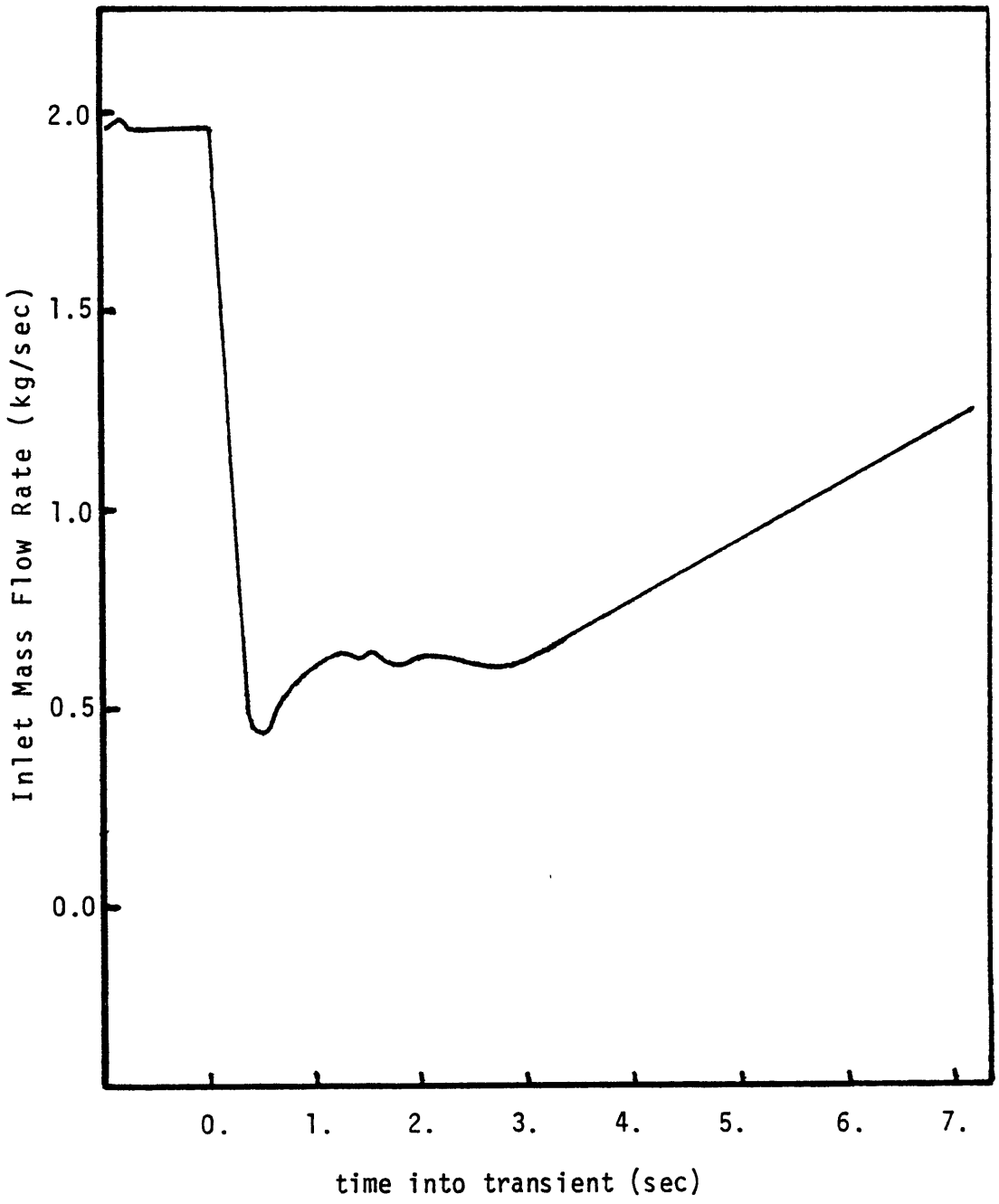


Figure 6.7 W-1 SLSF Test LOPI 2A Experiment Inlet Mass Flow Rate

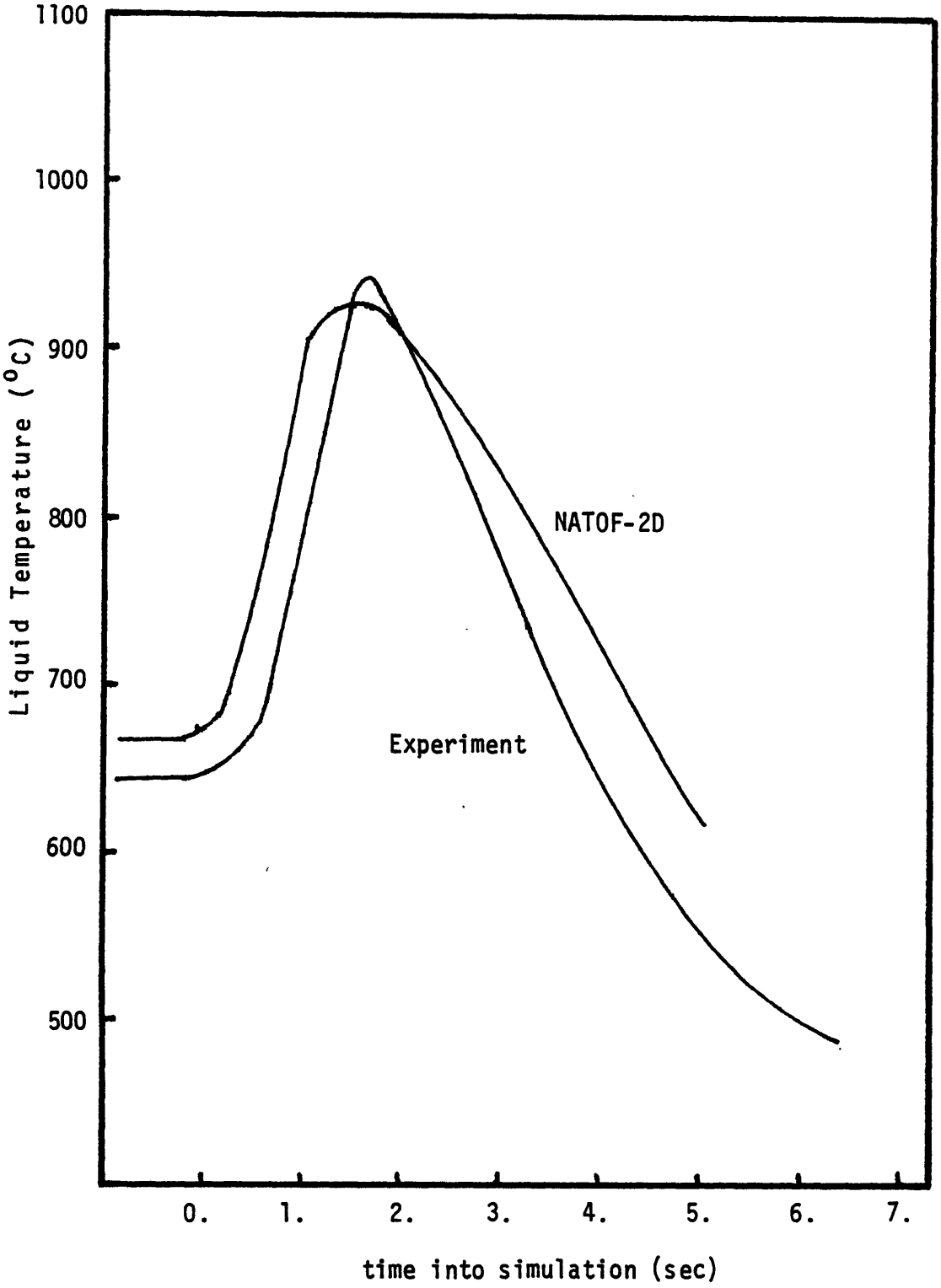


Figure 6.8 A Comparison Between Experiment and NATOF-2D Predicted Temperature Histories at the End of the Heated Zone for the Central Channel (W-1 SLSF Test LOPI 2A)

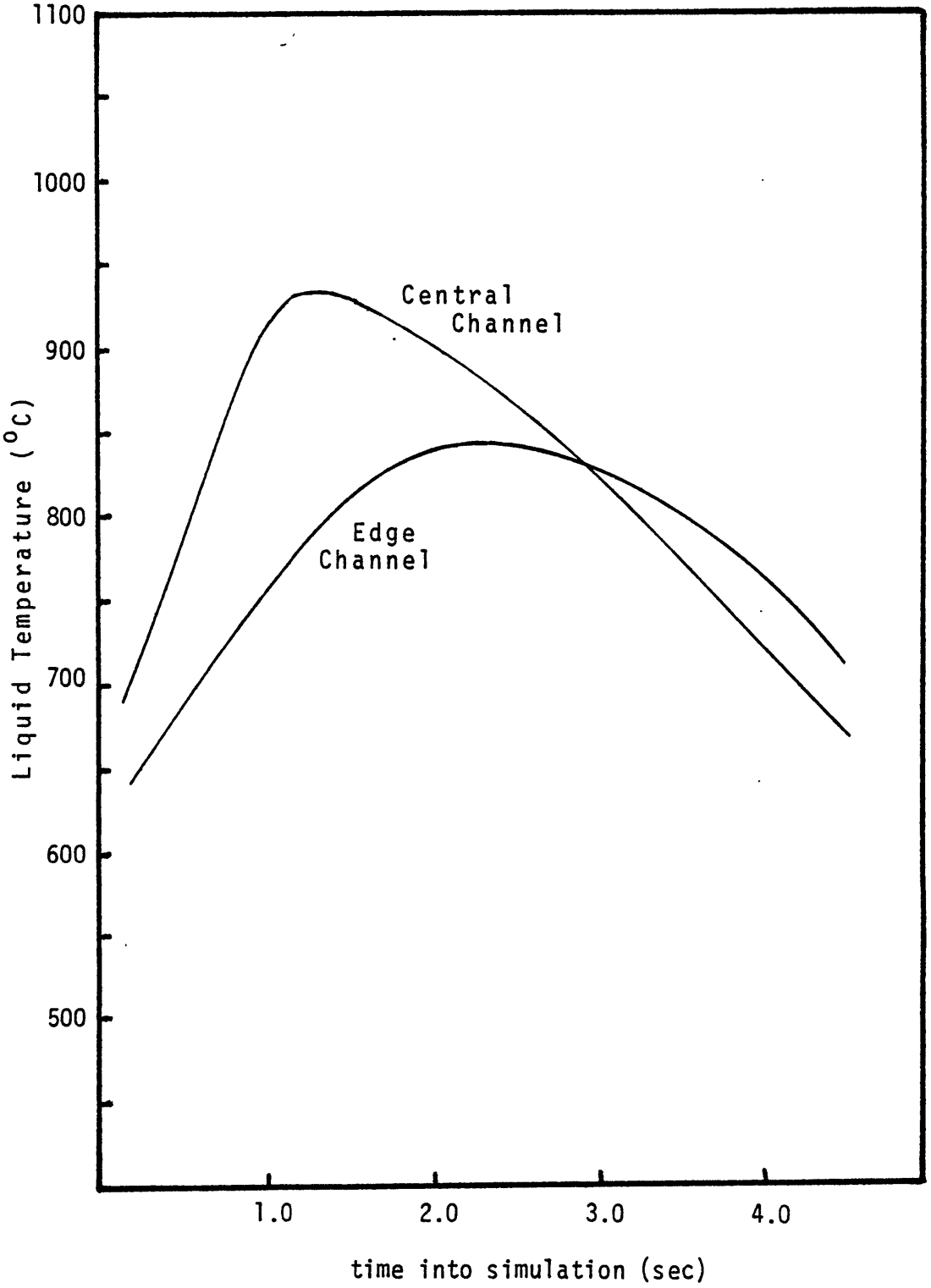


Figure 6.9 A Comparison of NATOF-2D Central and Edge Channel Temperature Histories at the End of the Heated Zone (W-1 SLSF Test LOPI 2A)

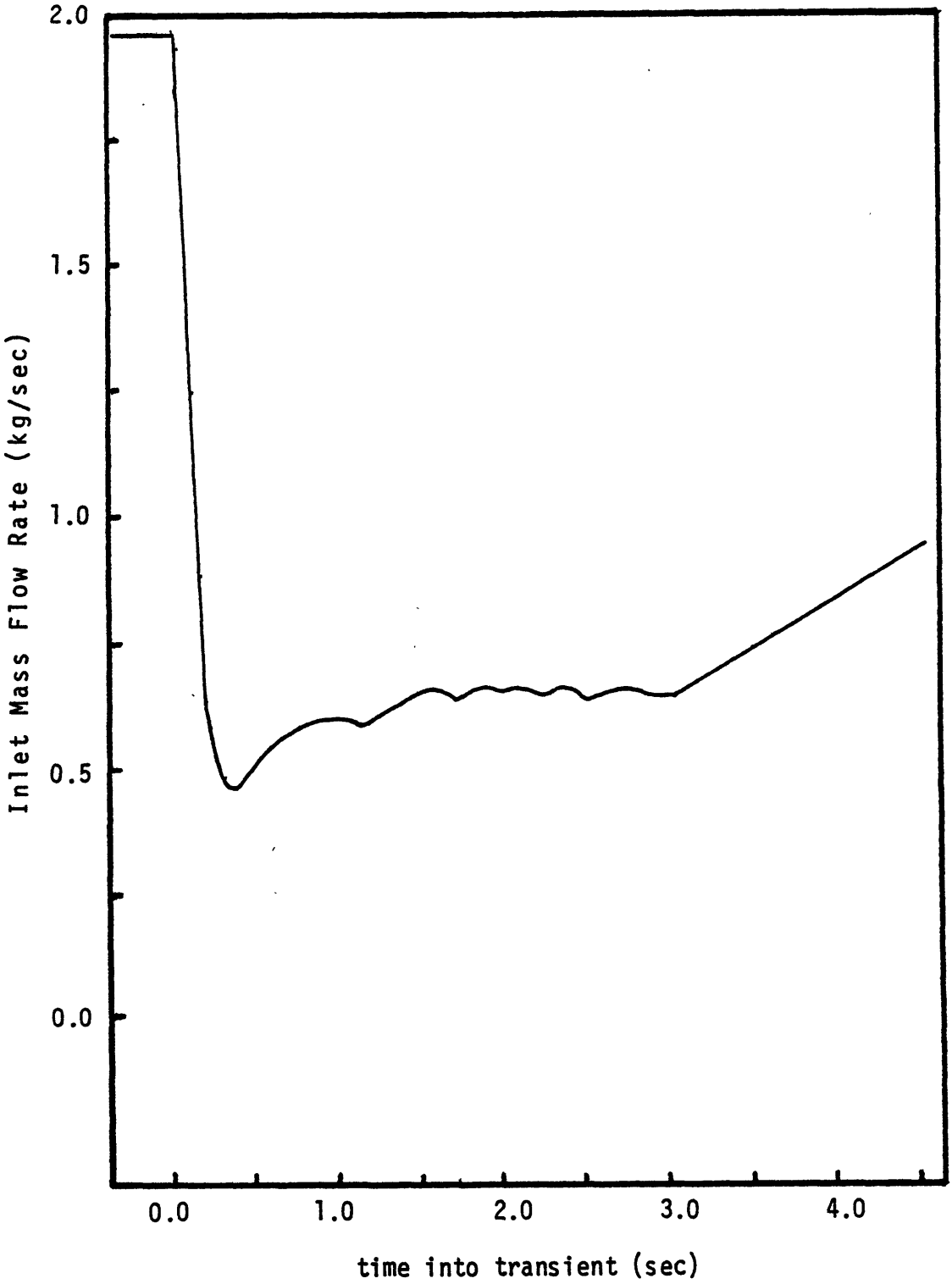


Figure 6.10 W-1 SLSF Test LOPI 4 Experiment Inlet Mass Flow Rate

6.11), but not as well for the edge channel (Figure 6.12).

Void maps for the central and middle channels are shown in Figure 6.13. Although these maps are for voids of 0.1, they nearly overlap the void maps for 0.9. A negligible void was found in the edge channel. In and near the voided regions the radial velocities were large, as the coolant traveled from the center of the bundle to the edge. This demonstrates the value of a two-dimensional model in simulating boiling transients.

6.4.3 BWT 2'

The BWT 2' simulation was a low power-low flow transient. The test section inlet mass flow rate is shown in Figure 6.14. The temperature histories at the midplane and end of the heated zone for the central channel are shown in Figures 6.15 and 6.16. Unlike the experiment, which had 0.8 seconds of boiling, no sodium boiling occurred in the simulation. The liquid reached a maximum temperature of 931^{o}C, approximately 20^{o}C below saturation.

The exact reason for this result is not known, but possibly may be associated to overestimating the heat capacity of the fuel rods. The reported inlet mass flow rate also could have been too high. In a low power transient such as this, the necessity of using cell averaged temperatures, results in lower peak temperatures. Even

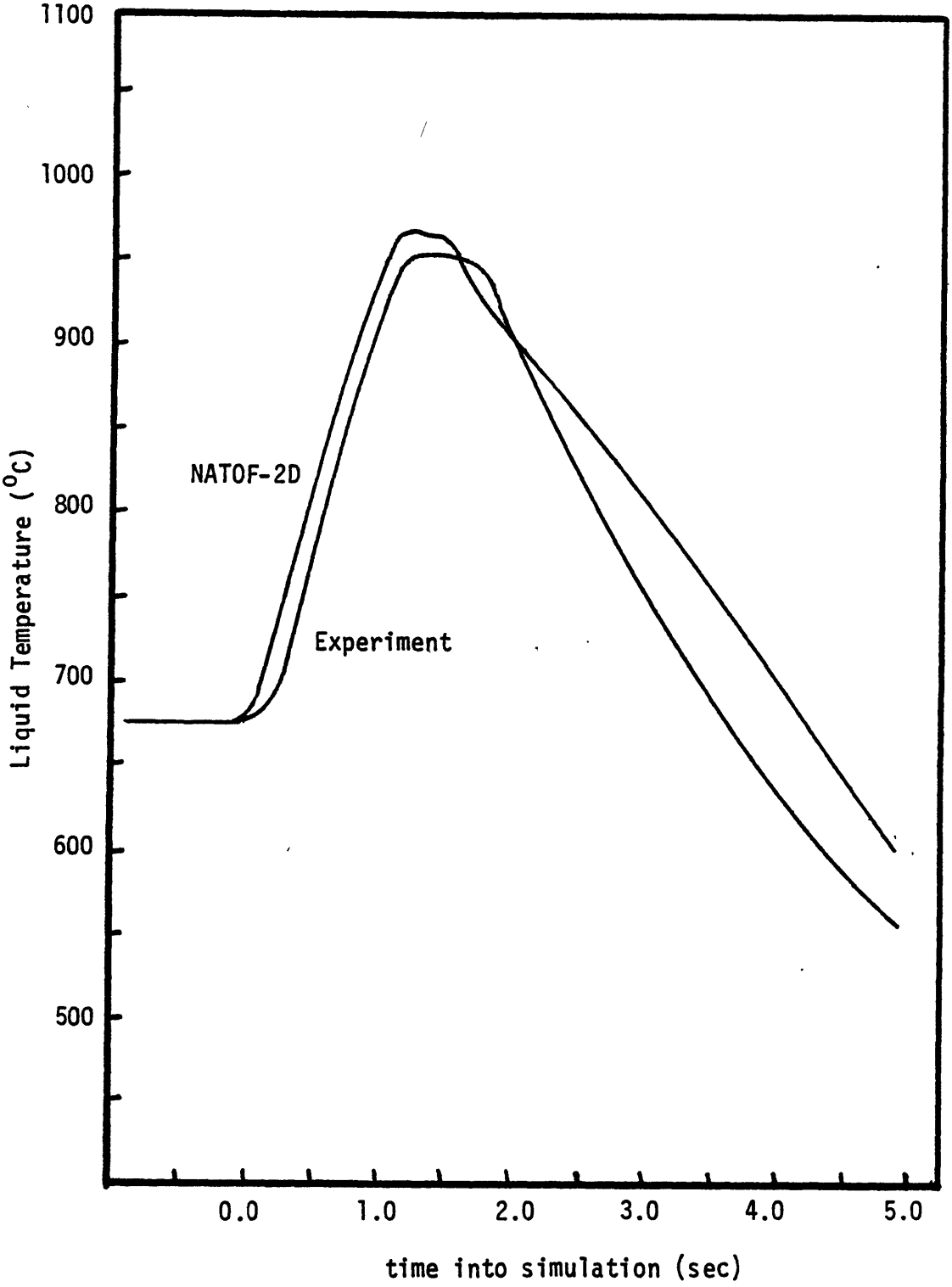


Figure 6.11 A Comparison Between W-1 SLSF Test LOPI 4 and NATOF-2D Temperature Histories at the End of the Heated Zone for the Central Channel

