ANL-79-91

ARGONNE NATIONAL LABORATORY
9700 South Cass Avenue
Argonne, Illinois 60439

**LIMBO Computer Code for
Analyzing Coolant-voiding Dynamics
in LMFBR Safety Tests**

by

G. L. Bordner

Reactor Analysis and Safety Division

October 1979

TABLE OF CONTENTS

	<u>Page</u>
ABSTRACT	5
I. INTRODUCTION	5
II. GOVERNING EQUATIONS	7
A. Description of the Model	7
B. Voiding Kinematics	7
C. Hydrodynamics	8
D. Fuel-element Heat Transfer	9
E. Coolant Liquid and Vapor Energy	10
F. Shroud-assembly Temperatures	11
III. METHODS OF SOLUTION	13
A. Voiding Equation	13
B. Structure and Coolant Temperatures	14
C. Channel Velocity and Pressure Distributions	14
D. Stability of Explicit Difference Equations	15
IV. SAMPLE RESULTS OF LIMBO CODE	16
A. P2 Posttest Analysis	16
B. P3A Pretest Analysis	18
APPENDIX A: A User's Guide for LIMBO	19
APPENDIX B: Listing of LIMBO Deck	21
REFERENCES	32

LIST OF FIGURES

<u>No.</u>	<u>Title</u>	<u>Page</u>
1.	Nodal Configuration for LIMBO LMFBR Subassembly Model	7
2.	Schematic Depicting Coolant Heat-transfer Paths within a Partially Voided Control Volume ...	10
3.	Schematic of Fluid Motion at an Advancing Void Front	13
4.	Comparison of SAS-3A Predicted and Observed Bundle Inlet Flow during P2 LOF Transient ...	16
5.	Effects of Test-section Boundary Conditions on Bundle Inlet Flow as Predicted by LIMBO	17
6.	Comparison of Measured and Computed Bundle Inlet Flow during P2 LOF Transient	17
7.	Comparison of Measured and Computed Bundle Exit Temperatures during P2 LOF Transient .	18
8.	Comparison of Measured and Predicted Bundle Inlet Flow during P3A LOF Transient	18

LIST OF TABLES

<u>No.</u>	<u>Title</u>	<u>Page</u>
I.	Data Input List for LIMBO	19
II.	List of LIMBO Output Variables	20

LIMBO Computer Code for Analyzing Coolant-voiding Dynamics in LMFBR Safety Tests

by

G. L. Bordner

ABSTRACT

This report describes the LIMBO (liquid metal boiling) code for the analysis of two-phase flow phenomena in an LMFBR reactor coolant channel. The code uses a nonequilibrium, annular, two-phase flow model, which allows for slip between the phases. Furthermore, the model is intended to be valid for both quasi-steady boiling and rapid coolant voiding of the channel.

The code was developed primarily for the prediction of, and the posttest analysis of, coolant-voiding behavior in the SLSF P-series in-pile safety test experiments. The program was conceived to be simple, efficient, and easy to use. It is particularly suited for parametric studies requiring many computer runs and for the evaluation of the effects of model or correlation changes that require modification of the computer program. The LIMBO code, of course, lacks the sophistication and model detail of the reactor safety codes, such as SAS, and is therefore intended to complement these safety codes.

I. INTRODUCTION

The analysis of boiling in an LMFBR reactor coolant channel is made only in connection with reactor safety or the evaluation of safety-related experiments because the severe conditions of overheating or undercooling required for boiling can only occur during various hypothesized accident situations. Several flow regimes are encountered in boiling transients. The onset of boiling is usually followed by a quasi-steady boiling phase, which is characterized by a relatively uniform axial (mass) flow distribution. In severe transients, the rate of vapor generation can increase to the point of channel pressurization and bubble coalescence, which results in an inlet-flow reversal and the rapid expulsion of liquid coolant from the subassembly.

The early sodium-boiling programs were patterned after the continuous-void-distribution models used previously for the analysis of boiling-water reactors. Examples of this genre of programs are TRANS-UGUE-5,¹ SAS-1A,² and the unnamed program of Siegmann.³ The developers of these programs encountered several problems which are peculiar to the analysis of sodium boiling. The first problem is related to the fact that these models are formulated in terms of parameters that are discontinuous across a voiding front (mass flow, vapor quality, mean density, etc.), which results in numerical problems when analyzing the voiding phase of a transient. MacFarlane, in the SAS-1A coolant dynamics model, overcame this problem by partitioning the channel into single-phase and voided sections and by using separate sets of equations for each section.

Another problem encountered in the development of sodium-boiling codes is a numerical instability due to the strong coupling between the momentum and energy equations, which is associated with the large volume change accompanying the vaporization of sodium at nominal pressures. Noyes et al.¹ ignored the pressure dependence of the saturation temperature, which uncoupled the above-mentioned equations but resulted in the overprediction of the rate of coolant expulsion. MacFarlane stabilized the solution by using a semiexplicit (Crank-Nicolson) numerical scheme.

The single-bubble models offer an alternative approach to the modeling of coolant dynamics and more closely simulate the physics of the coolant expulsion phase. BURP,¹ FLOW 2,⁴ and SLUG⁵ represent successive refinements of the single-bubble class of models. These models typically allow for liquid superheat (nonequilibrium), which affects the initial voiding rate as well as the choking velocity for two-phase flow. In addition, the last two models referred to above allow for a residual stagnant liquid film, which provides the major source of vapor during the latter stages of coolant expulsion.

The coolant dynamics models in SAS-2A⁶ and SAS-3A⁷ represent further refinements of the earlier bubble models. The SAS-2A code has a multibubble model which is necessary for the simulation of the early quasi-steady phase of the boiling transient. The SAS-3A model also allows for film motion, which has the effect of advancing the predicted dryout time for the bundle and consequently reducing the vapor production rate during the final stages of the voiding transient. The SAS programs use two sets of equations for computing liquid temperatures: Eulerian equations for the preboiling phase and the lower liquid slug, and Lagrangian equations for the bubble interface and upper liquid slugs.

The LIMBO code of the present study borrows its physical model for annular, nonequilibrium two-phase flow from the SAS bubble model, but uses the simpler Eulerian mathematical representation of the flow field which characterizes the continuous-void-distribution codes used for boiling-water channels. The governing equation for the void fraction (vapor volume fraction) can be solved in either of two modes: as a continuous-void-distribution model or as a coolant-expulsion model with discontinuities in the void fraction at the void fronts. With these two modes for solving the void fraction equation, the model should be valid for both the quasi-steady boiling and the coolant-expulsion phases of the boiling transient.

LIMBO avoids the complication of having two separate equations for the single- and two-phase regions by formulating the equations in terms of parameters, such as velocity, temperature, and pressure, which are continuous across a voiding front (the void-fraction parameter is, of course, an exception). The finite-difference equations for most of the parameters are of the simple explicit form; the momentum and vapor energy equations, however, are solved implicitly to eliminate the numerical instability due to the strong coupling of these two equations.

II. GOVERNING EQUATIONS

A. Description of the Model

The LIMBO model is a single-channel representation of an SLSF test subassembly. The configuration of the model is shown in Fig. 1. The model is presently set up to analyze the unprotected loss-of-flow transient and consequently assumes that the bundle power and bundle-inlet temperature are fixed. The flow coastdown is simulated by varying the bundle-inlet and -exit pressures; the model computes the subassembly flow to satisfy the pressure boundary conditions.

Fig. 1. Nodal Configuration for LIMBO LMFBR Subassembly Model

Some features of the computer model are:

1. A void fraction model, which can simulate either quasi-steady boiling or single-bubble coolant expulsion.
2. A liquid heat transfer model, which allows for liquid superheat.
3. The Lockhart-Martinelli correlation for interphase slip, which allows for film motion.
4. A pressure-dependent boiling temperature.
5. A simple Eulerian finite-difference representation of the flow and temperature fields.

B. Voiding Kinematics

The continuity equations for the liquid and gas phases, respectively, are given by

$$\left. \begin{aligned} & \frac{\partial}{\partial t}[\rho_L(1 - \alpha)] + \frac{\partial}{\partial z}[\rho_L(1 - \alpha)V_L] = -\dot{w}/A \\ & \text{and} \\ & \frac{\partial}{\partial t}(\rho_G \alpha) + \frac{\partial}{\partial z}(\rho_G \alpha V_G) = \dot{w}/A \end{aligned} \right\}, \quad (1)$$

where $\dot{w}$ is the mass rate of vaporization per unit length of channel.

