archives

THERMOHYDRAULIC ANALYSIS OF U-TUBE STEAM GENERATORS

by

Hugo Cardoso da Silva Jr.

Energy Laboratory Report No. MIT-EL 84-005

April 1984

VOLUME I

THERMOHYDRAULIC ANALYSIS OF U-TUBE STEAM GENERATORS
Volume I

by

Hugo Cardoso da Silva Jr.

Principal Investigators

Prof. David Lanning
Prof. John Meyer
Prof. Neil Todreas

Energy Laboratory
and
Department of Nuclear Engineering

Massachusetts Institute of Technology
Cambridge, Massachusetts 02139

Sponsored by:

Northeast Utilities Service Company
Yankee Atomic Electric Company

under the

MIT Energy Laboratory Electric Utility Program

MIT Energy Laboratory Report No. MIT-EL 84-005

April 1984

THERMOHYDRAULIC ANALYSIS OF U-TUBE STEAM GENERATORS

by

Hugo Cardoso da Silva, Jr.

ABSTRACT

Recent trends in plant safety analysis reveal a need for benchmark analytical representations of the steam generators to aid in the improvement of system codes and of fast codes for operator assistance. A model for such applications should exhibit four characteristics. First, it should be capable of representing the entire unit. Second, it should be based on detailed physical models, supplemented by well-tested empirical correlations and utilize a reliable numerical method, while still allowing for the assessment of potentially simplifying assumptions. Third, it should be validated. Fourth, it should provide a basic framework for expansion to severe transient (accident) analysis.

A model satisfying these characteristics has been developed. The downcomer, evaporator, and riser are treated by the two-fluid, three-dimensional code THERMIT. A zero-dimensional calculation closes the natural circulation loop by linking the riser to the downcomer. Effects included are: condensation, flashing, structure and liquid heat sinks and compressibility in the steam dome. The primary-side representation allows for any number of tubes per secondary-side computational cell. For each tube, four temperatures are calculated: primary fluid, primary wall, intermediate wall, and secondary wall.

The capability for calculating parameter distributions in steady-state at full and half power has been verified. Results are in excellent agreement with measurements conducted at the Westinghouse Model Boiler No. 2, as well as with calculations by the ATHOS code. Global parameter computations for both mild and severe operational transients have also been verified. Calculations compare well with plant start-up data gathered at the Arkansas Nuclear One-Unit 2 facility.

The present research has produced the first integrated U-tube steam generator model which both utilizes the porous body two-fluid formulation and has validated capability of application to operating transients.

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to Professor David Lanning for his unrelenting and kind support over the years.

To Professor Neil Todreas, for his friendship and suggestions, I convey my deeply felt appreciation. It has been a privilege to have been able to develop professionally in his company.

I am deeply indebted to Professor John Meyer for his always generously given time and professional advice at many difficult stages of this project.

Special thanks are due to Mr. Yehia Khalil for his invaluable assistance in the computer-related activities.

I must also acknowledge Dr. Walter Strohmayer and Dr. Derek Ebeling-Koning, former fellow students in the department, who took time from their own work to share with me hard earned information and instructive discussions.

To Marsha Levine and Rachel Morton I leave a thank you note for being so constantly helpful.

The painstaking care with which Karen Luxton typed this thesis, and including my constant revisions, is truly appreciated.

Thanks are also due to the many friends, both in this country and in Brazil, who over the years, one way or another, offered me their kindness. Here I include:

João, Renato, Rodolfo, Sandra, Cristina, Peter and Brenda Roberts, Buzz Nedry, Jack, Jean, Heidi and Bonnie Mohr, Rafael e Lucia, Sergio e Sonia, Leo, Ricardo, Zé Mauro, Rosinha, tia Elza, Fernanda, Xito, Ana Cristina, Augusto, Mário, Nobre, Peninha, Marta e Sávio, Horácio e Corina, Gail, Aunt Caroline and Uncle Bob, Anita and John, Richard and Leisa, Carmen, Margaret Helen,

Patti, who gave it a new meaning and

my mother, for what she is, her support, her concern.

The Westinghouse Model Boiler No. 2 experiments and code verification results were obtained through Dr. Allan Y. Lee of Westinghouse.

The financial contribution of Northeast Utilities Service Co. and Yankee Atomic Electric Co. to the computer and editing expenses of this work is acknowledged. In particular, I would like to thank Dr. Mario Bonaca, Dr. Don Dube, Mr. Marty Pitek, and Mr. Al Chyra of Northeast as well as Mr. Paul Bergeron and Mr. James Kendall of Yankee for reading interim reports, attending presentations and continuously recommending support for this research. In this regard I must also thank

Mrs. Barbara McWilliams of the MIT Energy Laboratory for the dedication she exerted in insuring all project contractual deadlines were met.

My entire doctoral program has been funded by a fellowship from CAPES, an agency of the Brazilian Ministry of Education. I gratefully acknowledge this support. Here I must thank Ms. Cristina A. Cononelli, head of the "Bolsas no Exterior" program, for her continued recommendations that my fellowship be renewed. Partial support to my doctoral program has also come from the Brazilian Nuclear Energy Commission (CNEN) through the PRONUCLEAR program and from the Catholic University of Rio de Janeiro (PUC/RJ) as well. Bearing in mind the economic hardships Brazil has been facing, these contributions are hereby most sincerely acknowledged.

Finally, I am grateful to Professor Pedro Carajilescov of the Instituto Technologico de Aeronautica who made it all possible.

Hugo C. Silva Jr.

Hugo Cardoso da Silva, Jr.

Cambridge, February 10, 1984

Table of Contents

Table of Contents (Continued) Page

Table of Contents (Continued) Page

Table of Contents (Continued) Page

Table of Contents (Continued) Page

List of Figures

List of Figures (continued) Page

List of Figures (continued)

List of Figures (continued)

List of Figures (continued) Page

List of Figures (continued) Page

List of Tables

List of Tables (continued) Page

List of Tables (continued) Page

Chapter 1

INTRODUCTION

1.1 Background

The objective of nuclear plant safety research is to continuously verify that the public is well protected if any of a variety of anticipated or postulated accidents should occur. Protection is accomplished by establishing operational and design limits that are based on conservative assumptions for both the individual and combined behavior of the plant components. In a more specific sense, the objective of nuclear plant safety is to continuously verify the appropriateness of these limits within a framework of expanding knowledge and desire for increased plant availability.

With this objective, several analytical tools have been developed to calculate the integrated behavior of all the plant components. Computer programs which do this are known as 'system' or 'loop' codes. Since these programs must calculate the overall performance of the plant, they cannot calculate in detail the behavior of any one component. Nevertheless, it has become increasingly evident in recent years that a more detailed representation of the steam generator, an important component on the secondary-side (see Fig. 2.1-1), gives loop codes the capability to predict more accurately both the reactor and the overall system response (L1)(L2)(W1). This permits (e.g. Ref. (H6)) an improved assessment of several current operational safeguards. In order to assess the various ways of modeling the steam generators in the system codes, it is important to have a detailed steam generator analytical model for benchmarking. One of the motivations for the present work is the need for such models.

Another important area in which the safety of nuclear power reactors is being improved is the development of systems to assist operators in taking appropriate corrective action during certain transients. Several of these systems are based on the creation and maintenance of a validated data base to be used in estimates of parameters relevant to plant safety (S2). One of the ways of achieving high reliability in the knowledge of those parameters is to compare several measurements with an analytical calculation.

The analytical calculation is essentially a computer model of a plant component. The steam generators are key components for assessing plant response in many transients. A computer model for this use must, for obvious reasons, be able to run in real time or faster. Furthermore, the time constraint must be met with methods that are reliable. This necessarily requires that these models involve a lumped parameter formulation and must therefore be supplied with information they cannot generate themselves (e.g. parameter distributions).

In this context there is another incentive for the development of the detailed benchmark steam generator component model. It can be used for testing key assumptions and developing insight into steam generator behavior. Hence, the detailed steam generator model is an important part of the operator assistance program as well.

1.2 Research Objectives

The goal of the present research is to develop an analytical, benchmark U-tube steam generator model for the previously described applications. Such a model should incorporate four important features.

First, it should be capable of representing the whole steam generator rather than only certain regions. This is a necessary condition for the model to be used in conjunction with a system code. It must respond directly to the various system boundary conditions.

Second, it should be based on detailed physical models, supplemented by well tested empirical correlations and utilize a reliable numerical method. This combination should involve a minimum of assumptions while permitting the assessment of many potential simplifications.

Third, it should be validated for steady-state parameter distribution calculations, as well as for calculations of global parameters during operational transients.

Fourth, it should provide the basic framework for expansion to severe transient (accident) analysis.

In order to meet the second requirement, the computer code THERMIT (R1)(K1) was selected as the framework within which to develop the integrated U-tube steam generator model. THERMIT was originally developed at MIT under sponsorship of the Electric Power Research Institute. This light water reactor thermohydraulic analysis code solves the three-dimensional, two-fluid equations describing the two-phase flow and heat transfer dynamics. The two-fluid model uses separate partial differential equations expressing conservation of mass, momentum, and energy for each individual fluid phase. This allows for thermal and mechanical non-equilibrium representation, which is of particular interest in the study of severe transients. This framework therefore meets the fourth requirement above. THERMIT combines an advanced two-phase model with well tested empirical correlations and a reliable numerical method. For all these reasons, which show this code satisfies both the second and fourth desired features of the model to be developed, and because of its ready availability at MIT, THERMIT was chosen as the starting point for the present study.

Within this framework four major developments are necessary to create an integrated analytical model for U-tube steam generator analysis.

First, it is necessary to develop a mathematical model describing the dynamics of the steam dome and of the upper downcomer (see Figs. 2.1-2 and 2.2-1), as well as the recirculating flow, into the downcomer from the steam separators (see Fig. 4.3-1). In addition, this model must have the same range of applicability as the basic two-fluid model.

Second, the fuel pin representation in THERMIT must be replaced by an appropriate representation of the U-tubes and the primary fluid within them.

Third, it is necessary to integrate the various regional models. In order to do this it is also necessary to identify and utilize the best combination of system boundary conditions for a given type of application.

Finally, in order to demonstrate that these three steps have been correctly executed it is necessary to validate the integrated model for global parameter predictions in transients, as well as for the calculation of parameter distributions in steady-state.

There are many thermohydraulic analysis codes which are physically and numerically well verified and capable of severe transient analyses. However, these codes are either for reactor core applications or they are general thermohydraulic codes which cannot directly represent the whole steam generator, only certain regions of it. In fact, an additional contribution of the present work is that the overall method could be used to transform many of these codes into an integrated U-tube steam generator model.

Of the codes written specifically for steam generator analysis, TRANSG (L2) is the only one which has objectives similar to those of the present work. However, TRANSG is simpler, as discussed in Chapter 3, in order that it be economically advantageous to run it coupled to a system code. Because of this characteristic it still incorporates many simplifications, although it is more complete than the system code representations of the steam generators.

One multi-dimensional integrated steam generator model incorporating a two-fluid formulation is the COBRA-TF (S1)(S9) model. This model, however, is mostly directed toward very small-scale studies such as the determination of local phase velocities and void fraction predictions around the tube support plate structure.

Finally, an important effort in steam generator modeling is the ATHOS (S3) code. The implicit characteristic of its numerical method, along with some simplifications in the treatment of the steam dome and downcomer, however, seem to indicate that ATHOS, at least in its present form, is only appropriate for steady-state and very slow transient analyses.

To summarize: the objective of the present work is to develop a benchmark integrated U-tube steam generator model utilizing the porous body two-fluid formulation with the demonstrated capability of calculating global parameters for both mild and severe operational transients, and of calculating steady-state parameter distributions with intermediate resolution. This is the first model with these characteristics.

A byproduct of this research is that the groundwork for developments in two important areas is also established. One area is the analysis of very severe (accident) transients. Another is the study of microscale problems by utilizing intermediate resolution fields, as boundary conditions for finer scale calculations.

1.3 Report Structure

Chapter 2 provides a general description of a U-tube steam generator and indicates the various regions into which these units are usually divided for analytical modeling.

Chapter 3 is a review of previous work from two perspectives. In one of them, the present model is situated in the context of steam generator models in general. In another, the previously developed THERMIT code is briefly discussed.

Chapter 4 is an overall description of the integrated analytical model showing the interfaces between the various regional models. Boundary conditions are discussed here as well.

Chapters 5 and 6 are in-depth presentations of the recirculation and primary temperature models, respectively.

Chapter 7 presents the assessment study.

Chapter 8 summarizes the work, presents conclusions and recommendations for future work.

Appendix F is a user's manual. It provides a description of the input for the computer program developed and discusses important program details for its preparation. Samples of input and output are also given.

Appendix G is a listing of the computer program in the present work.

The remaining Appendices, A, B, C, D, E, and H, give important mathematical derivations.

Chapter 2

STEAM GENERATOR MODELING: OVERVIEW

2.1 Description of a U-Tube Steam Generator

In pressurized water reactor plants, the heat generated in the reactor core is removed by the primary coolant system. The Steam Generator is a heat exchanger, transferring heat from the primary coolant system to the secondary system to produce steam to drive the turbine generator set. This arrangement is shown schematically in Fig. 2.1-1.

A typical U-tube steam generator (UTSG) is shown in Fig. 2.1-2. The primary fluid enters the unit typically at about 15 MPa and 590°K, through a plenum. The topmost boundary of the plenum is the tube sheet, where the flow is forced to distribute among the U-tubes. The primary coolant then flows axially upward within the hot side U-tubes, proceeds through the U-bend region from hot to cold side, and finally descends through the cold side, exiting by way of the outlet plenum, at temperatures of about 560°K.

The secondary fluid, in subcooled condition (typically 6 MPa, 500°K), is forced by the feedwater pumps into the feedwater nozzle and circumferentially partitioned into the feedwater mixing chamber by means of a distribution ring. There, it mixes with the recirculating saturated liquid (at about 6 MPa) returning from the steam separation equipment. The resulting subcooled liquid flows downward through the

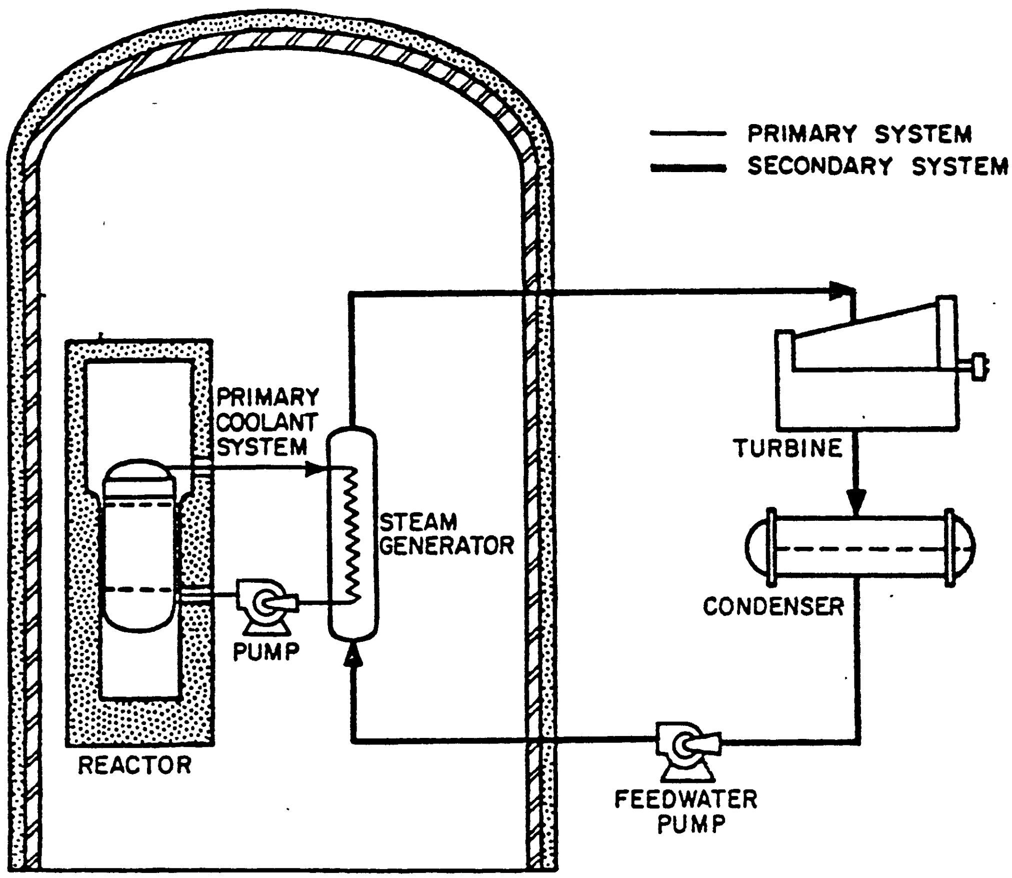


Figure 2.1-1. Schematic pressurized water reactor power plant showing primary and secondary systems (adapted from Ref. N1).

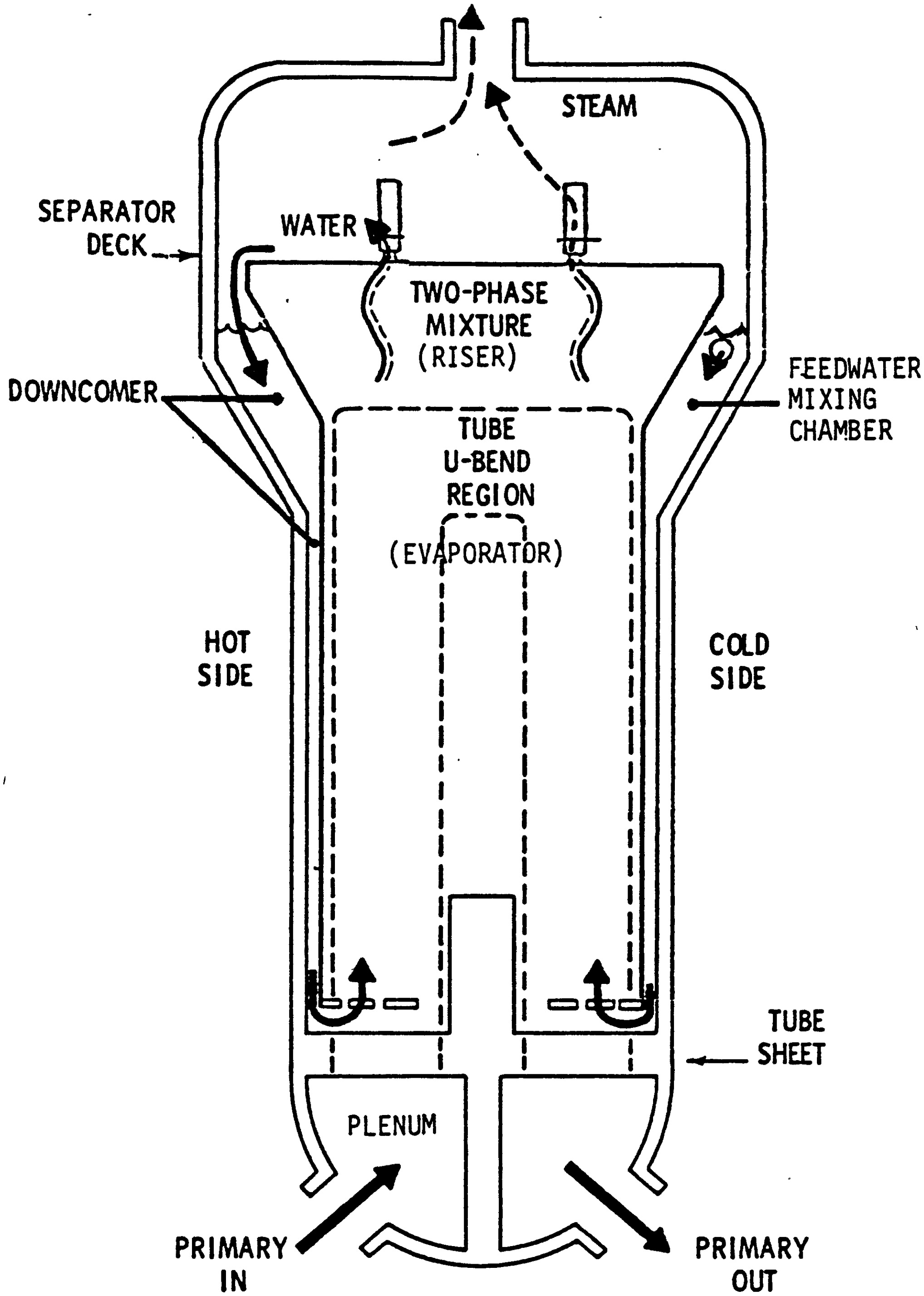


Figure 2.1-2. Schematic representation of a U-tube steam generator (UTSG) (adapted from Ref. S9)

annulus formed between the outer and inner shells of the steam generator (the downcomer). At the bottom of the downcomer, just at the tube sheet level, the flow proceeds radially into the shell side of the tube bundle (evaporator) while gradually turning upward. It continues up through the evaporator where it is heated to saturation and boiled. The two-phase mixture leaving the tube U-bend region then flows through the riser entering the primary steam separators at qualities between 20 percent and 33 percent. At this point most of the liquid is mechanically removed from the mixture (modern separators achieve at least 99.75 percent exit quality) and redirected into the downcomer, partly flowing across the separator deck, partly through drain pipes connecting the deck to the feedwater mixing chamber.