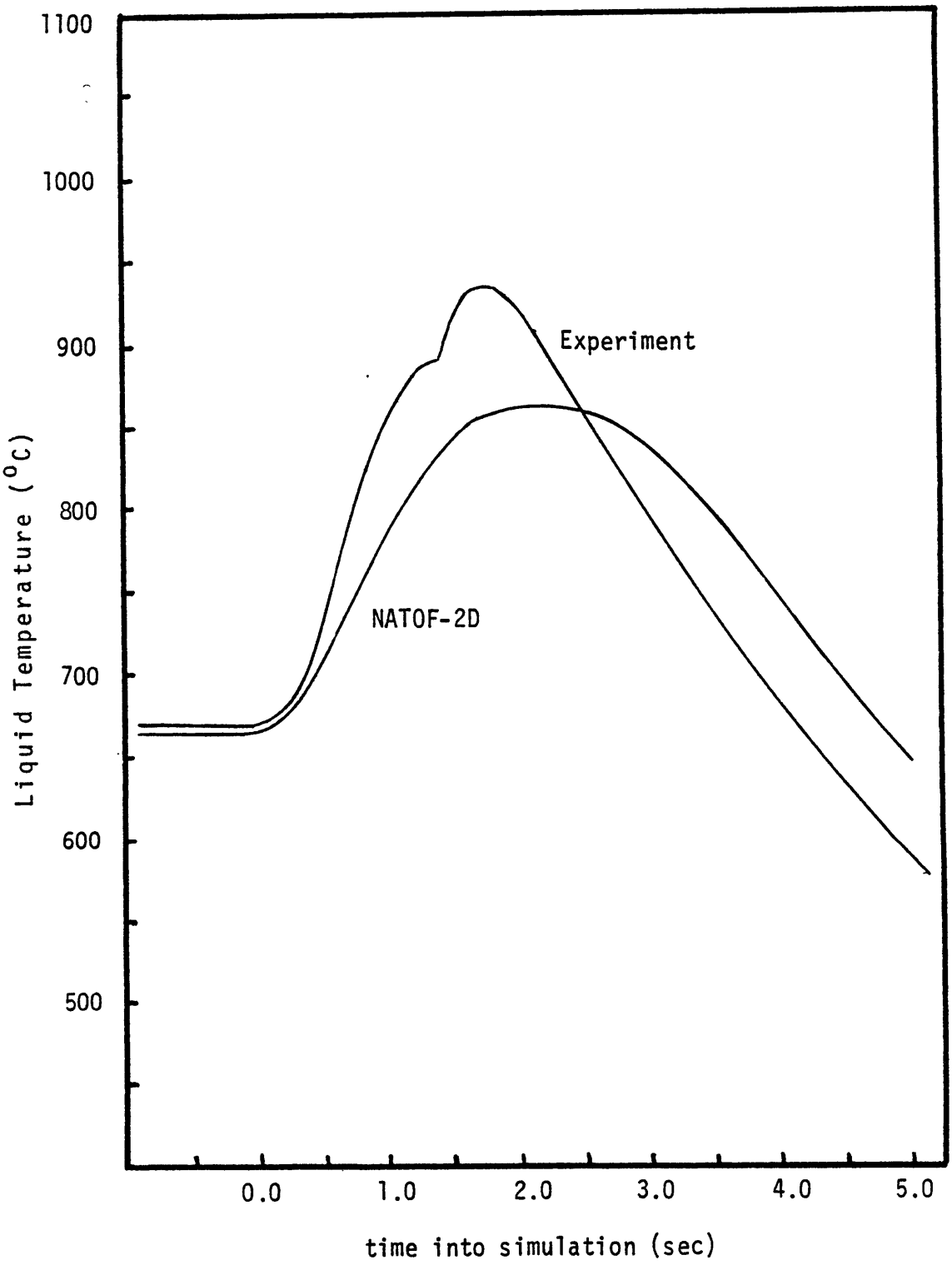
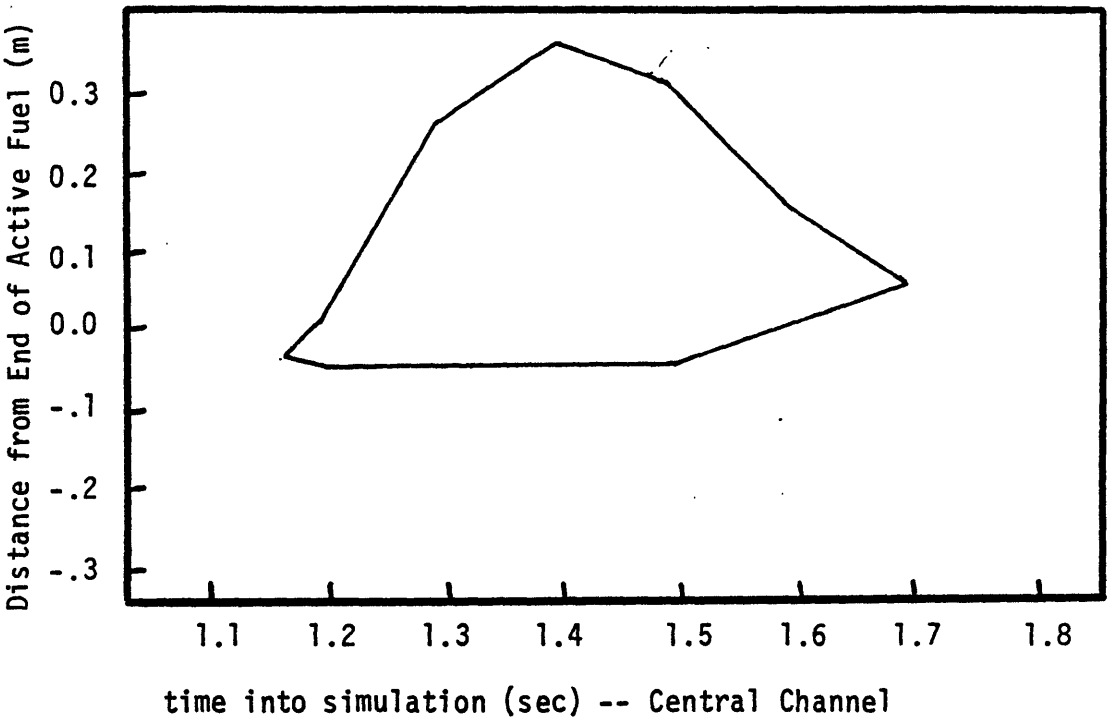


Figure 6.12 A Comparison Between W-1 SLSF Test LOPI 4 and NATOF-2D Temperature Histories at the End of the Heated Zone for the Midplane Channel



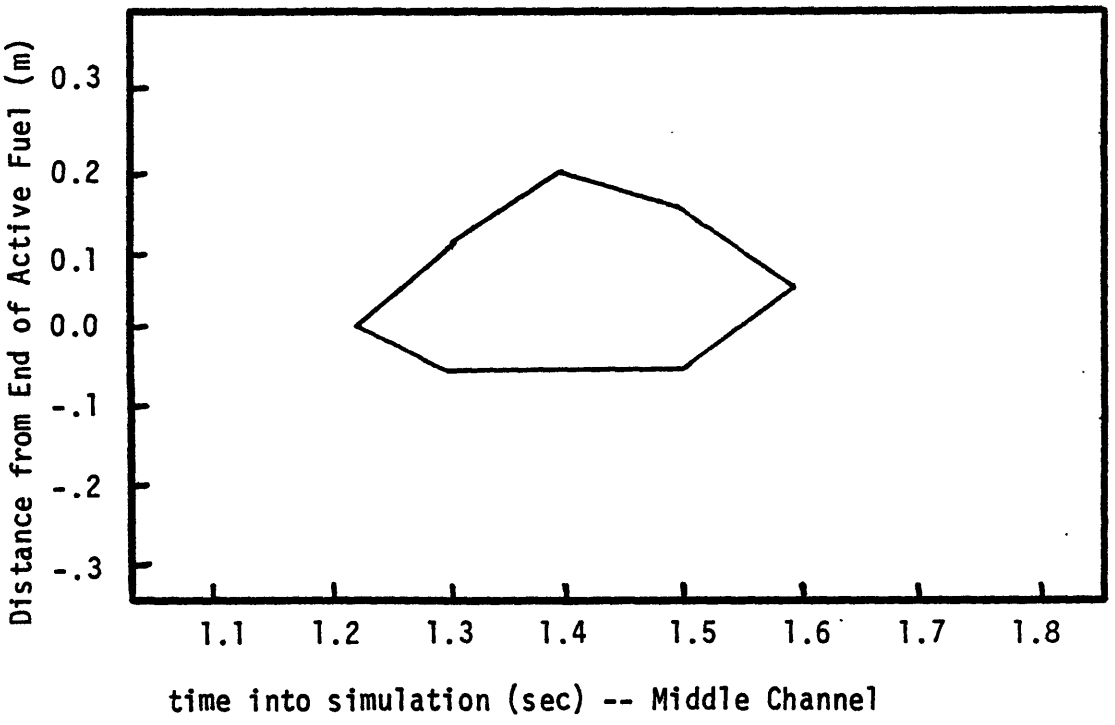


Figure 6.13 NATOF-2D Void Maps for W-1 SLSF Test LOPI 4 for the Central and Middle Channels
Void = 0.1

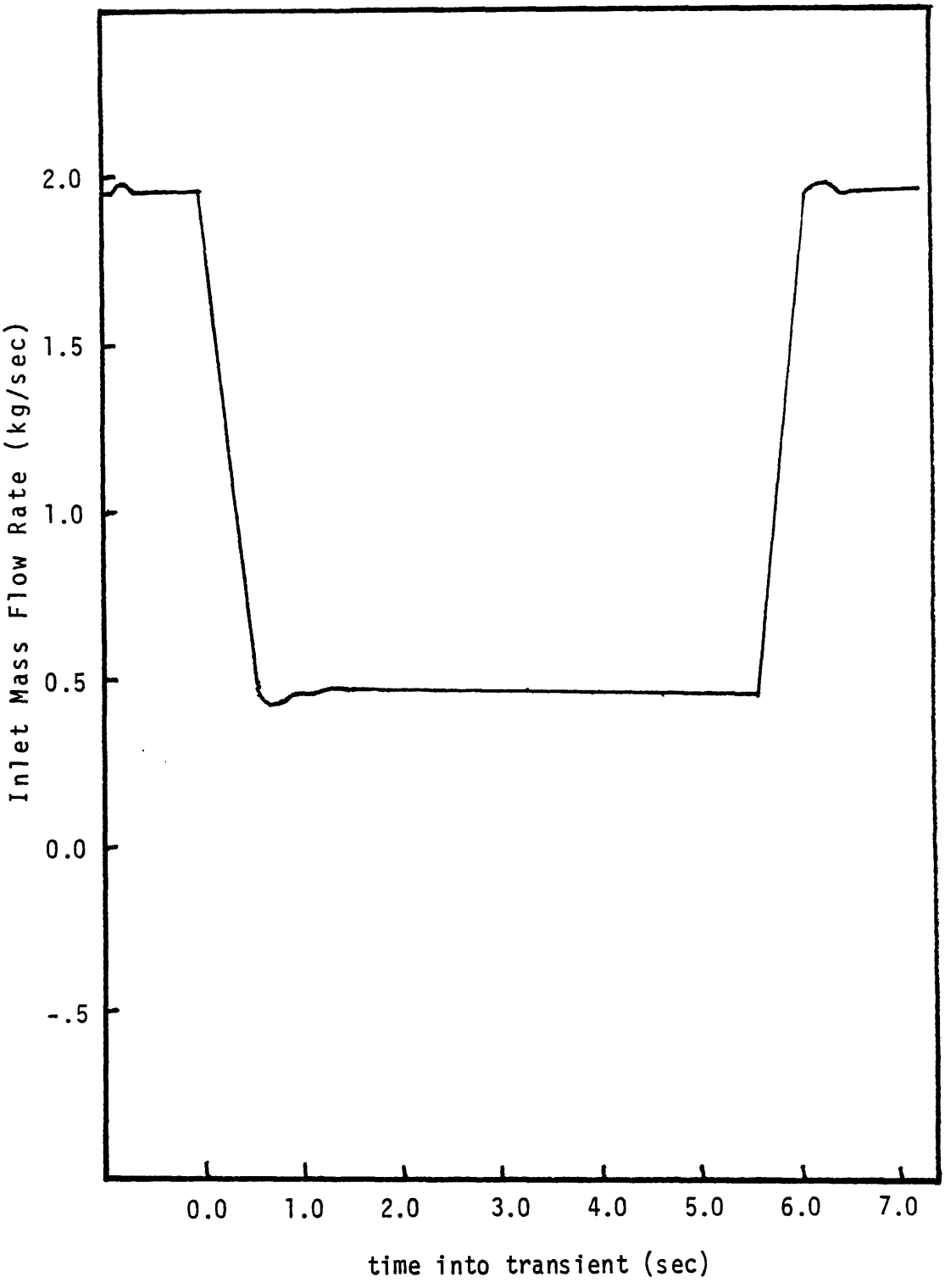


Figure 6.14 W-1 SLSF Test BWT 2' Inlet Mass Flow Rate

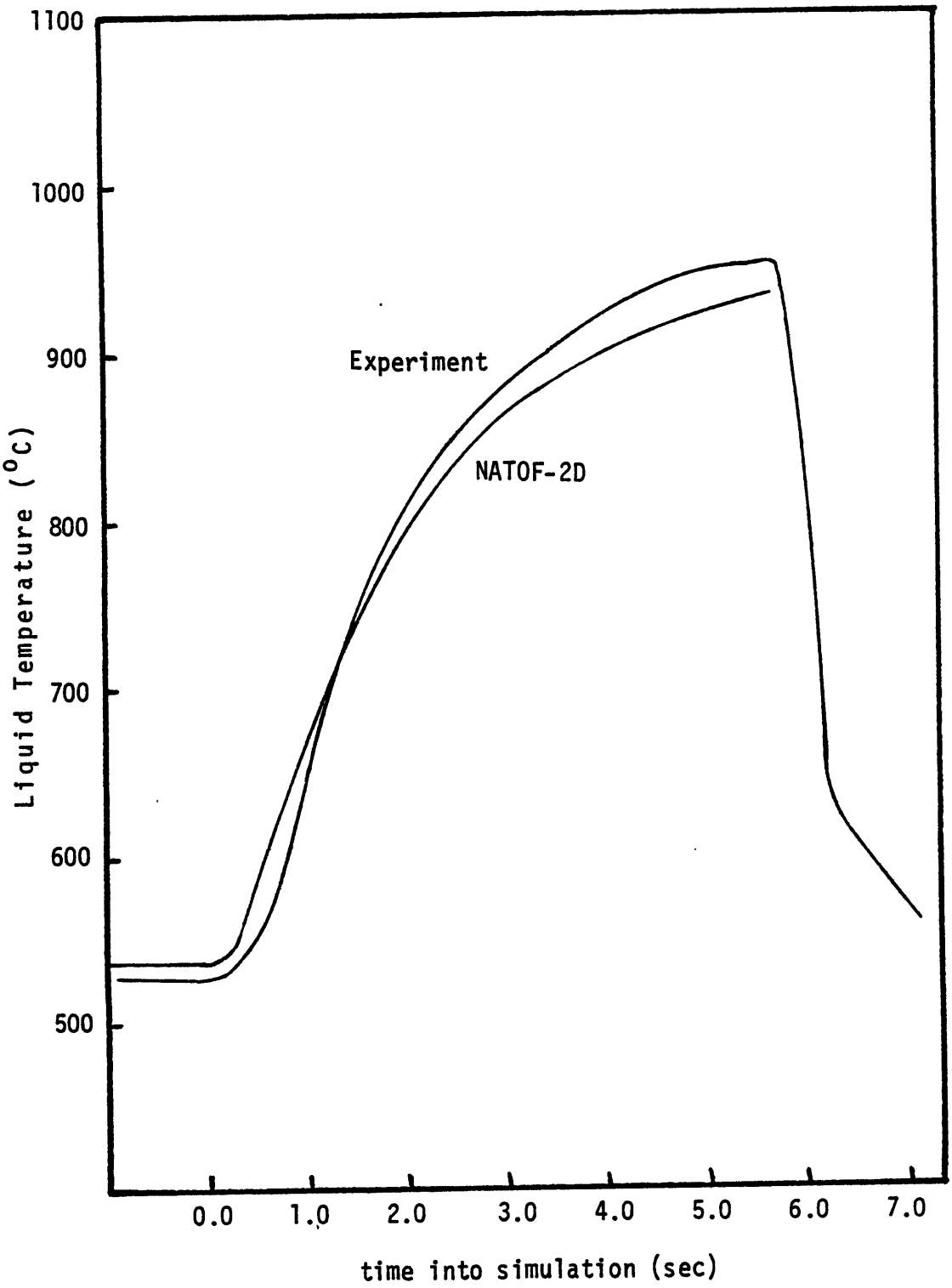


Figure 6.15 A Comparison Between W-1 SLSF Test BWT 2' and NATOF-2D Temperature Histories at the End of the Heated Zone for the Central Channel

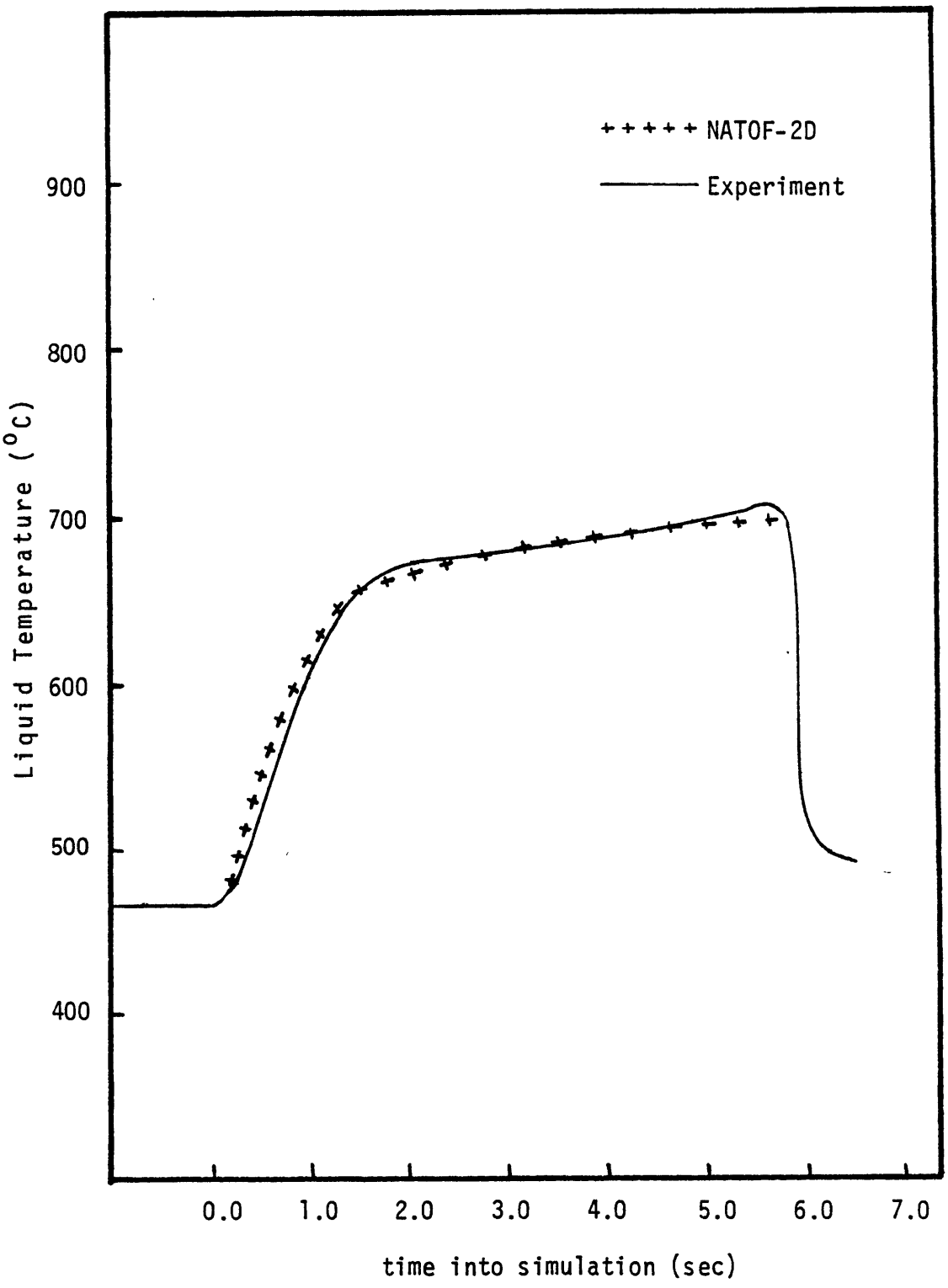


Figure 6.16 A Comparison Between W-1 SLSF Test BWT 2' and NATOF-2D Temperature Histories at the end of the Heated Zone for the Central Channel

using cells 5 cm in length resulted in axial cell to cell temperature differences of $12^{o}C$.

Further simulations of this experiment have revealed that this result is not caused by overestimating the radial fluid-to-fluid heat transfer or the hexcan heat capacity.

6.4.4 BWT 7B'

BWT 7B' was a high power boiling window transient. The simulation was carried out under velocity/pressure boundary conditions until the point of boiling inception, when pressure/pressure boundary conditions were used. When the code reached the dryout point, at which time the switch is made in interfacial mass and energy exchange correlations, the code reduced to extremely small timesteps (10^{-7} sec) which were below an acceptable level.