Rearranging the above equations and assuming constant phase densities, one obtains the relations

$$\frac{\partial \bar{V}}{\partial z} = (\beta - 1)S \quad (2)$$

and

$$\frac{\partial \alpha}{\partial t} = \frac{\partial F}{\partial z} + S, \quad (3)$$

where

$$\bar{V} = \alpha V_G + (1 - \alpha)V_L,$$

$$\beta = \rho_L / \rho_G,$$

$$S = \dot{w}/A\rho_L,$$

and

$$F = (1 - \alpha)V_L.$$

The mean velocity $\bar{V}$ is continuous across a voiding front and is constant when there is no vaporization such as for air/water-mixture flows. From the above definition of $\bar{V}$, we obtain the following relation for the liquid-phase velocity in terms of the mean velocity:

$$V_L = \bar{V}/(1 - \alpha + \sigma\alpha), \quad (4)$$

where

$$\sigma = V_G/V_L.$$

The relationship between the void fraction and slip ratio σ is obtained from MacFarlane's² empirical fit of the Martinelli correlation:

$$\frac{1}{(1 - \alpha)^2} = 1 + \frac{21}{X_{tt}} + \frac{1}{X_{tt}^2}. \quad (5)$$

The Lockhart-Martinelli two-phase flow parameter X_{tt} is expressed in terms of the slip ratio and void fraction by

$$X_{tt} = \beta^{0.4} \sigma^{0.9} \left(\frac{\mu_L}{\mu_G} \right)^{0.1} \left(\frac{1 - \alpha}{\alpha} \right)^{0.9}. \quad (6)$$

C. Hydrodynamics

The pressure distribution in a boiling channel is governed by the two-phase hydrodynamic equation

$$\alpha \rho_G \frac{\partial V_G}{\partial t} + (1 - \alpha) \rho_L \frac{\partial V_L}{\partial t} + \alpha \rho_G V_G \frac{\partial V_G}{\partial z} + \beta \frac{\partial \rho}{\partial z} = -g(1 - \alpha)\rho_L - f_0 \rho_L V_0^{0.2} |V_L|^{0.8} V_L / 2D \quad (7)$$

where f_0 is the (constant) friction factor based on the reference velocity V_0 for single-phase liquid flow. A term involving $\partial V_L / \partial z$ has been omitted from Eq. 7 by virtue of the incompressibility of the liquid slugs. However, the term involving the gradient of V_G has been retained because this velocity varies in the voided region due to vaporization and condensation.

The model allows for inertial and friction effects external to the coolant channel. In the equations below, p_{in} and p_{out} are the given inlet and exit pressures. The inertial and friction losses are expressed in terms of equivalent channel lengths; the Z 's and L 's are the inertial lengths and friction lengths, respectively. Single-phase flow outside of the channel is assumed.

$$p = p_{in} - (Z_{in} \rho_L / g) \frac{dV_{in}}{dt} - f_0 L_{in} \rho_L V_0^{0.2} |V_{in}|^{0.8} V_{in} / 2gD, \text{ at } z = 0, \quad (8)$$

$$p = p_{out} + (Z_{out} \rho_L / g) \frac{dV_{out}}{dt} + f_0 L_{out} \rho_L V_0^{0.2} |V_{out}|^{0.8} V_{out} / 2gD, \text{ at } z = L. \quad (9)$$

D. Fuel-element Heat Transfer

The fuel is represented by a single radial node, which is sufficient for slow transients such as the hypothetical loss-of-flow accident. The governing equation for the local mean fuel temperature is

$$(MC)_f \frac{dT_f}{dt} = q' - (UA)_f (T_f - T_{cl}), \quad (10)$$

where

$$(MC)_f = n\pi d_i^2 \rho_f C_f / 4,$$

$$(UA)_f = n\pi d_i / [d_i / 8k_f + 1/h_f + (d_0 - d_i) / 4k_{cl}],$$

and q' is the total bundle power per unit length at the elevation of the particular fuel node.

The cladding temperature for single-phase flow ($\alpha = 0$) is given by the following equation. This equation is not used, per se, in the program, because it is built into the two-phase equation, which is given later.

$$(MC)_{cl} \frac{dT_{cl}}{dt} = (UA)_f (T_f - T_{cl}) - (UA)_l (T_{cl} - T_L), \quad (11)$$

where

$$(MC)_{cl} = n\pi (d_0^2 - d_i^2) \rho_{cl} C_{cl} / 4,$$

and

$$(UA)_l = n\pi d_0 / [(d_0 - d_i) / 4k_{cl} + 1/h_L].$$

Figure 2 shows the assumed configuration for a partially voided control volume where $0 < \alpha < \alpha_c$. This situation assumes that part of the heat lost from the cladding is transferred directly to the liquid/vapor interface within the voided fraction of the control volume, equal to α/α_c . The remaining heat loss is transferred to the liquid slug which occupies a fraction of $(1 - \alpha/\alpha_c)$ of the length of the control volumes. With these considerations, the cladding temperature is given by

$$(MC)_{cl} \frac{dT_{cl}}{dt} = (UA)_f(T_f - T_{cl}) - (1 - \alpha/\alpha_c)(UA)_1(T_{cl} - T_L) - (\alpha/\alpha_c)(UA)_2(T_{cl} - T_{sat}), \quad (12)$$

for

$$0 < \alpha < \alpha_c.$$

Fig. 2. Schematic Depicting Coolant Heat transfer Paths within a Partially Voided Control Volume

After complete voiding of control volume, $\alpha \geq \alpha_c$, the small amount of remaining liquid coolant is lumped with the cladding and the hexcan. This eliminates the instability of the coolant-temperature equation associated with the vanishing heat capacity of the remaining liquid. Defining $(1 - \alpha_D)$ as the liquid-void fraction associated with the hexcan and $(\alpha_D - \alpha)$ as that associated with the cladding, the cladding temperature is given by

$$[(MC)_{cl} + (\alpha_D - \alpha)(MC)_L] \frac{dT_{cl}}{dt} = (UA)_f(T_f - T_{cl}) - (UA)_2(T_{cl} - T_{sat}), \quad (13)$$

for

$$\alpha_c \leq \alpha < \alpha_D,$$

where

$$(MC)_L = A \rho_L C_L.$$

The final case of $\alpha \geq \alpha_D$ corresponds to the local dryout of the fuel pin. For this situation, the cladding outer surface is assumed to be adiabatic and the cladding temperature is given by

$$(MC)_{cl} \frac{dT_{cl}}{dt} = (UA)_f(T_f - T_{cl}), \quad (14)$$

for

$$\alpha \geq \alpha_D.$$

E. Coolant Liquid and Vapor Energy

The vaporization rate is zero for $T_L < T_{sat}$ (and $\alpha = 0$) and $\alpha > \alpha_D$, which corresponds to dryout.

When the local liquid temperature exceeds the boiling temperature, boiling is assumed to begin. From the onset of boiling until the control volume is completely voided there are three paths for heat transfer to the void interface (see Fig. 2). They are through the interface of the liquid films on the cladding, and on the hexcan, and through the end of the void. The transfer of heat from the liquid slug to the interface,

through the end of the bubble, is necessary to initiate the growth of the bubble, but contributes little to the vaporization rate once the void has grown to a significant size. With these considerations, the initial vaporization rate is given by

$$\lambda \dot{W} = (\alpha/\alpha_c)[(UA)_2(T_{cl} - T_{sat}) + (UA)_{sh1}(T_{sh} - T_{sat})] + (1 - \alpha/\alpha_c)(UA)_3(T_L - T_{sat}) \quad (15)$$

for

$$0 < \alpha < \alpha_c,$$

where λ is the latent heat of vaporization

Following voiding of the control volume and before dryout, the vaporization rate is governed by the relationship

$$\lambda \dot{W} = (UA)_2(T_{cl} - T_{sat}) + (UA)_{sh1}(T_{sh} - T_{sat}), \quad (16)$$

for

$$\alpha_c < \alpha < \alpha_D.$$

The local saturation temperature is allowed to depend upon the local pressure in the coolant channel. The following simple relation for the saturation temperature is valid for moderate pressure ranges:

$$T_{sat} = C_1 + C_2 \ln(p). \quad (17)$$

The liquid-coolant temperature is governed by the equation

$$\begin{aligned} (1 - \alpha)(MC)_L \left(\frac{\partial T_L}{\partial t} + V_L \frac{\partial T_L}{\partial z} \right) = (1 - \alpha/\alpha_c) \left[(UA)_1(T_{cl} - T_L) \right. \\ \left. + (UA)_{sh1}(T_{sh} - T_L) + (UA)_3(T_L - T_{sat}) \right], \end{aligned} \quad (18)$$

for

$$0 < \alpha < \alpha_c.$$

For single-phase flow ($\alpha = 0$, $T_L < T_{sat}$), Eq. 15 is used with the last term deleted. After voiding of the control volume ($\alpha > \alpha_c$), the remaining liquid is lumped with the cladding and the hexcan, as was indicated previously.