The flow path from this chamber and ending again at the chamber constitutes a natural circulation loop, where the pressure gain due to the hydrostatic head in the downcomer is the driving force that overcomes the various pressure losses along the circuit.

The steam leaving the separators fills the upper dome of the generator. Because of the large sensitivity of turbine blades to steam moisture, a very small amount (less than 0.25 percent) liquid is further removed from the steam by a set of chevron type dryers. These are usually placed at the top of the dome near the steam exit nozzle.

2.2 Steam Generator Regions

For the purpose of developing models for the steam generators, it is appropriate to divide the whole unit into several regions. Depending upon the purpose of the model, each region can then be analyzed with any desired level of resolution. For example: the evaporator is often taken to constitute a region. In lumped parameter models it is treated as a single node, while in distributed parameter models several nodes may be used to represent it. The criterion for deciding what constitutes a region relates primarily to the physical processes involved and somewhat to the level of spatial resolution desired. The regions into which the steam generator is usually divided are: primary, tube bundle or evaporator, riser, separators, steam dome, and downcomer. Figure 2.2-1 depicts these regions.

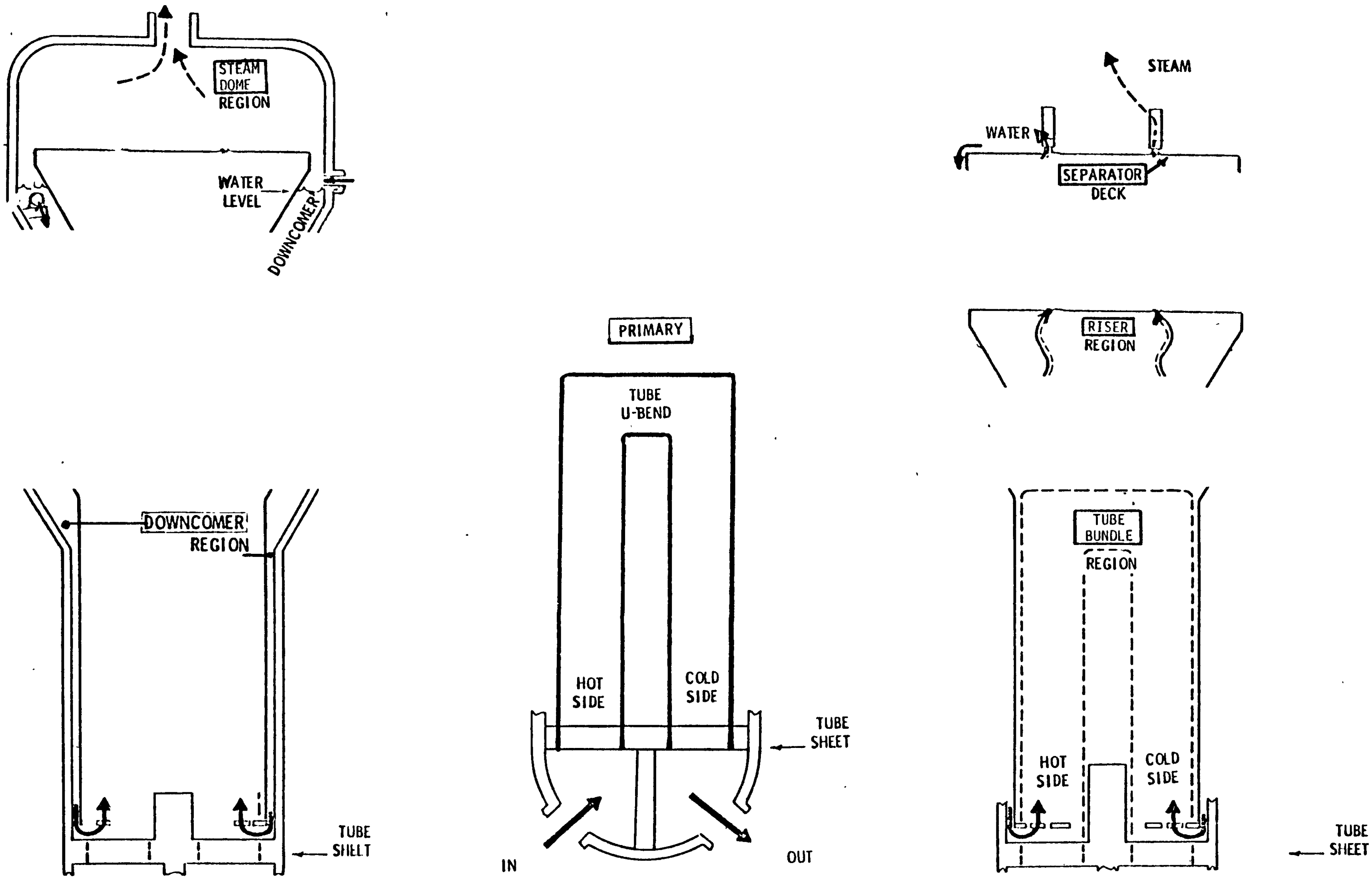


Figure 2.2-1. Steam generator regions.

Chapter 3

COMPARATIVE REVIEW OF PREVIOUS WORK

3.1 Representative Steam Generator Models

3.1.1 Overall Perspective

In recent years, there has been increasing interest in the thermal-hydraulic modeling of steam generators. It is possible (L1) to place these models in one of two broad categories: macroscale or microscale. These terms specify whether the model's main purpose is to represent the steam generator as a whole (macroscale) or a limited region of the unit (microscale).

For an example of the microscale approach, the COBRA-TF (S1) model has been used in the analysis of the local flow field a few tube diameters above and below the tube support plates. Microscale work has been motivated by problems such as tube wall thinning. The cause of wall thinning appears to be local corrosion. The location of the defects, within the tube-to-support plate crevice, at the periphery of the bundle, suggests that flow patterns within the bundle, leading to local flow starvation, may play an important role in the damage mechanism. Reference (W1) provides a detailed discussion on problems requiring microscale resolution.

For the macroscale category approach, it is helpful to consider two further subdivisions: detailed and fast. The term detailed refers mainly to the degree of complexity with which the physical processes in the various steam generator regions are modeled. The term also implies spatial resolution, although usually at least one order of magnitude above the microscale.

Recently, the development of systems to assist operators in taking corrective action during off-normal transients has stimulated the development of fast models (S2)(C3). Systems such as the disturbance analysis surveillance system (DASS) and the Nuclear Regulatory Commission-mandated safety parameter display system (SPDS) require non-linear component models capable of operating in real time for data reliability assessment. The time constraint which implies few and large nodes must be achieved with methods that are reliable. This often requires that parameter distribution information be incorporated into the volume integration procedure over each node. This makes the detailed model a very useful tool in the same program because it can generate time dependent profiles. Furthermore, it can aid in assessing the level of detail needed in the steam generator description for each intended application of the fast model.

Fast codes can also be used to generate information for guiding the development and refinement of detailed codes: for example, in input sensitivity studies over long time periods. In such a context these two tools are essentially symbiotic in the process of developing insight into steam generator dynamics. This constitutes an application distinct from data reliability assessment. A greater understanding of the physics of the steam generator could lead in the long run to improved control systems or trip set points.

A third application for detailed models is in conjunction with system codes. For example, optimal nodalization schemes in a system code can be sought by comparing the system code steam generator response to that of the detailed component model. Nodal parameters in the system code can be adjusted as well, to better match the component response in a given base case transient. These parameters would then be used in all system analyses for transients of that type.

Finally, an important application of detailed analyses is design. Such applications include comparison of alternate design performances, parametric studies to alter design, and predictions of performance under off-design conditions.

Figure 3.1-1 summarizes the various types of models and their applications. The code developed in this work is well suited for all the mentioned applications for detailed models. It would be optimally used for operational transients. Application to severe accident predictions, such as a steam line break, is beyond the scope of the present work.

3.1.2 Comparison of Representative Models

Within the microscale category two representative efforts are COBRA-TF (S1), and DUVAL (D1). COBRA-TF has been applied in the calculation of detailed local phase velocity and void fraction predictions around the annular tube support plate

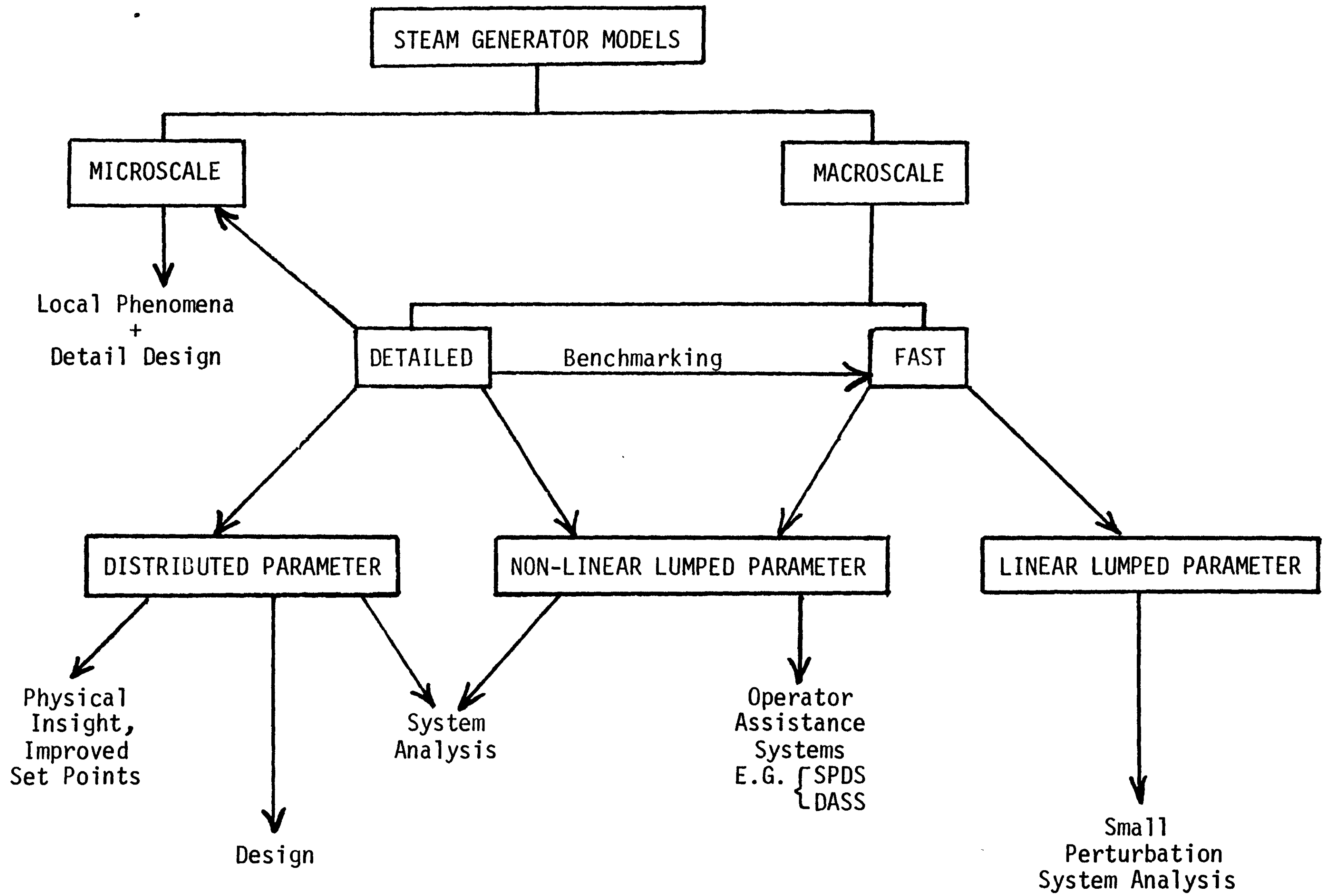


Figure 3.1-1. Steam generator models and representative applications.

structure. The motivation for that work has been to search for the potential existence of dryout areas near tubes leading to local steam superheat. These conditions are known to be conducive to corrosion and denting. DUVAL (D1) has been applied to the calculation of the mixture velocity field (it uses the homogeneous equilibrium model), enthalpy, pressure, and quality contours within a typical steam generator preheater. Figure 3.1-2 illustrates the problem size for the two simulations described. Other microscale models are discussed in (L1).

A representative work in the fast code area is a thesis recently concluded at MIT (S2). In this work the steam generator is divided into the regions of Fig. 2.2-1. There are additional models for the steam removal system and feedwater control. Each region is represented by one node except for the primary, which includes one node for each plenum and one node for the tubes. In this formulation, considerable effort was exercised in incorporating profile information into the spatial averaging of parameter values within each node. This technique proved itself very successful in the analysis of several transients of interest.

It is in the area of detailed macroscopic simulation that most of the steam generator simulation effort has been deployed. Within this category, a significant spectrum of detail exists.

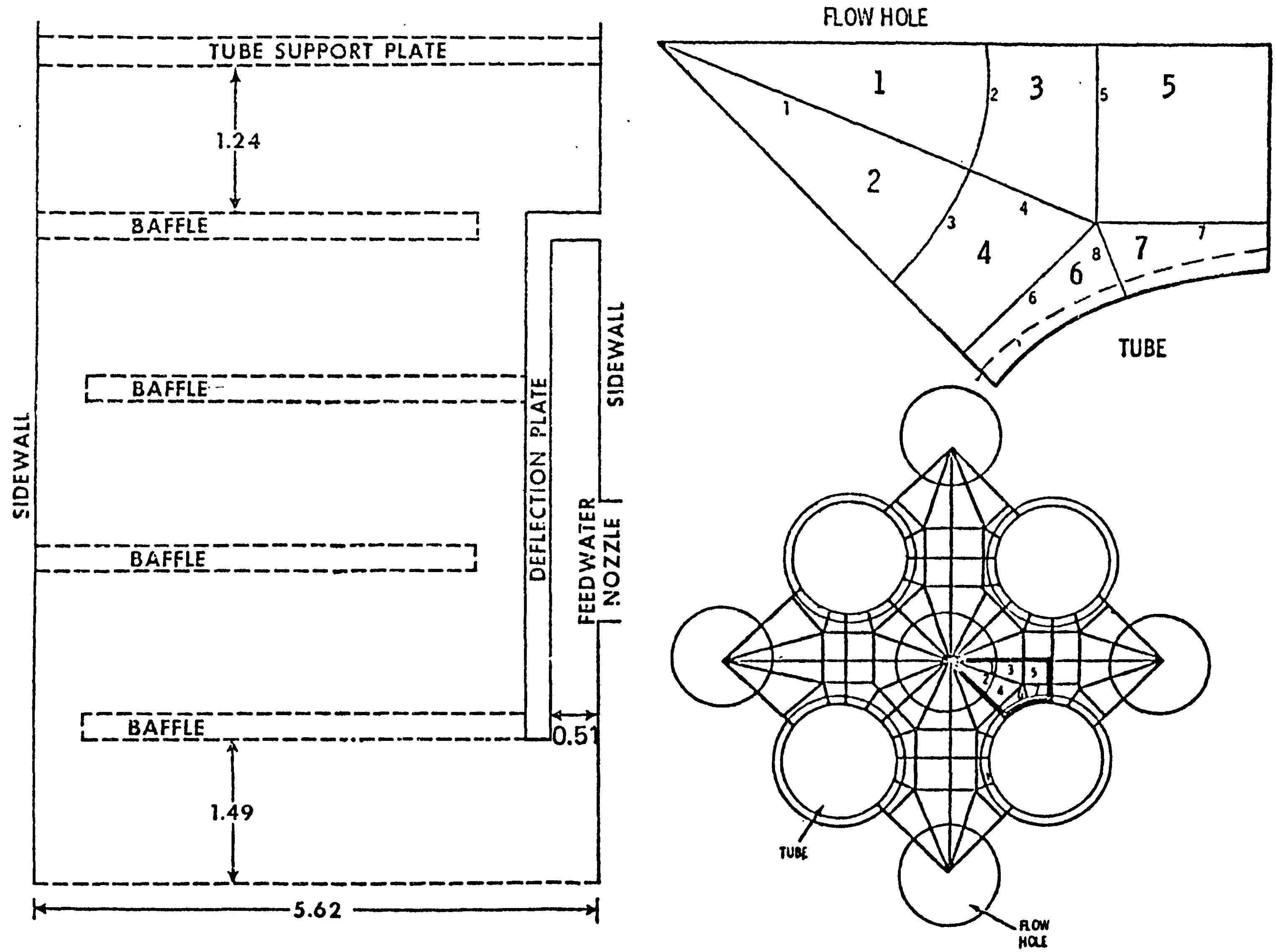


Figure 3.1-2. Examples of microscale size
Left: DUVAL (D1)
Right: COBRA-TF (S1)

At the least detailed end, a representative model is the code UTSG, developed in Germany by Hoeld (H1). In this model, the steam generator is again partitioned into the basic regions of Fig. 2.2-1. The evaporator is further divided into a subcooled water node, a transition node, a node with two phase flow, for which a drift flux formulation is used. There is also a model for the steam removal system.

Similar to Hoeld's model is that due to Bruens (B2). In this model the evaporator is divided into two radial regions: one for secondary flow parallel to primary, the other for countercurrent flow. Mixing occurs only in the riser. Bruens' and Hoeld's predictions have been compared (B2) to each other and to turbine trip data for Biblis A. These models are essentially lumped parameter non-linear representations. They require little computation time and are nearly capable of real time simulations on large machines.

At the higher detail end of the spectrum are the distributed parameter models. They range in complexity from the one-dimensional separate flow model (T2), e.g. TRANSG (L2), to three-dimensional, more complex algebraic slip models in the evaporator region, e.g. ATHOS (S3). It is interesting to note that the levels of model complexity have not been consistent for all the regions in the various codes. For example, the steam dome and downcomer representation of TRANSG incorporates more of the actual physics of the region than ATHOS. The reverse is true for the evaporator.

The question of how much detail is needed for the steam dome model relates to the time scale of interest. If transients of interest are such that the transport time, τ in the steam dome, where

$$\tau = \int \rho dV/w \qquad (3.1\text{-}1)$$

is small compared to the time of significant transient changes, then compressibility effects in the steam dome might be ignored. For a 1300 MWt U-tube steam generator delivering saturated steam at 6.2 MPa, the transport time is on the order of ten seconds. Thus, a detailed steam dome model is required in the present work, since the transients of interest exhibit significant variations over that same time frame.

In developing the models for the present study, the basic guideline has been to utilize, in all regions, specific models consistent with the capability in the two-fluid model, so as to retain the ability to predict severe operational transients.

To illustrate the range of complexity, the limitations, and the function of models of the various steam generator regions, a brief review of the approaches adopted in (a) ATHOS (S3) and (b) TRANSG-01 (L2) will be given.

3.1.2.1 ATHOS

Figure 3.1-3 shows the regions of a UTSG with economizer, as modeled in ATHOS. This configuration is presented here,

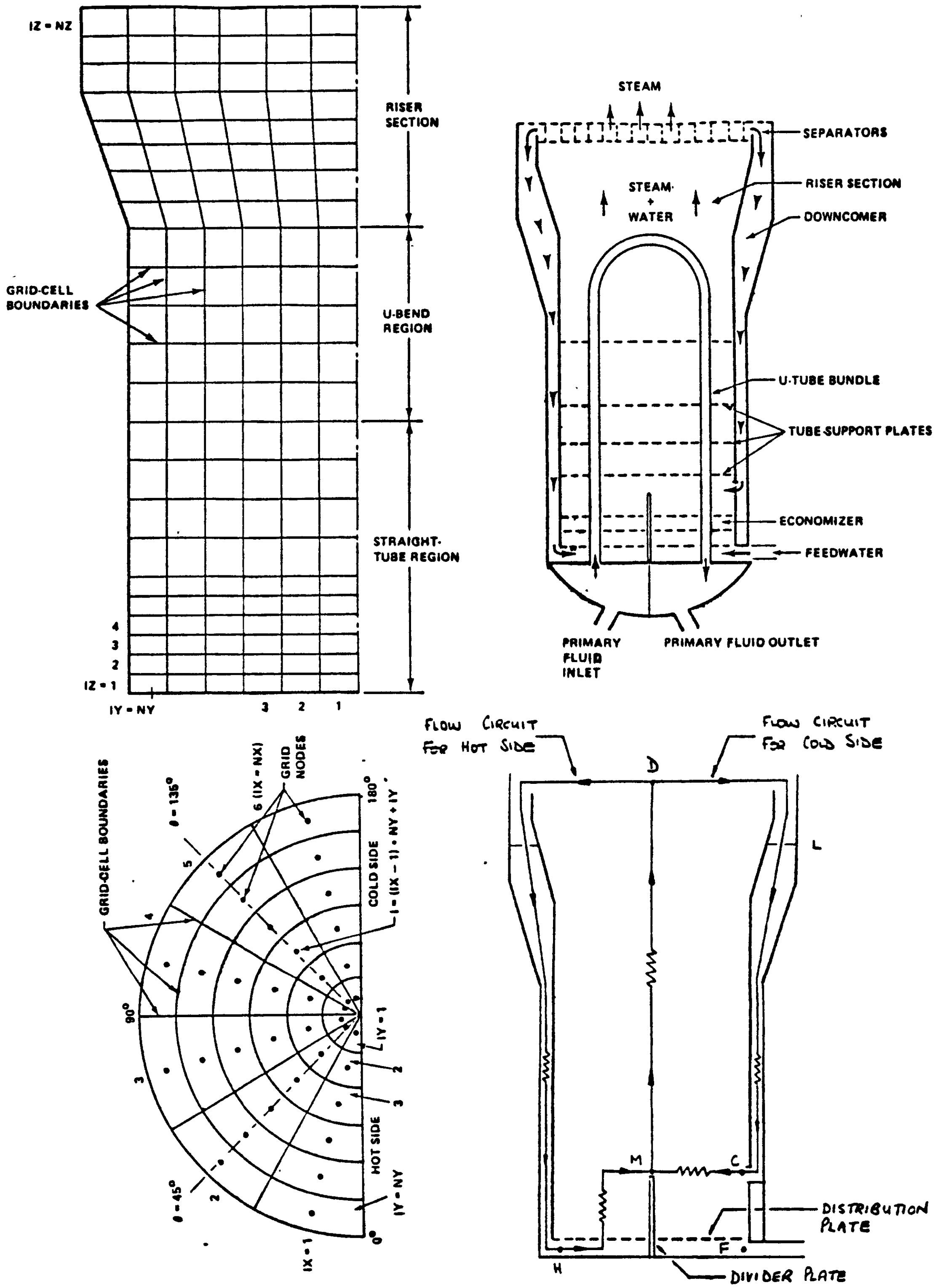


Figure 3.1-3 ATHOS (S3). Clockwise from top right: Overall Model Regions, CAP Reference points, Radial Mesh, Axial Mesh.

although not subject of present study, because it is well suited to illustrate two downcomer region models, with no loss of generality. The boundary conditions are: primary pressure, flow rate and inlet temperature, steam dome pressure, feedwater flow rate, and temperature. There is no model for steam dome dynamics. The calculated steam flow is located at the separator deck, as shown in Fig. 3.1-3. For this reason and until such a model is incorporated, ATHOS is not expected to make good predictions of outlet nozzle steam flows in more severe operational transients, where dome compressibility effects are important, as for example in a turbine trip.