The reason for this can again be traced to the problems of boiling and condensation. Even at high void fractions, the liquid phase is often subcooled to a large extent. Thus the switch in correlations requires drastic changes in temperature. The vapor phase must act as the source of this heat, and since it obeys the perfect gas law, large pressure changes are necessary.

In order to obtain results, the convergence criteria was relaxed ($\delta P = 10^{-2}$). It was found that with this large convergence criteria it was possible to set CF = 1.0 and run

for 0.7 seconds during boiling, until small timesteps were required. The simulation offers some interesting observations when compared to the same case which was run with CF = 0.01.

For CF = 0.01, flow reversal occurred at 0.25 seconds after boiling inception, while for CF = 1.0, flow reversal didn't occur until 0.45 seconds. Figure 7.17 compares the experiment's flow rate to that predicted by NATOF-2D (for CF = 1.0). A comparison of void maps is given in figures 6.18 and 6.19 for the central and edge channels. As can be seen the large condensation rate keeps the void centralized.

Figure 6.20 compares the temperatures at the end of the heated zone for the central channel.

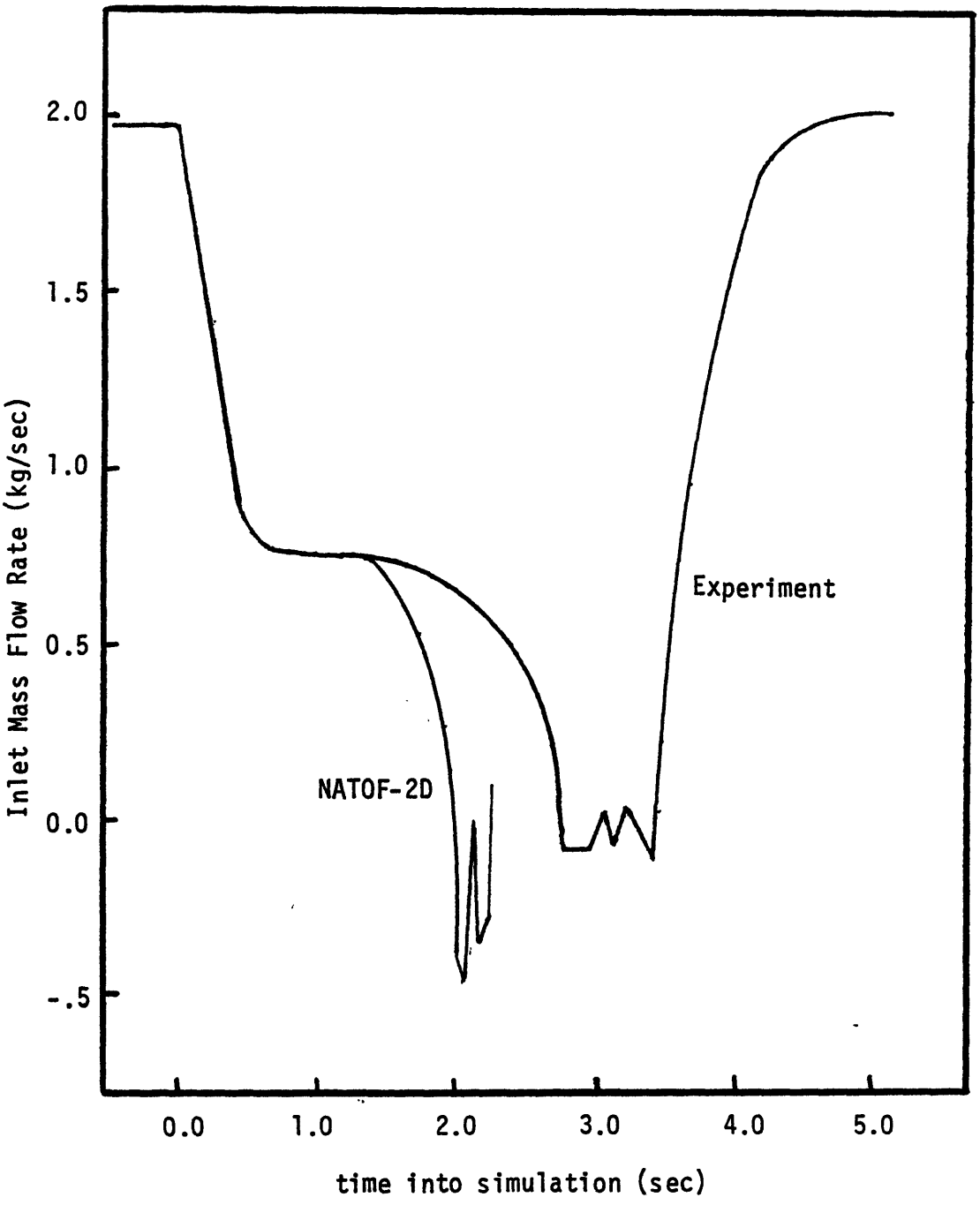


Figure 6.17 A Comparison Between W-1 SLSF BWT 7B' and NATOF-2D Predicted Inlet Mass Flow Rates

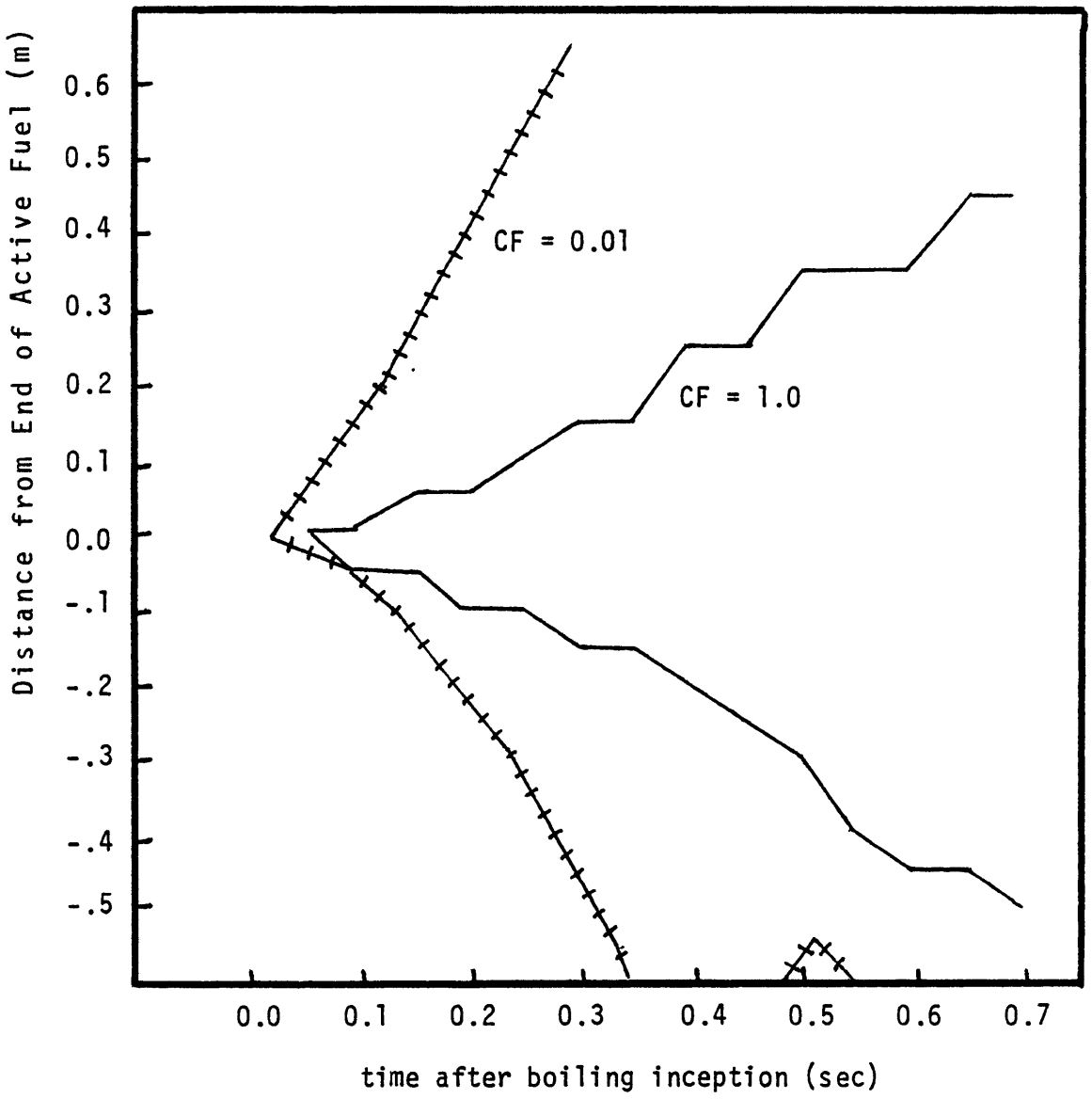


Figure 6.18 A Comparison of NATOF-2D Central Channel Void Maps for CF = 1.0 and CF = 0.01 -- Void = 0.9 (W-1 SLSF Test BWT 7B')

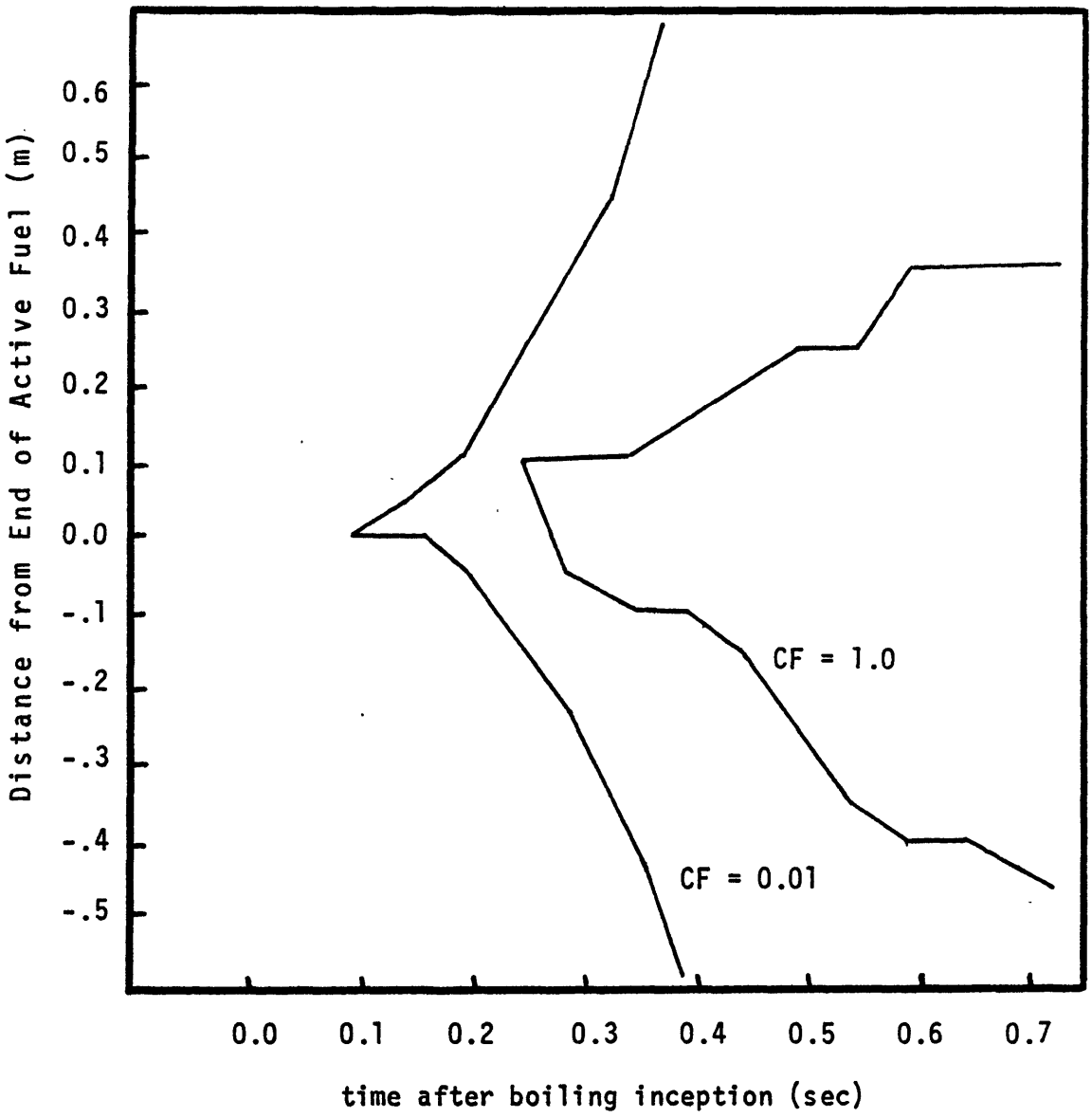


Figure 6.19 A Comparison of NATOF-2D Edge Channel Void Maps for CF = 1.0 and CF = 0.01 -- Void = 0.9 (W-1 SLSF Test BWT 7B')

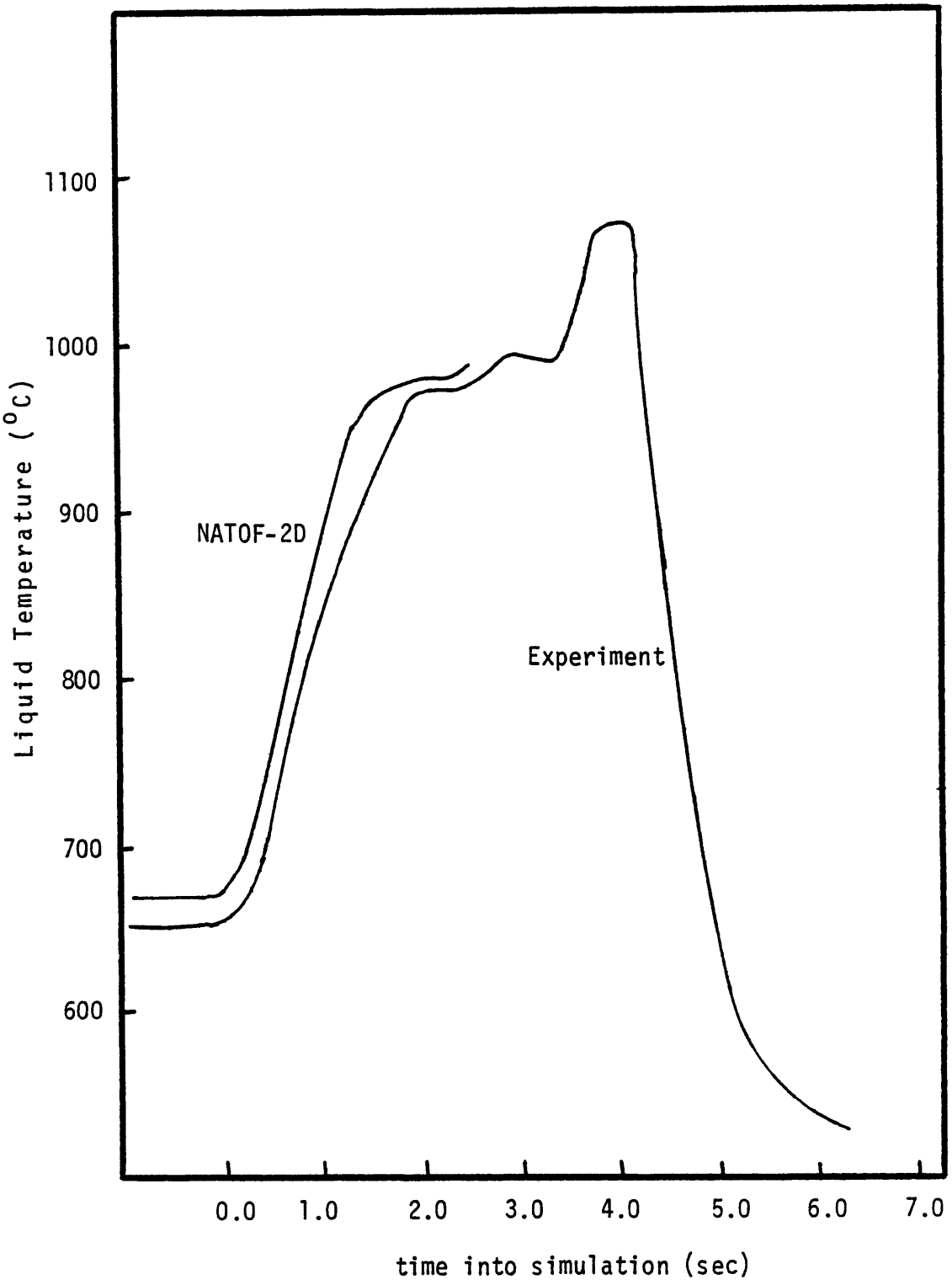


Figure 6.20 A Comparison Between NATOF-2D and W-1 SLSF BWT 7B' Temperature Histories at the End of Heated Zone for the Central Channel

Chapter 7
CONCLUSIONS AND RECOMMENDATIONS

7.1 Conclusions

The models implemented in NATOF-2D performed well during the simulations described in Chapter 6. In particular, the fluid heat conduction model gave a much improved temperature distribution comparable to the gradients found in the experiments. The direct solution technique allowed the simulations to be performed in a more detailed fashion, and within the limits imposed by computational costs.

Little more can be said about the interfacial momentum and energy exchange rates than what is described in Chapter 2. The effects of these terms are difficult to correlate with experimental results, since two-phase flow interactions are almost never directly measured. However, the sodium boiling simulations showed that the models gave physically reasonable results.

Problems were encountered during boiling in the test simulations, and most of these were covered in Chapter 5. For this reason, it is difficult to judge the effect of the mass exchange model of the code. However, it was possible to achieve full condensation with the model for a relaxed convergence criteria.

The smaller the axial mesh cell spacing in the simulation, the more confined is the voided areas of the core, and better condensation is achieved. Since the smaller volume of the cell leads to reduced pressure drop changes upon boiling, increased convergence could be attained.

7.2 Recommendations

NATOF-2D, in its present form. is an adequete model for single phase calculations. However, before extensive use of the code can occur, it will be necessary to overcome the difficulties of boiling and condensation. It is a requirement to have the evaporation and condensation rates be well behaved, since they effect the void fraction, flow reversal and heat transfer from the fuel pins. Once this has occurred, NATOF-2D can be a valuable tool in LMFBR accident analysis.

Throughout the present effort, there has persisted the problem of modelling the transition between the vapor and liquid phases without resorting to very small timesteps. Initially, the focus of this work had been the development of constituative relations in the belief that this problem could be overcome. However, continued work has shown that it is not the properties of the constituative relations which cause the difficulties, but rather the properties of the transients being simulated. The numerical scheme utilized by NATOF-2D, particularly the Newton Iteration method, is extremely powerful for well behaved functions. However, the properties of sodium and LMFBRs have severely tested the limits of the method, and have posed a choice: either small timesteps must be taken to achieve full condensation, or else the relaxed convergence criteria associated with larger timesteps must be accepted.

The results of this work demonstrate the need for improving the capability of the iteration scheme. Rather than the simple reduction of timestep sizes, it would be cost effective to develop a method which is able to solve for the unknowns of the problem at each timestep, even if more iterations than before are required. Of particular value in this line of development would be to use the semi-implicit nature of the calculation to cause the derivatives of the problem to be well behaved.

The NATOF-2D simulation of THORS Bundle 6A Test 71H, Run 101 demonstrated the need for improving the model for heat losses to the hexcan, and for including heat losses to the environment.

Another area for future work would be the development of a mechanism for assuming a temperature gradient in the cell, replacing the assumed flat profile presently used in the code. This would be necessary to achieve greater detail in the simulation results since further decreases in the axial mesh spacing beyond those used in the simulations are not possible due to computational constraints. An assumed gradient would allow boiling inception to occur in a more localized manner, reducing the effects of full cell boiling.

References

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

2. Engel, F.C., Markley, R.A., and Minushkin, B. "The Effect of Heat Input Patterns on Temperature Distributions in LMFBR Blanket Assemblies," Westinghouse Electric Corp. Advanced Reactor Division, Madison, Pennsylvania (1978)

3. Yeung.M.R., "A Multicell Slug Flow Heat Transfer Analysis for Finite LMFBR Bundles," PhD. Thesis, Department of Nuclear Engineering, MIT, (1978)

4. Ribando, R.J., et al.,"Sodium boiling in a Full-Length 19-Pin Simulated Fuel Assembly (THORS Bundle 6A)", ORNL/TM-6553 (1979)

5. Rothrock, R.B. and Henderson, J.M.,"Description and Results of the W-1 SLSF Experiment Fuel Pin Heat Release Characteristics and Sodium Boiling Stability", Hanford Engineering Development Laboratory, Energy Contract No. EY-76-C-14-2170

6. Schrage, R.W., A Theoretical Study of Interphase Mass Transfer, Columbia University Press, N.Y. (1953)

7. Collier, J.G., Convective Boiling and Condensation, McGraw-Hill, United Kingdom, 1972

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

9. Knight,D.D., "SLSF W-1 LOPI Experiments Preliminary Evaluation of Data," ST-TN-80015, October 1979

10. Martin,R.S., and Wilkinson, J.H., "Solution of Symmetric and Unsymmetric Band Equations and the Calculation of Eigenvectors of Band Matrices," Numerische Mathematic 9(4) 1967

11. Franklin, Joel N., Matrix Theory, Prentice Hall. Inc., Englewood Cliffs, N.J. (1968)

12. Bell, W.W., Matrices for Scientist and Engineers, Van Nostrand Reinhold Co., New York, N.Y. (1975)

13. Nero, Anthony V., A Guidebook to Nuclear Reactors, University of California Press, Berkley and Los Angeles, California (1979)

14. Weast, Robert C. et al., CRC Handbook of Chemistry and Physics, The Chemical Rubber Company, Cleveland, Ohio (1969)

Appendix A

SPECIFIED INLET VELOCITY AND MASS FLOW RATE

A.1 Introduction

In the simulation of sodium boiling transients, it is necessary to establish the proper flow rate through the bundle in order to obtain a temperature distribution which corresponds to the experimental results. Until now, this could only be accomplished by specifying the pressures at the inlet and at the outlet, and allowing the ΔP across the core to determine the mass flow rate. One difficulty associated with this method is that the pressures at the inlet and outlet are not always provided, and when provided, they often do not specify the effects of pressure drops due to valve throttling and fuel pin spacers. Another difficulty is that the flow rate is very sensitive to the ΔP across the core, and thus any small inaccuracies in the specified pressures can lead to large flow rate discrepencies. This has resulted in the use of a trial and error process to determine the necessary inlet and outlet pressures.