The inlet-boundary condition of $T_L = T_{in}$ at $Z = 0$, where T_{in} is a constant, is applied when the inlet velocity is positive. If backflow should occur at the channel exit, the entering coolant at $Z = L$ is assumed to be at the original steady-state discharge temperature.

F. Shroud-assembly Temperatures

The LIMBO model analyzes the shroud configuration used in the SLSF tests, as shown in Fig. 1, which consists of a hexcan, a ceramic insulator, and an outer shroud tube, which is assumed to be at the bypass coolant temperature. The following relations govern the temperature of the hexcan:

$$\begin{aligned} (MC)_{sh} \frac{dT_{sh}}{dt} = (1 - \alpha/\alpha_c)(UA)_{sh1}(T_L - T_{sh}) + (\alpha/\alpha_c)(UA)_{sh1} \\ \cdot (T_{sat} - T_{sh}) - (UA)_{sh2}(T_{sh} - T_{ins}), \end{aligned} \quad (19)$$

for

$$0 \leq \alpha < \alpha_c,$$

and

$$[(MC)_{sh} + (1 - \alpha_D)(MC)_L] \frac{dT_{sh}}{dt} = (UA)_{sh1}(T_{sat} - T_{sh}) - (UA)_{sh2}(T_{sh} - T_{ins}), \quad (20)$$

for

$$\alpha \geq \alpha_c.$$

The local radially averaged insulator temperature is given by

$$(MC)_{ins} \frac{dT_{ins}}{dt} = q'_{ins} + (UA)_{sh2}(T_{sh} - T_{ins}) - (UA)_{sh3}(T_{ins} - T_{in}). \quad (21)$$

III. METHODS OF SOLUTION

A. Voiding Equation

The finite-difference representation of Eq. 3 is

$$\alpha'_i = \alpha_i + (F_i - F_{i-1})\Delta t/\Delta z + S_i\Delta t, \quad (22)$$

where the prime on α denotes the updated (at $t + \Delta t$) value. The function F , as given by

$$F_i = (1 - \alpha_i)v_{L,i}, \quad (23)$$

is used for the following cases:

1. For all nodes during the period of quasi-steady flow (continuous void distribution).
2. For nodes within the voided (annular flow) section of the channel during the coolant expulsion phase of the transient.

The flow configuration for the leading leveling edge of the void is shown in Fig. 3. The local void fraction behind the void front is α_B , and the average void fraction for the node is α_i , where

$$\alpha_i = \delta\alpha_B/\Delta z$$

and δ is the position of the void front measured from z_{i-1} at time t .

Fig. 3. Schematic of Fluid Motion at an Advancing Void Front

The motion of the void front is simulated by special relations for F_i used at the void front. Note that the equation for α is in conservative form and, as a consequence, the changing method of computing F does not violate the laws of conservation of mass. Two alternative values of F_i are used, depending upon whether the void front crosses the boundaries of the control volume.

The following approximations apply to the control volume at the node front. The front moves with the vapor velocity

$$\frac{\partial \alpha_i}{\partial t} \approx \frac{\alpha_B}{\Delta z} \frac{d\delta}{dt} = \alpha_B v_{G,i}/\Delta z, \quad (24)$$

and F_{i-1} is given approximately by

$$F_{i-1} \approx (1 - \alpha_B) V_{L,i}. \quad (25)$$

The value of F_i that yields Eq. 24, using Eqs. 22 and 25 and ignoring S_i , is

$$F_i = \bar{V}_i,$$

for

$$0 < \alpha_i < \alpha_B \text{ and } \delta' < \Delta z.$$

For the situation in which the void front crosses the control-volume boundaries, the correct value of F_i is that which yields $\alpha'_i \approx \alpha_B$ from Eqs. 22 and 25:

$$F_i = \frac{\Delta z}{\Delta t} (\alpha_B - \alpha_i) + \frac{\bar{V}_i (1 - \alpha_B)}{1 + (\sigma - 1) \alpha_B} \quad (27)$$

for

$$0 < \alpha_i < \alpha_B \text{ and } \delta' > \Delta z.$$

B. Structure and Coolant Temperatures

Equations 10-14 and 18-21, which comprise the coolant and structure energy relations, are solved by an explicit or Eulerian integration in time. The space derivative in the convective term of the coolant temperature equation is represented by a backward (or upstream) difference scheme.

The finite-difference representation for the fuel temperature (Eq. 10) is given below as an example. The time derivative is approximated by

$$\frac{\partial T_f}{\partial t} \approx \frac{T'_{f,i} - T_{f,i}}{\Delta t}, \text{ at } z = z_i.$$

Evaluating Eq. 10 at time t and elevation z_i , substituting the above relation for the derivative, and solving for $T'_{f,i}$, we obtain

$$T'_{f,i} = T_{f,i} + [q_i - (UA)_f (T_{f,i} - T_{cl,i})] \Delta t / (MC)_f. \quad (28)$$

C. Channel Velocity and Pressure Distributions

The pressure distribution in the channel is determined by the hydrodynamic Eq. 7. In addition, this equation is indirectly affected by the pressure through its effect on the vaporization rate, and hence, on the velocity distribution. This strong coupling between Eqs. 2, 7, and 15-17 would produce numerical instability if the equations were evaluated at time t (explicit). To avoid this difficulty, the offending equations are evaluated at time $t + \Delta t$ (implicit). This scheme, of course, requires the simultaneous solution of these equations.

The solution of these equations at each time step, subject to the boundary conditions 8 and 9, begins with an assumed inlet velocity and inlet pressure. The exit pressure obtained by integration of the equations does not, in general, agree with the value given by the boundary condition. The inlet velocity is subsequently adjusted by a Newton-Raphson iteration technique to obtain the instantaneous velocity that satisfies the channel end conditions on pressure. Nonlinear terms involving the variables p'_i and $V'_{L,i}$ have been linearized and, as a result, the converged value of the inlet velocity is obtained on the first iteration.

D. Stability of Explicit Difference Equations

The LIMBO code uses different time increments during the single-phase and boiling stages of the transient to provide maximum program-running efficiency. During the single-phase stage of the transient, the maximum stable time step is controlled by the liquid-temperature equation and is given approximately by

$$\Delta t < [V_L/\Delta z + (UA_L + UA_{shl})/(MC)_L]^{-1}.$$

The stability of the void-fraction equation becomes a consideration with the onset of boiling, and the time step must be reduced from the single-phase value. Based on the fact that the void front propagates with the vapor velocity, a reasonable estimate for the stable time step for two-phase flow is

$$\Delta t < \Delta z/V_G.$$

IV. SAMPLE RESULTS OF LIMBO CODE

Sample results are given for the posttest analysis of the SLSF P2 experiment and the pretest analysis of the P3A experiment. Additional results can be found in the cited references. The discussion of these results will provide the prospective user of the code with an indication of both the capabilities and the limitations of LIMBO as an analytical tool.

A. P2 Posttest Analysis

Both the pretest predictions⁸ that used SAS3A and the test results from the R-series TREAT experiments⁹ had indicated that the onset of boiling in the P2 loss-of-flow experiment would be immediately followed by an inlet-flow reversal and a rapid explosion of coolant from both the heated section and the gas-plenum section of the test bundle. The comparison of the predicted and of several bundle inlet flows in Fig. 4 shows a slower-than-predicted flow decay after boiling at 6.3 s. Other atypical results of the P2 test were a slow boilout of sodium from the heated sodium and an incomplete voiding of the upper bundle.

Fig. 4. Comparison of SAS-3A Predicted and Observed Bundle Inlet Flow during P2 LOF Transient. ANL Neg. No. 900-77-1811.

The LIMBO program was used in the P2 posttest analysis¹⁰ to test various hypotheses that were proposed to explain the anomalous voiding behavior. The effect of these hypotheses on the computed bundle inlet flow is shown in Fig. 5. Case 1 is approximately equivalent to the pretest analysis case in which the hexcan is adiabatic and the void fraction is radially homogeneous ($\alpha_D = 1$). Case 2 shows the effect of a radial heat loss through the shroud due to a previous sodium leak into this structure. Case 3 shows the additional effect of radial inhomogeneity ($\alpha_D = 0.9$). These results indicate that both the effects of the radial heat loss and the radial inhomogeneity need to be accounted for (Case 3) to predict the voiding behavior in P2.