The present discussion on ATHOS is divided into three subjects: (a) downcomer, (b) primary and heat transfer, (c) evaporator and riser.

(a) <u>Downcomer</u>

There are two alternative procedures for treating the downcomer in ATHOS. Both are point model (i.e. zero dimensional) formulations. One of them, called CAP (circuit adjustment procedure), determines the mass flow rates entering the evaporator from hot and cold side, respectively. The other procedure, unnamed, calculates entry velocities into the evaporator. Both procedures are discussed in reasonable detail in this section.

The choice between these two methods depends on the solution method that is used in the evaporator region. There are two solution procedures for the evaporator: slab by slab and simultaneous. The slab by slab method was utilized in the URSULA2 (S7) code, the parent of ATHOS. The CAP method is used in conjunction with this approach. The simultaneous method is more recent and utilizes the unnamed procedure to obtain the entry velocities. It is not possible to give an idea of the slab by slab and simultaneous solution procedures without getting into details of the numerical solutions in ATHOS. Such a discussion would be out of place in the present work. A detailed description of the slab by slab and simultaneous solution procedures is given in Ref. (S3). The unnamed procedure is better suited than the CAP as described in (S3), for units without economizer. The CAP could be adapted for use in other types of steam generators, but it appears that this is not done.

Whether the CAP or the unnamed procedure is utilized, the enthalpy of the liquid entering the evaporator must be known. In ATHOS this quantity is calculated from mass and energy balances over a control mass corresponding to the water in the downcomer, as shown in Fig. 3.1-4. This approach assumes instantaneous mixing of the feed and recirculating flows into the whole downcomer volume, and that the properties of the liquid entering the evaporator are at their instantaneous mixed mean values.

A description of the CAP and of the unnamed procedure follows.

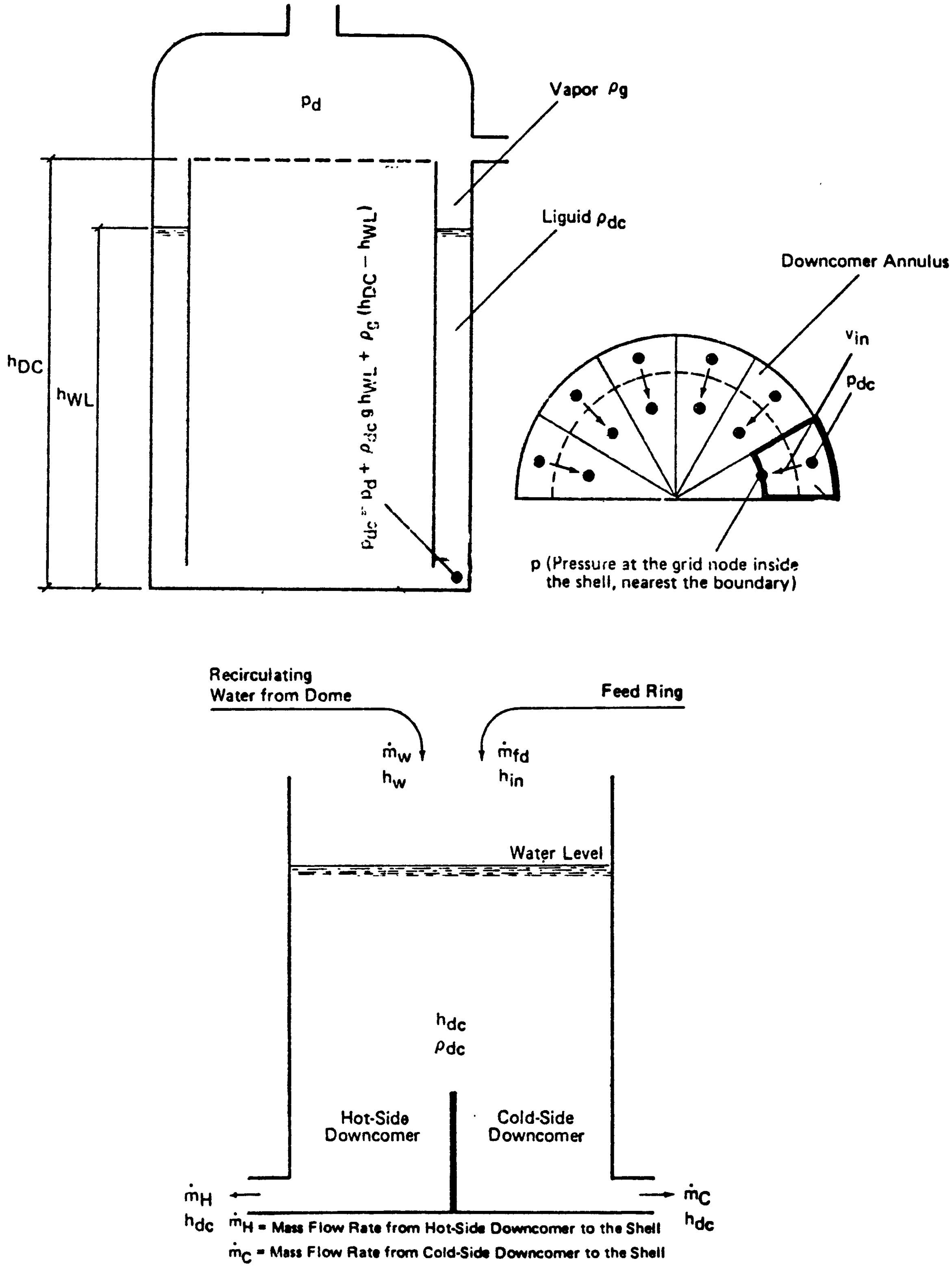


Figure 3.1-4. ATHOS (S3)
Above: Notation for shell-entry velocity calculation.
Below: Mass and heat balances in downcomer.

In the unnamed procedure the shell-entry velocities are calculated based on the local pressure differences across the entry face. An entry face is the imaginary surface just above and perpendicular to the tube sheet, where the fluid enters the evaporator flowing radially inward from the downcomer. The relevant quantities are shown in Fig. 3.1-4. An entry velocity is computed from the momentum balance equation for a control volume as shown in Fig. 3.1-4 by:

$$v_{in} = \frac{(p_{dc} - p) + C\rho_{dc} v_{in}^{*2} + Kv_{in}^{*} + Iv_{in-}}{\rho_{dc} v_{in}^{*} + 2C\,\rho_{dc} v_{in}^{*} + K + 1}$$

In this expression, C is the loss coefficient for the passage, K is an under-relaxation constant, and I is the coefficient of the time derivative term. The pressure at the bottom of the downcomer is computed as

$$p_{dc} = p_d + \rho_{dc} g h_{WL} + \rho_{dc} g (h_{dc} - h_{WL}).$$

The method clearly neglects all non-gravity terms in the downcomer momentum balance. While friction, drag, and acceleration are normally comparatively small, the same is not necessarily true of the inertia component. Hence, this procedure for calculating the downcomer pressure could introduce error for transients in which the downcomer flow is significantly accelerated, as during the dome pressure surge that usually follows turbine load rejection.

When the slab by slab method of solution is used in the evaporator region, the CAP is used to update the flow rates $\dot{m}_C$ and $\dot{m}_H$ entering that region from the downcomer, at points C and H shown in Fig. 3.1-3. In the following description of the CAP the nomenclature used is:

$\dot{m}_C^*$ = guessed (or stored from the previous iteration) value of $\dot{m}_C$

$\Delta\dot{m}_C$ = correction or adjustment to $\dot{m}_C^*$

Δp_C = pressure imbalance in the cold side circuit, due to pressure drops having been calculated based on approximate flow rates ($\dot{m}_C^*$ and $\dot{m}_H^*$)

$\dot{m}_H^*$, $\Delta\dot{m}_H$, Δp_H have analogous meanings.

The total flow rate is given by:

$$\dot{m}_T = \dot{m}_H + \dot{m}_C + \dot{m}_F \ . \tag{3.1-2}$$

A momentum balance around the natural circulation loop yields:

$$(p_D - p_C)_{\text{through downcomer}} = (P_D - P_c)_{\text{through shell}} \ .$$

This equality is then written in the form:

$$\rho_1 g(h_L - h_c) - K_{D,C}\dot{m}_C^{\ 2} = K_C\dot{m}_C^{\ 2} + K\dot{m}_T^{\ 2} + \sum_{Z=Z_C}^{Z_D} \bar{\rho} g \Delta Z \ ,$$

where h_L and h_C are respectively the heights of the downcomer water level and the cold-side downcomer entry port; $K_{D,C}$ is a constant for downcomer flow resistance; and K and K_C are the analogous constants for flow between M and D and C and M, respectively. The exact values of $\dot{m}_C$ and $\dot{m}_H$ are not known a priori. The above equation is written for guessed or approximate flow rates in the form

$$\Delta p_C = K_{D,C}\dot{m}_C{}^{*2} + K_C\dot{m}_C{}^{*2} + K\dot{m}_T{}^{*2} + \sum_{Z_C}^{Z_D} \bar{\rho} g \Delta Z - \rho_1 g(h_L - h_c) \qquad (3.1\text{-}3)$$

and for the hot side circuit

$$\Delta p_H = K_{D,H}\dot{m}_H{}^{*2} + K_H\dot{m}_H{}^{*2} + K\dot{m}_T{}^{*2} + \sum_{Z_H}^{Z_M} \bar{\rho} g \Delta Z \ +$$

$$+ \sum_{Z_M}^{Z_D} \bar{\rho} g \Delta Z - \rho_1 g(h_L - h_H) \qquad (3.1\text{-}4)$$

The key assumption to the CAP is that

$$\Delta p_i = \frac{\partial(\Delta p_i)}{\partial \dot{m}_i} \Delta \dot{m}_i + \frac{\partial(\Delta p_i)}{\partial \dot{m}_j} \Delta \dot{m}_j$$

where i = H, j = C for one equation , and i = C and j = H for another.

The four circuit-pressure-correction coefficients $\frac{\partial \Delta p_i}{\partial \dot{m}_j}$ with i = C,H and j = C,H are obtained by differentiating the two expressions for the Δp_i's: Eqs. (3.1-3) and (3.1-4). For example, from Eq. (3.1-3), with Eq. (3.1-2) in mind, it can be seen that:

$$\frac{\partial \Delta p_C}{\partial m_H} = 2K\dot{m}_T^* .$$

The coefficient K is determined as follows. Let,

$$\Delta p = R + B + M,$$

where Δp is the pressure drop between two points, say M and P; R is resistance; B is buoyancy; and M is momentum. The momentum component is argued to be small and neglected. Then

$$R \equiv \Delta p - B \tag{3.1-5}$$

with

$$B = g \int \bar{\rho} dz$$

where,

$$\bar{\rho} = \frac{\int\int \rho R d\theta dZ}{\int\int R d\theta dZ} .$$

Making use of the formulation:

$$R = K\dot{m}_T^{\,2}$$

it is possible to estimate

$$K = \frac{\Delta p - B}{\dot{m}_T^{\,2}} .$$

The remaining coefficients are evaluated analogously.

As a final remark regarding the CAP, it should be noted that the downcomer flow inertia effects are also neglected, as in the unnamed procedure. This is evidenced in Eq. (3.1-5).

(b) Primary and Heat Transfer

The primary model in ATHOS is based on the temperature equation for the fluid within one tube:

$$\rho c r \frac{\partial T_p}{\partial t} + cG\frac{\partial T_p}{\partial s} = \dot{q}_p \ , \qquad (3.1\text{-}6)$$

where c is specific heat, r is the volume of primary fluid per unit of shell volume, G is the primary mass flow rate per area of shell space in the axial direction, s is the path, and $\dot{q}_p$ is the rate of heat transfer per unit shell volume into the primary fluid.

For the tube metal it is written:

$$\rho_m c_m r_m \frac{\partial T_m}{\partial t} = -\dot{q}_p - \dot{q}_s \ , \qquad (3.1\text{-}7)$$

where the subscript m stands for metal, s for secondary.

The two volumetric rates of heat transfer, $\dot{q}_p$ and $\dot{q}_s$, are written in terms of fluid and metal temperature by lumping half the tube thermal resistance into each fluid. Thus,

$$q_i = \frac{1}{\left(\frac{1}{h_i} + \frac{1}{2}\frac{1}{H_i}\right)} \left(\frac{A}{V}\right) (T_m - T_i) \qquad (3.1\text{-}8)$$

where i stands for s (secondary) and p (primary) alternately, H_i is the reciprocal of the resistance to heat transfer exerted by the metal tube-wall itself, A is tube outer heat transfer area, and V is shell volume.

Equation (3.1-6) is discretized in a fully implicit upwind scheme over a node which corresponds to the secondary-side cell. Hence, the resulting algebraic equation represents the average primary temperature within that secondary-side cell. At each node, this equation plus its counterpart for the secondary-side fluid temperature, plus Eq. (3.1-8) applied for each of the q's constitutes an algebraic system of four equations with five unknowns: T_p, T_{sec}, $\dot{q}_p$, $\dot{q}_s$, T_m. By inserting Eq. (3.1-8) written for both i = s and i = p into Eq. (3.1-7), an expression for the metal temperature is obtained. This expression is then substituted into the equation for $\dot{q}_p$. This result is inserted into the primary temperature equation. The result is an expression relating primary and secondary temperatures, old time tube metal temperatures, and upstream values of the primary temperature.

In the overall solution procedure, the primary temperature is the first variable solved for. The energy equation for the secondary side is then solved for secondary side enthalpies. The source term for this equation, $\dot{q}_s$, is available from the new primary temperatures through the substitution procedure previously described. The momentum equations are solved last. The whole cycle is repeated until a convergence criterion is satisfied. In ATHOS there is one average primary temperature per secondary cell. This averaging might lead to artificial primary temperature predictions, particularly in cells in the U-tube bend regions in the cold side, if cells are too large. It also may affect the local heat flux prediction because of its effect on the heat transfer coefficient.

(c) Evaporator-Riser

ATHOS offers the options of homogeneous equilibrium and algebraic slip for the evaporator and riser analyses. Figure 3.1-3 illustrates the axial and radial nodalization schemes. Conservation equations for mass, momentum, and energy are integrated over each cell volume and a porous body formulation is used. The time discretization is fully implicit. The numerical method employed in the solution is an expansion of that developed by Patankar et al. (P1). The procedure is described in detail for the simpler case of single phase incompressible flow in (P2). The analysis in this region is clearly the most complicated and a more detailed discussion of ATHOS' method in this region is beyond the scope of this work. Reference (S3) provides a detailed description of this issue.

(d) Summary

In view of the preceding analysis and of recent code assessment work (H5), the following conclusions can be drawn of ATHOS as described in (S3).

First, that it is a powerful tool for a detailed analysis of the evaporator in steady state conditions. In this context it is a formidable calculational package for the comparison of performance of alternate designs, parametric studies to alter a design, prediction of performance under off-design conditions.

Its accuracy for transient analyses is restricted to long and slow transients, primarily by the simplicity of its models for steam dome, downcomer, and shell-entry velocities. Also, because ATHOS is discretized in a fully implicit manner, there are no upper limits on time step size. However, by virtue of this same feature, the computation time on a per time step basis is on the order of 30 seconds. This high value implies that ATHOS is not constructed for situations in which small time steps are required, even if the steam dome and downcomer models were improved to give it this capability. Nevertheless, for mild operational transients where time steps of tens of seconds can be taken and analyses on the order of hours are desired, ATHOS seems to have all the most appropriate features.

3.1.2.2 TRANSG-01

In contrast with ATHOS, this code incorporates a reasonably detailed model of the physical processes occurring in the steam dome and downcomer. These features, along with its numerical method, indicate that this code is applicable to more severe transient analyses over shorter time periods.

Figure 3.1-5 illustrates the regions represented in TRANSG as well as the mesh scheme used. The boundary conditions are: primary pressure, flow rate and inlet temperature, steam flow (at the outlet nozzle), feedwater flow rate, and temperature.

The present discussion on TRANSG is divided into three parts: (a) steam dome and downcomer, (b) primary and heat transfer, (c) evaporator-riser. In addition to the main reference on this code (L2), an extensive independent review is also available in Ref. (H6).

(a) Steam Dome and Downcomer

For UTSG applications, the link between the downcomer and the riser is provided by a recirculation (or steam dome) model. This model is described in detail in (L2). Briefly, it consists of conservation equations of mass and energy written for two control volumes with a moving boundary between them. The union of the two control volumes is the steam dome plus feed chamber volume. It is assumed that one of the volumes is of saturated steam and the other of subcooled water. The water is assumed subcooled because the feedwater mixing takes place in this liquid control volume. The energy equation for the steam is redundant since the steam is assumed saturated. Thus, there are four unknowns: dome pressure, water enthalpy, and the volumes of steam and water. In addition to the three conservation equations mentioned, an equation stating that the sum of the two volumes is constant in time provides closure to the set.

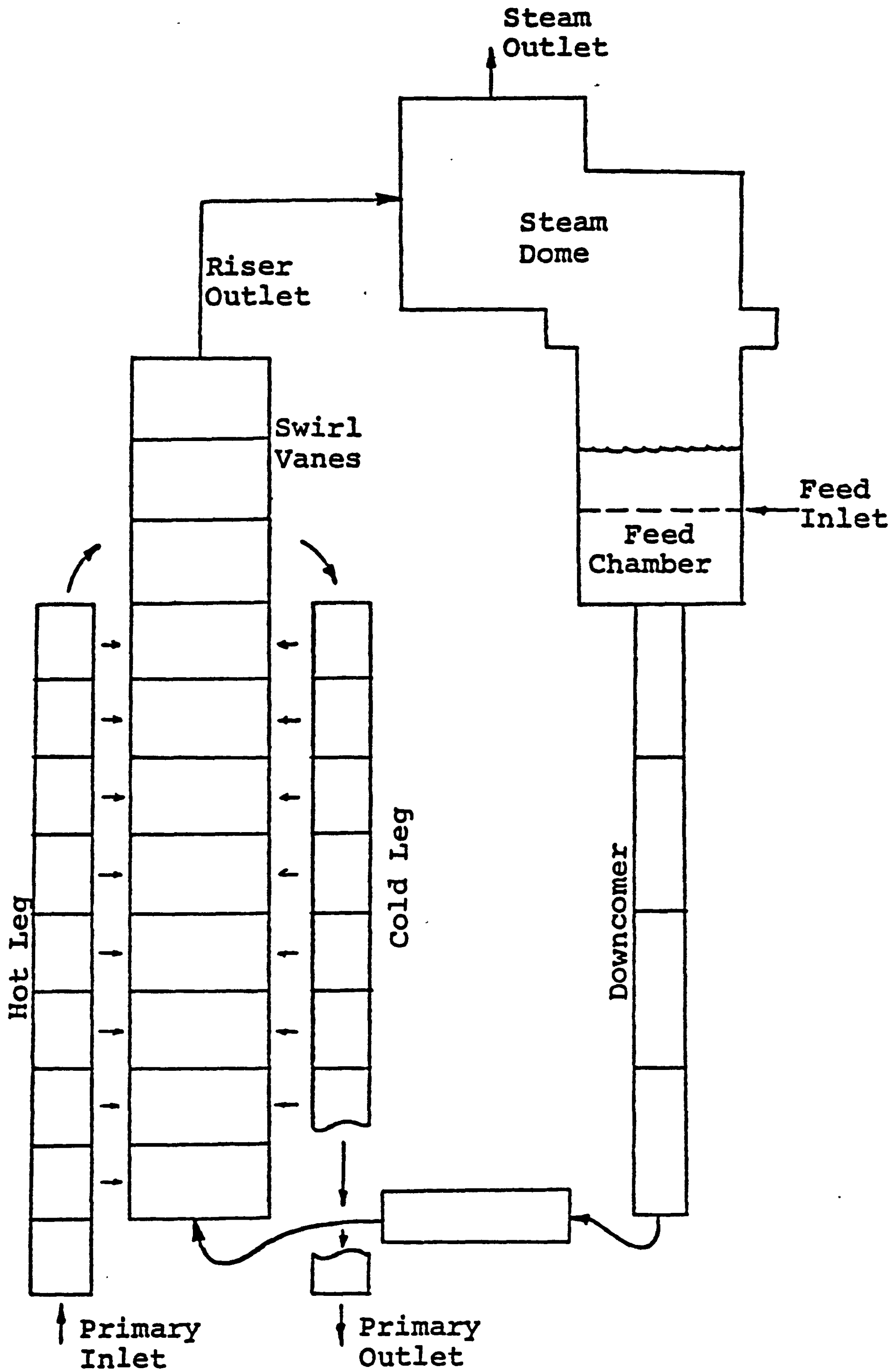


Figure 3.1-5. Model regions for TRANSG (L2).