An alternative to this method would be to specify the velocity of the fluid or the total mass flow rate at the inlet, and infer the required inlet pressure from this. The advantage of these methods is that the proper flow rate

could always be maintained.

This appendix will describe how the relationship between pressures and fluid velocity is treated in NATOF-2D, and show how these equations can be modified in order that an inlet velocity or an inlet mass flow rate can be imposed.

A.2 Treatment of the Momentum Equation

As described in the introduction to Chapter Four, the solution scheme employed by NATOF-2D relies on only the pressures relating the interactions between cells. This reduction to a pressure field solution is accomplished by treating the velocities at a cell boundary only as a function of the pressures in the two neighboring cells in the momentum equation, and then substituting these relations into the mass and energy conservation equations. This section will describe the treatment of the momentum equations to obtain this result. Since the focus of this appendix is on the calculations of the velocity at the inlet, only the z-direction liquid and vapor momentum equations need be considered.

The time discretinized, finite difference form of the momentum equation for the vapor and liquid phases in the z-direction are:

vapor phase

$$(\alpha\rho_v)^n_{i+\frac{1}{2}j}\left[\frac{(U^{n+1}_{vz}-U^n_{vz})}{\Delta t}_{i+\frac{1}{2}j} + U_{vz\ i+\frac{1}{2}j}\frac{(\Delta_z U^n_{vz})_{i+\frac{1}{2}j}}{\Delta z} + U^n_{vr\ i+\frac{1}{2}j}\frac{(\Delta_r U_{vz})_{i+\frac{1}{2}j}}{\Delta r}\right] + \alpha^n_{i+\frac{1}{2}j}\frac{(P^{n+1}_{i+1j}-P^n_{ij})}{\Delta z_{i+\frac{1}{2}}} +$$

$$(\alpha\rho_v)^n_{i+\frac{1}{2}j}g = -(M_{wzv}+M_{v\ell z})_{i+\frac{1}{2}j} \qquad (A.1)$$

liquid phase

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

$$\left. + U_{\ell r\ i+\frac{1}{2}j}\ \frac{(\Delta_r U_{\ell z})^n}{\Delta r}_{i+\frac{1}{2}j}\right) + (1-\alpha)^n_{i+\frac{1}{2}j}\ \frac{(P^n_{i+1j} - P^n_{ij})}{\Delta z_{i+\frac{1}{2}}}$$

$$+ ((1-\alpha)\rho_\ell)^n_{i+\frac{1}{2}\ j} g = -(M_{wz\ell} - M_{\ell vz}) \qquad (A.2)$$

where the interface momentum exchange terms are given by:

$$M_{v\ell z} = (K + \eta\Gamma)^n (U_{vz} - U_{\ell z})^{n+1} \qquad (A.3)$$

$$M_{\ell vz} = (K - (1-\eta)\Gamma)^n (U_{vz} - U_{\ell z})^{n+1} \qquad (A.4)$$

and the wall friction terms are given by:

$$M_{wzv} = f^n_v \cdot U^n_{vz} \cdot U^{n+1}_{vz} \qquad (A.5)$$

$$M_{wz\ell} = f^n_\ell \cdot U^n_{\ell z} \cdot U^{n+1}_{\ell z} \qquad (A.6)$$

A detailed description of the donor cell technique used to evaluate the terms $\Delta U^n_{vz_{i+\frac{1}{2}j}}$, etc. is given in Reference 1. Since these terms are treated explicitly while the main focus of this section is the treatment of the implicit variables, the technique used need not be repeated. The locations used to evaluate the terms are shown in figure A.1.

For greater clarity, equation A.1 can be rearranged so

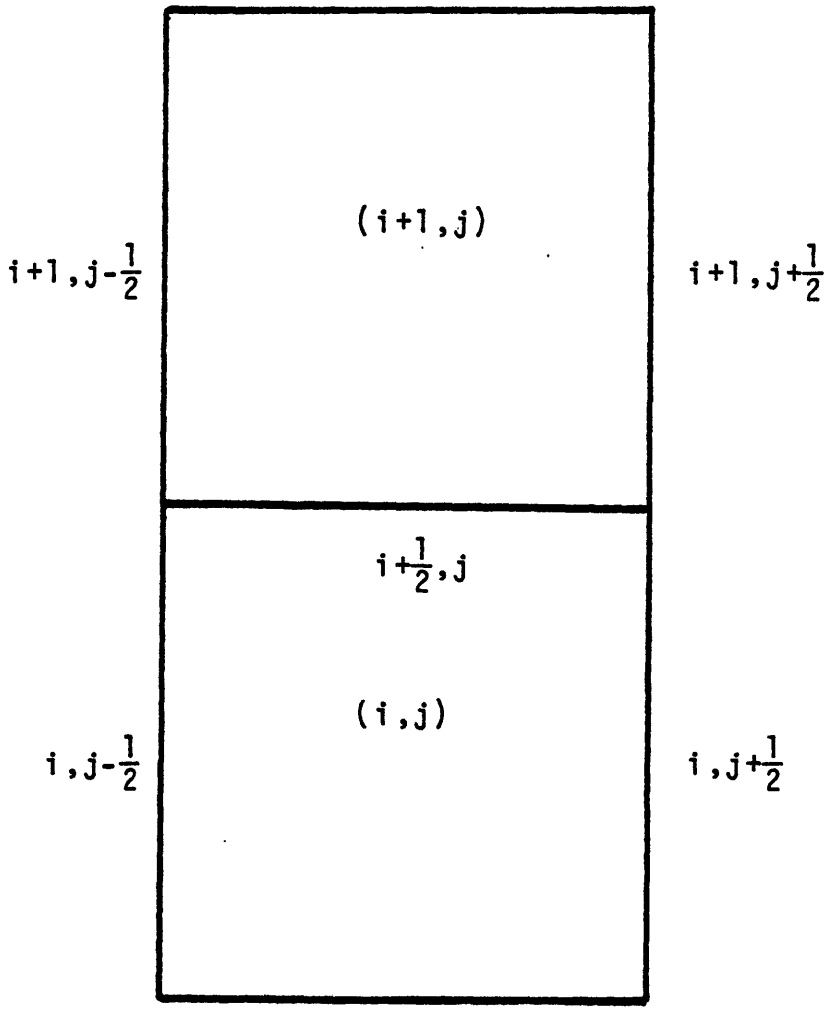


Figure A.1 Positions Used for the Evaluation of Variables

that all implicit terms appear on the left hand side. When the relations given by equations A.3 and A.5 are substituted into equation A.1 the result is

$$\left[\frac{(\alpha\rho_v)}{\Delta t} + f_v U_{vz} + (K+\eta\Gamma)\right]^{n}_{i+\frac{1}{2}j} U_{vz\ i+\frac{1}{2}j} + \alpha^{n}_{i+\frac{1}{2}j} \frac{(P^{n+1}_{i+1j} - P^{n+1}_{ij})}{\Delta z_{i+\frac{1}{2}}}$$

$$-(K + \eta\Gamma)U^{n+1}_{\ell z\ i+\frac{1}{2}j} = \text{EXPLICIT TERMS} \qquad (A.7)$$

As can be seen, the specific choice of the time step discretinization has made the velocities only dependent on the pressures. Since there is also an equation corresponding to A.7 for the liquid phase, the term $U^{n+1}_{\ell z\ i+\frac{1}{2}j}$ can be replaced with a term only dependent on U^{n+1}_{vz}, P^{n+1}_{i+1j}, and P^{n+1}_{ij}. Thus knowledge of any two variables in equation A.7 allows the calculation of the third.

The iteration scheme for the solution of the eight conservation equatons (2 mass, 2 energy, and 4 momentum) is an extension of the Newton iteration solution of algebraic equations. The equations are cast in a form similar to that of A.1 and A.2, and then a vector $\vec{X}$ of the unknowns is defined such that

$$\vec{X} = \left(\alpha,\ P,\ T\ ,\ T_v,\ U_{vz},\ U_{vr},\ U_{\ell z},\ U_{\ell r} \right)^{n+1} \qquad (A.8)$$

The finite difference equations can be written in the form

$$F_p(\vec{X}) = 0$$

where

$p = 1,2,\ldots 8$

At an iteration k, an approximate solution will be obtained. $F_p(\vec{X}^k)$ will not equal zero in general, so this is not an exact solution. A Taylor expansion around the point X^k is made to obtain

$$F_p(X^{k+1}) = F_p(X^k) + \sum_{q=1}^{8} \frac{\partial F_p}{\partial X_q}\Big|_{X_q} \cdot (X_q^{k+1} - X_q^k) \qquad (A.10)$$

If X^{k+1} is required to be the solution of A.9 then

$$\sum_{q=1}^{8} \frac{\partial F_p}{\partial X_q}\Big|_{X_q} \cdot (X_q^{k+1} - X_q^k) = -F_p(X^k) \qquad (A.11)$$

Defining $\delta X = X_q^{k+1} - X_q^k$, equation A.11 can be written explicitly for the vapor and liquid momentum equations. The result is:

vapor phase

$$\frac{(\alpha\rho_v)_{i+\frac{1}{2}j}}{\Delta t} \cdot \delta U_{vz} + \left(\frac{\alpha}{\Delta z}\right)_{i+\frac{1}{2}j} \cdot (\delta P_{i+1j} - \delta P_{ij}) + \frac{\partial M_{wzv}}{\partial U_{vz}} \cdot \delta U_{vz}$$

$$+ \frac{\partial M_{v\ell z}}{\partial U_{vz}} \cdot \delta U_{vz} + \frac{\partial M_{v\ell z}}{\partial U_{\ell z}} \cdot \delta U_{\ell z} = -F_1 \qquad (A.12)$$

liquid phase

$$\frac{((1-\alpha)\rho_\ell)_{i+\frac{1}{2}j}}{\Delta t} \cdot \delta U_{\ell z} + \left(\frac{1-\alpha}{\Delta z}\right)_{i+\frac{1}{2}j} \cdot (\delta P_{i+1j} - \delta P_{ij}) + \frac{\partial M_{wz\ell}}{\partial U_{\ell z}} \cdot \delta U_{\ell z}$$

$$- \frac{\partial M_{\ell vz}}{\partial U_{\ell z}} \cdot \delta U_{\ell z} - \frac{\partial M_{\ell vz}}{\partial U_{vz}} \cdot \delta U_{vz} = -F_2 \qquad (A.13)$$

Solving equations A.12 and A.13 for the velocities yields

$$\delta U_{vz_{i+\frac{1}{2}j}} = W_{vz_{i+\frac{1}{2}j}} (\delta P_{i+1j} - \delta P_{ij}) + R_{vz} \qquad (A.14)$$

$$\delta U_{\ell z_{i+\frac{1}{2}j}} = W_{\ell z_{i+\frac{1}{2}j}} (\delta P_{i+1j} - \delta P_{ij}) + R_{\ell z} \qquad \text{(A.15)}$$

where

$$W_{vz_{i+\frac{1}{2}j}} = \frac{-\left(\frac{\alpha}{\Delta z}\left(\frac{(1-\alpha)\rho_\ell}{\Delta t} + \frac{\partial M_{wz\ell}}{\partial U_{\ell z}} - \frac{\partial M_{\ell vz}}{\partial U_{\ell z}}\right) - \frac{\partial M_{v\ell z}}{\partial U_{\ell z}} \cdot \frac{(1-\alpha)}{\Delta z}\right)}{\left(\left(\frac{\alpha\rho_v}{\Delta t} + \frac{\partial M_{wzv}}{\partial U_{vz}} + \frac{\partial M_{v\ell z}}{\partial U_{vz}}\right)\left(\frac{(1-\alpha)\rho_\ell}{\Delta t} + \frac{\partial M_{wz\ell}}{\partial U_{\ell z}} + \frac{\partial M_{\ell vz}}{\partial U_{\ell z}}\right) + \frac{\partial M_{v\ell z}}{\partial U_{\ell z}}\frac{\partial M_{\ell vz}}{\partial U_{vz}}\right)}$$

$$R_{vz} = \frac{F_1 \cdot \left(\frac{(1-\alpha)\rho_\ell}{\Delta t} + \frac{\partial M_{wz\ell}}{\partial U_{\ell z}} + \frac{\partial M_{\ell vz}}{\partial U_{\ell z}}\right) - F_2 \cdot \frac{\partial M_{v\ell z}}{\partial U_{\ell z}}}{\left(\left(\frac{\alpha\rho_v}{\Delta t} + \frac{\partial M_{wzv}}{\partial U_{vz}} + \frac{\partial M_{v\ell z}}{\partial U_{vz}}\right)\left(\frac{(1-\alpha)\rho_\ell}{\Delta t} + \frac{\partial M_{wz\ell}}{\partial U_{\ell z}} + \frac{\partial M_{\ell vz}}{\partial U_{\ell z}}\right) + \frac{\partial M_{v\ell z}}{\partial U_{\ell z}}\frac{\partial M_{\ell vz}}{\partial U_{vz}}\right)}$$

and similarly for the liquid phase.

Equation A.14 and A.15 are the form of the momentum equations which will be used in the following two sections to describe the inlet velocity and mass flow rate boundary conditions.

A.3 Inlet Velocity Boundary Condition

An inlet velocity boundary condition refers to a user defined fluid velocity, given as a function of time, which is constant across the bundle inlet. Only the inlet velocity and the outlet pressure need to be specified, since the inlet pressure no longer enters into the calculation. With these two parameters, the iteration scheme can calculate the pressure field distribution. After the iteration is performed, the inlet pressure necessary to generate the specified velocity can be inferred. However, since a constant velocity across the bundle is assumed, in general there will be a different inlet pressure for each cell.

The inlet velocity condition is imposed by setting $W_{vz\ i+\frac{1}{2}j}$ and $W_{\ell z\ i+\frac{1}{2}j}$ in equation A.14 and A.15 to zero, and then setting

$$R_{vz} = U_{inlet} - U_{vz\ 1+\frac{1}{2}j} \tag{A.16}$$

$$R_{\ell z} = U_{inlet} - U_{\ell z\ 1+\frac{1}{2}j} \tag{A.17}$$

Since R_{vz} and $R_{\ell z}$ represent the error term from the previous iteration, one can see that as R_{vz} and $R_{\ell z}$ go to zero, the Newton iteration converges on the exact solution. Equation A.7 is then used to update the boundary pressure, P_{1j}, since $P_{2,j}$, $U_{vz\ 1+\frac{1}{2}j}$, and $U_{\ell z\ 1+\frac{1}{2}j}$ are known.

The inlet velocity boundary condition calculation is

simple to perform, but it has the disadvantage of preventing localized flow reversal during sodium boiling transients. The second method, inlet mass flow rate, does not have this restriction.

A.4 Specified Inlet Mass Flow Rate Boundary Condition

The method for specifying the inlet mass flow rate was developed by Andrei L. Schor at the Massachusetts Institute of Technology. For the cell numbering scheme shown in figure A.2, equations A.1 and A.2 can be written for the bottom row of real cells in the form:

$$U_{\ell j} = a_{\ell j} \cdot (P_j - P_0) + r_{\ell j} \tag{A.18}$$

$$U_{vj} = a_{vj} \cdot (P_j - P_0) + r_{vj} \tag{A.19}$$

$$j = 1,..nj$$

The mass flow rate of each phase into the cell is given by:

$$W_{\ell j} = A_j \cdot (1 - \alpha) \rho_\ell U_{\ell j} \tag{A.20}$$

$$W_{vj} = A_j \cdot \alpha \cdot \rho_v U_{vj} \tag{A.21}$$

where

$W_{\ell j}$ = liquid mass flow rate into cell j

W_{vj} = vapor mass flow rate into cell j

A_j = flow area of cell j

The total mass flow rate into the cell, W_j, is:

$$W_j = W_{\ell j} + W_{vj} \tag{A.22}$$

$$W_j = A_j(1 - \alpha)\rho_\ell(a_{\ell j}(P_j - P_0) + r_{\ell j}) + A_j \alpha \rho_v (a_{vj}(P_j - P_0) + r_{vj}) \tag{A.23}$$

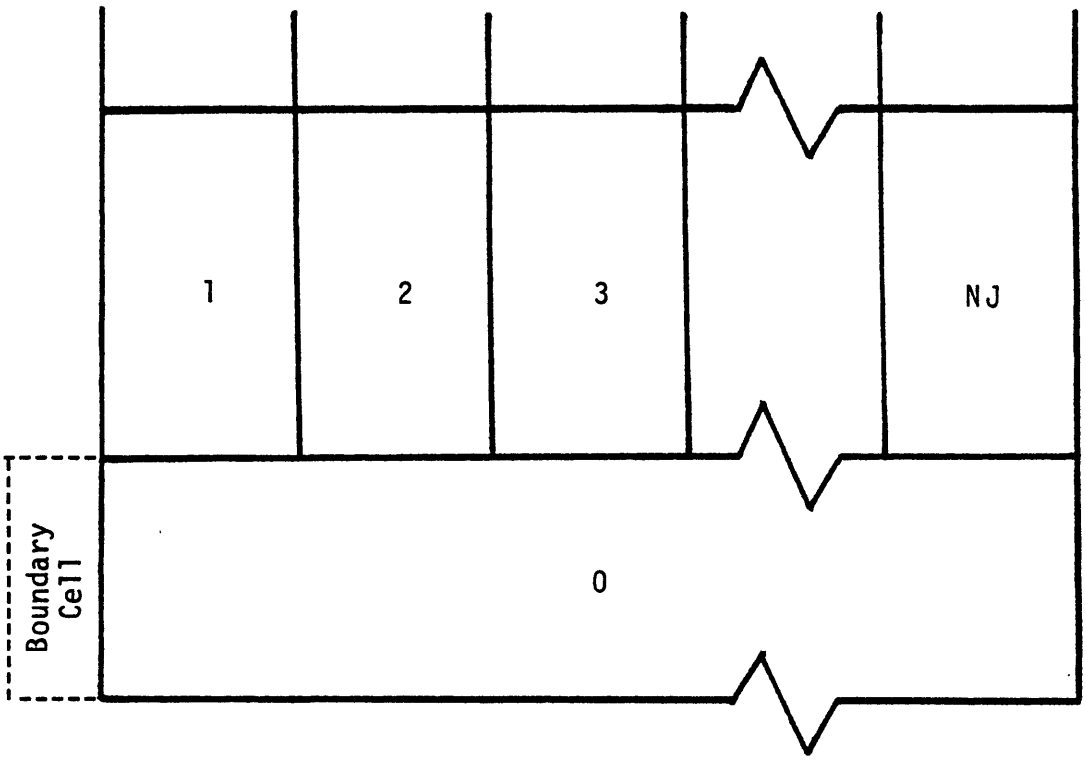


Figure A.2 Example Cell Numbering Scheme for Flow Boundary Calculation

$$W_j = A_j((1-\alpha)\rho_\ell a_{\ell j} + \alpha\rho_v a_{vj})(P_j - P_0) + A_j((1-\alpha)\rho_\ell r_{\ell j} + \alpha\rho_v r_{vj}) \quad (A.24)$$

And the total mass flow rate into the bundle then becomes:

$$W_T = \sum_{j=1}^{nj} W_j \quad (A.26)$$

Defining

$$a_j' = A_j((1-\alpha)\rho_\ell a_{\ell j} + \alpha\rho_v a_{vj}) \quad (A.27)$$

$$r_j' = A_j((1-\alpha)\rho_\ell r_{\ell j} + \alpha\rho_v r_{vj}) \quad (A.28)$$

W_T is then given by:

$$W_T = \sum_{j=1}^{nj} a_j'(P_j - P_0) + \sum_{j=1}^{nj} r_j \quad (A.29)$$

$$= \sum_{j=1}^{nj} a_j' P_j - P_0 \sum_{j=1}^{nj} a_j' + \sum_{j=1}^{nj} r_j \quad (A.30)$$

Rearranging equation A.30 yields

$$\left(-\sum_{j=1}^{nj} a_j'\right) P_0 + a_1' P_1 + \ldots + a_{nj}' P_{nj} = W_T - \sum_{j=1}^{nj} r_j' \quad (A.31)$$

When written in incremental form, equation A.31 becomes

$$\left(-\sum_{j=1}^{nj} a_j'\right)\delta P_0 + a_1'\delta P_1 + a_2'\delta P_2 + \ldots a_{nj}'\delta P_{nj} = 0 \quad (A.32)$$

Thus, for the specified mass flow rate boundary condition, there will be an additional pressure field equation to solve. This equation is added to the pressure field matrix described in Chapter 4. For the configuration of cells shown in figure A.3, the resulting matrix is shown in figure A.4.