Fig. 5. Effects of Test-section Boundary Conditions on Bundle Inlet Flow as Predicted by LIMBO

Figure 6 demonstrates the reasonable agreement between LIMBO (Case 3) and the observed bundle inlet flow. The comparison of predicted and observed bundle exit temperatures in Fig. 7 shows that LIMBO correctly predicts a subcooled bundle exit temperature which is a result of the incomplete voiding of the upper bundle. Other LIMBO results¹² similarly show good agreement with the data.

Fig. 6. Comparison of Measured and Computed Bundle Inlet Flow during P2 LOF Transient

Fig. 7. Comparison of Measured and Computed Bundle Exit Temperatures during P2 LOF Transient

B. P3A Pretest Analysis

The pretest LIMBO prediction of the P3A voiding behavior¹² was the first opportunity to verify the LIMBO program and the interpretation of the P2 data. The model and assumptions used for P3A are the same as those used in P2, except for adjustments in the input to reflect the differences in the test configurations. The larger number of pins in P3A (37 versus 19 for P2) suggested a larger α_D (~ 0.93) to reflect the larger area fraction occupied by internal bundle coolant channels.

The LIMBO prediction for P3A was a rapid flow decay after boiling accompanied by a fast bundle coolant expulsion. The faster expulsion in P3A, compared with P2, is due principally to the negligible bundle heat loss in P3A. Figure 8 shows good agreement between the predicted and observed bundle inlet flow during the P3A safety test. Other comparisons between LIMBO and the P3A data are given in Ref. 13. The data input for the P3A analysis is listed in Appendix B after the program listing.

Fig. 8. Comparison of Measured and Predicted Bundle Inlet Flow during P3A LOF Transient

APPENDIX A

A User's Guide for LIMBO

This appendix tabulates the input and output variables, along with the definitions of the variables and the units to be used. Variables are listed in the order in which they appear in the code input or output. Note that English units were used. The conversion factors and physical constants are all located at the front of the FORTRAN source program before the initial-condition section to facilitate conversion of the deck to metric units.

TABLE I. Data Input List for LIMBO

Item	Definition	Units
IE	Number of axial nodes	-
IFB	Bottom fuel node	-
IFT	Top fuel node	-
NT	Number of time steps in the transient	-
NDATA	Number of pressure-vs-time entries	-
DELT1	Time step for nonboiling phase of transient	s
DELT2	Time step for boiling phase of transient	s
LNGLTH	Total bundle length	in.
AREA	Bundle coolant flow area	in. ²
DHYD	Bundle hydraulic diameter	in.
ZINER1,2	Equivalent inertial lengths for channel inlet and exit, respectively	in.
ZFRIC1,2	Equivalent friction lengths for channel inlet and exit, respectively	in.
VREF	Reference velocity for friction factor	ft/s
FF	Moody friction factor at reference velocity	-
TIN	Bundle inlet coolant temperature	°F
UAF	Fuel-to-cladding heat-transfer factor	Btu/s-ft-°F
UA1	Cladding-to-liquid coolant heat-transfer factor	Btu/s-ft-°F
UA2	Cladding-to-void interface heat-transfer factor	Btu/s-ft-°F
UA3	Liquid-to-hubble heat-transfer factor	Btu/s-ft-°F
EMCF	Thermal capacity of fuel	Btu/ft-°F
EMCL	Thermal capacity of cladding	Btu/ft-°F
RHOL	Liquid coolant density	lb/ft ³
BETA	Ratio of liquid and vapor densities	-
CP	Specific heat capacity of coolant	Btu/lb-°F
LMBDA	Coolant latent heat of vaporization	Btu/lb
C1, C2	Constants in Eq. 17 for TSAT	-
VISL	Liquid viscosity	any consistent
VISG	Gas viscosity	units
ALFAB	Void fraction at void front	-
ALFAC	Void fraction marking transition to annular flow	-
ALFAD	Void fraction at bundle dryout	-
UASH1	Coolant-to-hexcan heat-transfer factor	Btu/s-ft-°F
UASH2	Hexcan-to-insulator heat-transfer factor	Btu/s-ft-°F
UASH3	Insulator-to-bypass heat-transfer factor	Btu/s-ft-°F
EMCSH	Thermal capacity of hexcan	Btu/ft-°F
EMCINS	Thermal capacity of insulator	Btu/ft-°F
QP(I)	Nodal fuel heat generation	Btu/s-in.
TIME(N)	Times for pressure boundary conditions	s
PIND(N)	Inlet pressure data	psia
POUTD(N)	Outlet pressure data	psia

TABLE II. List of LIMBO Output Variables

Item	Definition	Units
TIME	Time from start of transient	s
FLØ	Bundle inlet flow rate	lb/s
PIN	Instantaneous inlet pressure	psfa
PØUT	instantaneous outlet pressure	psfa
UØUT	Bundle outlet velocity	ft/s
Z(I)	Nodal elevations in bundle	ft
ALFA(I)	Void volume fraction	-
VAVG(I)	Mean velocity $\bar{V}$	ft/s
VL(I)	Liquid phase velocity	ft/s
VG(I)	Gas phase velocity	ft/s
P(I)	Local pressure	psfa
TF(I)	Radial-averaged fuel temperature	°F
TCL(I)	Cladding temperature	°F
TL(I)	Liquid coolant temperature	°F
TSH(I)	Shroud hexcan temperature	°F
TINS(I)	Shroud insulator temperature	°F
TSAT(I)	Local saturation temperature	°F

APPENDIX B**Listing of LIMBO Deck**

The following is a listing of FORTRAN source deck and data-input deck that were used for the LIMBO pretest analysis of the SLSF-P3A safety experiment.

```

      COMPILER OPTIONS - NAME=  MAIN,OPT=00,LINECNT=57,SIZE=0000K,
                           SOURCE,EBCDIC,NOLIST,NODECK,LOAD,MAP,NOEDIT,IO,NOXREF
ISN 0002      IMPLICIT REAL*8 (A-H,O-W,Z)
ISN 0003      REAL*8  LENGTH,LMBDA ,LTOT
ISN 0004      COMMON PIN,RHOL,ZINER1,ZINER2,GC,DELT,DELZ,BETA,ALFAC,FF,DHYD,
1VL(42),VG(42),S(41),ALFA(41),TLP(41),TCLP(41),TSAT(41),PP(41)
2 ,LMBDA,AREA,VLP(42),      P(41),TSATP(41),C1,C2,VGP(42)
3 ,UA1,UA2,UA3,VAVG(42),ZFRIC1,ZFRIC2,TSHP(41),UASH1,VOUT,VOUTP
4 ,VISL,VISG,VREF,ALFAD,ALFAB
ISN 0005      DIMENSION TIMED(20),      Z(41),TL(41),TCL(41),QP(41),F(42) ,TF(41)
1 ,TFP(41),POUTD(20),QL(41),TSH(41),TINS(41),TINSP(41),PIND(20)
2 ,X(600),Y(600)
ISN 0006      IDEL = 200
ISN 0007      IDEL2 = 20
ISN 0008      GC = 32.2
ISN 0009      IPRINT = 1
ISN 0010      IBOIL = 0
ISN 0011      DELV = 0.1
ISN 0012      TIME = 0.
ISN 0013      ICOUNT = 1
ISN 0014      IPLOT = 1
ISN 0015      READ(5,103) IE,IFB,IFT,NT,NDATA
ISN 0016      READ(5,100) DELT1,DELT2,LENGTH,AREA,DHYD,ZINER1,
1 ZINER2,ZFRIC1,ZFRIC2,VREF,FF,TIN,
2 UAF,UA1,UA2,UA3,EMCF,EMCL,
3 RHOL,BETA,CP,LMBDA,C1,C2
4 ,VISL,VISG,ALFAB,ALFAC,ALFAD,DUMMY
5 ,UASH1,UASH2,UASH3,EMCSH,EMCINS
DO 220 I = 1,IE
ISN 0017      220 QP(I) = 0.
ISN 0018      READ(5,100) (QP(I),I=IFB,IFT)
ISN 0019      READ(5,100)(TIMED(I),PIND(I),POUTD(I),I=1,NDATA)
ISN 0020      103 FORMAT(5I5)
ISN 0021      100 FORMAT(6F10.0)
ISN 0022      WRITE(6,108)
ISN 0023      WRITE(6,104) IE,IFB,IFT,NT,NDATA
ISN 0024      WRITE(6,109)
ISN 0025      WRITE(6,105) DELT1,DELT2,LENGTH,AREA,DHYD,ZINER1
ISN 0026      WRITE(6,110)
ISN 0027      WRITE(6,105) ZINER2,ZFRIC1,ZFRIC2,VREF,FF,TIN
ISN 0028      WRITE(6,111)
ISN 0029      WRITE(6,105) UAF,UA1,UA2,UA3,EMCF,EMCL
ISN 0030      WRITE(6,112)
ISN 0031      WRITE(6,105) RHOL,BETA,CP,LMBDA,C1,C2
ISN 0032      WRITE(6,114)
ISN 0033      WRITE(6,105) VISL,VISG,ALFAB,ALFAC,ALFAD
ISN 0034      WRITE(6,113)
ISN 0035      WRITE(6,105) UASH1,UASH2,UASH3,EMCSH,EMCINS
ISN 0036      WRITE(6,106) (I,QP(I),I=IFB,IFT)
ISN 0037      WRITE(6,107)(TIMED(I),PIND(I),POUTD(I),I=1,NDATA)
ISN 0038      104 FORMAT(5I10)
ISN 0039      105 FORMAT(6F12.4/)
ISN 0040      108 FORMAT(5X,'INPUT DATA'//7X,'IE',8X,'IFB',7X,'IFT',7X,'NT',7X,'NDAT
ISN 0041      1A')