For the evaporator model, the steam dome pressure is used as the boundary condition at the riser outlet. The other hydrodynamic boundary condition for the secondary side loop is at the top of the downcomer. This is also a pressure, which is determined by adding the hydrostatic head in the feed chamber to the dome pressure. The feed chamber enthalpy, which also results from the steam dome calculation is also a needed boundary condition.

(b) Primary and Heat Transfer

The primary is analyzed with the same equations, discretization procedure and numerical method as the secondary except that single phase flow is imposed. A one-dimensional slab heat conduction model is used to represent the thermal behavior of primary tube walls. The primary side and secondary side solution procedures are decoupled. This is done by utilizing the primary side wall heat flux explicitly in the primary side energy equation and analogously for the secondary. The wall temperatures are updated by using the two advanced fluid temperatures as boundary conditions. Only one primary tube is represented.

(c) Evaporator-Riser

Given the riser and downcomer top pressures, the feed chamber enthalpy, and the previous time wall heat flux, the evaporator, riser, and downcomer regions are all solved using the same one-dimensional discretization procedure and numerical method, as indicated in Fig. 3.1-5. In these regions, the conservation equations are formulated for Thom's (T2) separated flow model. Obviously, the homogeneous equilibrium model is easily invoked by setting the slip ratio to unity. The numerical method is the EICE (Extended Implicit Continuous Eulerian) which allows for a dependency of density on enthalpy not available in the original ICE (H2) formulation.

(d) Summary

To summarize the discussion on TRANSG the following points are worthy of notice.

The steam dome and downcomer model allows the analysis of reasonably severe operational transients. There is a possibility of under-predicting the pressure in cases where the steam outlet nozzle flow drops abruptly because the steam in the dome is not allowed to become superheated. This can occur although conservation equations for the dome accurately predict the internal energy. The reason is that the saturation assumption yields a lower pressure for a given internal energy than if superheat were allowed. Data comparison with predictions of a turbine trip test at the Donald C. Cook plant, published in Ref. (L1), are consistent with this observation.

Finally, the liquid control volume properties, corresponding to feed chamber enthalpy, are always added into the topmost computational cell in the downcomer. This can have an impact on predictions where the transport time for these properties from their mixing position to evaporator inlet is important.

To conclude, in spite of simplifications, the steam dome and downcomer representations in TRANSG do model the actual physical processes. This, along with the availability of many heat conduction nodes in the tube walls and the EICE solution procedure, indicate that this code is most suitable for short time frame analyses of severe operational transients, as suggested by published calculation (L2).

3.2 The Two-Fluid Model

Prior to examining the models developed in this work for the various regions of the U-tube steam generator, it is appropriate to review the framework into which all the models were fit. This framework has been provided by THERMIT2 (K1), a modified version of THERMIT (R1): a two-fluid, three-dimensional code developed at MIT under EPRI sponsorship.

3.2.1 General Characteristics

THERMIT2 (from now on referred to as THERMIT) solves the three-dimensional, two-fluid equations describing two-phase flow and heat transfer dynamics. The two-fluid model uses separate partial differential equations to express the conservation of mass, momentum, and energy for the vapor and for the liquid. The model allows for thermal and mechanical non-equilibrium between the phases. In this context, constitutive relations for the exchange of mass, momentum, and energy between the phases are required. These relations are briefly reviewed in Section 3.2.3.

A second important characteristic of THERMIT is that it offers the choice of either pressure or velocity boundary conditions at the top and bottom of the solution domain. This feature permits realistic modeling as well as integration of downcomer, evaporator, and riser regions of the U-tube steam generator into one continuous solution domain.

A third important characteristic of THERMIT is its continuous general boiling curve. The boiling curve is an adaptation, (R1), (K1), of the BEEST, (B3), model. Basically, the model allows four regimes: convection to liquid, subcooled and nucleate boiling, transition, and convection to vapor. The heat being transferred from the tube walls and into the secondary fluid in the evaporator is represented by a heat transfer coefficient (obtained from a correlation appropriate to each heat transfer regime) times the outer wall-fluid temperature difference. An exception to this procedure is made in the transition region, where the heat flux is prescribed directly, by linearly interpolating between the critical heat flux and the heat flux at the minimum stable film boiling temperature.

Finally, the numerical method used in THERMIT is a semi-implicit technique with the following stability restriction on the maximum allowable time step:

$$\Delta t < (\Delta/v)_{min},$$

where Δ is some mesh length and v is the largest phase velocity in the direction of Δ. Convergence can always be obtained provided the time step is sufficiently small. The numerical method was originally conceived (R1) and recently improved for computational speed (S4) for severe transient analysis. There are, however, other limitations to this severity as discussed in Section 3.2.3. THERMIT combines an advanced two-phase flow model with a very reliable numerical method. However, this combination alone does not guarantee accurate predictions. Constitutive equations must also be suitable for the given application. Furthermore, local boundary conditions, which in the present UTSG applications are computed by other models, must also be accurately supplied.

3.2.2 Conservation Equations

The conservation equations of mass, momentum, and energy of the two-fluid model in THERMIT have been derived (K1) by averaging local instantaneous equations first over time and then over an arbitrary volume. The volume and time averaged equations are reproduced here for completeness.

Conservation of Vapor Mass

$$\frac{\partial}{\partial t}(\alpha\rho_v) + \nabla \cdot (\alpha\rho_v\vec{V}_v) = \Gamma - W_{tv}$$

Conservation of Liquid Mass

$$\frac{\partial}{\partial t}[(1-\alpha)\rho_l] + \nabla \cdot [(1-\alpha)\rho_l\vec{V}_l] = -\Gamma - W_{tl}$$

Conservation of Vapor Energy

$$\frac{\partial}{\partial t}(\alpha\rho_v e_v) + \nabla \cdot (\alpha\rho_v e_v\vec{V}_v) + P\,\nabla \cdot (\alpha\vec{V}_v) + P\frac{\partial\alpha}{\partial t}$$

$$= Q_{wv} + Q_i - Q_{tv}$$

Conservation of Liquid Energy

$$\frac{\partial}{\partial t}[(1-\alpha)\rho_l e_l] + \nabla \cdot [(1-\alpha)\rho_l e_l\vec{V}_l] = +P\,\nabla \cdot [(1-\alpha)\vec{V}_l]$$

$$-P\frac{\partial\alpha}{\partial t} = Q_{wl} - Q_i - Q_{tl}$$

Conservation of Vapor Momentum

$$\alpha\rho_v\frac{\partial\vec{V}_v}{\partial t} + \alpha\rho_v\vec{V}_v \cdot \nabla\vec{V}_v + \alpha\,\nabla P = -\vec{F}_{wv} - \vec{F}_{iv}$$

$$+ \alpha\rho_v\vec{g} - \vec{F}_{tv}$$

Conservation of Liquid Momentum

$$(1-\alpha)\rho_l\frac{\partial\vec{V}_l}{\partial t} + (1-\alpha)\rho_l\vec{V}_l \cdot \nabla\vec{V}_l + (1-\alpha)\,\nabla P =$$

$$-\vec{F}_{wl} - \vec{F}_{il} + (1-\alpha)\rho_l\vec{g} - \vec{F}_{tl}$$

These equations incorporate two assumptions: (a) that viscous stress and energy dissipation can be neglected and (b) that the liquid and vapor pressures are equal within a control volume. None of these assumptions is likely to pose any restrictions for UTSG applications, provided the nodes are not longer than a couple of meters in the direction of the gravitational field.

3.2.3 Constitutive Equations

To provide closure to the set of conservation equations written for each phase, constitutive relations are required representing various transport processes. These processes occur in the porous body formulation of the THERMIT model at three types of interfaces: (a) wall to coolant and (b) liquid to vapor, both within cell volume; and (c) inter-cell at cell boundaries. Table 3.2-1 classifies the various processes according to this criterion.

A brief discussion of the main features of these processes is given next. For detailed discussion on correlation selection criteria, the reader is referred to (R1), (K1), and (E1). Meticulous analyses on numerical aspects of expressions for the interactions are available in (K1) and (S4).

(a) Wall to Coolant Interactions

As shown in Table 3.2-1 there are two types of wall to coolant interactions: hydrodynamic (wall friction and form losses) and heat transfer (wall heat transfer).

Table 3.2-1.

Summary of transport processes.

Wall to Coolant

F_{wl} - Wall Frictional Force on the Liquid

F_{wv} - Wall Frictional Force on the Vapor

Q_{wl} - Wall Heat Transfer to the Liquid

Q_{wv} - Wall Heat Transfer to the Vapor

Liquid to Vapor

Γ_i - Interfacial Mass Transfer Rate

F_i - Interfacial Momentum Exchange Rate

Q_i - Interfacial Heat Exchange Rate

Inter-Cell

W_{tv} - Turbulent Vapor Mass Exchange Rate

W_{tl} - Turbulent Liquid Mass Exchange Rate

Q_{tv} - Turbulent Vapor Energy Exchange Rate

Q_{tl} - Turbulent Liquid Energy Exchange Rate

F_{tv} - Turbulent Vapor Momentum Exchange Rate

F_{tl} - Turbulent Liquid Momentum Exchange Rate

The wall friction terms in THERMIT, F_{wl} and F_{wv} are determined by apportioning known two-phase friction pressure gradient correlations between liquid and vapor. Thus,

$$\left.\frac{\partial p}{\partial z}\right|_f = \frac{fG^2}{2D_h\rho_l}\phi_{lo}^2 = F_{wv} + F_{wl} \ .$$

The apportioning is accomplished through the use of a parameter θ referred to as the liquid contact fraction. In practice this parameter is usually set to unity. This means that all the wall friction is experienced by the liquid, which will then partially transmit this force to the vapor via the interfacial momentum exchange term. Exceptions to this norm occur when the only phase present is vapor and between the critical heat flux and the minimum stable film boiling points, where θ is made to vary linearly from one to zero respectively.

Three types of expressions are included in the THERMIT wall friction subroutine. One is axial friction, with three choices for the two-phase friction multiplier. A second is form loss associated with spacer grids, which is uniformly distributed along a computational momentum cell. A third type of expression is transverse friction, for which there is one correlation with one two-phase multiplier choice. Table 3.2-2 summarizes the available options.

Recently, (R2) THERMIT has been equipped with the option of refining the computation of wall-fluid interactions in cases

Table 3.2-2. Summary of wall-fluid correlations available for parallel and perpendicular, single- and two-phase flows.

		TWO PHASE			SINGLE PHASE
		Levy	Martinelli-Nelson	Martinelli-Nelson-Jones	
AXIAL FRICTION	F^z_{wl}	$\frac{\theta f_l}{2D_h}\rho_l \lvert v^z_l\rvert v^z_l$	$v^z_l\left\{\frac{\theta f_l}{2D_h}(1-\alpha)^2\rho_l v^z_l + \frac{\theta}{2D_h}c\alpha(1-\alpha)(f_v f_l\rho_v\rho_l)^{1/2}\lvert v^z_v\rvert\right\}$	$v^z_l\left\{\frac{\theta f_l}{2D_h}[(1-\alpha)^2\rho_l\lvert v^z_l\rvert - (1-\alpha)\alpha\rho_v\lvert v_v{}^z\rvert]\right\}\phi_{lo}{}^2$	$\frac{\theta f_l}{2D_h}\rho_l v^z_l\lvert v^z_l\rvert$
	F^z_{wv}	$\frac{(1-\theta)f_v}{2D_h}\rho_v\lvert v^z_v\rvert v^z_v$	$\frac{(1-\theta)f_v}{2D_h}\alpha^2\rho_v\lvert v^z_v\rvert v^z_v$	$\frac{(1-\theta)f_v}{2D_h}\alpha\rho_v\lvert v^z_v\rvert v^z_v$	$\frac{(1-\theta)}{2D_h}\rho_v\lvert v^z_v\rvert v^z_l$
2Ø MULTIPLIER	ϕ^2_{lo}	$\left(\frac{1-x}{1-\alpha}\right)^{2-b}$	$1+\frac{c}{X_{tt}}+\frac{1}{X^2_{tt}}$ (c=21)	$F(G,p)\left\{1.2\left[\left(\frac{\rho_l}{\rho_v}\right)-1\right]x^{.824}\right\}+1$	
TRANSVERSE FRICTION	F^x_{wl}	(Not Available) NA	Same as above but with $v^x_{l,max}$ instead of v^z_l (c=8)	NA	$\frac{\theta f_l}{2D_v}\rho_l\lvert v^x_{l,max}\rvert v^x_{l,max}$
	F^x_{wv}	NA	idem for $v^x_{v,max}$ and v^z_v (c=8)	NA	$\frac{(1-\theta)}{2D_v}f_v\rho_v\lvert v^x_{v,max}\rvert v^x_{v,max}$
FORM LOSS	$F^z_{wl,local}$	$\frac{K}{2\Delta z}[(1-\alpha)\lvert G\rvert + \alpha(1-\alpha)\rho_l\lvert v^z_v\rvert]v^z_l$			$\frac{K}{2\Delta z}\lvert G\rvert v^z_l$
	$F^z_{wv,local}$	$\frac{K}{2\Delta z}\alpha^2\rho_v\lvert v_v{}^z\rvert v_v{}^z$			$\frac{K}{2\Delta z}\rho_v\lvert v^z_v\rvert v^z_v$
PARAMETERS		$1/X_{tt}{}^2 = \left(\frac{x}{1-x}\right)^{2-b}\left(\frac{\rho_l}{\rho_v}\right)\left(\frac{\mu_v}{\mu_l}\right)^b$ $G = \alpha\rho_v v^z_v + (1-\alpha)\rho_l\lvert v^z_l\rvert$ $G_0 = 950\ Kg/m^2/s$ $F(G,p) = \begin{cases}1.43+\frac{G-G_0}{G_0}(.07-7.35\times10^{-8}p) & G < G_0\\ 1.43+\left(\frac{G_0}{G}-1\right)(.17-6\times10^{-8}p) & G > G_0\end{cases}$			$f_i = 64(Re^z_i)^{-1}$ $Re_i < Re^*$ $f_i = a(Re^z_i)^{-b}$ $Re_i > Re^*$ $i = l,v$ $Re^* = 1353$ $Re^z_i = \rho_i v^z_i D_h \mu_i$ $a=b=.2$

where the flow is neither parallel nor perpendicular to the tube bundle, as in the U-bend region of steam generators. The physical bases for the method are extensively discussed in (E1). Basically, a coordinate transformation is utilized to compute a wall-fluid force through oblique rods. The calculation procedure is done in three stages. First, the flow velocity in the code mesh coordinate system is tranformed into the coordinates parallel and perpendicular to the rods. Next, the transformed velocities are used to calculate friction (parallel component) and drag (normal component) forces using the appropriate correlations. Finally, the resulting force on the fluid is converted back to the original coordinate system.

The wall-fluid thermal interaction terms are determined through a set of flow regime dependent heat transfer correlations. An elaborate heat transfer logic is used in the determination of the appropriate flow regime. This so-called "heat transfer package" (B3), (R1), (K3) is based on the construction of a boiling curve at each location; and then for each local flow condition an appropriate heat transfer coefficient is selected. Figure 3.2-1 summarizes the heat transfer logic for regime determination and Table 3.2-3 presents the heat transfer coefficient correlations used in each regime.

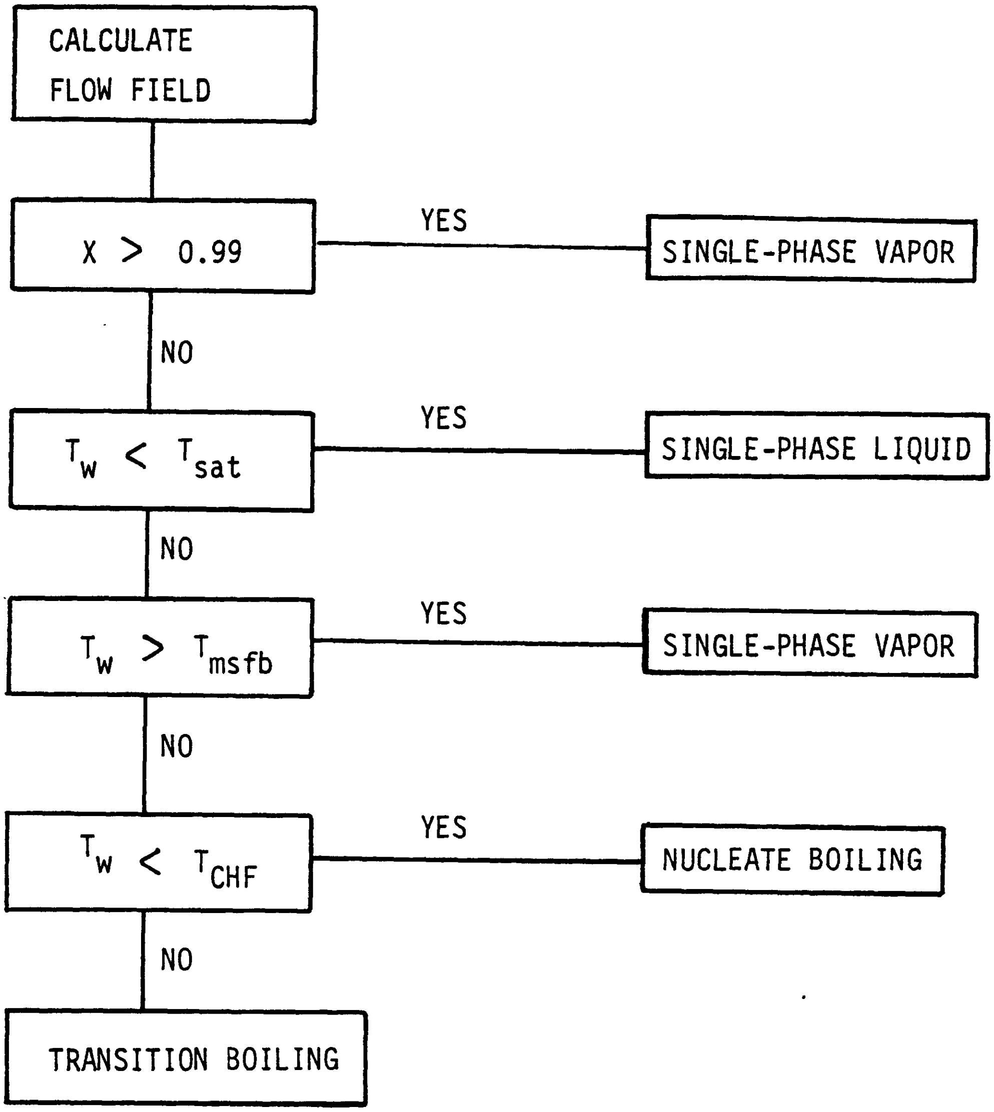


Figure 3.2-1. Heat transfer logic (Ref. K1).

Table 3.2-3. Heat transfer correlations.