The coefficients c_{01}, c_{02} and c_{03} are given by equation

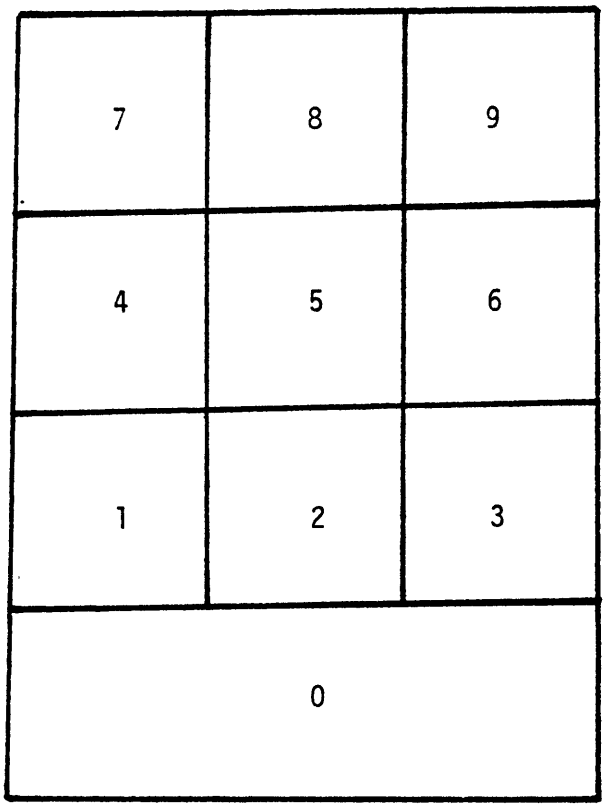


Figure A.3 Cell Configuration for Matrix Shown in Figure A.4

$$
\begin{vmatrix}
1 & C_{01} & C_{02} & C_{03} & 0 & 0 & 0 & 0 & 0 & 0 \\
C_{10} & a_{1,1} & a_{1,2} & 0 & a_{1,4} & 0 & 0 & 0 & 0 & 0 \\
C_{20} & a_{2,1} & a_{2,2} & a_{2,3} & 0 & a_{2,5} & 0 & 0 & 0 & 0 \\
C_{30} & 0 & a_{3,2} & a_{3,3} & 0 & 0 & a_{3,6} & 0 & 0 & 0 \\
0 & a_{4,1} & 0 & 0 & a_{4,4} & a_{4,5} & 0 & a_{4,7} & 0 & 0 \\
0 & 0 & a_{5,2} & 0 & a_{5,4} & a_{5,5} & a_{5,6} & 0 & a_{5,8} & 0 \\
0 & 0 & 0 & a_{6,3} & 0 & a_{6,5} & a_{6,6} & 0 & 0 & a_{6,9} \\
0 & 0 & 0 & 0 & a_{7,4} & 0 & 0 & a_{7,7} & a_{7,8} & 0 \\
0 & 0 & 0 & 0 & 0 & a_{8,5} & 0 & a_{8,7} & a_{8,8} & a_{8,9} \\
0 & 0 & 0 & 0 & 0 & 0 & a_{9,6} & 0 & a_{9,8} & a_{9.9}
\end{vmatrix}
$$

Figure A.4 Pressure Field Matrix with Flow Boundary Calculation

A.8. The coefficients c_{10}, c_{20} and c_{30} are the coefficients of the momentum equation at the boundary. Previously, these terms were not used, since the boundary pressure was constant during a timestep. With the flow boundary condition, however, these terms are used to relate the pressure in the boundary cell to the bottom row of real cells, and in this way the boundary cell pressure can be updated.

Note that the bandwidth of the matrix remains the same, and therefore the additional cpu requirements are negligible.

This method offers an advantage over a specified inlet velocity, since it allows the boundary pressure to adjust itself to the conditons prevalent in the bundle. Thus, in principle, it is possible to have flow reversal in some channels, while still maintaining a net positive flow.

A.5 Programming Information

The boundary condition at the inlet is specified by setting the input parameter, itbd, to -1, 0 or 1 to indicate a velocity, pressure, or flow boundary condition.

For the velocity boundary condition, the velocity at the inlet is input, and the velocities are updated in subroutine BC. These values are passed to subroutine ONESTP, where the differnece between U_b and U_j (i = 1,nj) is used to calcualte ΔU, rather than ΔP in the boundary cell momentum equations.

For the flow boundary condition, the desired flow input for the timestep is used along with the pressures of the previous timestep to get a first estimate for P_0 in subroutine BC (that is, solving equation A.31 for P_0). The new P_0 is passed to subroutine ONESTP, where it is treated the same as a pressure boundary condition. The difference is that δP_0 is calculated in subroutine DIRECT, and added to P_0 to get a better estimate. This is continued until convergence is attained. This option is only available when the direct method pressure field solution is used.

Appendix B

****
********** N A T O F - 2 D - INPUT DESCRIPTION **********
****

SECTION I

The following cards are read via namelist input. A total of four namelists are used: "restrt", "unos", "duos", and "tres".

The input should look like:

$"namelist" f1,f2,f3,...,fn,$end

for each namelist, where each fI is a field consisting of:

all blanks, or
name = constant, or
name = list of constants.

The order of input is immaterial; as many cards as needed may be used; the $end signifying the end of the namelist input should appear only on the last card, for each namelist.

For additional details on the use of namelist input, the user is referred to a standard fortran manual.

Group No.	Format	Contents
1	namelist	$restrt,nres ,$end
		nres = steady state or transient indicator (1/0)
2	namelist	$unos ,ni ,nj ,ncf ,ncld ,itm1 , igauss,dtmax ,dtmin ,eps1 ,eps2 ,nset , tset ,indgs ,sprint,itbd ,$end
		ni = number of mesh cells in the axial direction
		nj = number of mesh cells in the radial direction
		ncf = number of mesh cells in the fuel
		ncld = number of mesh cells in the clad
		itm1 = maximum number of iterations in the Newton iterative solution
		igauss = maximum number of iterations in the pressure problem solution (only necessary when indgs = 1)

indgs = indicator for direct or gausian solution (0/1)
dtmax = maximum value of the time step increment
dtmin = minimum time step increment
eps1 = convergence criterion for the Newton iteration (N/m-2)
eps2 = convergence criterion for the pressure problem (N/m-2)
sprint = time interval between monitoring prints
itbd = inlet boundary condition number
-1 = velocity boundary condition
0 = pressure boundary condition
1 = flow boundary condition

The following two corresponding cards can be incremented from i = 1 to 40, and control the printed output. The code will print nset times the flow map at an increment of tset.

nset(i) = number of printouts
tset(i) = time between printouts

3 namelist

```
$duos  ,nrow  ,pitch ,d     ,e     ,ad    ,
 apu   ,dil   ,radr  ,thc   ,thg   ,lss   ,
 tinit ,ntcd  ,lp    ,rnusl1,rnusl2,alpdry,
 $end
```

nrow = number of rows of fuel pins in the fuel assembly
pitch = distance between fuel pin centerlines (m)
d = diameter of the fuel pin (m)
e = minimum distance between fuel pin surface and hex can wall (m)
ad = fraction of theoretical density of fuel
apu = fraction of plutonium in the fuel
dil = fraction of helium gas in gas compositon
radr = fuel pin outside radius (m)
thc = clad thickness (m)
thg = gap thickness (m)
lss = transient or steady state indicator (0/1)
tinit = initial starting time (sec)
ntcd = number of boundary condition cards
lp = partial or full boundary calculation (0/1)
rnusl1 = effective nusselt number for radial

heat conduction -- inner cells
0 > implicit calculation
0 = bypass calculation
0 < explicit calculation

rnusl2 = effective nusselt number for radial heat conduction -- edge cells

alpdry = dryout void fraction

The following cards are required only for an initial start, and appear at the very end of the input file.

3 namelist $tres ,pin ,pout ,tin ,tav ,$end

pin = initial inlet pressure to the fuel assembly (N/m2)

pout = initial outlet pressure to the fuel assembly (N/m2)

tin = initial inlet temperature of the coolant (K)

tav = average temperature of the fuel assembly (K)

SECTION II

The following cards are read via NIPS free-format input processor. Fields are separated by blanks. Entry (or group of entries) repetition is allowed; for example n(a b m(c d e) f) where: a,b,c,d,e,f are entries (integer or real) and n,m are integers representing the number of repetitions; note that no blanks must appear between a left parenthesis and the integer preceding it. Up to 10 levels of nesting are permitted.

The end of a group is marked by a $-sign.

The following cards govern the boundary conditions of the problem as a function of time. These cards are always required. The order of input must be maintained.
Those cards marked with a * are necessary only for a full boundary calculation (lp = 1)

For a partial boundary calculation (lp = 0), the boundary is calculated as follows:

$$X = X1(L)*dtime + X2(L)$$

and for a full boundary calculation:

$$X = (X1(L)*dtime + X2(L))*exp(OMX(L)*dtime) + X3(L)$$

where

dtime = time - tb(L-1)
L = index of currect time segment
tb(L) = time at the end of segment L
X1, X2, X3, OMX = input parameters

1 tb(ntcd) = time at the end of a time segment

2 bnb1(ntcd) = velocity at inlet (m/s) (itbd = -1)
= pressure at bottom of fuel assembly (N/m2) (itbd = 0)
= flow at inlet (kg/s) (itbd = 1)
bnb2(ntcd) =
* bnb3(ntcd) =
* omb(ntcd) =

3 pnt1(ntcd) = pressure at top of fuel assembly (N/m2)
pnt2(ntcd) =
* pnt3(ntcd) =
* omt(ntcd) =

4 alb1(ntcd) = void fraction at the inlet of the fuel assembly
alb2(ntcd) =
* alb3(ntcd) =
* oma(ntcd) =

5 tvb1(ntcd) = vapor temperature at the inlet (K)
tvb2(ntcd) =
* tvb3(ntcd) =
* omv(ntcd) =

6 tlb1(ntcd) = liquid temperature at the inlet(K)
tlb2(ntcd) =
* tlb3(ntcd) =
* oml(ntcd) =

7 hnb1(ntcd) = power density in the fuel pins (w/m2)
hnb2(ntcd) =
* hnb3(ntcd) =
* omh(ntcd) =

The following cards are always required for a steady state calculation, but not for a transient calculation. The dimensions are given by:

ni = number of axial cells
nj = number of radial cells
npin = ncf + ncld + 2
nn = ni*nj

8 n(19) = row numbers where the boundary

between cell J and cell J + 1 lies (n = 2,...)

9 dz(ni) = axial mesh spacing of cells (m)
10 tcan(ni) = heat capacity of hex can per unit area
11 shape(nn) = power density shape in fuel assembly
12 sppd(nn) = spacer pressure drop
13 ppp(npin) = radial power profile inside fuel pin (fuel, gap and clad)
14 lplnm(ni) = axial composition of the fuel pin
0 = gas compositon
1 = mixed oxide U,PuO2

RESTART OPTION

For a restart of a previous calculation, the namelists restrt,unos and duos are required, for a selected number of cards.

For namelist restrt, the following previously defined card is required:

nres = 0

For namelist unos, the following previously defined cards are required:

nset(i)=
tset(i)=

The following cards are optional inputs:

eps1 ,eps2 ,dtmax ,igauss,itm1 ,
dtmin ,sprint,itbd ,indgs

For namelist duos, the following previously defined cards are required

lss = 0
tinit =
ntcd =
lp =
rnusl1 =
rnusl2 =
alpdry =

Boundary condition cards are always required.

* * *

As of September 1, 1981

Appendix C.1: Westinghouse Blanket Heat Transfer Test Program Run 544

```
$restrt
ntype  = 0
nres   = 1
$end
$unos
sprint = 0.1
ni     = 14
nj     = 5
ncf    = 4
ncld   = 2
itm1   = 8
indgs  = 0
igauss = 1
dtmax  = 1.0
dtmin  = 1.e-8
eps1   = 10.
eps2   = 0.001
nset   = 1
tset   = 3.0
$end
$duos
rnusl1 = 22.0
rnusl2 = 28.0
tsr    = 0.0
nrow   = 5
pitch  = 1.4216e-2
d      = 1.32e-2
e      = 8.60e-4
ad     = 0.95
apu    = 0.0
dil    = 0.9
radr   = 6.60e-3
thc    = 8.61e-4
thg    = 1.36e-4
lss    = 1
tinit  = 0.0
ntcd   = 1
lp     = 0
$end
3.5                                                    $tb
.0                                                     $pnb1
1.888e5                                                $pnb2
0.0                                                    $pnt1
1.5e5                                                  $pnt2
0.0                                                    $alb1
0.0                                                    $alb2
0.0                                                    $tvb1
589.14                                                 $tvb2
0.0                                                    $tlb1
589.14                                                 $tlb2
0.0                                                    $hnw1
8.930e7                                                $hnw2
2 3 4 5 15(0)                                          $n
14(.1633)                                              $dz
14(6500.)                                              $tcan
5(2(0.0) .38 .71 .92 1.0 .92 .71 .38 5(0.0))           $shape
4(0.0 61.0 12(0.0)) 0.0 89. 12(0.0)                    $sppd
4(1.0) 4(0.0)                                          $ppp
9(1) 5(0)                                              $lplmn
$tres
pin    = 1.888e5
pout   = 1.5e5
tin    = 589.14
tav    = 700.0
$end
```

Appendix C.2: Westinghouse Blanket Heat Transfer Test Program Run 545

```
$restrt
ntype  = 0
nres   = 1
$end
$unos
sprint = 0.1
ni     = 14
nj     = 5
ncf    = 4
ncld   = 2
itm1   = 8
indgs  = 0
igauss = 100
dtmax  = 1.0
dtmin  = 1.e-8
eps1   = 10.
eps2   = .001
nset   = 1
tset   = 3.0
$end
$duos
rnusl1 = -22.0
rnusl2 = -23.0
tsr    = 0.0
nrow   = 5
pitch  = 1.4216e-2
d      = 1.32e-2
e      = 8.60e-4
ad     = 0.95
apu    = 0.0
dil    = 0.9
radr   = 6.60e-3
thc    = 8.61e-4
thg    = 1.36e-4
lss    = 1
tinit  = 0.0
ntcd   = 1
lp     = 0
$end
3.5                                          $tb
0.0                                          $pnb1
1.838e5                                      $pnb2
0.0                                          $pnt1
1.5e5                                        $pnt2
0.0                                          $alb1
0.0                                          $alb2
0.0                                          $tvb1
589.14                                       $tvb2
0.0                                          $tlb1
589.14                                       $tlb2
0.0                                          $hnw1
8.3765e7                                     $hnw2
2 3 4 5 15(0)                                $n
14(.1633)                                    $dz
14(6500.)                                    $tcan
2(0.0) .31 .58  .76  .82  .76 .58 .31 5(0.0)
2(0.0) .34 .64  .83  .91  .83 .64 .34 5(0.0)
2(0.0) .39 .72  .93 1.02  .93 .72 .39 5(0.0)
2(0.0) .42 .79 1.02 1.11 1.02 .79 .42 5(0.0)
2(0.0) .46 .87 1.12 1.22 1.12 .87 .46 5(0.0)

5(0.0 61.0 12(0.0))                          $shape
4(1.0) 4(0.0)                                $sppd
9(1) 5(0)                                    $ppp
                                             $lplmn
$tres
pin    = 1.838e5
pout   = 1.5e5
tin    = 589.14
tav    = 700.0
$end
```

Appendix C.3: THORS Bundle 6A Test 71H Run 101 - Steady State

```
$restrt
nres   =  1
$end
$unos
sprint =  8.
ni     =  42
nj     =  3
ncf    =  4
ncld   =  2
itmi   =  8
indgs  =  0
igauss =  0
dtmax  =  1.0
eps1   =  11.
eps2   =  0.002
nset   =  5
tset   =  1.2
itbd   = -1
$end
$duos
rnusl1 = -13.
rnusl2 = -13.
nrow   =  3
pitch  =  7.265e-3
d      =  5.842e-3
e      =  7.113e-4
ad     =  0.95
apu    =  0.0
dil    =  0.9
radr   =  2.921e-3
thc    =  0.381e-3
thg    =  0.6e-4
lss    =  1
tinit  =  0.0
ntcd   =  1
lp     =  0
$end
11.0                                                    $tb
0.0                                                     $fb1
1.02504                                                 $fb2
0.0                                                     $pnt1
1.4445e5                                                $pnt2
0.0                                                     $alb1
0.0                                                     $alb2
0.0                                                     $tvb1
660.91                                                  $tvb2
0.0                                                     $tlb1
660.91                                                  $tlb2
0.0                                                     $hnw1
4.90971831e8                                            $hnw2
2 3 17(0)                                               $n
7(.05398) 25(.0508) 10(.07232)                          $dz
7(0.0) 21(650.) 14(0.0)                                 $tcan
3( 7(0.0) .43 .515 .64 .73 .815 .885 .94 .98
1.0 1.0 .98 .94 .885 .815 .73 .64 .515 .43
17(0.0))                                                $shape
2(0.0 50. 40(0.0)) 0.0 50. 40(0.0)                      $sppd
4(1.0) 4(0.0)                                           $ppp
28(1) 14(0)                                             $iplnm
$tres
pin    =  1.7277e+5
pout   =  1.4445e+5
tin    =  660.91
tav    =  900.0
$end
```

THORS Bundle 6A Test 71H Run 101
- Transient (0.0 - 11.0 sec)

```
$restrt
nres     = 0
$end
$unos
itm1     = 5
igauss   = 100
dtmax    = 1.0
dtmin    = 1.0e-04
nset(1)  = 12
nset(2)  = 0
tset(1)  = 1.0
tset(2)  = 0.0
indgs    = 0
sprint   = 10.0
eps1     = 250.
eps2     = 1.e-2
itbd     = -1
$end
$duos
rnusl1   = -13.0
rnusl2   = -13.0
lss      = 0
tinit    = 0.0
ntcd     = 3
lp       = 0
$end
3.2 6.8 30.0                          $tb
0.0 -0.197185 0.0                     $fb1
1.02504 1.02504 .315174               $fb2
0.0 -3347.222 0.0                     $pnt1
1.4445e5 1.4445e5 1.324e5             $pnt2
3(0.0)                                $alb1
3(0.0)                                $alb2
3(0.0)                                $tvb1
3(660.91)                             $tvb2
3(0.0)                                $tlb1
3(660.91)                             $tlb2
3(0.0)                                $hnb1
3(4.90971831e8)                       $hnb2
```

THORS Bundle 6A Test 71H Run 101
- Transient (11.0 - 20.0 sec)

```
$restrt
nres     =  0
$end
$unos
itm1     =  5
igauss   =  100
dtmax    =  1.0
dtmin    =  1.0e-08
nset(1)  =  50
nset(2)  =  0
tset(1)  =  0.1
tset(2)  =  0.0
indgs    =  0
sprint   =  15.
eps1     =  250.
eps2     =  1.e-2
itbd     =  0
$end
$duos
rnus11   = -13.0
rnus12   = -13.0
lss      =  0
tinit    =  0.0
ntcd     =  3
lp       =  0
$end
5.0 10. 25.                                  $tb
3(0.0)                                       $pnb1
3(1.6113e5)                                  $pnb2
3(0.0)                                       $pnt1
3(1.324e5)                                   $pnt2
3(0.0)                                       $alb1
3(0.0)                                       $alb2
3(0.0)                                       $tvb1
3(660.91)                                    $tvb2
3(0.0)                                       $tlb1
3(660.91)                                    $tlb2
3(0.0)                                       $hnb1
3(4.90971831e8)                              $hnb2
```

The flow map output for this problem at Time = 11.3415 sec appears in Appendix F.