```

```

ISM 0042      109 FORMAT(5X,'DELT1',7X,'DELT2',7X,'LNGTH',7X,'AREA',8X,'DHYD',8X,'ZINER1')
ISM 0043      110 FORMAT(5X,'ZINER2',6X,'ZFRIC1',6X,'ZFRIC2',6X,'VREF',8X,'FF',10X,'LTIN')
ISM 0044      111 FORMAT(5X,'UAF',9X,'UA1',9X,'UA2',9X,'UA3',9X,'EMCF',8X,'EMCL')
ISM 0045      112 FORMAT(5X,'RHOL',8X,'BETA',8X,'CP',10X,'LMBDA',7X,'C1',10X,'C2')
ISM 0046      114 FORMAT(5X,'VISL',8X,'VISG',8X,'ALFAB',7X,'ALFAC',7X,'ALFAD')
ISM 0047      113 FORMAT(5X,'UASH1',7X,'UASH2',7X,'UASH3',7X,'EMCSH',7X,'EMCINS')
ISM 0048      106 FORMAT(1H0,4X,'I',9X,'QP'/(110,F10.4))
ISM 0049      107 FORMAT(1H0,6X,'TIMED',6X,'PIND',6X,'POUTD'/(3F10.4))

C CONVERT INCHES TO FEET
ISM 0050      ZINER1 = ZINER1/12.
ISM 0051      ZINER2 = ZINER2/12.
ISM 0052      ZFRIC1 = ZFRIC1/12.
ISM 0053      ZFRIC2 = ZFRIC2/12.
ISM 0054      DHYD = DHYD/12.
ISM 0055      LNGTH = LNGTH/12.
ISM 0056      AREA = AREA/144.
ISM 0057      DO 218 I = 1,NDATA
ISM 0058      PIND(I) = PIND(I)*144.
ISM 0059      218 POUTD(I) = POUTD(I)*144.
ISM 0060      DO 219 I = 1,IFB,IF7
ISM 0061      219 QP(I) = QP(I)*12.

C INITIAL CONDITIONS
ISM 0062      DELT = DELT1
ISM 0063      IM = IE - 1
ISM 0064      DELZ = LNGTH/FLOAT(IM)
ISM 0065      PIN = PIND(1)
ISM 0066      POUT = POUTD(1)
ISM 0067      LTOT = ZFRIC1 + LNGTH + ZFRIC2
ISM 0068      VIN = 2.*DHYD*GC*(PIN - POUT - LNGTH*RHOL)/(FF*RHOL*LTOT)
ISM 0069      VIN = DSQRT(VIN)
ISM 0070      VOUT = VIN
ISM 0071      FLO = AREA*RHOL*VIN
ISM 0072      EMC = AREA*RHOL*CP
ISM 0073      Z(1) = 0.0
ISM 0074      TL(1) = TIN
ISM 0075      P(1) = PIN - FF*ZFRIC1*RHOL*VIN**2/(2.*GC*DHYD)
ISM 0076      DELP = FF*DELZ*RHOL*VIN**2/(2.*GC*DHYD) + RHOL*DELZ
ISM 0077      UASH = 1./(1./UASH1 + 1./UASH2 + 1./UASH3)
ISM 0078      DO 200 I = 2,IF
ISM 0079      Z(I) = Z(I-1) + DELZ
ISM 0080      QCL = (QP(I-1) + QP(I))/2.
ISM 0081      TL(I) = TL(I-1) + (QCL - UASH*(TL(I-1) - TIN))/(FLO*CP/DELT
1 + UASH/2.)
ISM 0082      P(I) = P(I-1) - DELP
ISM 0083      200 CONTINUE
ISM 0084      TOUT = TL(IE)
ISM 0085      DO 201 I = 1,IE
ISM 0086      VAVG(I) = VIN
ISM 0087      VL(I) = VIN
ISM 0088      VG(I) = VIN
ISM 0089      ALFA(I) = 0.
ISM 0090      S(I) = 0.
ISM 0091      TSAT(I) = C1 + C2*DLOG( P(I)/144.)

```

```

ISM 0092      QCL = QP(I)
ISM 0093      TCL(I) = TL(I) + QCL/UA1
ISM 0094      TF(I) = 0.
ISM 0095      IF(1.GE.IFB.AND.I.LE.IFT) TF(I) = TCL(I) + QP(I)/UAF
ISM 0097      TSH(I) = TL(I) - UASH*(TL(I) - TINS)/UASH1
ISM 0098      TINS(I) = TIN + UASH*(TL(I) - TINS)/UASH3
ISM 0099      201 CONTINUE
ISM 0100      VAVG(IE+1) = VIN
ISM 0101      VL(IE+1) = VIN
ISM 0102      ZFRONT = 36.

C TRANSIENT EQUATIONS
ISM 0103      WRITE(6,115)
ISM 0104      115 FORMAT(1H1)
ISM 0105      WRITE(6,101) TIME ,FLO,PIN,POUT,VOUT,ZFRONT
ISM 0106      WRITE(6,102) (I,Z(I),ALFA(I),VAVG(I),VL(I),VG(I),P(I),TF(I),TCL
1 (I),TL(I),TSH(I),TINS(I),TSAT(I),I=1,IE)
ISM 0107      101 FORMAT(1H0,5X,'TIME =',F7.3,5X,'FLO =',F7.3,5X,'PIN =',F6.0,5X,'PO
1UT =',F6.0,5X,'VOUT =',F6.2,5X,'ZFRONT =',F6.2/)
ISM 0108      102 FORMAT(5X,'I',8X,'Z',6X,'ALFA',6X,'VAVG',8X,'VL',8X,'VG',8X,'P',6X
1,'TF',6X,'TCL',5X,'TL',5X,'TSH',5X,'TINS',5X,'TSAT',/((15,2F10.4,4F1
20.2,6F8.1)))
ISM 0109      X(1) = 0.
ISM 0110      Y(1) = FLO
ISM 0111      ALFAX = .1
ISM 0112      DO 300 IT = 1,NT
ISM 0113      TIME = TIME + DELT
ISM 0114      DO 202 I = 1,IE

C FUEL MODEL
ISM 0115      IF(1.GE.IFB.AND.I.LE.IFT) TFP(I) = TF(I) + (QP(I) - UAF*(TF(I) -
1 TCL(I)))*DELT/EMCF

C CLAD MODEL
ISM 0117      QCL = 0.
ISM 0118      IF(1.GE.IFB.AND.I.LE.IFT) QCL = UAF*(TF(I) - TCL(I))
ISM 0120      QCLP = (1. - ALFA(I))/ALFAC*UA1*(TCL(I) - TL(I)) + ALFA(I)*UA2*
1 (TCL(I) - TSAT(I))/ALFAC
ISM 0121      IF(ALFA(I).GT.ALFAC) QCLP = UA2*(TCL(I) - TSAT(I))
ISM 0123      TEST = ALFAD - 0.1
ISM 0124      IF(ALFA(I).GT.TEST) QCLP = QCLP*(ALFAD - ALFA(I))/0.1
ISM 0126      IF(ALFA(I).GT.ALFAD) QCLP = 0.
ISM 0128      EMCTOT = EMCL
ISM 0129      IF(ALFA(I).GT.ALFAC) EMCTOT = EMCL + EMC*(ALFAD - ALFA(I))
ISM 0131      TCLP(I) = TCL(I) - (QCL - QCLP)*DELT/EMCTOT

C SHROUD EQUATION
ISM 0132      QSH = UASH1*(TL(I) - TSH(I))*(1. - ALFA(I))/ALFAC + UASH1
1 *(TSAT(I) - TSH(I))*ALFA(I)/ALFAC
ISM 0133      IF(ALFA(I).GT.ALFAC) QSH = UASH1*(TSAT(I) - TSH(I))
ISM 0135      EMCTOT = EMCSH
ISM 0136      IF(ALFA(I).GT.ALFAC) EMCTOT = EMCSH + (1. - ALFAD)*EMC
ISM 0138      TSHP(I) = TSH(I) + (QSH - UASH2*(TSH(I) - TINS(I)))*DELT/EMCTOT
ISM 0139      TINS(I) = TINS(I) + (UASH2*(TSH(I) - TINS(I)) - UASH3*(TINS(I) -
1 TIN))*DELT/EMCINS

ISM 0140      202 CONTINUE
ISM 0141      DO 216 I = 1,IE
ISM 0142      QL(I) = UA1*(TCL(I) - TL(I))
1 = UASH1*(TL(I) - TSH(I))