Regime	Correlation	Author
1. Forced convection to single phase liquid	$h_{ST} = 0.023 \frac{k}{D} Re^{.8} Pr^{.33} (\mu/\mu_w)^{.14}$ (Fluid properties at bulk fluid temperature, except μ_w at T_w.)	Sieder-Tate (S6)
2. Natural convection to single phase liquid	$h_{MA} = 0.13\, k[\rho^2 g\beta(T_w - T)Pr/\mu^2]^{.33}$ (Fluid properties should be at a fluid film temperature.)	McAdams (M1)
3. Subcooled boiling	$q_{Chen} = h_{fc}(T_w - T_l) + h_{nb}(T_w - T_{sat})$ $\quad h_{fc} = 0.023 \frac{k_l}{D} Re_l^{.8} Pr_l^{.4} F$ $h_{nb} = 0.00122\, S \left[\frac{k_l c_{pl}}{\sigma}\right]^{.5} Pr_l^{-.29} \rho_l^{.25} (P_w - P)^{.75} \left[\frac{c_{pl}(T_w - T_{sat})\rho_l}{h_{vl}\rho_v}\right]^{.24}$	Chen (C1)
4. Nucleate boiling	$F = \begin{cases} 1 & X_{tt}^{-1} \leq 0.1 \\ 2.35(X_{tt}^{-1} + .213)^{.736} & X_{tt}^{-1} > 0.1 \end{cases}$ $\quad S = \begin{cases} [1 + .12Re_{TP}^{1.14}]^{-1} & Re_{TP} < 32.5 \\ [1 + .42Re_{TP}^{.78}]^{-1} & 32.5 \leq Re_{TP} \leq 70 \\ 0.1 & Re_{TP} > 70 \end{cases}$ $Re_{TP} = 10^{-4} F^{1.25} (1-\alpha) \rho_l v_l D/\mu_l$ $\quad X_{tt}^{-1} = [x/(1-x)]^{.9} (\rho_l/\rho_v)^{.5} (\mu_v/\mu_l)^{.1}$	
5. Transition	Interpolation between heat fluxes at the minimum stable film boiling point and at the critical heat flux point.	- -
6. Forced convection to single phase vapor	Same as Number 1.	Sieder-Tate
7. Natural convection to single phase vapor	Same as Number 2.	McAdams

(b) Liquid to Vapor Interactions

As noted in Table 3.2-1, three interphase exchange terms are needed for closure: the mass transfer rate, the momentum exchange rate, and the energy exchange rate.

The energy exchange rate between liquid and vapor is modeled with different but equivalent formulations depending on whether the flow regime is subcooled or nucleate boiling or whether it is post critical heat flux droplet vaporization. For UTSG applications the former regimes are more important and only those will be reviewed briefly. In the present formulation, the interfacial energy exchange rate for subcooled and saturated boiling is given by:

$$Q_i = H_{iv}(T_{sat} - T_v) + \Gamma h_g$$

where Γ is the mass transfer rate (to be discussed later in this subsection), h_g is the vapor saturation enthalpy at the prevailing pressure and H_{iv} is an interface to vapor heat transfer coefficient. The interface to vapor formulation is chosen for these flow regimes because the vapor is saturated while the bulk of the liquid may be subcooled. By setting H_{iv} to a very large value (e.g. 10^{11} W/m^3) the vapor is forced to saturation while the liquid, whose bulk temperature is not used in this equation, is unconstrained and may therefore be subcooled.

The momentum exchange rate between liquid and vapor is represented by a force per unit free cell volume of one phase on the other. At least five types of forces have been postulated to exist: viscous, inertial, bouyancy, virtual mass, and Basset. The last two apply to rapidly accelerating flows which are anticipated in blowdown transients and may be important for the analysis of steam line breaks. They are not included in the version of THERMIT upon which the present work is based. However, a detailed study on this problem is given in Ref. (N2) in connection with sodium boiling. In the current version, only the first two forces are included in the interfacial momentum exchange rate viz.: viscous and inertial forces. There are however two models which combine these forces differently: the MIT model and the Los Alamos model. Both are described in detail in Ref. (K1).

To conclude the discussion on liquid to vapor interactions, it remains to review the mass exchange rate term, Γ.

The mass exchange rate models incorporated into THERMIT vary with three heat transfer regimes: subcooled boiling, saturated flow boiling, and droplet vaporization. Condensation is also modeled.

In the subcooled boiling regime, both the vapor generation on the heated walls as well as condensation in the bulk of the liquid are modeled. Vapor generation is based on the model of Ahmad (A1) in which it is assumed that the vaporization rate increases linearly between a bubble departure temperature,

T_d, and the saturation temperature, T_s. For bulk liquid temperatures below T_d the net vaporization is zero. For $T_l = T_s$, all the wall heat flux leads to vapor generation. This equilibrium vaporization rate is written as

$$\Gamma_s = q_w/h_{fg} \quad ,$$

where q_w is the wall heat flux and h_{fg} the latent heat of vaporization. The model used to represent condensation is a conduction term divided by the latent heat viz.,

$$\Gamma_c = A_i H_i (T_l - T_v)/h_{fg}$$

where A_i is an interfacial area based on bubble radius (A1) and H_i is an interfacial heat transfer coefficient (K3). The subcooled vapor generation model should be applicable to all non-depressurization transients in steam generators. For those cases, which include the steam line break, flashing becomes the dominant vaporization mechanism and other vapor generation rate terms should be developed.

The droplet vaporization regime applies to post critical heat flux conditions. This regime is not within the scope of the applications presented in this work. Hence, the model (S5) will not be discussed here.

(c) Inter-Cell Interactions

Inter-cell interactions result from the mixing of fluid between adjacent channels. This mixing can be forced or natural. Forced mixing is caused by solids such as grids and wire wraps. THERMIT has the capability to incorporate forced mixing by appropriately defining flow areas. This capability is sufficient for the simulation of such items as baffles and diverters. Natural mixing includes diversion crossflow and turbulent mixing. THERMIT incorporates both forms of natural mixing. Diversion crossflow is caused by transversal pressure gradients and incorporated into the analysis via transverse momentum equations. Turbulent mixing results from interchannel eddy transport. Turbulence is important when eddy and mesh scales are comparable. In the steam generator evaporator, where the mesh scale is one or two orders of magnitude larger than the pitch between U-tubes, which can be taken as representative of the order of magnitude of eddy size in the tube bundle region, these terms are normally negligible. However, in the riser region the flow is comparatively unobstructed and larger eddies can be expected. There, eddy and channel scales could be comparable and turbulence could have an effect.

The turbulent mixing incorporated into THERMIT (K3) is an extension of what would be called in single-phase flow a zero equation or algebraic model. This means that an algebraic equation (R4) is used to assess the turbulent velocity, ε/l.

The generalized mixing flux is assessed from an expression of the type:

$$\psi = \sum_j \frac{S_{ij}}{A_i} \left\{ \frac{\varepsilon}{T} [(\phi P)_i - (\phi P)_j] \right\}_{ij} \tag{3.2-1}$$

where S_{ij} is the gap between rods, A_i is the channel flow area, and the parameters in Eq. (3.2-1) are given below.

ψ	ϕ	P
W_{tv}	1	αP_v
W_{tl}	1	$(1-\alpha)P$
Q_{tv}	e	αP_v
Q_{tl}	e	$(1-\alpha)P$
F_{tv}	v	αP_v
F_{tl}	v	$(1-\alpha)P_v$

The mixing velocity is given by (R4):

$$\varepsilon/l = \frac{1}{2} \lambda R_e^{-0.1} \left[1 + \left(\frac{D_j}{D_i}\right)^{1.5}\right] \frac{D_i}{D_{fs}} \frac{G_i}{\rho} \tag{3.2-2}$$

where,

$$\lambda = 0.0058 \left(\frac{S_{ij}}{D_{fs}}\right)^{-1.46}$$

D_{fs} = fuel rod diameter.

The extension to two-phase flow (K1) adds two physical effects: vapor diffusion and the dependency of the turbulent velocity on the flow regime. This formulation, however, is at present only applicable to subchannel analysis because Eq. (3.2-2) has been experimentally correlated from measurements in that type of configuration.

Chapter 4

THERMIT-UTSG: OVERALL STRUCTURE

4.1 Introduction

The U-tube steam generator regions represented in THERMIT-UTSG include: evaporator, riser, steam dome, downcomer, the primary fluid and the metal of the U-tubes. These regions, as shown in Figs. 2.1-2 and 2.2-1, come together to form a complete steam generator. THERMIT-UTSG constitutes thus a macroscale representation of the whole unit. However, the boundaries between the solution methodologies are not necessarily the same as the region boundaries. For example, the evaporator, riser, and downcomer are all incorporated into the domain of the two-fluid model reviewed in Chapter 3. The complete solution, however, is only possible after the development of the recirculation model, decribed in Chapter 5, which closes the natural circulation loop by linking riser outlet to downcomer. The two-fluid model also requires the power level and distribution at each time interval. These are obtained from the primary- side model, as described in Chapter 6.

The purpose of this chapter is to clearly define the domain and the function of each calculation procedure, viz.:

1. two-fluid model
2. recirculation model
3. primary-side model

as well as to present an overview on how they are linked to form the overall structure of THERMIT-UTSG. Details regarding each of these procedures and their coupling are given in Chapters 3, 5, and 6. System boundary conditions for transient and steady-state analyses are also presented and discussed in this chapter.

The term 'system boundary condition' is used to designate input required for the overall model. A distinction is made in this work between this term and the term 'local boundary condition.' The latter refers to a parameter at the interface between calculation procedures. Thus, the heat input to a secondary-side cell is a local boundary condition, whereas the primary inlet temperature is a system boundary condition.

4.2 Two-Fluid Model Domain

Figure 4.2-1 is the side view of a representative design U-tube steam generator. Figure 4.2-2 is the calculational model of the same UTSG including the downcomer, riser, and evaporator as represented by the two-fluid model in THERMIT-UTSG. The cross-hatched zones in Fig. 4.2-2 indicate fractions of downcomer cell volumes that are blocked off to the flow. There are other inaccessible regions to flow: namely, those areas of evaporator and riser which are occupied by structural material (including the U-tubes) such as can be seen in Fig. 4.2-1. While these areas are not shown in Fig. 4.2-2, for the sake of simplicity, they are nevertheless taken into

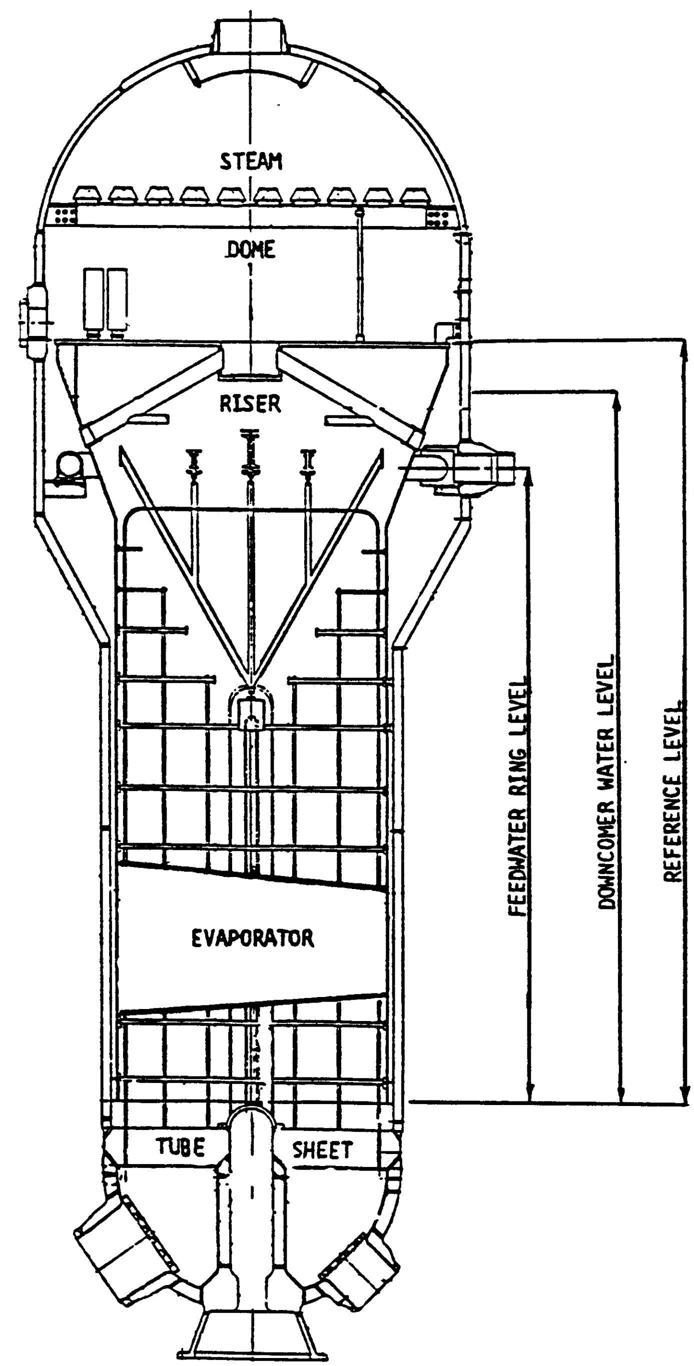


Figure 4.2-1. UTSG in side view
(adapted from Ref. G2).

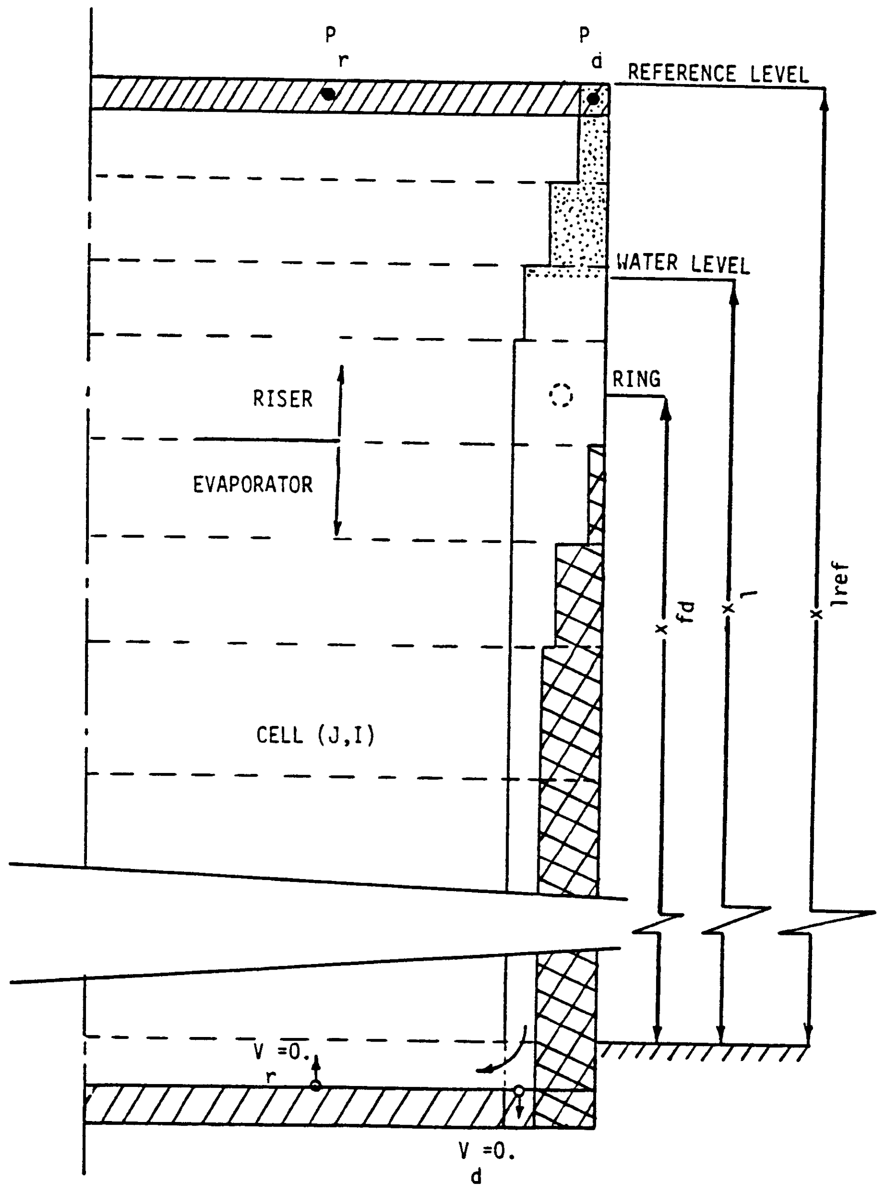


Figure 4.2-2. THERMIT-UTSG representation of two-fluid model solution domain.

consideration. By appropriately inputting free volumes and areas, in each cell, the presence of solids in the flow path is accounted for in the porous body formulation of the two-fluid equations.

Figure 4.2-3 shows in cross section a typical channel layout in the two-fluid model domain, at an elevation where the tubes are straight. In this configuration four channels represent the downcomer (1, 2, 4, and 6), channel 3 represents evaporator hot-side, and channel 5 the cold-side.

The number of tubes shown in Fig. 4.2-3 is only schematic. As described in Chapter 3, the two-fluid conservation equations have been volume integrated according to the porous body method where the location of solids within each channel is not relevant.

The hatched areas in Fig. 4.2-2 represent fictitious cells. The two-fluid equations are not solved there but the cells are used to define boundary conditions for that model. For instance, by defining a zero vertical velocity at the top of the bottom fictitious cells, as shown in Fig. 4.2-2, the flow is forced to turn horizontally from the downcomer into the evaporator. In the top fictitious cells, and in the steam dome as well, the relevant quantities are calculated by the recirculation model.

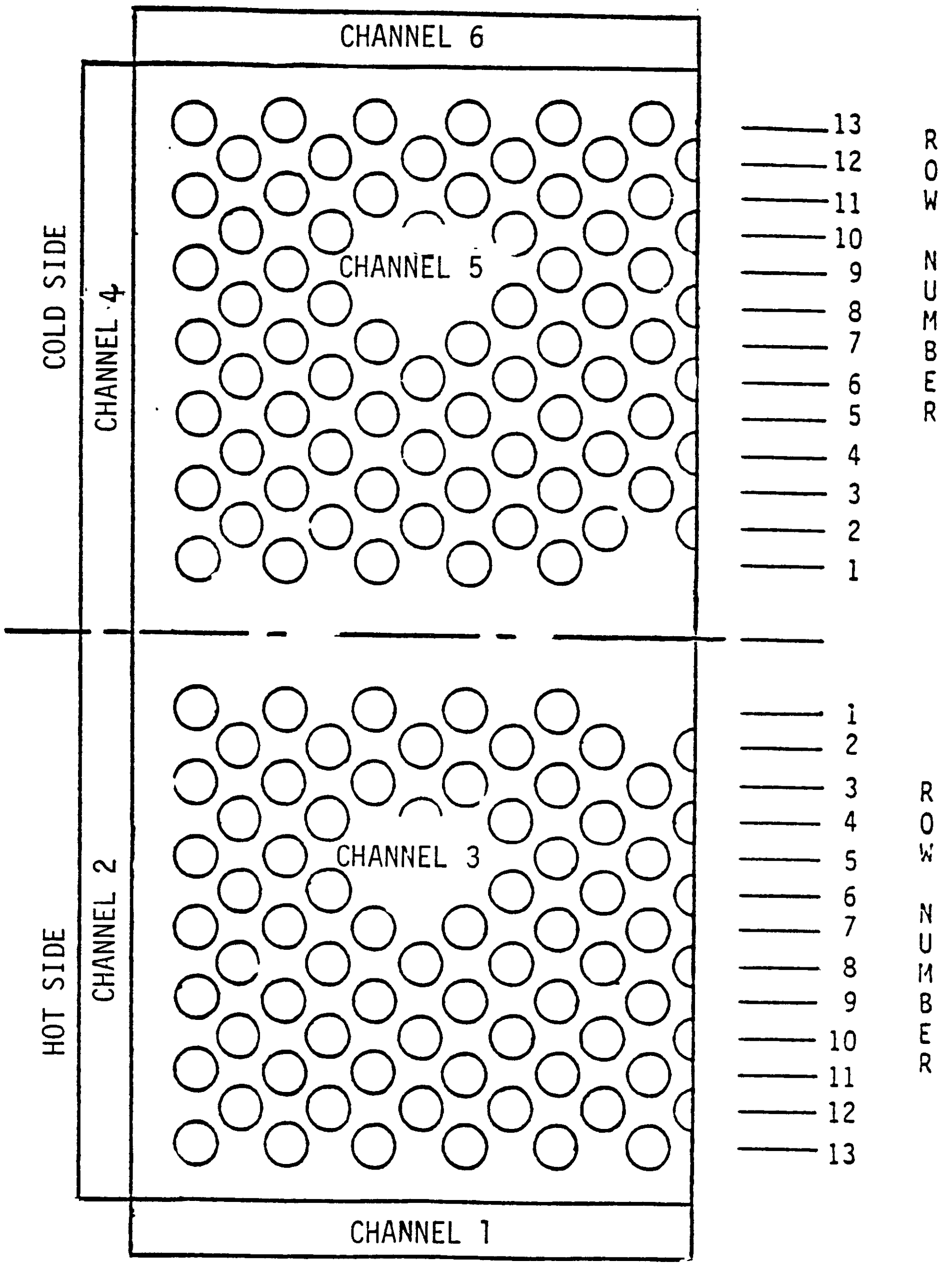


Figure 4.2-3. Typical channel layout for THERMIT-UTSG.

4.3 Recirculation Model Domain

The regions of the steam generator for which the recirculation calculation procedure is used are shown in Fig. 4.3-1. They include the steam dome and part of the downcomer, down to slightly below the water level.

The purpose of this model can be seen as twofold. One function is to compute important global parameters of the steam generator such as the downcomer water level and outlet nozzle steam flow. Another purpose is to provide the needed local boundary conditions for the two-fluid calculation. An overview of the latter function is necessary to describe the overall code structure. This section provides such an overview. A detailed description of the interfacing of the two calculational procedures is given in Chapter 5.

The two-fluid computation requires local boundary conditions at the topmost fictitious cells in all channels. The required local boundary conditions depend on whether there is inflow or outflow at that cell.