Appendix C.4: W-1 SLSF LOPI 2A - Steady State

```
$restrt
nres    = 1
$end
$unos
sprint  = 0.1
ni      = 40
nj      = 3
ncf     = 4
ncld    = 2
itm1    = 6
indgs   = 0
igauss  = 500
dtmax   = 1.0
eps1    = 10.0
eps2    = 0.005
nset(1) = 1
tset(1) = 6.0
itbd    = -1
$end
$duos
rnusl1  = -13.
rnusl2  = -13.
nrow    = 3
pitch   = 0.7264d-2
d       = 0.584d-2
e       = 0.711d-3
ad      = 0.954
apu     = 0.25
dil     = 0.9
radr    = 0.2921d-2
thc     = 0.381d-3
thg     = 0.6d-4
lss     = 1
tinit   = 0.0
ntcd    = 1
lp      = 0
$end
8.0                                                    $tb
0.0                                                    $fb1
5.97742777                                             $fb2
0.0                                                    $pnt1
2.776e+5                                               $pnt2
0.0                                                    $alb1
0.0                                                    $alb2
0.0                                                    $tvb1
661.14                                                 $tvb2
0.0                                                    $tlb1
661.14                                                 $tlb2
0.0                                                    $hnw1
1.907849e+9                                            $hnw2
2 3 17(0)                                              $n
6(.05948) 23(.0508) 11(.10033)                         $dz
29(0.85e+4) 11(0.0)                                    $tcan
3( 6(0.0) .66 .835 .98 1.11 1.22 1.3 1.37 .141 1.42
1.41 1.38 1.32 1.25 1.14 1.02 .85 .7 .5 16(0.0))  $shape
2(0.0 0.0 38(0.0)) 0.0 0.0 38(0.0)                     $sppd
4(1.0) 4(0.0)                                          $ppp
29(1) 11(0)                                            $lplnm
$tres
pin     = 6.0d+5
pout    = 2.776d+5
tin     = 661.14
tav     = 900.0
$end
```

W-1 SLSF LOPI 2A
- Transient (0.0 - 4.5 sec)

```
$restrt
nres     = 0
$end
$unos
itm1     = 5
igauss   = 100
dtmax    = 1.0
dtmin    = 1.0e-06
nset(1)  = 45
nset(2)  = 0
tset(1)  = 0.1
tset(2)  = 0.0
indgs    = 0
sprint   = 0.1
eps1     = 70.0
eps2     = 1.e-2
itbd     = -1
$end
$duos
rnusl1   = -13.0
rnusl2   = -13.0
lss      = 0
tinit    = 0.0
ntcd     = 7
lp       = 0
$end
0.33 0.5 0.65 0.85 1.33 3.0 6.0                          $tb
-13.8911754 -.81961177 1.85778667 1.393341
0.29027917 0.0 0.4180020                                 $vb1
5.9774277 1.3933398 1.2540058 1.5326738
1.811342 1.950676 1.950676                               $vb1
2(-2.1356e5) 5(0.0)                                      $pnt1
2.776e5 2.071252e5 5(1.7082e5)                           $pnt2
7(0.0)                                                   $alb1
7(0.0)                                                   $alb2
7(0.0)                                                   $tvb1
7(661.14)                                                $tvb2
7(0.0)                                                   $tlb1
7(661.14)                                                $tlb2
3(0.0) -8.346839e9 3(0.0)                                $hnb1
4(1.907849e9) 3(2.384811e8)                              $hnb2
```

Appendix C.5: W-1 SLSF LOPI 4 - Steady State

```
$restrt
nres    = 1
$end
$unos
sprint  = 0.1
ni      = 40
nj      = 3
ncf     = 4
ncld    = 2
itm1    = 6
indgs   = 0
igauss  = 500
dtmax   = 1.0
eps1    = 10.0
eps2    = 0.005
nset(1) = 1
tset(1) = 5.0
itbd    = -1
$end
$duos
rnusl1  = -13.
rnusl2  = -13.
nrow    = 3
pitch   = 0.7264d-2
d       = 0.584d-2
e       = 0.711d-3
ad      = 0.954
apu     = 0.25
dil     = 0.9
radr    = 0.2921d-2
thc     = 0.381d-3
thg     = 0.6d-4
lss     = 1
tinit   = 0.0
ntcd    = 1
lp      = 0
$end
8.0                                                          $tb
0.0                                                          $fb1
5.97742777                                                   $fb2
0.0                                                          $pnt1
2.776e+5                                                     $pnt2
0.0                                                          $alb1
0.0                                                          $alb2
0.0                                                          $tvb1
661.14                                                       $tvb2
0.0                                                          $tlb1
661.14                                                       $tlb2
0.0                                                          $hnw1
2.0332515e+9                                                 $hnw2
2 3 17(0)                                                    $n
6(.05948) 23(.0508) 11(.10033)                               $dz
29(0.85e+4) 11(0.0)                                          $tcan
3( 6(0.0) .66 .835 .98 1.11 1.22 1.3 1.37 .141 1.42
1.41 1.38 1.32 1.25 1.14 1.02 .85 .7 .5 16(0.0))             $shape
2(0.0 0.0 38(0.0)) 0.0 0.0 38(0.0)                           $sppd
4(1.0) 4(0.0)                                                $ppp
29(1) 11(0)                                                  $lplnm
$tres
pin     = 600000.0
pout    = 277600.0
tin     = 661.14
tav     = 900.0
$end
```

W-1 SLSF LOPI 4
- Transient (0.0 - 5.0 sec)

```
$restrt
nres    = 0
$end
$unos
itm1    = 5
igauss  = 100
dtmax   = 1.0
dtmin   = 1.0e-08
nset(1) = 30
nset(2) = 6
tset(1) = 0.1
tset(2) = 0.333333333
indgs   = 0
sprint  = 10.
eps1    = 700.
eps2    = 1.e-2
itbd    = -1
$end
$duos
rnus11  = -13.0
rnus12  = -13.0
lss     = 0
tinit   = 0.0
ntcd    = 7
lp      = 0
$end
0.33 0.5 0.65 0.85 1.33 3.0 6.0                          $tb
-13.8911754 -.81961177 1.85778667 1.393341
0.29027917 0.0 0.4180020                                 $vb1
5.9774277 1.3933398 1.2540058 1.5326738
1.811342 1.950676 1.950676                               $vb1
2(-2.1356e5) 5(0.0)                                      $pnt1
2.776e5 2.071252e5 5(1.7082e5)                           $pnt2
7(0.0)                                                   $alb1
7(0.0)                                                   $alb2
7(0.0)                                                   $tvb1
7(661.14)                                                $tvb2
7(0.0)                                                   $tlb1
7(661.14)                                                $tlb2
3(0.0) -8.8954752e9 3(0.0)                               $hnb1
4(2.0332515e9) 3(2.5415644e8)                            $hnb2
```

Appendix C.6: W-1 SLSF BWT 2'
- Steady State

```
$restrt
nres    = 1
$end
$unos
sprint  = 6.0
ni      = 40
nj      = 3
ncf     = 4
ncld    = 2
itm1    = 6
indgs   = 0
igauss  = 500
dtmax   = 1.0
eps1    = 1000.0
eps2    = 0.005
nset(1) = 1
tset(1) = 3.0
itbd    = -1
$end
$duos
rnusl1  = -13.
rnusl2  = -13.
nrow    = 3
pitch   = 0.7264d-2
d       = 0.584d-2
e       = 0.711d-3
ad      = 0.954
apu     = 0.25
dil     = 0.9
radr    = 0.2921d-2
thc     = 0.381d-3
thg     = 0.6d-4
lss     = 1
tinit   = 0.0
ntcd    = 1
lp      = 0
$end
8.0                                                    $tb
0.0                                                    $fb1
5.97742777                                             $fb2
0.0                                                    $pnt1
2.776e+5                                               $pnt2
0.0                                                    $alb1
0.0                                                    $alb2
0.0                                                    $tvb1
661.14                                                 $tvb2
0.0                                                    $tlb1
661.14                                                 $tlb2
0.0                                                    $hnw1
9.3999172e+8                                           $hnw2
2 3 17(0)                                              $n
6(.05948) 23(.0508) 11(.10033)                         $dz
29(100.) 11(1.0)                                       $tcan
3(6(0.0) .66 .835 .98 1.11 1.22 1.3 1.37 1.41 1.42
1.41 1.38 1.32 1.25 1.14 1.02 .85 .7 .5 16(0.0))       $shape
2(0.0 30. 38(0.0)) 0.0 30.0 38(0.0)                    $sppd
4(1.0) 4(0.0)                                          $ppp
29(1) 11(0)                                            $lplnm
$tres
pin     = 600000.0
pout    = 277600.0
tin     = 661.14
tav     = 900.0
$end
```

W-1 SLSF BWT 2'
- Transient (0.0 - 3.0 sec)

```
$restrt
nres     = 0
$end
$unos
itm1     = 5
igauss   = 100
dtmax    = 1.0
dtmin    = 1.0e-06
nset(1)  = 27
nset(2)  = 0
tset(1)  = 0.11111111111
tset(2)  = 0.0
indgs    = 0
sprint   = 4.0
eps1     = 700.
eps2     = 1.e-2
itbd     = -1
$end
$duos
rnusl1   = -9.0
rnusl2   = -9.
lss      = 0
tinit    = 0.0
ntcd     = 2
lp       = 0
$end
0.5 5.0                                  $tb
-9.07342871 0.0                          $vb1
5.9774277 1.44071334                     $vb2
-2.1356e5 0.0                            $pnt1
2.776e5 1.7082e5                         $pnt2
2(0.0)                                   $alb1
2(0.0)                                   $alb2
2(0.0)                                   $tvb1
2(661.14)                                $tvb2
2(0.0)                                   $tlb1
2(661.14)                                $tlb2
2(0.0)                                   $hnb1
2(9.3999172e+8)                          $hnw2
```

W-1 SLSF BWT 2'
- Transient (3.0 - 5.5 sec)

```
$restrt
nres     = 0
$end
$unos
itm1     = 5
igauss   = 100
dtmax    = 1.0
dtmin    = 1.0e-09
nset(1)  = 23
nset(2)  = 0
tset(1)  = .11111111
tset(2)  = 0.0
indgs    = 0
sprint   = 10.
eps1     = 700.
eps2     = 1.e-2
itbd     = 0
$end
$duos
rnusl1   = -11.0
rnusl2   = -11.0
lss      = 0
tinit    = 0.0
ntcd     = 3
lp       = 0
$end
2.5 3.0 7.0                              $tb
0.0 6.9528e+5 0.0                        $pnb1
2.1087e5 2.1087e5 5.5878e5               $pnb2
0.0 2.1356e5 0.0                         $pnt1
1.7082e5 1.7082e5 2.776e5                $pnt2
3(0.0)                                   $alb1
3(0.0)                                   $alb2
3(0.0)                                   $tvb1
3(661.14)                                $tvb2
3(0.0)                                   $tlb1
3(661.14)                                $tlb2
3(0.0)                                   $hnb1
3(9.399917e+8)                           $hnb2
```

Appendix C.7: W-1 SLSF BWT 7B'
- Steady State

```
$restrt
nres    = 1
$end
$unos
sprint  = 10.
ni      = 40
nj      = 3
ncf     = 4
ncld    = 2
itm1    = 6
indgs   = 0
igauss  = 500
dtmax   = 1.0
eps1    = 1000.0
eps2    = 0.005
nset(1) = 1
tset(1) = 3.0
itbd    = -1
$end
$duos
rnusl1  = -13.
rnusl2  = -13.
nrow    = 3
pitch   = 0.7264d-2
d       = 0.584d-2
e       = 0.711d-3
ad      = 0.954
apu     = 0.25
dil     = 0.9
radr    = 0.2921d-2
thc     = 0.381d-3
thg     = 0.6d-4
lss     = 1
tinit   = 0.0
ntcd    = 1
lp      = 0
$end
8.0                                                    $tb
0.0                                                    $fb1
6.06976171                                             $fb2
0.0                                                    $pnt1
2.776e+5                                               $pnt2
0.0                                                    $alb1
0.0                                                    $alb2
0.0                                                    $tvb1
661.14                                                 $tvb2
0.0                                                    $tlb1
661.14                                                 $tlb2
0.0                                                    $hnw1
1.7860674e+9                                           $hnw2
2 3 17(0)                                              $n
6(.05948) 23(.0508) 11(.10033)                         $dz
29(100.) 11(1.0)                                       $tcan
3(6(0.0) .66 .835 .98 1.11 1.22 1.3 1.37 1.41 1.42
1.41 1.38 1.32 1.25 1.14 1.02 .85 .7 .5 16(0.0))       $shape
2(0.0 30. 38(0.0)) 0.0 30.0 38(0.0)                    $sppd
4(1.0) 4(0.0)                                          $ppp
29(1) 11(0)                                            $lplnm
$tres
pin     = 600000.0
pout    = 277600.0
tin     = 661.14
tav     = 900.0
$end
```

W-1 SLSF BWT 7B'
- Transient (0.0 - 1.5 sec)

```
$restrt
nres     = 0
$end
$unos
itm1     = 5
igauss   = 100
dtmax    = 1.0
dtmin    = 1.0e-06
nset(1)  = 15
nset(2)  = 0
tset(1)  = 0.1
tset(2)  = 0.0
indgs    = 0
sprint   = 2.
eps1     = 700.
eps2     = 1.e-2
itbd     = -1
$end
$duos
rnusl1   = -13.0
rnusl2   = -13.0
lss      = 0
tinit    = 0.0
ntcd     = 2
lp       = 0
$end
0.5 5.0                                  $tb
-7.6025298 0.0                           $vb1
6.06976171 2.2684968                     $vb2
-2.1356e5 0.0                            $pnt1
2.776e5 1.7082e5                         $pnt2
2(0.0)                                   $alb1
2(0.0)                                   $alb2
2(0.0)                                   $tvb1
2(661.14)                                $tvb2
2(0.0)                                   $tlb1
2(661.14)                                $tlb2
2(0.0)                                   $hnb1
2(1.7860674e+9)                          $hnw2
```

W-1 SLSF BWT 7B'
- Transient (1.5 - 5.0 sec)

```
$restrt
nres    = 0
$end
$unos
itm1    = 5
igauss  = 100
dtmax   = 1.0
dtmin   = 1.0e-09
nset(1) = 45
nset(2) = 0
tset(1) = 0.05
tset(2) = 0.0
indgs   = 0
sprint  = 10.
eps1    = 2000.
eps2    = 1.e-2
itbd    = 0
$end
$duos
rnusl1  = -11.0
rnusl2  = -11.0
alpdry = .0.957
lss     = 0
tinit   = 0.0
ntcd    = 3
lp      = 0
$end
2.0 2.5 7.0                                   $tb
0.0 6.6306e+5 0.0                             $pnb1
2(2.3684e5) 5.6837e5                          $pnb2
0.0 2.1356e5 0.0                              $pnt1
1.7082e5 1.7082e5 2.776e5                     $pnt2
3(0.0)                                        $alb1
3(0.0)                                        $alb2
3(0.0)                                        $tvb1
3(661.14)                                     $tvb2
3(0.0)                                        $tlb1
3(661.14)                                     $tlb2
0.0 -3.214921e9 0.0                           $hnb1
2(1.7860674e+9) 1.7860674e+8                  $hnb2
```

Appendix D

NATOF-2D HEXCAN MODEL

In NATOF-2D, the user specifies the heat capacity of the hexcan per unit area (J/m^2-^oK) for each axial level. Presently, this value is estimated by the user based on the properties of the can. However, this simple model has many limitations since it cannot take into account varying properties or dimensions of the hexcan, or heat losses to the environment.

The value chosen for the hexcan heat capacity has a pronounced effect of the temperature profile of the coolant. As an example, two cases were run with NATOF-2D for different values of the hexcan heat capacity. The simulation chosen for the test was BWT 2' (described in Chapter 6) and the input can be found in Appendix C.6. Table D.1 gives the liquid temperature at the end of the heated zone for the central channel at various points of time. The radial heat conduction nusselt numbers were $Nu_1 = 13.$ and $Nu_2 = 13.$

Table D.1

Liquid Temperature (^{o}C) at Various Points in Time

Time (sec)	HCAN = 100	HCAN = 8500
0.0	538.	538.
0.5	580.	578.
1.0	661.	649.
1.5	735.	708.
2.0	790.	756.
2.5	832.	794.

As can be seen, an inaccurate choice of the hexcan heat capacity can cause large liquid temperature differences. Thus, in two-phase transients, boiling inception time can be drastically altered.

Appendix E

SPACER PRESSURE DROP MODEL

The spacer pressure drop feature of NATOF-2D allows the user to simulate pressure drops which occur in the bundle due to valve throttling, spacer wires, etc. The spacer pressure drop is calculated as follows:

$$\Delta P = SPPD * \frac{\rho \cdot U^2}{2}$$

where SPPD is specified for each cell.

Flow reversal occurs in NATOF-2D when the pressure in the first real cell exceeds the pressure at the boundary. Thus, by specifying a large boundary pressure, and a large value of SPPD, it is possible to prevent flow reversal during boiling transients.

As a sample test of this feature, simulations were run in which a constant flow of 2 kg/sec was imposed on the bundle, with an outlet pressure of 2.5 bars. The geometric parameters used were from the W-1 SLSF Experiment. The value of SPPD for the bottom row of cells was varied over a wide range, and the inlet pressure necessary to maintain the flow rate was inferred. Table E.1 gives values of SPPDs and the corresponding inlet pressure.

Table E.1

Inlet Pressure vs. SPPD for Constant Flow Boundary Condition

SPPD	Inlet Pressure (bars)
0.0	5.16
10.0	5.26
100.0	6.12
500.0	9.96

As can be seen, it is possible to determine flow reversal time by the correct choice of the spacer pressure drop. Furthermore, if the experimental outlet and inlet pressure drops are accurately known, the spacer pressure drop feature can be used to obtain the correct inlet flow rate.

flow map at time = 11.3415 sec.

number of time steps = 161
number of iterations = 206
time step size = 0.2612D-02 sec.
cpu time = 712.16

inlet mass flow rate = 0.577033D-01 kg/sec
outlet mass flow rate = 0.757669D+00 kg/sec
total heat transfered = 0.361730D+05 watt

inlet enthalpy flow = 0.503501D+05 watt
outlet enthalpy flow = 0.982316D+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)							
42	1.3240	0.0025	914.304	723.717	912.005	722.192	2.80873	2.78910	0.00000	0.00000
41	1.3502	0.0025	914.304	723.717	914.303	722.192	2.93854	2.86145	0.00185	0.00185
40	1.3756	0.0219	916.493	731.763	916.490	730.203	3.63601	3.14366	0.00474	0.00474
39	1.4131	0.1406	919.664	739.740	919.664	738.455	4.88741	3.54369	-0.00620	-0.00620
38	1.4484	0.4837	922.590	747.438	922.590	746.526	6.84971	3.65853	-0.00175	-0.00174
37	1.4785	0.7753	925.046	755.750	925.046	755.160	9.26300	3.24684	0.01743	0.01726
36	1.4979	0.8795	926.604	764.217	926.604	763.753	11.15557	2.81298	-0.00296	-0.00291
35	1.5115	0.9099	927.685	774.945	927.685	773.502	12.64466	2.55882	-0.01069	-0.01046
34	1.5224	0.9376	928.546	795.077	928.546	784.533	14.59305	2.45305	0.00918	0.00892
33	1.5321	0.9447	929.312	825.230	929.312	796.585	16.13365	2.53376	0.00266	0.00258
32	1.5408	0.9480	929.992	858.958	929.992	809.545	16.77430	2.54972	-0.00716	-0.00693
31	1.5479	0.9494	930.547	880.073	930.547	819.322	17.27383	2.60706	-0.00413	-0.00400
30	1.5552	0.9494	931.117	892.759	931.117	829.350	17.66196	2.68181	-0.00271	-0.00262
29	1.5629	0.9491	931.712	899.557	931.712	839.675	18.03033	2.76169	0.00079	0.00076
28	1.5710	0.9488	932.334	904.565	932.334	837.319	18.38645	2.83970	0.00354	0.00343
27	1.5795	0.9485	932.985	913.784	932.985	861.088	18.62471	2.89438	0.00364	0.00352
26	1.5882	0.9484	933.656	924.738	933.656	887.445	18.68256	2.90893	0.00143	0.00138
25	1.5972	0.9482	934.336	934.664	934.336	949.574	18.07009	2.83700	-0.00168	-0.00163
24	1.6058	0.9472	934.990	935.600	934.990	960.766	16.99696	2.70355	-0.00413	-0.00400
23	1.6139	0.9458	935.601	936.620	935.601	971.042	15.25347	2.46241	-0.00784	-0.00760
22	1.6210	0.9435	936.137	937.665	936.137	974.239	12.62162	2.06388	-0.01515	-0.01472
21	1.6266	0.9397	936.559	938.932	936.559	972.108	8.62277	1.41161	-0.02123	-0.02069
20	1.6303	0.9271	936.833	941.047	936.833	957.825	2.39687	0.37249	0.00124	0.00122
19	1.6325	0.7176	936.993	938.730	936.993	939.721	-0.19455	0.06971	0.05584	0.05571
18	1.6312	0.0169	936.900	915.544	936.900	917.795	0.21688	0.22406	0.01677	0.01677
17	1.6293	0.0000	939.245	887.411	936.761	890.151	0.25595	0.25595	0.00227	0.00227
16	1.6280	0.0000	939.215	853.287	936.658	856.202	0.23974	0.23974	-0.00069	-0.00069
15	1.6267	0.0000	939.198	812.318	936.565	815.276	0.22233	0.22233	-0.00076	-0.00076