```



```

ISM 0143      IF(ALFA(I).GT.1.D-05) QL(I) = (1. - ALFA(I)/ALFAC)*QL(I) - UA3*
              1 (TL(I) - TSAT(I))
ISM 0145      IF(ALFA(I).GT.ALFAC) QL(I) = 0.
ISM 0147      216 CONTINUE
ISM 0148      TLP(I) = TIM
ISM 0149      TLP(IE) = TOUT
ISM 0150      DO 203 I = 1,IE
ISM 0151      IF(ALFA(I).GT.ALFAC) GO TO 203
ISM 0153      FLO = AREA*RHOL*VL(I)*(1. - ALFA(I))
ISM 0154      IF(VL(I).LT.0.) GO TO 209
C COOLANT TEMPERATURE EQN. FOR FORWARD FLOW
ISM 0156      IF(I.EQ.1) GO TO 203
ISM 0158      EMCP = EMC*(1. - ALFA(I)/2. - ALFA(I-1)/2.0)
ISM 0159      TLP(I) = TL(I) + ((QL(I) + QL(I-1))/2.
              1 - FLO*CF*(TL(I) - TL(I-1))/DELZ)*DELT/EMCP
ISM 0160      GO TO 203
ISM 0161      209 CONTINUE
C COOLANT TEMPERATURE EQN. FOR REVERSE FLOW
ISM 0162      IF(I.EQ.1E) GO TO 203
ISM 0164      EMCP = EMC*(1. - ALFA(I)/2. - ALFA(I+1)/2.0)
ISM 0165      TLP(I) = TL(I) + ((QL(I) + QL(I+1))/2.
              1 - FLO*CP*(TL(I+1) - TL(I))/DELZ)*DELT/EMCP
ISM 0166      203 CONTINUE
C VOID DISTRIBUTION
ISM 0167      IF(IPOIL.EQ.2) ALFAX = ALFAB
ISM 0169      F(I) = VL(I)
ISM 0170      DO 212 I = 1,IE
ISM 0171      F(I+1) = 0.
ISM 0172      IF(VL(I).LT.0.) GO TO 215
ISM 0174      F(I+1) = VL(I+1)*(1. - ALFA(I))
ISM 0175      IF(ALFA(I).GE.ALFAX) GO TO 212
ISM 0177      F(I+1) = VAVG(I+1)
ISM 0178      FP = (1. - ALFA(I)/ALFAX)*DELZ/DELT
ISM 0179      IF(FP.LT.F(I+1)) F(I+1) = FP
ISM 0181      GO TO 212
ISM 0182      215 F(I) = VL(I)*(1. - ALFA(I))
ISM 0183      IF(ALFA(I).GE.ALFAX) GO TO 212
ISM 0185      F(I) = VAVG(I)
ISM 0186      212 CONTINUE
ISM 0187      IF(VL(I).GE.0.) ALFA(I) = 0.
ISM 0189      DO 213 I = 1,IE
ISM 0190      ALFA(I) = ALFA(I) + (S(I) + (F(I+1) - F(I))/DELZ)*DELT
C CORRECTION FOR COLLAPSED VOIDS
ISM 0191      IF(ALFA(I).LT.0.) TLP(I) = TLP(I) + AREA*RHOL*LMBDA*ALFA(I)/(EMC*
              1 (BETA - 1.))
ISM 0193      IF(ALFA(I).LT.0.) ALFA(I) = 0.
ISM 0195      213 CONTINUE
C ROUTINE TO SET CONDITIONS FOR ONSET OF BOILING
ISM 0196      DO 206 I = 1,IM
ISM 0197      TSUPHT = TL(I) - TSAT(I)
ISM 0198      206 IF(TSUPHT.GT.0.0.AND.ALFA(I).LT.1.D-05) ALFA(I) = 0.001
C EQUATIONS FOR FLOW DURING BOILING
ISM 0200      CALL INTER(TIMED,PIND,TIME,PIN,NDATA)
ISM 0201      CALL INTER(TIMED,POUTD,TIME,POUT,NDATA)
ISM 0202      VIN = VL(1)

```

```

ISM 0203      CALL PRESS(VIN,PA)
ISM 0204      VIN = VL(1)* DELV
ISM 0205      CALL PRESS(VIN,PB)
ISM 0206      VIN = VL(1)* (POUT - PA)*DELV/(PB - PA)
ISM 0207      CALL PRESS(VIN,PC)
ISM 0208      FLO = AREA*RHOL*VIN
C  SAVE DATA
ISM 0209      VOUT = VOUTP
ISM 0210      DO 204 I = 1,IE
ISM 0211      TCL(I) = TCLP(I)
ISM 0212      TL(I) = TLP(I)
ISM 0213      VL(I) = VLP(I)
ISM 0214      VG(I) = VGP(I)
ISM 0215      P(I) = PP(I)
ISM 0216      TSH(I) = TSHP(I)
ISM 0217      TINS(I) = TINSP(I)
ISM 0218      IF(PP(I).LE.0.) GO TO 204
ISM 0220      TSATP(I) = C1 + C2*DLOG(PP(I)/144.)
ISM 0221      IF(ALFA(I).GT.ALFA) TL(I) = TSATP(I)
ISM 0223      TSAT(I) = TSATP(I)
ISM 0224      204 CONTINUE
ISM 0225      VL(IE + 1) = VLP(IE + 1)
ISM 0226      VG(IE + 1) = VGP(IE + 1)
ISM 0227      DO 217 I = IFB,IFT
ISM 0228      217 TF(I) = TFP(I)
ISM 0229      DO 221 I = IFB,IFT
ISM 0230      J = I + 1
ISM 0231      221 IF(ALFA(J).GT.1.D-05) GO TO 222
ISM 0233      ZFRONT = (J - IFB - ALFA(J)/ALFAD)*DELZ*12.
ISM 0234      IPRINT = IPRINT + 1
ISM 0235      IF(IPRINT.LE.IDEL) GO TO 205
ISM 0237      WRITE(6,101) TIME ,FLO,PIN,POUT,VOUT,ZFRONT
ISM 0238      WRITE(6,102) (I,Z(I),ALFA(I),VAVG(I+1),VL(I+1),VG(I+1),P(I),TF(I)
1 ,TCL(I),TL(I),TSH(I),TINS(I),TSAT(I),I=1,IE,2)
ISM 0239      IPRINT = 1
ISM 0240      205 CONTINUE
ISM 0241      IPLGT = IPLOT + 1
ISM 0242      IF(IPLOT.LE.IDEL2) GO TO 207
ISM 0244      ICOUNT = ICOUNT + 1
ISM 0245      X(ICOUNT) = TIME
ISM 0246      Y(ICOUNT) = FLO
ISM 0247      IPLOT = 1
ISM 0248      207 CONTINUE
ISM 0249      IF(IBOIL.EQ.1) DELT = DELT2
ISM 0251      IF(FLO.LT. 0.0) IBOIL = 2
ISM 0253      300 CONTINUE
ISM 0254      CALL STRTPL
ISM 0255      CALL LIMPLT(X,Y,ICOUNT)
ISM 0256      CALL DONEPL
ISM 0257      STOP
ISM 0258      END