For the top boundary cells in the riser, an outflow boundary cell, only the pressure, p_r, need be prescribed. This is done in such a way that

$$p_r = p + \Delta p_{sep} \; .$$

The steam dome pressure, p, is taken as a system boundary condition or it can be calculated from the outlet nozzle steam

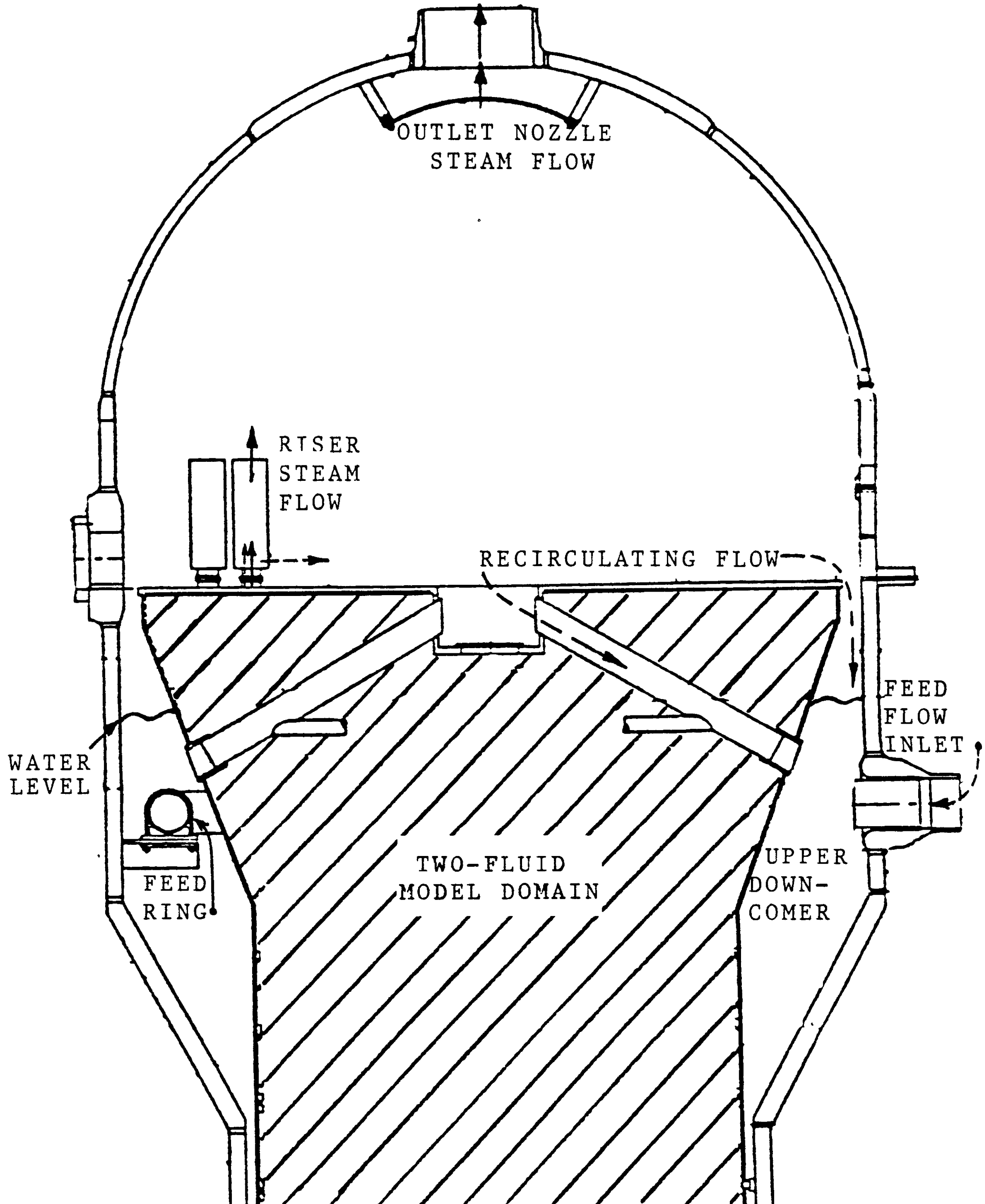


Figure 4.3-1. Recirculation model domain.

flow as described in Chapter 5. The pressure drop across the separators, Δp_{sep}, varies considerably with device design. Different designs will have different pressure losses for the same flow conditions even if separator efficiencies are comparable (R5). The basic expression used in this work is a correlation due to Burley (B4), viz.:

$$\Delta p_{sep} = \rho(C_1 . Q^2 + C_2)$$

where

Q = volumetric flow rate per separator (m^3/s)

ρ = mixture density (kg/m^3)

The pressure drop Δp_{sep} is in Pascal and nominally:

$C_1 = 4494.5$ and

$C_2 = 30.5$.

If the circulation ratio is known, it is recommended that C_1 and C_2 be adjusted so as to best approach the data, since the details of separator design are often not known (i.e. proprietary).

In the topmost downcomer cells, the two-fluid model also requires that the pressure, p_d, be prescribed. Furthermore, comparison of Fig. 4.2-2 and Fig. 4.3-1 reveals an overlapping of domains in the upper downcomer, represented by the dotted area in Fig. 4.2-2. The zone of overlap is represented by the recirculation model. The cells in the two-fluid calculation within that zone are fictitious. This is in the sense that these cells constitute a boundary where there is flow into the solution domain of the two-fluid model. Since the recirculation model describes the region, the property values for the two-fluid calculation must be prescribed there at every time step to insure correct results downstream. This is a consequence of the donor cell logic utilized in the determination of fluid properties convected into each cell. Chapter 5 describes in detail the motivation for the overlap zone and the coupling procedure of the recalculation model with the two-fluid model in that region.

4.4 Primary Model Domain

The primary model domain includes primary coolant and tube metal, from inlet to outlet plenum inclusive. The model can be seen as having a twofold purpose. In one sense, that is to compute the temperatures of:

1. primary fluid
2. primary-side wall
3. secondary-side wall
4. intermediate wall

at each representative primary tube, in each secondary-side cell, in the tube bundle region. Also, to determine the

5. outlet plenum temperature.

In a different sense, the primary-side model must provide the required power level and distribution for the two-fluid model solution in the evaporator.

The U-tubes which bend within a given axial level are grouped together into one representative tube, which is called a 'tube bank.' There is one representative tube per mesh level in the U-bend section of the tube bundle region. This arrangement is illustrated in Fig. 4.4-1 and Fig. 4.2-2, where the regions marked, cell (J, I), are mutually correspondent. Thus, there are as many primary temperatures computed per secondary-side cell as there are tube banks in that cell. For the case of Fig. 4.4-1, there are three temperature groups in each cell in the straight tube region, where each temperature group includes the four temperatures previously listed.

The length of each representative tube (tube bank) is determined in such a way that its heat transfer area, when multiplied by the number of tubes it represents, yields the heat transfer area of the group. This calculation is dependent on steam generator geometry and on mesh layout as well. A Fortran routine to perform this calculation is given in Appendix D along with the assumptions required for its use.

The average primary mass flux is split among the representative tubes based on an equal frictional pressure drop criterion, for all tubes, between plena.

Metal and fluid properties are computed at each temperature and at system pressure.

It has been noticed (S2) that commonly used temperature sensors, namely resistance temperature detectors, can have non-negligible response times in comparison with time spans for

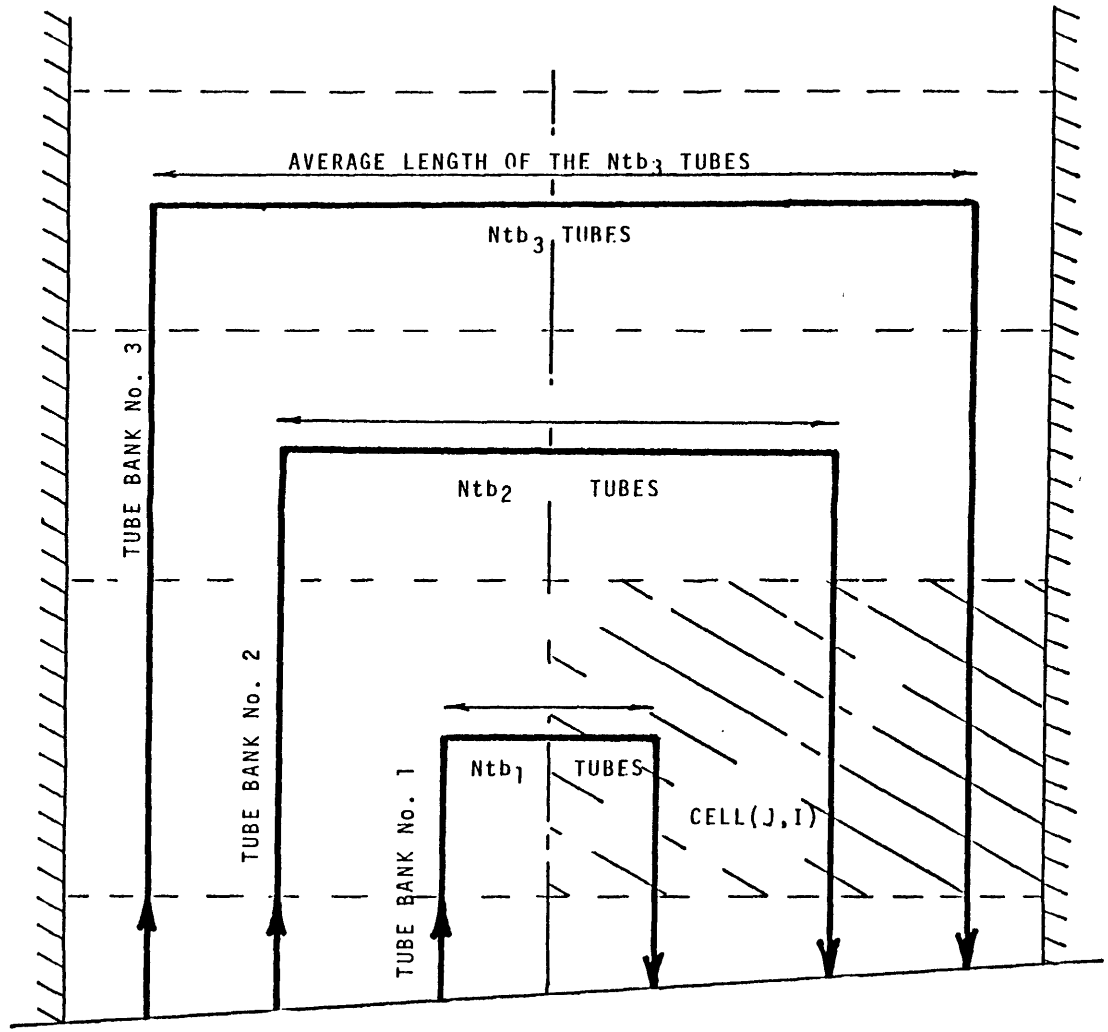


Figure 4.4-1. Primary model domain showing three tube banks.

transients of interest. Thus a sensor model using a first order lag equation is incorporated to adjust inlet and outlet temperatures for comparison with data.

4.5 System Boundary Conditions

System boundary conditions used in this work are listed in Table 4.5-1.

For steady-state calculations it is possible to utilize the primary average temperature in lieu of the inlet temperature. The advantage of using T_{avg} is that it can be related to reactor power by a primary system model, thereby creating the potential to eliminate an input. However, since the average temperature is defined:

$$T_{avg} = 0.5\ (T_{in} + T_{out}) \qquad (4.5\text{-}1)$$

and since T_{out} is a calculated quantity, prescribing T_{avg} involves an iterative procedure that is unwarranted vis a vis the currently envisioned applications of THERMIT-UTSG as a component benchmark model. If it should be desired to operate in conjunction with a system code, T_{avg} can be used instead of T_{in} with the understanding that the equation above becomes a convergence criterion, to be met by appropriately guessing T_{in} and subsequently computing T_{out}.

The feedwater flow rate in steady-state is determined by the relation:

Table 4.5-1. System boundary conditions for THERMIT-UTSG.

STEADY STATE		TRANSIENT
100% Power	Other Power Levels	
1. Primary Inlet Temperature	1. Primary Inlet Temperature	1. Primary Inlet Temperature
2. Average Primary Mass Flux	2. Average Primary Mass Flux	2. Average Primary Mass Flux
3. Primary System Pressure	3. Primary System Pressure	3. Primary System Pressure
4. Power Level	4. Power Level	4. - - -
5. Steam Dome Pressure	5. - - -	5. Steam Dome Pressure (Outlet Nozzle Steam Flow)
6. Downcomer Water Level	6. Downcomer Water Level	6. - - -
7. Feedwater Temperature	7. Feedwater Temperature	7. Feedwater Temperature
8. - - -	8. - - -	8. Feedwater Flow Rate
9. - - -	9. Fouling Coefficient	9. Fouling Coefficient

$$W_{fd} = W_s = \frac{Q_B}{(h_s - h_{fd})} \tag{4.5-2}$$

where

Q_B = Tube Bundle Power

W_s = Steam Flow Rate

W_{fd} = Feedwater Flow Rate

h_s = Steam Enthalpy

h_{fd} = Feedwater Enthalpy

Since in steady-state

h_s = Saturated Vapor Enthalpy (at the steam dome pressure)

all quantities on the right hand side of Eq. (4.5-2) can be calculated from the input given in Table 4.5-1. Hence, steam and feed flow need not be given in steady-state.

A steady-state analysis at a power level other than 100 percent requires the same seven items as input, with one difference: the steam dome pressure is calculated by using the full power fouling coefficient as input. To clarify why this should be the case, a simple analysis is presented. In steady-state,

$$Q_B = W_p \bar{C}_p (T_{in} - T_{out}) \quad (4.5\text{-}3)$$

where

W_p = Primary Flow Rate

T_{in} = Primary Inlet Temperature

T_{out} = Primary Outlet Temperature

$\bar{C}_p$ = Average Specific Heat Capacity of Primary Fluid

With W_p, Q_B, T_{in}, and the primary pressure given as indicated in Table 4.5-1, the outlet temperature is immediately determined from the above equation, i.e.:

$$T_{out} = T_{in} - \frac{Q_B}{W_p \bar{C}_p} \quad (4.5\text{-}4)$$

On the secondary-side, assuming that the heat transfer coefficients and thermal conductivities are constant and the temperature is uniformly at saturation, T_{sat}, it can be written (H4), (R6), (T1):

$$Q_B = UA\Delta T_{lm} \quad (4.5\text{-}5)$$

where,

U = Overall Heat Transfer Coefficient

A = Heat Transfer Surface

ΔT_{lm} = $(T_{in}-T_{out})/\ln[(T_{in}-T_{sat})/(T_{out}-T_{sat})]$.

Combining Eqs. (4.5-3), (4.5-4), and (4.5-5) it can be obtained that,

$$U = \frac{W_p\bar{C}_p}{A} \ln \left(\frac{T_{in} - T_{sat}}{T_{in} - \frac{Q_B}{W_p\bar{C}_p} - T_{sat}} \right) . \qquad (4.5\text{-}6)$$

The key to the rationale for the steady-state boundary conditions is in the relationship shown above between T_{sat} and the overall heat transfer coefficient. If the steam dome pressure is given, then T_{sat} is determined. This implies that the overall heat transfer coefficient's value must correspond to the value given in Eq. (4.5-6). In order for this to happen, it is necessary to adjust the overall heat transfer coefficient. In this work the adjustment is made through a factor introduced in the thermal conductivity of the outer region of the tube metal, as described in Chapter 6. This adjustment is physically justified by phenomena such as crud deposition, tube wall thinning, or any other factor which may contribute to uncertainty in that parameter. Once the full power calculation is made and the factor established, the same value is used at other powers. This implies that T_{sat} is no longer arbitrary: it is determined by Eq. (4.5-6), since all other quantities are known. Hence, the steam dome pressure ceases to be an input.

For transient analyses, the system boundary conditions are also given in Table 4.5-1. It is demonstrated in Chapter 5 that outlet steam flow and steam dome pressure are equivalent from closure standpoint. Nevertheless, the pressure condition is preferred in this work because as an independent variable in recirculation model state equations, an iteration process can be bypassed when it is given.

Chapter 5
RECIRCULATION MODEL

5.1 Motivation

The recirculation model and calculational procedure are motivated by the need to model certain characteristics of the steam dome and upper downcomer, down to the neighborhood of the water level, in order to analyze transients of the steam generator. These characteristics are time dependent and include: the recirculation into the downcomer of the liquid removed from the steam at the separators, its mixing with the feedwater, the interaction of liquid with steam in the dome, the heat transfer of steam with structure and the downcomer water level rate of change. Relevant parameters for the study of these processes are shown in Figs. 4.3-1 and 5.2-1.

In a different sense, the present model formulation is also motivated by the potential it offers to increase overall computing efficiency. Computer time savings are expected because the present model permits the restriction of the solution domain of the more time consuming three-dimensional, two-fluid calculation to the evaporator, riser, and downcomer regions of the steam generator. The model separately describes the steam dome and the upper downcomer. Hence, the recirculation model provides a descriptive and even didactic view of the dynamics of the steam dome, whose equations are solved decoupled from those of the two-fluid model. It is important to note, however, that no unwarranted simplifications are carried out. As in the two-fluid calculation, the recirculation model includes separate equations for the liquid and the vapor, with flashing and condensation allowed.

The apparent complexity of utilizing two approaches, one for the evaporator, riser, and downcomer (two-fluid model) and one for the steam dome and upper downcomer (recirculation model) is outweighed by a reduction in computing cost and a more descriptive insight into steam dome and upper downcomer macroscopic dynamics. The reduction is in comparison with the alternative of having the complete three-dimensional two-fluid model describe these regions. Use of this model also obviates the need for major changes in the two-fluid equations, which would otherwise be required. These points are to be further illustrated.

If the steam dome is included in the two-fluid domain, the steam separators are left inside this domain. In order to model this situation, a mass sink term is required in the conservation equations to account for the liquid removed by the separators. In the solution of these equations, serious numerical difficulties could be expected, due to the inherently destabilizing nature of source terms, not to mention the cost of the additional iterations which are certainly required in any iterative numerical scheme where source terms are included which vary rapidly over a large range.

A second inconvenience of including the steam dome in the two-fluid solution domain, would arise in the cell(s) representing the upper downcomer, where again a mass source term would be required to represent the liquid returning from the separators and the feedwater. Furthermore, an energy source term would also be needed to insure simultaneous conservation of mass and energy in the computational cell containing the water level. To clarify why this should be the case, consider the dotted region in Fig. 4.2-2. If this space were filled with steam in the two-fluid representation, as it is in the actual unit, mass conservation in the cell in question, shown in Fig. 5.1-1, requires in a steady state situation that:

$$w_d^+ = w_d^- .$$

Furthermore, energy conservation requires that:

$$w_d^+ h_d^+ = w_d^- h_d^- ,$$

a condition which cannot be satisfied if

$$h_d^+ = \text{vapor enthalpy}$$

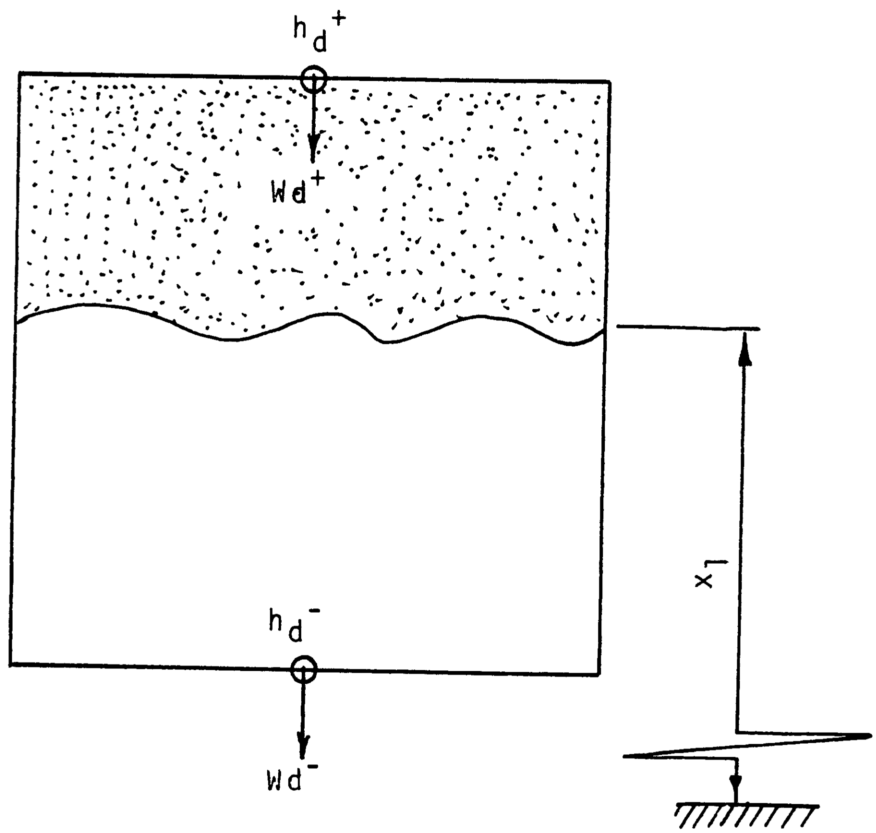


Figure 5.1-1.