16	1.6280	0.0000	939.215	853.287	936.658	856.202	0.23974	0.23974	-0.00069	-0.00069
15	1.6267	0.0000	939.198	812.318	936.565	815.276	0.22233	0.22233	-0.00076	-0.00076
14	1.6255	0.0000	939.136	765.934	936.474	768.824	0.21106	0.21106	-0.00039	-0.00039
13	1.6243	0.0000	939.004	716.213	936.384	718.979	0.20355	0.20355	-0.00016	-0.00016
12	1.6231	0.0000	938.794	664.447	936.296	667.045	0.19773	0.19773	-0.00007	-0.00007
11	1.6220	0.0000	938.507	611.886	936.210	614.262	0.19274	0.19274	-0.00005	-0.00005
10	1.6209	0.0000	938.191	560.093	936.125	562.240	0.18825	0.18825	-0.00004	-0.00004
9	1.6198	0.0000	937.718	509.717	936.043	511.548	0.18441	0.18441	-0.00003	-0.00003
8	1.6187	0.0000	937.517	462.993	935.962	464.689	0.18099	0.18099	-0.00002	-0.00002
7	1.6176	0.0000	935.479	416.119	935.880	416.466	0.18042	0.18042	0.00001	0.00001
6	1.6165	0.0000	935.045	398.087	935.796	398.212	0.18100	0.18100	0.00005	0.00005
5	1.6154	0.0000	935.666	390.952	935.712	390.990	0.18249	0.18249	0.00009	0.00009
4	1.6142	0.0000	935.616	388.616	935.628	388.626	0.18417	0.18417	0.00010	0.00010
3	1.6131	0.0000	935.541	387.967	935.543	387.969	0.18467	0.18467	0.00003	0.00003
2	1.6120	0.0000	935.457	387.808	935.457	387.808	0.18292	0.18292	-0.00010	-0.00010
1	1.6113	0.0000	387.770	387.770	935.406	387.808	0.18292	0.18292	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)--------							
42	1.3240	0.0032	914.297	724.332	912.005	722.680	2.76187	2.73827	0.00000	0.00000
41	1.3502	0.0032	914.297	724.332	914.297	722.680	2.89537	2.79878	0.00251	0.00251
40	1.3754	0.0246	916.472	733.267	916.469	731.532	3.70797	3.08188	-0.00313	-0.00313
39	1.4133	0.1954	919.684	742.515	919.684	741.160	5.23751	3.54216	-0.00152	-0.00152
38	1.4484	0.5861	922.596	751.942	922.596	751.028	7.86026	3.73747	0.03048	0.03039
37	1.4778	0.8179	924.992	762.102	924.992	761.498	10.16014	3.32169	0.02046	0.02028
36	1.4980	0.8626	926.613	771.577	926.613	770.883	10.84887	2.80917	-0.06563	-0.06462
35	1.5119	0.9151	927.716	783.069	927.716	781.281	12.66733	2.49359	0.00142	0.00138
34	1.5220	0.9400	928.520	806.522	928.520	792.532	14.66586	2.48910	0.02238	0.02176
33	1.5320	0.9403	929.305	826.719	929.305	805.750	15.96186	2.57973	-0.00466	-0.00452
32	1.5410	0.9494	930.011	864.631	930.011	819.832	16.77386	2.53586	0.00442	0.00427
31	1.5480	0.9498	930.557	882.380	930.557	829.535	17.34419	2.59818	0.00128	0.00124
30	1.5553	0.9505	931.124	894.599	931.124	839.339	17.78258	2.66169	0.00143	0.00139
29	1.5629	0.9502	931.710	902.509	931.710	848.931	18.14940	2.74732	0.00406	0.00392
28	1.5709	0.9492	932.325	907.412	932.326	844.384	18.45993	2.84136	0.00571	0.00553
27	1.5793	0.9484	932.976	915.767	932.977	866.811	18.64648	2.90881	0.00460	0.00445
26	1.5882	0.9479	933.653	925.415	933.653	891.275	18.67160	2.92745	0.00092	0.00089
25	1.5972	0.9479	934.340	934.675	934.340	948.597	18.05545	2.85070	-0.00240	-0.00232
24	1.6059	0.9460	935.000	935.609	935.000	958.983	17.01460	2.71860	-0.00554	-0.00537
23	1.6141	0.9452	935.620	936.669	935.620	968.288	15.26988	2.47962	-0.01208	-0.01173
22	1.6215	0.9421	936.172	937.913	936.172	970.950	12.35471	2.03568	-0.02705	-0.02630

26	1.5882	0.9479	933.653	925.415	933.653	891.275	18.67160	2.92745	0.00092	0.00089
25	1.5972	0.9479	934.340	934.675	934.340	948.597	18.05545	2.85070	-0.00240	-0.00232
24	1.6059	0.9460	935.000	935.609	935.000	958.983	17.01460	2.71860	-0.00554	-0.00537
23	1.6141	0.9452	935.620	936.669	935.620	968.288	15.26988	2.47962	-0.01208	-0.01173
22	1.6215	0.9421	936.172	937.913	936.172	970.950	12.35471	2.03568	-0.02705	-0.02630
21	1.6273	0.9364	936.609	939.654	936.609	968.521	7.23057	1.22169	-0.04532	-0.04420
20	1.6303	0.9242	936.830	941.021	936.830	954.083	0.56214	0.17885	-0.00820	-0.00814
19	1.6306	0.5223	936.854	935.666	936.854	937.146	0.23920	0.23592	0.01277	0.01276
18	1.6306	0.0039	936.856	913.266	936.856	915.593	0.25300	0.25495	0.00997	0.00997
17	1.6293	0.0000	938.798	884.788	936.755	887.539	0.23780	0.23780	-0.00004	-0.00004
16	1.6280	0.0000	938.750	850.373	936.660	853.351	0.22265	0.22265	-0.00131	-0.00131
15	1.6267	0.0000	938.733	809.189	936.567	812.186	0.21310	0.21310	-0.00081	-0.00081
14	1.6255	0.0000	938.668	762.942	936.475	765.848	0.20640	0.20640	-0.00036	-0.00036
13	1.6243	0.0000	938.537	713.496	936.385	716.278	0.20090	0.20090	-0.00015	-0.00015
12	1.6231	0.0000	938.343	661.989	936.296	664.601	0.19599	0.19599	-0.00007	-0.00007
11	1.6220	0.0000	938.088	609.711	936.210	612.099	0.19151	0.19151	-0.00005	-0.00005
10	1.6209	0.0000	937.810	558.244	936.125	560.399	0.18741	0.18741	-0.00005	-0.00005
9	1.6198	0.0000	937.402	508.289	936.043	510.122	0.18383	0.18383	-0.00005	-0.00005
8	1.6187	0.0000	937.206	462.128	935.962	463.820	0.18056	0.18056	-0.00003	-0.00003
7	1.6176	0.0000	935.520	416.149	935.880	416.502	0.17989	0.17989	0.00001	0.00001
6	1.6165	0.0000	935.653	398.270	935.795	398.402	0.18018	0.18018	0.00008	0.00008
5	1.6153	0.0000	935.665	391.063	935.711	391.105	0.18147	0.18147	0.00017	0.00017
4	1.6142	0.0000	935.615	388.660	935.627	388.672	0.18371	0.18371	0.00025	0.00025
3	1.6131	0.0000	935.540	387.980	935.543	387.983	0.18544	0.18544	0.00017	0.00017
2	1.6120	0.0000	935.457	387.811	935.458	387.812	0.18290	0.18290	-0.00027	-0.00027
1	1.6113	0.0000	387.770	387.770	935.406	387.812	0.18290	0.18290	0.00000	0.00000

channel number 3

iz	p (bar)	void	tv	tl	tsat	twall	tcan	uvz (m/sec)	ulz (m/sec)
			(degree celsius)						
42	1.3240	0.0001	914.291	724.102	912.005	722.663	724.102	2.34813	2.34786
41	1.3501	0.0001	914.291	724.102	914.290	722.663	724.102	2.32161	2.31818
40	1.3755	0.0015	916.486	734.048	916.483	732.156	734.048	2.43372	2.38073
39	1.4134	0.0152	919.689	746.893	919.688	744.564	746.893	2.82541	2.53235
38	1.4475	0.1216	922.519	761.580	922.519	759.823	761.580	3.63084	2.69108
37	1.4772	0.4855	924.942	776.337	924.942	775.609	776.337	5.11298	2.56616
36	1.4999	0.8437	926.765	789.181	926.765	788.647	789.181	7.82457	2.14141