```

```

      COMPILER OPTIONS - NAME= MAIN,OPT=00,LINECNT=57,SIZE=0000X.
                        SOURCE,EBCDIC,NOLIST,NODECK,LOAD,MAP,NOEDIT,ID,NOXREF
ISM 0002      SUBROUTINE PRESS(VIN,PCALC)
ISM 0003      IMPLICIT REAL*8 (A-H,O-Z)
ISM 0004      REAL*8 LMBDA
ISM 0005      COMMON PIN,RHOL,ZINER1,ZINER2,GC,DELT,DELZ,BETA,ALFAC,FF,DHYD,
1VL(42),VG(42),S(41),ALFA(41),TCLP(41),TCLP(41),TSAT(41),PP(41)
2 ,LMBDA,AREA,VLP(42), P(41),TSATP(41),C1,C2,VGP(42)
3 ,UA1,UA2,UA3,VAVG(42),ZFRIC1,ZFRIC2,TSHP(41),UASH1,VOUT,VOUTP
4 ,VISL,VISG,VREF,ALFAD,ALFAB
5 ,IBOIL,IE,IFB,IFT
ISM 0006      RHOG = RHOL/BETA
ISM 0007      CONST = (VISL*BETA**4/VISG)**(1./9.)
ISM 0008      CONST2 = FF*RHOL/(2.*GC*DHYD)
ISM 0009      CONST3 = 1./(AREA*LMBDA*RHOL)
ISM 0010      S(1) = 0.
ISM 0011      VAVG(1) = VIN
ISM 0012      VGP(1) = VI*
ISM 0013      VLP(1) = VIN
ISM 0014      PP(1) = PIN - RIOL*ZINER1*(VLP(1) - VL(1))/(GC*DELT)
1 - CONST2*(DABS(VL(1)))**0.8*VREF**0.2*(1.8*VLP(1) - 0.8*VL(1))*
1 ZFRIC1
ISM 0015      TSATP(1) = TSAT(1) + C2*(PP(1)/P(1) - 1.)
ISM 0016      IX = IE + 1
ISM 0017      DO 100 I = 2,IX
ISM 0018      VAVG(I) = VAVG(I-1) + (BETA - 1.)*S(I-1)*DELZ
ISM 0019      SLIP = 1.
ISM 0020      ALFAX = ALFA(I-1)
ISM 0021      IF(VL(I).LT.0.D0.AND.I.LT.IX) ALFAX = ALFA(I)
ISM 0023      IF(ALFAX .LT.1.D-05.OR.ALFAX .GT.0.999) GO TO 103
ISM 0025      ARG = 437.*(1. - ALFAX )**2 + 4.
ISM 0026      XTT = 2.*(1. - ALFAX )/(DSQRT(ARG) - 21.*(1. - ALFAX ))
ISM 0027      SLIP = CONST*(1. - ALFAX )*XTT**(-10./9.)/ALFAX
ISM 0028      103 CONTINUE
ISM 0029      VLP(I) = VAVG(I)/(1. + (SLIP - 1.)*ALFAX )
ISM 0030      VGP(I) = SLIP*VLP(I)
ISM 0031      IF(ALFAX .GT.0.999) VLP(I) = 0.
ISM 0033      IF(I.EQ.IX) GO TO 100
ISM 0035      VBAR = (VL(I) + VL(I-1))/2.
ISM 0036      VBARP = (VLP(I) + VLP(I-1))/2.
ISM 0037      F1 = (1. - ALFA(I-1))*RHOL*(VLP(I) - VL(I))/(GC*DELT)
1 + CONST2*(DABS(VBAR))**0.8*VREF**0.2*(1.8*VBARP - 0.8*VBAR)
ISM 0038      IF(VL(I).LT.0.) F1 = (1. - ALFA(I-1))*RHOL*(VLP(I-1) - VL(I-1))
1/(GC*DELT)
1 + CONST2*(DABS(VL(I-1)))*(2.*VLP(I-1) - VL(I-1))
ISM 0040      F2 = ALFA(I-1)*RHOG*(VGP(I-1) - VG(I-1))/(GC*DELT) + RHOL*(1. -
1 ALFA(I-1))
ISM 0041      PP(I) = PP(I-1) - (F1 + F2)*DELZ
ISM 0042      TSATP(I) = TSAT(I) + C2*(PP(I)/P(I) - 1.)
ISM 0043      101 S(I) = 0.
ISM 0044      IF(ALFA(I) .LT.1.D-05) GO TO 102
ISM 0046      IF(ALFA(I).GT.1.0) GO TO 102
ISM 0048      S(I) = CONST3*UA2*(TCLP(I) - TSATP(I))
1 + CONST3*UASH1*(TSHP(I) - TSATP(I))
ISM 0049      IF(ALFA(I).LT.ALFAC) S(I) = S(I)*ALFA(I)/ALFAC + CONST3*UA3*

```

```

1 (TLP(I) - TSATP(I))*(1. - ALFA(I)/ALFAC.
ISN 0051   CONST4 = (ALFAD - ALFA(I))/0.1
ISN 0052   TEST = ALFAD - 0.1
ISN 0053   IF (ALFA(I).GT.TEST) S(I) = CONST3*(CONST4*UA2*(TCLP(I) - TSATP(I))
          1 + UASH1*(TSHP(I) - TSATP(I)))
ISN 0055   IF (ALFA(I).GT.ALFAD) S(I) = CONST3*UASH1*(TSHP(I) - TSATP(I))
ISN 0057   IF (IBOIL.EQ.0) IBOIL = 1
ISN 0059   102 CONTINUE
ISN 0060   100 CONTINUE
ISN 0061   VOUTP = (1. - ALFA(IE))*VLP(IX) + ALFA(IE)*VGP(IX)/BETA
ISN 0062   VOUTP = VAVG(IX)
ISN 0063   PCALC = PP(IE) - RHOL*ZINER2*(VOUTP - VOUT)/(GC*DELT)
          1 - CONST2*(DABS(VOUT))*0.8*VREF*0.2*(1.8*VOUTP - 0.8*VOUT)*
          2 ZFRIC2
ISN 0064   RETURN
ISN 0065   END

```

LEVEL 21.7 (JAN 73)

05/360 FORTRAN H

DATE 78.251/10.14.05

COMPILER OPTIONS - NAME= MAIN,OPT=00,LINECNT=57,SIZE=0000K,
SOURCE,EBCDIC,NOLIST,NODECK,LOAD,MAP,NODEDIT,ID,NOXREF

```
ISN 0002      SUBROUTINE INTER(X,Y,XIN,YOUT,N)
ISN 0003      IMPLICIT REAL*8 (A-H,O-Z)
ISN 0004      DIMENSION X(20),Y(20)
ISN 0005      M = N - 1
ISN 0006      DO 100 I = 1,M
ISN 0007      100 IF(X(I+1).GT.XIN) GO TO 101
ISN 0009      101 DIFF = (XIN - X(I))/(X(I+1) - X(I))
ISN 0010      YOUT = Y(I) + DIFF*(Y(I+1) - Y(I))
ISN 0011      IF(XIN.GE.X(N)) YOUT = Y(N)
ISN 0013      RETURN
ISN 0014      END
```

INPUT DATA

IE	IFB	IFT	HT	NDATA	
41	5	20	1	9	
DELT1	DELT2	LNETH	AREA	DHYD	ZINER1
0.0025	0.0005	90.0000	1.1000	0.1190	40.0000
ZINER2	ZFRIC1	ZFRIC2	VREF	FF	TIN
36.0000	71.7000	12.2000	23.6000	0.0213	792.0000
UAF	UA1	UA2	UA3	EMCF	EMCL
0.1710	10.1000	0.8000	0.1000	0.4630	0.2340
RHOL	BETA	CP	LMBDA	C1	C2
50.9000	1640.0000	0.3010	1640.0000	1004.0000	226.0000
VISL	VISG	ALFAB	ALFAC	ALFAD	
0.1670	0.0233	0.2000	0.6000	0.9280	
UASH1	UASH2	UASH3	EMCSH	EMCINS	
0.4000	0.0040	0.0050	0.1780	0.8830	

I	-QP
5	16.6000
6	23.1000
7	28.6000
8	33.1000
9	36.0000
10	39.6000
11	41.4000
12	42.3000
13	42.3000
14	41.4000
15	39.6000
16	36.8000
17	33.1000
18	28.6000
19	23.1000
20	16.6000

TIMED	PIND	POUTD
0.0	125.4000	27.5000
0.9000	107.9000	25.7000
2.7000	76.6000	23.2000
4.5000	56.7000	21.6000
6.3000	47.0000	20.8000
8.1000	40.4000	20.3000
9.9000	36.2000	19.9200
11.7000	33.1000	19.5100
18.9000	33.1000	19.5100