Two-fluid model downcomer cell at the water level.

and

$h_d^- = $ liquid enthalpy

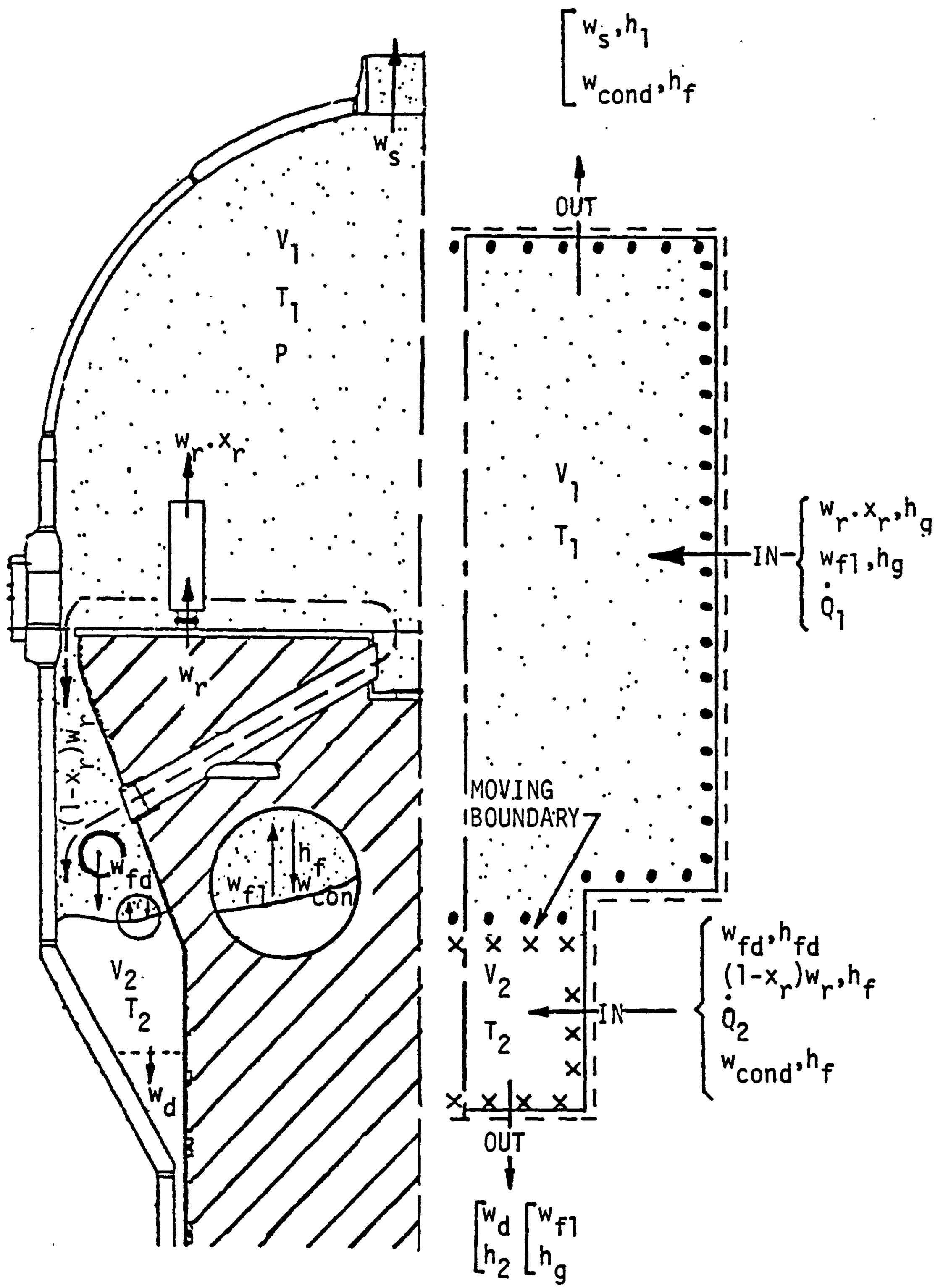
unless a source term equal to $w_d \cdot h_{fg}$ is added to the energy equation above equality. The introduction of such a term would significantly change the structure of the two-fluid model solution. What is done, then, is to fill the space between x_1 and x_{1ref}, represented by the dotted region in Fig. 4.2-2, with fictitious water. Special care must be taken in the determination of the pressure, p_r, to be prescribed at the top of this water and in prescribing its temperature in the two-fluid model as well. These considerations are part of the recirculation model and are analyzed in detail in Section 5.4.3.

To conclude, expansion of the two-fluid model to provide closure of the natural circulation loop is more complex and costly in computation time than to describe the process by a separate model. Furthermore, there is nothing to be gained from the more complex approach. As long as the flow in the downcomer below the liquid level is accurately described by the two-fluid model, it is fair to assume that any volume above this level is filled with steam at the conditions prevailing in the dome. Therefore, to include that region in the two-fluid model domain is redundant. Furthermore, because the velocity field in the steam dome is of no practical concern in the context of this work, the momemtum equations in that region can be dropped. Thus, a separate model for the steam dome saves considerable computing time by eliminating two-fluid model computational cells, while adding a set of equations which is smaller, but sufficiently descriptive. The solution of the new set can be uncoupled from the two-fluid solution, yielding a net gain in computing time, which increases substantially, as the two-fluid model cell number is increased.

5.2 Geometric Representation and Thermohydraulic Parameters

Figure 5.2-1 is a schematic representation of the upper downcomer and of the steam dome, showing the parameters relevant to the analysis. These quantities are:

1. feedwater flow rate, w_{fd}
2. feedwater enthalpy, h_{fd}
3. steam dome pressure, p
4. outlet nozzle steam flow rate, w_s
5. two-phase flow rate leaving the riser, w_r
6. quality of the two-phase mixture leaving the riser, x_r
7. downcomer flow rate, w_d
8. mass rate of vaporization, w_{fl}
9. mass rate of condensation, w_{cond}
10. heat input by conduction to steam (includes heat source/sink effect of structure and of water), $\dot{Q}_1$
11. heat input by conduction to liquid (includes heat source/sink effect of structure and of steam), $\dot{Q}_2$



• Vapor Control Volume, V_1 (variable)
× Liquid Control Volume, V_2 (variable)
— Total CV: V_1+V_2 (constant over each time step)

Figure 5.2-1.

Identification of variables and control volumes for recirculation model.

12. portion of the control volume filled with steam, V_1

13. portion of the control volume filled with liquid, V_2

The objective of this model is to calculate the time dependent response of:

1. steam flow, w_s
2. water control volume, V_2
3. temperature of the steam in the dome, T_1
4. temperature of the water in V_2, T_2

for changes in steam dome pressure, exit quality, feedwater temperature and the various flow rates viz.: riser, downcomer, and feedwater.

Parameters listed above, such as the steam flow and the water level in the downcomer (equivalent to V_2) are frequently measured. Their calculation is therefore indispensable to the assessment of the overall steam generator model, particularly for transients.

Furthermore, local boundary conditions for the two-fluid calculation depend on the four parameters listed. These conditions are:

1. temperature of liquid entering the two-fluid domain
2. pressure at the topmost cells in the downcomer
3. pressure at the topmost cells in the evaporator.

5.3 Description of the Model

5.3.1 Conservation Equations

For each of the control volumes shown in Fig. 5.2-1, V_1 and V_2, mass and energy conservation equations are written. The equations incorporate the following assumptions:

1. no steam carry-under or liquid carry-over
2. transit time of liquid in separators is negligible
3. perfect mixing of steam in V_1 and liquid in V_2.

The first assumption is made because separators are designed to minimize liquid carry-over. The quality of the steam leaving the primary separators is higher than 0.9975 (W1) for modern equipment operating in steady-state. In transient situations, carry-over is highly equipment-dependent and information in this area is limited. Analytical separator models such as that of Ref. (V1) could, in principle, be used to calculate liquid carry-over. However, such calculations are extremely time consuming and the geometrical information needed is usually proprietary. The computational effort required in assessing this term would not be commensurate with its importance. As for steam carry-under, there is not enough known about this quantity to justify its inclusion in the analysis. The study of steam separation is a field in itself. Publications in this area include Refs. (R5), (R7), (B4), (V1), (P3), and (C4). In this work, both carry-over and carry-under are assumed small in comparison with steam and recirculating flows. The same applies to liquid possibly held up in separator return paths or structure surfaces. This justifies the second assumption above. The third assumption is embodied in the choice of only one control volume for each phase. The motivation for using the point model is given in Section 5.1.

Based on these assumptions, and for the control volumes V_1 and V_2 shown in Fig. 5.2-1, the equations of mass conservation are:

$$\dot{M}_2 = w_{fd} + (1-x_r)w_r + w_{cond} - w_{fl} - w_d \qquad (5.3\text{-}1)$$

and

$$\dot{M}_1 = w_r x_r - w_s - w_{cond} + w_{fl} \qquad (5.3\text{-}2)$$

for the liquid and vapor respectively.

Reference (B5) presents the following general partial differential equation describing energy conservation:

$$\frac{\partial}{\partial t}(\rho h) = -(\nabla.(\rho h\vec{v})) - \nabla.\vec{q} - (\mathcal{T}: \nabla.v) + \frac{Dp}{Dt}\,. \qquad (5.3\text{-}3)$$

This equation will be integrated over each control volume to obtain the working form of the energy conservation equations.

The first step in the process is to neglect viscous dissipation, i.e.,

$$\tau : \nabla . \vec{v} \sim o \ . \qquad (5.3\text{-}3a)$$

By hypothesis, the pressure is uniform within both control volumes, hence,

$$\frac{Dp}{Dt} = \frac{dp}{dt} \qquad (5.3\text{-}3b)$$

Integration of the first term on the left hand side of Eq. (5.3-3) yields:

$$\int_{v_i} [\frac{\partial}{\partial t} \rho h] \partial v_i = \frac{d}{dt}(Mh)_i - (\rho h)_i \frac{dv_i}{dt} \qquad (5.3\text{-}3c)$$

The first term on the right hand side can be integrated as,

$$RHS_I = \int_{v_i} (\nabla . \rho h\vec{v}) dv_i = \int_{cs} \rho h\vec{v} . \, d\vec{A} \ ,$$

where $\vec{v}$ is the velocity in a fixed reference frame. To perform the surface integration, it is illustrative to break it up into the relative velocity of the fluid with respect to the boundary and the boundary velocity in the fixed referential. Hence,

$$RHS_I = \int_{cs} \rho h(\vec{v}_{rel} + \vec{v}_{bound}) \cdot d\vec{A} =$$

$$= - \sum_{inlets} wh + \sum_{outlets} wh + \int_{\substack{moving\\boundary}} \rho h \vec{v}_{bound} \cdot d\vec{A} =$$

$$= - \sum_{inlets} wh + \sum_{outlets} wh + \rho h \frac{dv_i}{dt} \qquad (5.3\text{-}3d)$$

Defining as positive the heat entering each control volume leads to:

$$\int_{v_i} (\nabla . \vec{q}) dv_i = -\dot{Q}_i \ . \qquad (5.3\text{-}3e)$$

And finally,

$$\int_{v_i} (\frac{dp}{dt}) dv_i = v_i \frac{dp}{dt}. \qquad (5.3\text{-}3f)$$

Combining the results of Eq. (5.3-3a) through Eq. (5.3-3f) into the framework provided by Eq. (5.3-3) results in:

$$\frac{d}{dt}(M_i h_i) - (\rho_i h_i)\frac{dv_i}{dt} =$$

$$= -\sum_{\text{outlets}} wh + \sum_{\text{inlets}} wh - (\rho_i h_i)\frac{dv_i}{dt} + \dot{Q}_i + v_i\frac{dp}{dt} \qquad (5.3\text{-}4)$$

Specializing this equation for the liquid in V_2 yields:

$$\frac{d}{dt}(M_2 h_2) = w_{fd}h_{fd} + w_{cond}h_f - w_d h_2 - w_{f1}h_g +$$

$$+ (1-x_r)w_r h_f + \dot{Q}_2 + (M_2/\rho_2)\dot{P} \quad . \qquad (5.3\text{-}5)$$

Similarly, for the vapor, Eq. (5.3-4) can be written in the form:

$$\frac{d}{dt}(M_1 h_1) = w_r x_r h_g + w_{f1}h_g - w_{cond}h_f + \dot{Q}_1 + (M_1/\rho_1)\dot{P} - w_s h_1 \qquad (5.3\text{-}6)$$

The thermodynamic property routine used in the two-fluid model has entries in pressure and temperature. In order to utilize the same routine, the state equations are written in the form:

$$h_2 = h_2(T_2, P) \qquad (5.3\text{-}7)$$

$$\rho_2 = \rho_2(T_2, P) \qquad (5.3\text{-}8)$$

$$h_1 = h_1(T_1, P) \qquad (5.3\text{-}9)$$

$$\rho_1 = \rho_1(T_1, P) \qquad (5.3\text{-}10)$$

An additional equation is obtained by imposing

$$\frac{d}{dt}\left(\frac{M_1}{\rho_1} + \frac{M_2}{\rho_2}\right) = 0 \qquad (5.3\text{-}11)$$

where (M_1/ρ_1) is the actual dome volume occupied by steam, and (M_2/ρ_2) is a liquid volume bounded above by the interface with the vapor and below by an imaginary horizontal plane. The axial location of this lower boundary, for the purpose of the equations derived in this section, is evidently arbitrary, since there is nothing fixing its position in the derivation just described.

The basic model equations are nine: (5.3-1), (5.3-2), (5.3-5) to (5.3-11), and the unknowns are thirteen: p (or w_s, if p is an unknown then w_s must be prescribed and vice versa), M_1, M_2, w_{cond}, w_{fl}, h_1, h_2, ρ_1, ρ_2, $\dot{Q}_2$, $\dot{Q}_1$, T_1, T_2.

5.3.2 Closure

Clearly, unless additional equations are specifed or simplfying assumptions are made, the system will remain undeterminate. In order to provide closure, both approaches will be adopted.

The first assumption for closure has to do with obtaining equations for the heat source/sink terms, $\dot{Q}_1$ and $\dot{Q}_2$. These are given by additional relations (to be discussed in the next section) which are independent of the unknowns. Consequently, setting these terms aside for the time being, the unknowns are reduced to eleven in number.

The second approach towards closure incorporates the assumption (M2) that neither phase can exist in non-stable form. This is to say the vapor can be saturated or superheated, but not subcooled, while the liquid can be saturated or subcooled, but not superheated. The assumption implies that flashing and condensation occur spontaneously within the bulk of liquid and vapor respectively. It also provides an extra set of constraints on four of the unknowns, viz.: w_{fl}, W_{cond}, T_1, T_2, as shown in Table 5.3-1.

Table 5.3-1. Model unknowns according to direction of pressure change, and initial conditions of steam and water in dome and upper downcomer, respectively.

CASE NUMBER	$\frac{\Delta p}{\Delta t}$	Initial Condition of		Value of				Variables Eliminated	Remaining Unknowns
		Vapor	Liquid	w_{fl}	w_{cond}	(T_1)	(T_2)		
1	>0	sat. or super.	sat. or subc.	0	0	?	?	$w_{fl} = 0$ $w_{cond} = 0$	$\mathring{p}$ or w_s, h_1, h_2, ρ_1, ρ_2, M_1, M_2, T_1, T_2
4	<0	sat.	sat.	?	?	(T_{sat})	(T_{sat})	$T_1 = T_{sat}$ $T_2 = T_{sat}$	$\mathring{p}$ or w_s, w_1, w_{cond}, ρ_1, ρ_2, M_1, M_2, h_1, h_2
2	<0	sat.	subc.	0	?	(T_{sat})	?	$w_{fl} = 0$ $T_1 = T_{sat}$	$\mathring{p}$ or w_s, w_{cond}, h_2, ρ_1, ρ_2, T_2, M_1, M_2, h_1
3	< 0	super.	sat.	?	0	?	(T_{sat})	$w_{cond} = 0$ $T_2 = T_{sat}$	$\mathring{p}$ or w_s, w_{fl}, h_1, ρ_1, ρ_2, T_1, M_1, M_2, h_2
5	< 0	super.	subc.	SAME AS CASE NUMBER 1					

To arrive at Table 5.3-1, consider Fig. 5.3-1 representing a temperature-entropy diagram. For case 4 in the table, for example, where $\Delta P/\Delta t < 0$, and the liquid and vapor are saturated (at the start of Δt), it is assumed that there will be flashing and condensation as shown in the figure. On the other hand, for case 1, where $\Delta P/\Delta t > 0$, there will be neither flashing nor condensation, since in a pressure rise, regardless of the initial state (only stable states allowed of course) the process towards equilibrium does not involve change of phase but only heat transfer. The constraints for the remaining cases shown in Table 5.3-1 follow analogously. These constraints are then used to reduce the number of unknowns to nine in each case, as indicated in the table.

The procedure is simple. Depending on the case number, two of the following four unknowns: W_{cond}, W_{fl}, T_1, T_2, are known a priori. For example, for case 1,

$$w_{cond} = w_{flash} = 0$$

and the set becomes seven by seven. Evidently, closure can always be obtained in this manner regardless of the case. For cases in which the pressue is fallowing (2, 3, and 5) flashing or condensation are disallowed, while physically these processes could indeed occur. For moderately small time steps,

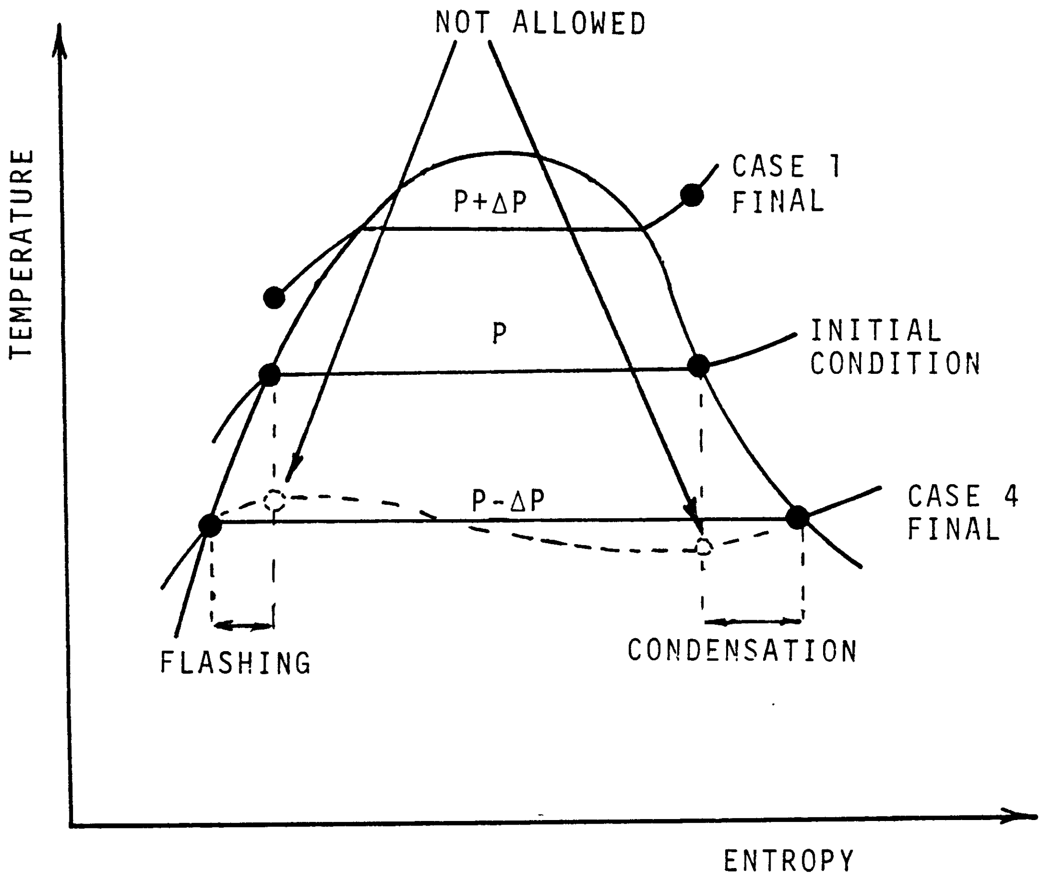


Figure 5.3-1

Illustration of the closure scheme.

this is not a problem because in these situations another case is initiated in the next time step, which will then account for the appropriate phase change mechanism.

It is illustrative to point out that the conditions at the beginning of each step are used to choose the case which will prevail within the step. A possible refinement would be to verify, after the equations are solved, whether the case choice was adequate. This has been found unnecessary because case changes do not occur with great frequency and an inaccurate choice made in one time step is corrected in the next.

5.3.3 The Heat Sink Terms

The purpose of this section is to present the procedure used to assess the quantities $\dot{Q}_1$ and $\dot{Q}_2$, the heat transfer rates into the steam and liquid volume respectively. It must be emphasized that the heat sink/source terms discussed in this section relate to the mechanism of conduction. Heat transfer by liquid-vapor interfacial flashing and/or condensation, is incorporated into the basic equations, as previously shown.

5.3.3.1 Heat Transfer Rate from the Vapor for Rising Pressure

It is expected in steam generator operational transients that, as the steam dome pressure rises, the vapor will superheat. Under these circumstances there are essentially two types of heat sinks available to reinstate equilibrium: structure and liquid. Therefore, the total heat transfer rate from the vapor control volume can be seen in terms of these two components, i.e.

$$\dot{Q}_1 = \dot{Q}_{1-s} + \dot{Q}_{1-w} \tag{5.3.12}$$

Each of the terms on the right-hand side will now be discussed in detail.