35	1.5118	0.9269	927.713	806.221	927.712	801.344	806.221	9.66715	1.99827
34	1.5214	0.9180	928.472	817.640	928.472	814.924	817.640	10.81656	2.13968
33	1.5321	0.9375	929.315	841.307	929.315	829.134	841.307	12.20306	2.12147
32	1.5409	0.9439	930.002	868.943	930.002	843.314	868.943	12.85329	2.11556
31	1.5480	0.9453	930.555	881.914	930.555	853.189	881.914	13.38528	2.15590
30	1.5553	0.9472	931.121	894.130	931.121	861.994	894.130	13.80544	2.19861
29	1.5628	0.9468	931.702	899.128	931.702	870.457	899.128	14.14821	2.27296
28	1.5707	0.9458	932.314	899.633	932.314	862.147	878.998	14.47696	2.36172
27	1.5792	0.9447	932.968	909.964	932.968	880.900	896.462	14.68455	2.43020
26	1.5882	0.9443	933.651	921.601	933.651	899.784	913.215	14.76841	2.45266
25	1.5973	0.9447	934.344	934.564	934.345	943.994	930.023	14.38129	2.39341
24	1.6061	0.9442	935.010	935.556	935.010	953.065	932.410	13.55669	2.29605
23	1.6144	0.9412	935.642	936.799	935.642	960.568	933.121	11.89976	2.11027
22	1.6221	0.9333	936.222	938.643	936.222	959.247	934.261	8.51273	1.67252
21	1.6284	0.9067	936.693	941.081	936.693	946.747	940.342	2.10520	0.65164
20	1.6305	0.7020	936.847	937.777	936.847	939.050	937.468	0.22167	0.29299
19	1.6303	0.2680	936.831	924.139	936.831	925.918	923.782	0.28323	0.27942
18	1.6303	0.0019	936.836	904.095	936.836	906.129	903.736	0.18297	0.18343
17	1.6293	0.0000	937.786	874.505	936.755	876.635	874.105	0.18005	0.18005
16	1.6280	0.0000	937.716	838.230	936.662	840.426	837.852	0.18869	0.18869
15	1.6268	0.0000	937.595	797.743	936.568	800.003	797.409	0.19229	0.19229
14	1.6255	0.0000	937.452	753.030	936.476	755.304	752.745	0.19148	0.19148
13	1.6243	0.0000	937.308	704.672	936.385	706.882	704.431	0.18871	0.18871
12	1.6231	0.0000	937.153	654.042	936.297	656.122	653.842	0.18541	0.18541
11	1.6220	0.0000	936.976	602.703	936.210	604.603	602.545	0.18221	0.18221
10	1.6209	0.0000	936.790	552.252	936.126	553.963	552.133	0.17933	0.17933
9	1.6198	0.0000	936.545	503.598	936.043	505.048	503.527	0.17694	0.17694
8	1.6187	0.0000	936.377	458.816	935.962	460.152	458.768	0.17464	0.17464
7	1.6176	0.0000	935.680	415.774	935.880	416.088	415.774	0.17395	0.17395
6	1.6165	0.0000	935.700	398.497	935.795	398.636	398.497	0.17292	0.17292
5	1.6153	0.0000	935.675	391.241	935.711	391.291	391.241	0.17110	0.17110
4	1.6142	0.0000	935.615	388.736	935.626	388.751	388.736	0.16851	0.16851
3	1.6131	0.0000	935.539	388.004	935.542	388.008	388.004	0.16678	0.16678
2	1.6120	0.0000	935.458	387.816	935.459	387.817	387.816	0.16960	0.16960
1	1.6113	0.0000	387.770	387.770	935.406	387.817	387.816	0.16960	0.16960

```
      subroutine read2(p,tv,tl,alfa,uvz,ulz,uvr,ulr,dh,dv,
     *                 qsi,tr,dtr,tw,sppd,tcan,tinit,dtmax,
     *                 np,ntr,npin,npm1,nn,ncan,nres,itbd)
c
c   this subroutine reads all other information, controls the
c   writing of input data for a restart, and calculates
c   parameters which will remain constant throughout the
c   problem.
c
      implicit real*8 (a-h,o-z)
      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),
     *               vb1(51),vb2(51),vb3(51),omvb(51),flb1(51),
     *               flb2(51),flb3(51),omfb(51),lmax,lp
      common /pshape/ shape(500)
      common /dim/ dz(150),dz1(150),dro(150),dr1(150),dr2(150),dr3(150),
     *             dr4(150),dr5(150),dr6(150),ni,nj,nim1,nim2,njm1,nni,
     *             nnj,nnjj
      common /pin0/ 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(150)
      common /iconst/ ncf,ncc,ng
      common /pd/ d4,pod2
      common /dryout/ alpdry
      common /poverd/ r
      common /hxcn/ acov
      common /stst/ tafp,lss
      common /eccof/ cefl,ccfv
      common /extra/ con
      common /nussy/ rnusl1,rnusl2
      common /tbound/ tbc(50),tbmax,tsr
      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)
      namelist/duos/nrow,pitch,d,e,ad,apu,dil,radr,thc,thg,lss,
     *              tinit,ntcd,lp,cefl,ccfv,con,rnusl1,rnusl2,
     *              alpdry,tsr,itbc
      namelist/tres/pin,pout,tav,tin
c
      fa0 = 1.81d+06
      fa1 = 3.72d+03
      fa2 = -2.510d0
```

```
      fa3 = 6.59d-04
      fb0 = 10.80d0
      fb1 = -8.84d-03
      fb2 = 2.25d-06
c
      ca0 = 4.28d+06
      ca1 = 3.75d+02
      ca2 = -7.45d-03
      ca3 = zero
      cb0 = 16.27d0
      cb1 = zero
      cb2 = zero
      cb3 = zero
      alpdry = 0.957d0
c
      tb(1) = zero
c
      read(5,duos)
c
      call nips(4h  tb,4h    ,idum,tb(2),ntcd,ierr,0)
      if(itbd) 41,42,43
   41 continue
      call nips(4h   v,4hb1  ,idum,vb1(2) ,ntcd,ierr,0)
      call nips(4h   v,4hb2  ,idum,vb2(2) ,ntcd,ierr,0)
      if(lp.eq.0) go to 2
      call nips(4h   v,4hb3  ,idum,vb3(2) ,ntcd,ierr,0)
      call nips(4h  om,4hvb  ,idum,omvb(2),ntcd,ierr,0)
      go to 2
   42 continue
      call nips(4h  pn,4hb1  ,idum,pnb1(2),ntcd,ierr,0)
      call nips(4h  pn,4hb2  ,idum,pnb2(2),ntcd,ierr,0)
      if (lp.eq.0) go to 2
      call nips(4h  pn,4hb3  ,idum,pnb3(2),ntcd,ierr,0)
      call nips(4h omp,4h    ,idum,omp(2),ntcd,ierr,0)
      go to 2
   43 continue
      call nips(4h  fl,4hb1  ,idum,flb1(2),ntcd,ierr,0)
      call nips(4h  fl,4hb2  ,idum,flb2(2),ntcd,ierr,0)
      if (lp.eq.0) go to 2
      call nips(4h  fl,4hb3  ,idum,flb3(2),ntcd,ierr,0)
      call nips(4h  om,4hfb  ,idum,omfb(2),ntcd,ierr,0)
    2 continue
      call nips(4h  pn,4ht1  ,idum,pnt1(2),ntcd,ierr,0)
      call nips(4h  pn,4ht2  ,idum,pnt2(2),ntcd,ierr,0)
      if (lp.eq.0) go to 3
      call nips(4h  pn,4ht3  ,idum,pnt3(2),ntcd,ierr,0)
      call nips(4h omt,4h    ,idum,omt(2),ntcd,ierr,0)
    3 continue
```

```
      call nips(4h  al,4hb1  ,idum,alb1(2),ntcd,ierr,0)
      call nips(4h  al,4hb2  ,idum,alb2(2),ntcd,ierr,0)
      if (lp.eq.0) go to 4
      call nips(4h  al,4hb3  ,idum,alb3(2),ntcd,ierr,0)
      call nips(4h oma,4h    ,idum,oma(2),ntcd,ierr,0)
    4 continue
      call nips(4h  tv,4hb1  ,idum,tvb1(2),ntcd,ierr,0)
      call nips(4h  tv,4hb2  ,idum,tvb2(2),ntcd,ierr,0)
      if (lp.eq.0) go to 5
      call nips(4h  tv,4hb3  ,idum,tvb3(2),ntcd,ierr,0)
      call nips(4h omv,4h    ,idum,omv(2),ntcd,ierr,0)
    5 continue
      call nips(4h  tl,4hb1  ,idum,tlb1(2),ntcd,ierr,0)
      call nips(4h  tl,4hb2  ,idum,tlb2(2),ntcd,ierr,0)
      if (lp.eq.0) go to 6
      call nips(4h  tl,4hb3  ,idum,tlb3(2),ntcd,ierr,0)
      call nips(4h oml,4h    ,idum,oml(2),ntcd,ierr,0)
    6 continue
      call nips(4h  hn,4hw1  ,idum,hnw1(2),ntcd,ierr,0)
      call nips(4h  hn,4hw2  ,idum,hnw2(2),ntcd,ierr,0)
      if (lp.eq.0) go to 7
      call nips(4h  hn,4hw3  ,idum,hnw3(2),ntcd,ierr,0)
      call nips(4h omh,4h    ,idum,omh(2),ntcd,ierr,0)
    7 continue
      if(itbc.eq.0) go to 225
      call nips(4h  tb,4hc   ,idum,tbc(1),itbc,ierr,0)
      itbcm1 = itbc - 1
      tbmax = tsr
      do 22 i = 1,itbcm1,2
      tbmax = tbmax + tbc(i)*tbc(i+1)
   22 continue
  225 continue
c
      if (nres.eq.1) go to 23
      read(7,1003)nrow,pitch,d,e
      write(8,1003)nrow,pitch,d,e
      call redum1(n,dz,tcan,shape,sppd,ppp,lplnm,ncan,nn,npin,
     *            ad,apu,dil,radr,thc,thg,ni)
      go to 24
c
   23 continue
c
      call nips(4h   n,4h    ,n(1),rdum,19,ierr,1)
      call nips(4h  dz,4h    ,idum,dz(1),ni,ierr,0)
      call nips(4h  tc,4han  ,idum,tcan(1),ni,ierr,0)
      call nips(4h  sh,4hape ,idum,shape(1),nn,ierr,0)
      call nips(4h  sp,4hpd  ,idum,sppd(1),nn,ierr,0)
      call nips(4h ppp,4h    ,idum,ppp(1),npin,ierr,0)
```

```
      call nips(4h lpl,4hnm  ,lplnm(1),rdum,ni,ierr,1)
   24 continue
c
      lmax=ntcd + 2
      do 25 ko=1,nn
         qsi(ko) = (4.*d/(pitch - d))**2
   25 continue
c
      povd = pitch/d
      pod2 = povd*povd
      d4 = 4./d
      r = -16.15 + 24.96*povd - 8.55*povd*povd
c
      dz1(1) = dz(1)
      do 111 i = 2,ni
         dz1(i) = (dz(i) + dz(i-1))/2.0
  111 continue
c
      a1 = dsqrt(3.0d0)/2.0
      a2 = 3.1415927/4.0
      w = pitch - d
c
      x = (pitch*pitch*a1 - (d*d + w*w)*a2)/a2/d
      xi = 4.0/x
      xix = x/a1/pitch*6.
c
      do 8 j = 1,njm1
         do 8 i = 1,ni
            ko = (j-1)*ni + i
            dh(ko) = x
            dv(ko) = xi
    8 continue
      dr5(1) = xix*(n(1) - 1)*d*a2
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*3.0
         dr5(j) = xix*d*a2*n41
         dr6(j) = dr4(j)/1.5/(n41 +n42)/w
c
         nx = n(j) - n(j-1)
         nx1 = 2*n41
         nx2 = (2*n42 + nx)*nx
         dnx1 = nx1
         dr1(j) = dnx1/nx2/pitch/a1
```

```
193          dr2(j) = 2.0*n42/nx2/pitch/a1
194          dro(j) = pitch*a1*nx
195       9 continue
196 c
197          dn4 = (n(1) - 1)*(n(1) - 1)
198          dr4(1) = dn4*x*a2*d*3.0
199          dr6(1) = dr4(1)/1.5/(n(1) - 1)/w
200 c
201          dr1(1) = 2.0/pitch/a1/(n(1)-1)
202          dr2(1) = 0.0
203          dro(1) = pitch*a1*(n(1)-1)
204 c
205          b1 = (n(njm1) + nrow - 2)
206          b2 = (nrow - n(njm1))
207          b3 = (nrow - 1)
208 c
209          xx = b1*b2/2.0 + b3/2.0 + 1.0/6.0
210          pt = b3*pitch + (d/2.0 + e)/a1 + a2*d*xx*4.0
211          ac = (b1*pitch + (d/2.0 + e)/a1)*(b2*pitch*a1 + d/2.0 + e)*
212         *       0.50 - a2*(d*d + e*e)*xx
213          y = 4.0*ac/pt
214          pp = a2*d*xx*4.0
215          yy = pp/ac
216          arm = (one - a2/a1*(d*d + w*w)/(pitch*pitch))*
217         *       (n(njm1) - 1)*pitch
218 c
219          dr1(nj) = zero
220          dr2(nj) = arm/ac
221          dro(nj) = b2*pitch + d/2.0 + e
222          dr4(nj) = ac*6.0
223          acov = (b3*pitch + (d/2.0 + e)/a1)/ac
224 c
225          do 10 i = 1,ni
226             ko = njm1*ni + i
227             dh(ko) = y
228             dv(ko) = yy
229       10 continue
230 c
231          dr3(nj) = dro(nj)
232          dr6(nj) = dr4(nj)/1.5/(n(njm1) -1)/w
233          do 11 j = 1,njm1
234             dr3(j) = (dro(j) + dro(j+1))/2.0
235       11 continue
236 c
237          radfu = radr - thg - thc
238          radcl = radfu + thg
239          ncld  = npin - ncc
240          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))/2.0
         if(k.ne.ng) rodr(k) = (rad(k+1)+rad(k))/(rad(k+1)-rad(k))/2.0
   16 continue
c
      call inecho(nrow,pitch,d,e,ad,apu,dil,thc,thg,lss,tinit,lp)
c
      vm(1) = zero
      vp(1) = drf*drf/8.0
      rm    = (radr + rad(npm1))/2.0
      vm(npin) = (radr*radr + w*w/4.0- rm*rm)/2.0
      vp(npin) = zero
      do 17 k = 2,npm1
         rp = (rad(k+1) + rad(k))/2.0
         rm = (rad(k) + rad(k-1))/2.0
         vp(k) = (rp*rp -rad(k)*rad(k))/2.0
         vm(k) = (rad(k)*rad(k) - rm*rm)/2.0
   17 continue
      if(nres.eq.1) go to 18
      call redum2(tv,tl,p,alfa,uvz,ulz,uvr,ulr,tr,tcan,tw,nn,
     *            ntr,ncan,np,ni,nim2,nj,npin,tinit,lss)
      go to 20
c
   18 continue
c
      read(5,tres)
      qpp = hnw2(2)*radfu*radfu/radr/2.0
      call stead(pin,pout,tin,tav,qpp,p,tv,tl,uvz,ulz,uvr,ulr,
     *           alfa,tw,tr,dtr,dh,dv,nn,np,ntr,npin,npm1)
      write(8,1003)nrow,pitch,d,e
      call redum3(n,dz,tcan,shape,sppd,ppp,lplnm,ncan,nn,npin,
     *            ad,apu,dil,radr,thc,thg,ni)
   20 continue
c
c  computes the time step limitation imposed when the explicit
c  radial heat conduction option is utilized
c
```

```
      if(rnusl1.le.zero) return                                          289
      drmin = dr6(1)                                                     290
      do 37 i = 2,nj                                                     291
   37    drmin = dmin1(drmin,dr6(i))                                     292
      drmin = drmin*drmin/dmax1(rnusl1,rnusl2)*1.8623d+03                293
      dtmax = dmin1(dtmax,drmin)                                         294
      return                                                             295
 1003 format(i5,3d15.9)                                                  296
      end                                                                297
```

```
      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,gamo,nn)
      implicit real*8 (a-h,o-z)
      common /dim/ dz(150),dz1(150),dro(150),dri(150),dr2(150),dr3(150),
     *             dr4(150),dr5(150),dr6(150),ni,nj,nim1,nim2,njm1,nni,
     *             nnj,nnjj
      common /tempo/ time,dt,dto,dtls,sprint,ndt,nres
      common /number/ zero,one,big,small
      common /flbdry/ fbftr(20),explm(20),expvm(20)
      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),gamo(nn)
c
c  subroutine ws complete the evaluation of the explicit terms
c  involved in the solution of the problem stated with subroutine
c  donor.  here are set the terms containing the time increment
c  dt.it is written separately from subroutine donor in order to
c  allow a change in the value of dt when the problrm does not
c  converge with the previous dt.(see next coment in this subroutine.)
c
      do 3 jo = 1,nj
        do 3 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
c  calculate the interfacial and wall friction terms
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   since the program allows a change in the value of the time increment
c   dt,even if the time step is not completed,we put a check here to know
c   if such a change did occur (in this case ndt would be different than
c   zero) in case the test be true,we subtract the terms which have the
c   old dt and add them back with the new value of dt.
c
c   here we have added the effects of mass exchange on the
c   interfacial momentum exchange coefficient.  We assume
c   the interfacial velocity is given by ui = eta*uv +
c   (1 - eta)*ul  where 0 < eta < 1
c
      eta = 0.5
      wz4(ko) = c1z + eta*gamo(ko)
      wz6(ko) = c1z - (1. - eta)*gamo(ko)
      wr4(ko) = c1r + eta*gamo(ko)
      wr6(ko) = c1r - (1. - eta)*gamo(ko)
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
      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 continue
      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)
```

```
    3 continue
c
c   radial direction equations
c
      do 4 jo = 1,njm1
        do 4 io = 2,nim1
          ko = (jo-1)*ni + io
c
          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)
    4 continue
c
c  these terms are only used for the flow boundary condition
c  they are the explicit terms of the non-discretized liquid
c  and vapor momentum equations
c
      do 5 i = 1,nj
         ko = (i-1)*ni + 2
         explm(i) = -(wz7(ko)*wz6(ko) + wz3(ko)*wz8(ko))/wz11(ko)
         expvm(i) = -(wz7(ko)*wz5(ko) + wz4(ko)*wz8(ko))/wz11(ko)
    5 continue
      return
      end
```

```
      subroutine qcond(tl,tc,tcn,tr,all,alc,alr,qic1,qic2,
     *                 ind1,numr)
c
c   this subroutine calculates the radial heat transfer between
c   cells.  If vapor is present in a cell, no heat transfer is
c   assumed.  Both a partially implicit and a full explicit
c   calculation is possible.
c
c      qic1 =   heat transfered per unit volume
c      qic2 =   the derivative of qic1 with respect to tl
c
      implicit real*8 (a-h,o-z)
      common /dim/ dz(150),dz1(150),dro(150),dr1(150),dr2(150),dr3(150),
     *             dr4(150),dr5(150),dr6(150),ni,nj,nim1,nim2,njm1,nni,
     *             nnj,nnjj
      common /nussy/ rnusl1,rnusl2
      common /number/ zero,one,big,small
c
      qic1 = zero
      qic2 = zero
      rnu1 = dabs(rnusl1)
      ti   = tc
      if(ind1.eq.1)  ti = tcn
      if(numr.eq.nj) go to 50
      if(numr.ne.1)  go to 30
   10 continue
      if(alc.ne.zero.or.alr.ne.zero) return
      call htran(tc,tr,h,rnu1,1,2)
      qic2 = -one*dr5(1)*h/dr4(1)
      qic1 = qic2*(ti - tr)
      return
   30 continue
      if(alc.ne.zero) return
      if(all.ne.zero) go to 40
      call htran(tc,tl,h,rnu1,numr,(numr-1))
      qic2 = -one*dr5(numr - 1)*h/dr4(numr)
      qic1 = qic2*(ti - tl)
   40 continue
      if(alr.ne.zero) return
      if(numr.ne.njm1) rnu1 = dabs(rnusl2)
      call htran(tc,tr,h,rnu1,numr,(numr + 1))
      qic2 = qic2 - h*dr5(numr)/dr4(numr)
      qic1 = qic1 + h*dr5(numr)*(tr - ti)/dr4(numr)
      return
   50 continue
      if(alc.ne.zero.or.all.ne.zero) return
      rnu2 = dabs(rnusl2)
      call htran(tc,tl,h,rnu2,nj,njm1)
```

```
      qic2 = -one*dr5(njm1)*h/dr4(nj)
      qic1 = qic2*(ti - t1)
      return
      end
```

```
      subroutine htran(t1,t2,h,rnu,nu1,nu2)
c
c   this subroutine calculates the intercell heat transfer coefficient
c   for subroutine qcond
c
      implicit real*8 (a-h,o-z)
      common /dim/ dz(150),dz1(150),dro(150),dr1(150),dr2(150),dr3(150),
     *             dr4(150),dr5(150),dr6(150),ni,nj,nim1,nim2,njm1,nni,
     *             nnj,nnjj
c
      conv1 = condl(t1)/dr6(nu1)
      conv2 = condl(t2)/dr6(nu2)
c
      h = 2.*rnu*conv1*conv2/(conv1 + conv2)
      return
      end
```

```
      subroutine bc(p,tv,tl,alfa,alfaz,rhovz,rholz,uv,ul,wz9,wz10,
     *              time,itbd,nn)
c
c   this subroutine calculates the boundary conditions as a function
c   of time.  The inlet boundary condtion indicator, itbd, can be
c   either -1,0,1 to indicate a velocity, pressure, or flow boundary
c   condition at the inlet.
c
      implicit real*8 (a-h,o-z)
      common /bcx/ ulo
      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) ,
     *               vb1(51) ,vb2(51) ,vb3(51) ,omvb(51),flb1(51),
     *               flb2(51) ,flb3(51) ,omfb(51),lmax,lp
      common /dim/ dz(150),dz1(150),dro(150),dr1(150),dr2(150),dr3(150),
     *             dr4(150),dr5(150),dr6(150),ni,nj,nim1,nim2,njm1,nni,
     *             nnj,nnjj
      common /flbdry/ fbftr(20),explm(20),expvm(20)
      dimension p(nn),tv(nn),tl(nn),alfa(nn),alfaz(nn),rholz(nn),
     *          rhovz(nn),uv(nn),ul(nn),wz9(nn),wz10(nn)
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)
      if(itbd) 3,5,7
    3 continue
      vb  = vb1(l)*dtime + vb2(l)
      if(lp.eq.1) vb = dexp(omvb(l)*dtime)*vb + vb3(l)
      do 4 j = 1,(nn-nim1),ni
         ul(j) = vb
         uv(j) = vb
    4 continue
      go to 100
    5 continue
      pnb = pnb1(l)*dtime + pnb2(l)
      if(lp.eq.1) pnb = dexp(omp(l)*dtime)*pnb + pnb3(l)
      do 6 j = 1,(nn-nim1),ni
    6    p(j) = pnb
      go to 100
    7 continue
```

```
      flb = flb1(l)*dtime + flb2(l)
      if(lp.eq.1) flb = dexp(omfb(l)*dtime)*flb + flb3(l)
      fbftr1 = 0.d0
      fbftr2 = 0.d0
      do 8 j = 1,nj
         ko  = (j-1)*ni + 2
         fbftr(j) = dr4(j)*((1.d0-alfa(ko))*rholz(ko)*wz10(ko) +
     +              alfa(ko)*rhovz(ko)*wz9(ko))
         fbftr1 = fbftr1 + fbftr(j)
         fbftr2 = fbftr2 + dr4(j)*((1.d0-alfa(ko))*rholz(ko)*
     *             explm(j) + alfa(ko)*rhovz(ko)*expvm(j))
    8 continue
      flb = (fbftr2 - flb)/fbftr1
      do 9 j = 1,nj
        ko = (j-1)*ni + 2
        fbftr(j) = -fbftr(j)/fbftr1
        flb = flb - fbftr(j)*p(ko)
    9 continue
      do 10 j = 1,nj
        ko = (j-1)*ni + 1
        p(ko) = flb
   10 continue
  100 continue
      pnt = pnt1(l)*dtime + pnt2(l)
      alb = alb1(l)*dtime + alb2(l)
      tvb = tvb1(l)*dtime + tvb2(l)
      tlb = tlb1(l)*dtime + tlb2(l)
      if(lp.eq.0) go to 11
c
      pnt = dexp(omt(l)*dtime)*pnt + pnt3(l)
      alb = dexp(oma(l)*dtime)*alb + alb3(l)
      tvb = dexp(omv(l)*dtime)*tvb + tvb3(l)
      tlb = dexp(oml(l)*dtime)*tlb + tlb3(l)
c
   11 continue
      do 12 j = ni,nn,ni
         ko = j - nim1
         p(j)     = pnt
         alfa(ko) = alb
         tv(ko)   = tvb
         tl(ko)   = tlb
   12 continue
      return
      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
      common /dryout/ alpdry
      dimension s(5,2)
      data an,rgas /6.d-04,2.09882d-02/,half /0.50d0/
      data a1,b1 /12020.0,21.9358/
      data pi,sr3,cadry,adry /3.141592654,3.464101616,0.043,0.957/
      data h0,h1 /5.089d+06,-.1043d+04/
      data rnu /10.0/
      data h10,h11,h12,h13 /-6.75075d+04,1.63014d+03,
     *     -.416720,1.54272d-04/
c
c  subroutine noneq calculates the mass and energy exchange rates
c  and its derivatives.
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 = alfa
      if(alfa.lt.1.d-4) ax = 1.e-4
      if(alfa.gt.0.9999) ax = 0.9999
c
      xx = one/(sr3*pod2 - pi)
      ann = an*an*d4*d4*pi*xx/12.
      if(alfa.gt.ann) go to 10
c
c  incipient boiling
c
      area = 3.*ax/an
      darda = 3./an
      go to 60
   10 continue
      if(ax.gt.0.55) go to 20
c
c  bubbly flow correlation
c
      area = d4*dsqrt(3.*pi*xx*ax)
      darda = 0.5*d4*dsqrt(3.*pi*xx)/dsqrt(ax)
      go to 60
   20 continue
```

```
      if(ax.gt.0.65) go to 30
c
c  slug/churn flow transition
c
      ar1 = d4*dsqrt(3.*pi*xx*ax)
      darda1 = 0.5*d4*dsqrt(3.*pi*xx)/dsqrt(ax)
c
      ar3 = sr3*pi*pod2*xx*xx - pi*ax*xx
      ar2 = d4*dsqrt(ar3)
      darda2 = -0.5*d4/dsqrt(ar3)*pi*xx
c
      call poly(ar1,darda1,ar2,darda2,area,darda,alfa)
      go to 60
   30 continue
      if(ax.gt.0.957) go to 40
c
c  annular flow correlation
c
      ar1 = sr3*pi*pod2*xx*xx - pi*ax*xx
      area = d4*dsqrt(ar1)
      darda = -0.5*d4/dsqrt(ar1)*pi*xx
      go to 60
   40 continue
c
c  dryout correlation
c
      ar1    = sr3*pi*pod2*xx*xx - pi*ax*xx
      area   = d4*dsqrt(ar1)*dsqrt(1. - ax)*4.822
      darda1 = -2.411*d4/dsqrt(ar1)*pi*xx*dsqrt(1. - ax)
      darda2 = -2.411*d4*dsqrt(ar1)/dsqrt(1. - ax)
      darda  = darda1 + darda2
   60 continue
c
      ts     = sat(p)
      hlg    = h1*ts + h0
      ftr    = 0.1
      pstl   = dexp(b1 -a1/tl)
      srts   = dsqrt(ts)
c
      coef   = rgas*ftr/(1.0 - 0.5*ftr)
c
      ce = 0.0
      cc = 0.0
      if(alfa.eq.0.0) go to 70
      if(alfa.gt.alpdry) go to 85
      if(tl.gt.ts) ce = 1.0
      if(tl.le.ts) cc = 1.0
      go to 80
```

```
   70 continue
      if(tl.gt.ts) ce = 1.0
   80 continue
c
c  mass exchange rate
c
      ce1 = (pstl - p)/srts
      ce2 = area*coef*(ce + cc)
      ddp = dtsdp(p)
c
      s(1,1) = ce1*ce2
      s(2,1) = 0.0
      s(3,1) =  ce2*a1*pstl/tl/tl/srts
      s(4,1) = -ce2*(0.5*ce1*ddp/ts + 1./srts)
      s(5,1) = darda*coef*(ce + cc)*ce1
      go to 87
   85 continue
      if(tv.gt.ts) ce = 1.0
      if(tv.le.ts) cc = 0.01
      pstv = dexp(b1 - a1/tv)
c
c  mass exchange rate
c
      ce1 = (pstv - p)/srts
      ce2 = area*coef*(ce + cc)
      ddp = dtsdp(p)
c
      s(1,1) = ce1*ce2
      s(2,1) = ce2*a1*pstv/tv/tv/srts
      s(3,1) = 0.0
      s(4,1) = -ce2*(0.5*ce1*ddp/ts + 1./srts)
      s(5,1) = darda*coef*(ce + cc)*ce1
   87 continue
c
c                energy exchange rate
c
      u     = 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 + hl*ddp
      if(alfa.gt.alpdry) go to 90
      s(1,2) = s(1,1)*hv + area*u*(ts - tv)
      s(2,2) = s(2,1)*hv - u*area
      s(3,2) = s(3,1)*hv
      s(4,2) = s(4,1)*hv + s(1,1)*dhvdp + area*u*ddp
      s(5,2) = s(5,1)*hv + darda*u*(ts - tv)
      return
```

```
 90 continue                                                                  145-
    u       = condv(tl)*rnu*d4                                                146
    s(1,2) = s(1,1)*hl + area*u*(tl - ts)                                     147
    s(2,2) = s(2,1)*hl                                                        148
    s(3,2) = s(3,1)*hl + area*u                                               149
    s(4,2) = s(4,1)*hl + s(1,1)*dhldp - area*u*ddp                            150
    s(5,2) = s(5,1)*hl + darda*u*(tl - ts)                                    151
    return                                                                    152
    end                                                                       153
```

```
      subroutine poly(one,two,three,four,area,darda,alfa)
c
c   this subroutine performs a polynomial fit for the area and the
c   the derivative of the area in the bubbly/annular flow transition
c
      implicit real*8 (a-h,o-z)
      a = one
      b = two
      c = 3*(three - one) - four - 2.*two
      d = four +two + 2*(one - three)
c
      x = 10.*(alfa - .55)
c
      area = a +b*x + c*x*x + d*x*x*x
      darda = b + 2.*c*x + 3.*d*x*x
c
      return
      end
```

```
      subroutine direct(a1,a2,a3,a4,f,x,nc,z1,z2,z3,z4,bbb,
     *                  aaa,x1,dpbd,itbd,nband1,nband2,ndds)
      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,igauss,indgs
      common /flbdry/ fbftr(20),explm(20),expvm(20)
      dimension a1(nc),a2(nc),a3(nc),a4(nc),f(nc),x(nc),
     *          x1(ndds,nband2),z1(ndds),z2(ndds),z3(ndds),
     *          z4(ndds),bbb(ndds),aaa(ndds,nband1)
c
      dpbd = zero
   10 continue
c
c  this subroutine solves the pressure problem by use of a
c  direct solution, using library subroutine leqt1b.
c
c  rearrange numbering of cells to minimize the
c  bandwidth.
c
      do 30 j=1,nr
         do 20 i=1,nz
c
            inc1 = nr*(i-1) + j
            inc2 = nz*(j-1) + i
c
            z1(inc1) = a1(inc2)
            z2(inc1) = a2(inc2)
            z3(inc1) = a3(inc2)
            z4(inc1) = a4(inc2)
            bbb(inc1) = f(inc2)
c
   20    continue
   30 continue
c
      if(itbd.ne.1) go to 36
      do 35 i = 1,(ndds-1)
         ko = ndds + 1 - i
         z1(ko) = z1(ko - 1)
         z2(ko) = z2(ko - 1)
         z3(ko) = z3(ko - 1)
         z4(ko) = z4(ko - 1)
         bbb(ko) = bbb(ko - 1)
   35 continue
      z1(1) = zero
      z2(1) = zero
      z3(1) = zero
```

```
      z4(i) = zero
      bbb(i) = zero
   36 continue
c
c  set values for input data
c
      ier  = 0
      ijob = 0
      n    = ndds
      nlc  = nr
      nuc  = nlc
      ia   = ndds
      ib   = ndds
      m    = 1
c
c   set up matrix aaa
c
      j0 = nr + 1
      j1 = 1
      j2 = nr
      j3 = nr + 2
      j4 = 2*nr + 1
c
c  initialize matrix aaa to zero
c
      nband = 2*nr + 1
      do 50 j = 1,ndds
         do 40 k = 1,nband
            aaa(j,k) = zero
   40    continue
   50 continue
c
c  input band components
c
      do 60 i = 1,ndds
         aaa(i,j1) = z2(i)
         aaa(i,j2) = z1(i)
         aaa(i,j3) = z4(i)
         aaa(i,j4) = z3(i)
         aaa(i,j0) = one
   60 continue
c
c  for a flow boundary condition, it is necessary to add an
c  additional equation to the pressure field matrix, in
c  order to update the boundary pressure
c
      if(itbd.ne.1) go to 70
      do 63 j = 1,nr
```

```
847          aaa(1,(nband2+j)) = fbftr(j)
848       63 continue
849          do 68 j = 2,nr
850             nb1 = (2-j) + nr
851             aaa(j,nb1) = aaa(j,1)
852             aaa(j,1) = 0.0d0
853       68 continue
854       70 continue
855    c
856    c  at this point a call to the library subroutine
857    c  leqt1b is made
858    c
859          call leqt1b(aaa,n,nlc,nuc,ia,bbb,m,ib,ijob,xl,ier)
860    c
861    c   check the results
862    c
863          if (ier.eq.129) ierr = 1
864    c
865          if(itbd.ne.1) go to 90
866          dpbd = bbb(1)
867          do 80 i = 2,ndds
868             bbb(i-1) = bbb(i)
869       80 continue
870       90 continue
871    c
872    c     now convert back the results to the old numbering scheme
873    c
874          do 110 j = 1,nr
875             do 100 i = 1,nz
876    c
877                inc1 = nr*(i - 1) + j
878                inc2 = nz*(j - 1) + i
879    c
880                x(inc2) = bbb(inc1)
881      100    continue
882      110 continue
883          res = 0.0d0
884          do 120 i = 1,nc
885            xx = dabs(x(i))
886            if(xx.gt.res) res = xx
887      120 continue
888          return
889          end
```