TIME = 0.0 FLO = 9.189 PIN =18058. POUT = 3960. VOUT = 23.61 ZFRONT = 36.00												
I	Z	ALFA	VAVG	VL	VG	P	TF	TCL	TL	TSH	TINS	TSAT
1	0.0	0.0	23.61	23.61	23.61	12402.47	0.0	792.0	792.0	792.0	792.0	2011.0
2	0.1875	0.0	23.61	23.61	23.61	12215.47	0.0	792.0	792.0	792.0	792.0	2007.6
3	0.3750	0.0	23.61	23.61	23.61	12028.46	0.0	792.0	792.0	792.0	792.0	2004.1
4	0.5625	0.0	23.61	23.61	23.61	11841.46	0.0	792.0	792.0	792.0	792.0	2000.6
5	0.7500	0.0	23.61	23.61	23.61	11654.45	1983.4	818.5	798.8	798.7	795.0	1997.0
6	0.9375	0.0	23.61	23.61	23.61	11467.44	2463.4	842.3	814.9	814.8	802.1	1993.3
7	1.1250	0.0	23.61	23.61	23.61	11280.44	2876.9	869.9	835.9	835.7	811.4	1989.6
8	1.3125	0.0	23.61	23.61	23.61	11093.43	3223.1	900.3	861.0	860.6	822.5	1985.8
9	1.5000	0.0	23.61	23.61	23.61	10906.43	3515.6	933.1	889.4	888.9	835.1	1982.0
10	1.6875	0.0	23.61	23.61	23.61	10719.42	3746.5	967.5	920.5	919.8	848.8	1978.1
11	1.8750	0.0	23.61	23.61	23.61	10532.42	3907.9	1002.6	953.4	952.5	863.3	1974.1
12	2.0625	0.0	23.61	23.61	23.61	10345.41	4006.1	1037.7	937.4	986.3	878.4	1970.0
13	2.2500	0.0	23.61	23.61	23.61	10158.40	4040.5	1072.1	1021.8	1020.5	893.6	1965.9
14	2.4375	0.0	23.61	23.61	23.61	9971.40	4010.3	1105.0	1055.8	1054.4	908.6	1961.7
15	2.6250	0.0	23.61	23.61	23.61	9784.39	3914.7	1135.8	1088.7	1087.1	923.1	1957.4
16	2.8125	0.0	23.61	23.61	23.61	9597.39	3745.9	1163.5	1119.7	1117.9	936.9	1953.1
17	3.0000	0.0	23.61	23.61	23.61	9410.38	3510.3	1197.4	1148.1	1146.2	949.4	1948.6
18	3.1875	0.0	23.61	23.61	23.61	9223.37	3214.2	1227.1	1173.2	1171.1	960.5	1944.1
19	3.3750	0.0	23.61	23.61	23.61	9036.37	2842.6	1221.6	1194.1	1191.9	969.7	1939.5
20	3.5625	0.0	23.61	23.61	23.61	8849.36	2394.9	1229.9	1210.2	1207.9	976.8	1934.7
21	3.7500	0.0	23.61	23.61	23.61	8662.36	0.0	1216.9	1216.9	1214.6	979.8	1929.9
22	3.9375	0.0	23.61	23.61	23.61	8475.35	0.0	1216.8	1216.8	1214.5	979.8	1925.0
23	4.1250	0.0	23.61	23.61	23.61	8288.34	0.0	1216.8	1216.8	1214.4	979.7	1919.9
24	4.3125	0.0	23.61	23.61	23.61	8101.34	0.0	1216.7	1216.7	1214.4	979.7	1914.8
25	4.5000	0.0	23.61	23.61	23.61	7914.33	0.0	1216.7	1216.7	1214.3	979.7	1909.5
26	4.6875	0.0	23.61	23.61	23.61	7727.33	0.0	1216.6	1216.6	1214.2	979.7	1904.1
27	4.8750	0.0	23.61	23.61	23.61	7540.32	0.0	1216.5	1216.5	1214.2	979.6	1898.6
28	5.0625	0.0	23.61	23.61	23.61	7353.32	0.0	1216.5	1216.5	1214.1	979.6	1892.9
29	5.2500	0.0	23.61	23.61	23.61	7166.31	0.0	1216.4	1216.4	1214.1	979.6	1887.1
30	5.4375	0.0	23.61	23.61	23.61	6979.30	0.0	1216.3	1216.3	1214.0	979.6	1881.1
31	5.6250	0.0	23.61	23.61	23.61	6792.30	0.0	1216.3	1216.3	1213.9	979.5	1874.9
32	5.8125	0.0	23.61	23.61	23.61	6605.29	0.0	1216.2	1216.2	1213.9	979.5	1868.6
33	6.0000	0.0	23.61	23.61	23.61	6418.29	0.0	1216.1	1216.1	1213.8	979.5	1862.1
34	6.1875	0.0	23.61	23.61	23.61	6231.28	0.0	1216.1	1216.1	1213.7	979.4	1855.5
35	6.3750	0.0	23.61	23.61	23.61	6044.27	0.0	1216.0	1216.0	1213.7	979.4	1848.6
36	6.5625	0.0	23.61	23.61	23.61	5857.27	0.0	1216.0	1216.0	1213.6	979.4	1841.5
37	6.7500	0.0	23.61	23.61	23.61	5670.26	0.0	1215.9	1215.9	1213.5	979.4	1834.1
38	6.9375	0.0	23.61	23.61	23.61	5483.26	0.0	1215.8	1215.8	1213.5	979.3	1826.6
39	7.1250	0.0	23.61	23.61	23.61	5296.25	0.0	1215.8	1215.8	1213.4	979.3	1818.7
40	7.3125	0.0	23.61	23.61	23.61	5109.24	0.0	1215.7	1215.7	1213.4	979.3	1810.6
41	7.5000	0.0	23.61	23.61	23.61	4922.24	0.0	1215.6	1215.6	1213.3	979.2	1802.2

REFERENCES

1. R. C. Noyes, H. Lurie, and A. A. Jarrett, "The Development and Growth of In-core Voids Due to Boiling during Fast Reactor Transients," *Proc. Conf. on Safety, Fuels, and Core Design in Large Fast Power Reactors*, Oct 11-14, 1965, ANL-7120, pp. 881-889 (1965).
2. D. F. MacFarlane, *An Analytic Study of the Transient Boiling of Sodium in Reactor Coolant Channels*, ANL-7222 (June 1966).
3. E. R. Siegmann, *Theoretical Study of Transient Sodium Boiling in Reactor Coolant Channels Utilizing a Compressible Flow Model*, ANL-7842 (July 1971).
4. E. G. Schlechtendahl, *Theoretical Investigations on Sodium Boiling in Fast Reactors*, Nucl. Sci. Eng. 41, 99-114 (1970).
5. A. W. Cronenberg, H. K. Fauske, S. G. Bankoff, and D. T. Eggen, *A Single Bubble Model for Sodium Expulsion from A Heated Channel*, Nucl. Eng. Design 16, 285-293 (1971).
6. F. E. Dunn et al., *The SAS2A LMFBR Accident Analysis Computer Code*, ANL/RAS 73-39 (1973).
7. F. E. Dunn et al., *The SAS3A LMFBR Accident Analysis Computer Code*, ANL/RAS 75-17 (1975).
8. G. L. Bordner et al., *Pretest Analysis—SLSF In-reactor Experiment P2*, ANL/RAS 76-31 (1976).
9. M. A. Grolmes et al., "R-Series Loss-of-Flow Safety Experiments in TREAT," *Proc. Fast Reactor Safety Meeting*, Beverly Hills, CA (Apr 1974).
10. G. L. Bordner, "Voiding Behavior From P2 Posttest Analysis," *Reactor Development Program Progress Report: March 1977*, ANL-RDP-59, pp. 6.32-6.34 (May 20, 1977) and *Reactor Development Program Progress Report: May-June 1977*, ANL-RDP-61, pp. 6.74-6.76 (Aug 11, 1977).
11. G. L. Bordner, editor, *Summary and Evaluation of the SLSF In-Reactor Safety Experiment P2*, ANL/RAS 78-23 (1978).
12. G. L. Bordner, "Predicted Voiding Behavior during the P3A LOF Test," *Reactor Development Program Progress Report: July-August 1977*, ANL-R1 P-62, pp. 6.32-6.64 (Oct 5, 1977).
13. G. L. Bordner, "Boiling and Voiding Dynamics: LIMBO Analysis," *Reactor Development Program Progress Report: November 1977*, ANL-RDP-65, pp. 6.27-6.28 (Jan 24, 1978).