(a) Heat Transfer Rate to Structure

The first term on the right-hand side, $\dot{Q}_{1-s}$, is the heat transfer rate to structure. Structure includes the steam dome wall (shell), the separator deck, and separators themselves. Structure is modeled as several one-dimensional heat sinks (finite thickness slabs) with one wall at the saturation temperature at the prevailing dome pressure. The implicit assumption is that steam condenses on the walls under these circumstances which then become at T_{sat} themselves. In fact, the inner walls will be slightly below this temperature (J1); however the error introduced through this assumption has been found negligible in similar circumstances (G1).

The other wall is assumed insulated. Figures 5.3-2(a) and (b) illustrate the situation. For the shell, this is nearly the actual case. For the remaining structure, due to the uniformity of the pressure within the dome, surfaces exposed to steam on both sides can also be accurately modeled as above, as long as their half thickness is used instead of the actual thickness. This is because if both surfaces are at the same temperature, from symmetry considerations, it is obvious that half the slab with a no-heat flow condition at one wall will have a temperature distribution equal to that on each half of the actual slab.

An expression for the energy content of the wall at the end of N equal time intervals, Q_{1-s}^{N}, can be derived by integrating the temperature distribution shown in Fig. 5.3-2(a). The integration is performed assuming that the boundary temperature $T_b(t)$ is linear within each time interval Δt. The derivation, similar to the procedure suggested in (G1), is shown in Appendix B, and gives:

$$Q_{1-s}^{N} = \left\{ \sum_{n=1}^{N-1} T_b(n\Delta t) \sum_{k=0}^{\infty} \exp[-(N-n)a_k\Delta t][(\exp(a_k\Delta t)+\exp(-a_k\Delta t)-2)/a_k^2] \right.$$

$$\left. + T_b(N\Delta t) \sum_{k=0}^{\infty} [\exp(-a_k\Delta t)+a_k\Delta t-1] \right\} \frac{K}{\Delta t} \qquad (5.3\text{-}13)$$

where,

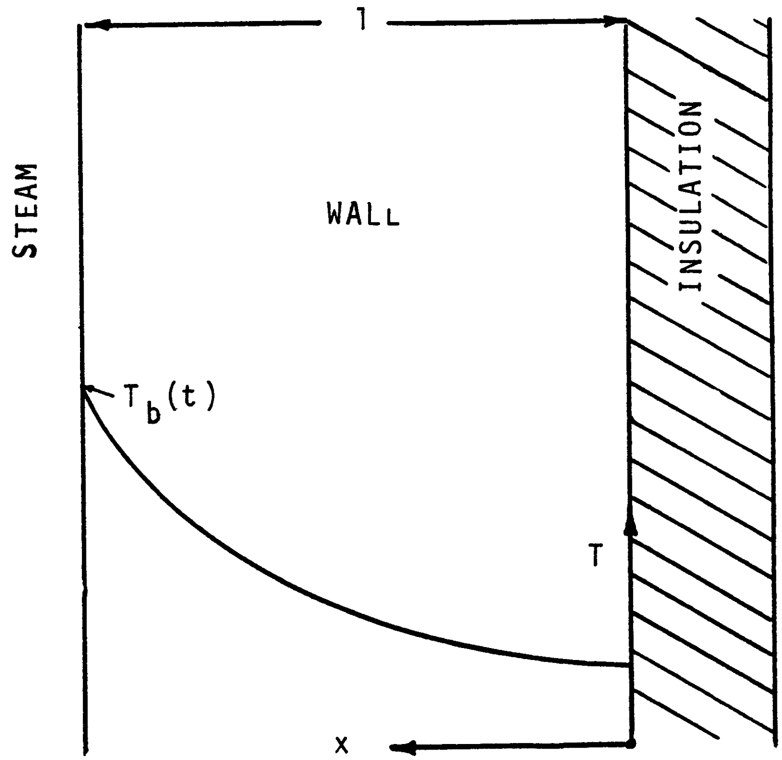
$$a_k = \left[\frac{2k+1}{2l}\pi\right]^2 \alpha$$

l = slab thickness

α = thermal diffusivity

K = thermal conductivity

The difference in heat content during any time interval divided by the size of the interval, Δt, yields the heat transfer rate to structure. For time $t_N = N.\Delta t$ this rate is:

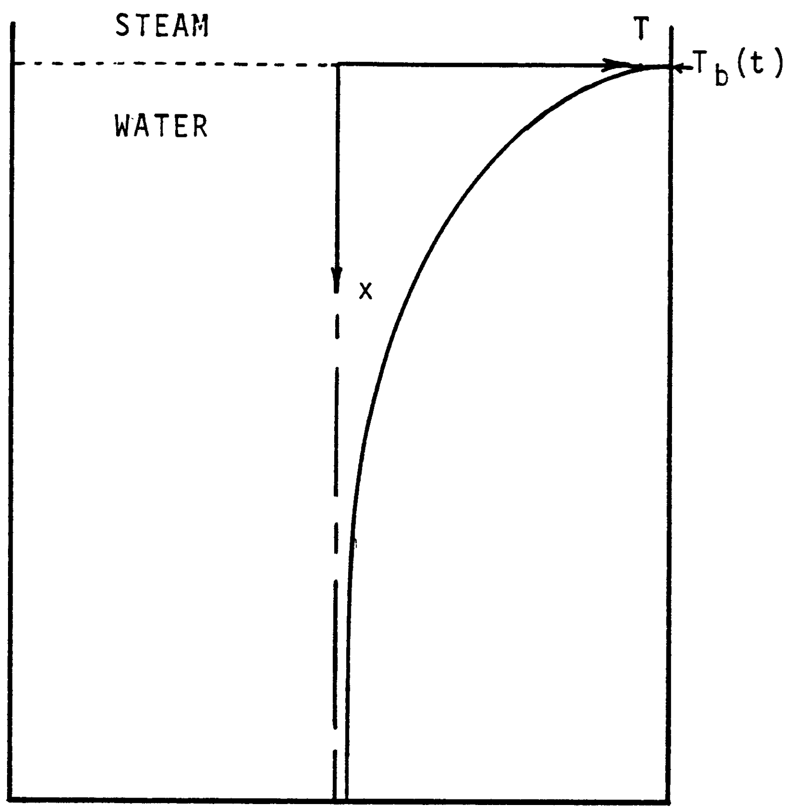


$$T(x,t) = \frac{2}{l} \sum_{k=0}^{\infty} \exp\left[-\alpha(2k+1)^2 \pi^2 t/4l^2\right] \cos \frac{(2k+1)\pi x}{2l} \cdot$$

$$\left[\frac{(2k+1)\pi\alpha(-1)^k}{4l} \int_0^t \exp[\alpha(2k+1)^2\pi^2 t'/4l^2] T_b(t')dt'\right]$$

Figure 5.3-2(a).

Wall temperature distribution with sinks initially at zero temperature, Eq. from Ref. (C2).



$$T(x,t) = \frac{x}{2\sqrt{\pi\alpha}}\int_0^t T_b(t')\,\frac{e^{\frac{-x^2}{4\alpha(t-t')}}}{(t-t')^{3/2}}\,dt'$$

Figure 5.3-2(b).

Water temperature distribution with sink taken as semi-infinite body initially at zero temperature, Eq. from Ref. (C2).

$$\dot{Q}_{1-s}(t_N) = [Q_{1-s}^{N} - Q_{1-s}^{N-1}] / \Delta t \ . \qquad (5.3\text{-}14)$$

Equation (5.3-14) determines most of the heat transfer rate from the steam control volume, viz. the first term on the right-hand side of Eq. (5.3-12). The term is easily calculated when the steam dome pressure is known as a function of time because

$$T_b(t) = T_{sat}(p(t)) \qquad (5.3\text{-}15)$$

is known from the equation for the saturation line. This permits the calculation of all the Q_{1-s}^{N} before actually analyzing a transient. When the transient is run, only Eq. (5.3-14) need be computed. To illustrate this point consider a pressure increase over a period of ten seconds. Arbitrarily the time step, Δt, for Eq. (5.3-13) can be set at 1 sec. Equation (5.3-15) is used to determine $T_b(N.\Delta t)$, N = 1,2...10 which corresponds to the saturation temperatue at every second on the second. Equation (5.3-13) is then used to calculate the Q_{1-s}^{N} for N = 1,2...10 and these values are stored. When analyzing the transient, let the current time be t=3.3s. Then the heat transfer rate is calculated from Eq. (5.3-14) with $\Delta t = 1$ and $N = 4$.

(b) Heat Transfer Rate to Liquid

For the heat transfer from the vapor to the liquid, the problem is modeled as that of flow of heat to a semi-infinite body, as shown in Fig. 5.3-2(b). This is justified by experimental evidence (G1) for pressurizer applications.

The expression for the energy content of the liquid in control volume V_2, at the end of N time intervals, Q^N_{1-w}, also derived in Appendix B, is:

$$Q^N_{1-w} = \frac{4}{3}\sqrt{\frac{\alpha}{\pi}\Delta t \quad \rho A C_p} \cdot$$

$$\cdot \left\{ \sum_{n=1}^{N-1} T_b(n\Delta t)[(N-n+1)^{3/2}+(N-n-1)^{3/2}-2(N-n)^{3/2}+T_b(N\Delta t) \right\} \tag{5.3-16}$$

where,

A	=	heat transfer area
C_p	=	specific heat capacity of liquid
α	=	thermal diffusivity of liquid
ρ	=	density of liquid
$T_b(n\Delta t)$	=	saturation temperature at steam dome pressure at $t = n.\Delta t$.

Analogously to the heat transfer rate to structure, $\dot{Q}_{1-s}$, the heat transfer rate to liquid, $\dot{Q}_{1-w}$, is calculated via:

$$\dot{Q}_{1-w}(t_N) = [Q_{1-w}^{N} - Q_{1-w}^{N-1}]/\Delta t \qquad (5.3\text{-}17)$$

where the stored energies, Q_{1-w}^{N}, are computed a priori as discussed in the previous section.

(c) <u>The Importance of Heat Sink Terms</u>

The inclusion of the heat sink term, $\dot{Q}_1$, into Eq. (5.3-6) demonstrates that, for operational transients, the importance of the heat transfer rates to structure, $\dot{Q}_{1-s}$, and to liquid, $\dot{Q}_{1-w}$, is small. This can be inferred from Fig. 5.3-3 and Fig. 5.3-4. These figures were generated using Eqs. (5.3-17) and (5.3-14) as described in conjuction with THERMIT-UTSG to compute the total heat loss to structure and to liquid for a turbine trip incident at the Arkansas Nuclear One power plant. Figure 5.3-3 shows that the total variation of the enthalpy in the steam dome volume viz., $\frac{d}{dt}(M_1 h_1)$ the term on the left hand side of Eq. (5.3-6), is considerably higher than the power loss to structure, $\dot{Q}_{1-s}$. Figure 5.3-4 shows that the heat loss to liquid, $\dot{Q}_{1-w}$ is an order of magnitude lower than the loss to structure.

These results, when viewed in the context of Eqs. (5.3-6) and (5.3-12) and while specific for a given transient, indicate nevertheless that heat sinks in the dome can only be counted on

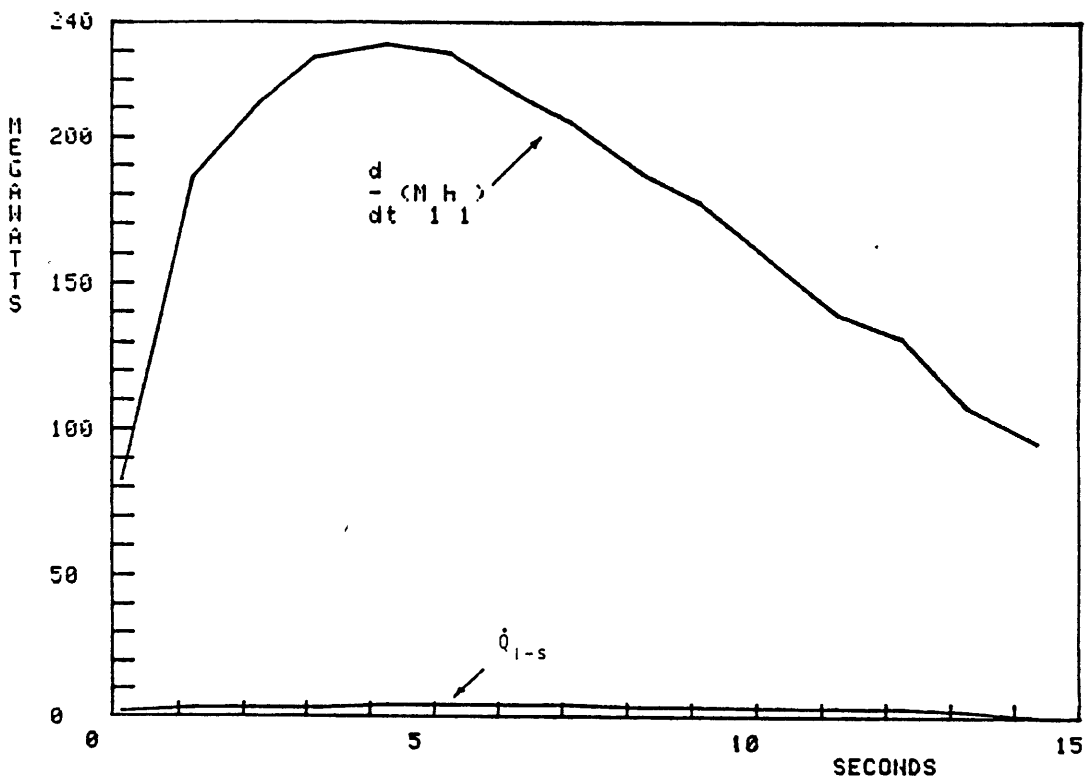


Figure 5.3-3.

Comparison of power loss to structure by conduction
with variation of steam dome enthalpy during a turbine trip test.

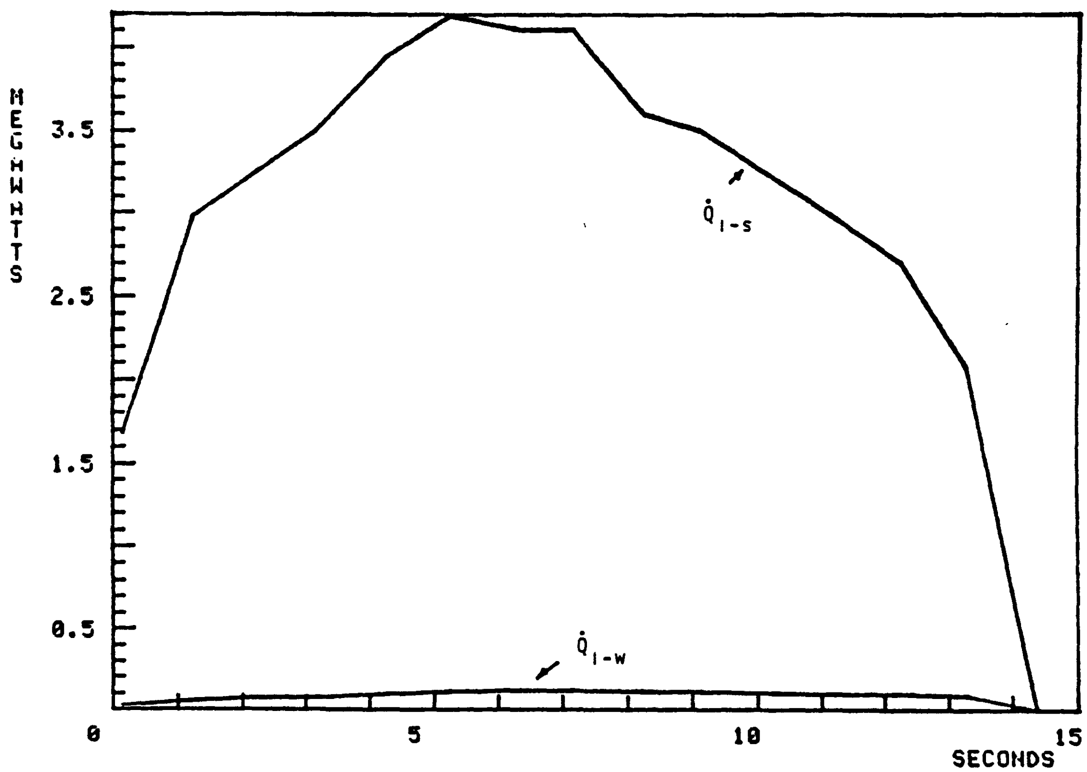


Figure 5.3-4.

Comparison of power loss to structure with power loss to liquid, both by conduction during a turbine trip test.

to remove a small fraction of steam superheat. Based on these results, $\dot{Q}_1$ in Eq. (5.3-6) could be neglected with reasonable confidence. Nevertheless, for small steam generators (model units) where the sink surface area to dome volume is much higher, or for a situation in which the unit becomes bottled up, the terms could become significant.

5.3.3.2 Heat Transfer Rate to the Vapor for Falling Pressure

Transfer of heat out of the vapor was discussed in the previous section. The situation here is considerably different since the mechanism of heat transfer between steam and wall through condensation will not be available. Heat will enter the vapor through sensible heat transfer only. In view of this relatively poor mechanism of heat transfer, the effect is neglected.

5.3.3.3 Heat Transfer to the Liquid for Rising Pressure

By analogy with Eq. (5.3-12), it can be written:

$$\dot{Q}_2 = \dot{Q}_{2-s} + \dot{Q}_{2-v} \tag{5.3-18}$$

where the subscript s refers again to structure and v to vapor.

As indicated in the previous section for the vapor, heat transfer from structure into the liquid in this case will take place by sensible heat transfer. For the same reasons described in Section 5.3.3.2, this term is neglected. Thus,

$$\dot{Q}_{2-s} = 0 .$$

The heat flow from the steam into the water is, of course

$$\dot{Q}_{2-v} = - \dot{Q}_{1-w}$$

with $\dot{Q}_{1-w}$ given by Eq. (5.3-17). And, therefore,

$$\dot{Q}_2 = - \dot{Q}_{1-w} .$$

5.3.3.4 Heat Transfer from the Liquid for Falling Pressure

In this case, some local boiling of the liquid at the walls could take place. However, the ratio of volume to surface area of contact between the liquid control volume V_2 and the wall is at least one order of magnitude less than that for the vapor. Therefore, even though this term could be calculated by utilizing Eq. (5.3-17) for the liquid, the results of Fig. 5.3-3 demonstrate it would be an excessive refinement to do so.

5.3.4 Solution of Model Equations

According to the closure precedure described in Section 5.3.2, each of the five cases listed in Table 5.3-1 will correspond to a different set of nine first order non linear differential equations. The set involves the nine unknowns given in that table for each case. In each of the five cases, a variable number of equations in the set are state equations. Because these functions are transcendental, an iterative scheme is used to solve the system.

This section presents the system of equations and the iterative solution procedure for each of the five cases as well as the input and control volume selection criteria. The system of equations is derived in Appendix A.

Once the equations are solved, the converged results are used to update steam dome and upper downcomer parameters, as well as to compute local boundary conditions, for advancing the solution in the evaporator, riser, and lower downcomer.

5.3.4.1 Input

The required input for the solution method is summarized in Table 5.3-2. It is of three origins:

(a) Two-Fluid Model

(b) Calculated

(c) External

Table 5.3-2. Summary of input parameters for solution to recirculation model equations.

ORIGIN	INPUT PARAMETERS
TWO-FLUID MODEL	w_d^n , W_r^n , x_r^n , Δt^{n+1}
CALCULATED	$\dot{Q}_1$, $\dot{Q}_2$ (Section 5.3.3) $h_{fd} = h_{fd}(p_{fd}^n, T_{fd}^{n+1})$ $p_{fd}^n = p^n + \rho_2^n.g.max((x_1-x_{fd}^n),0)$
EXTERNAL	Steady-State: T_{fd} , Q_B , x_1 , p
	Transient: T_{fd}^{n+1} , w_{fd}^{n+1} , W_s^{n+1} or p^{n+1}

The input needed from the two-fluid calculation includes: the flow rate in the downcomer, w_d; the flow rate of steam leaving the separators, $w_r x_r$; and the recirculating flow, $(1-x_r)w_r$. The calculated input includes: the heat sink terms, whose computation is described in Section 5.3.3, and the feedwater enthalpy. The latter is calculated from the state equation as a function of the prescribed temperature, T_{fd}, and a calculated pressure, p_{fd}. The calculated pressure is the steam dome pressure plus the hydrostatic head between the water level, x_1, and the feedwater distribution ring level, x_{fd}, as shown in Fig. 4.2-2.

The external input depends on the nature of the analysis: transient or steady-state. Input parameters and selection logic are presented in Fig. 5.3-5.

In transient analysis, feedwater temperature, T_{fd}, and flow rate, W_{fd}, must be prescribed. An additional input is needed in the form of either steam flow, W_s, or steam dome pressure, p. The two quantities are equivalent for transient analyses from the closure standpoint, but the pressure condition is preferred because, as an independent variable in the state equations, fewer iterations are required when it is known.

In a steady-state full power calculation the bundle power level, Q_B, is given along with the steam dome pressure, p; the feedwater temperature, T_{fd}; and the water level, x_1.

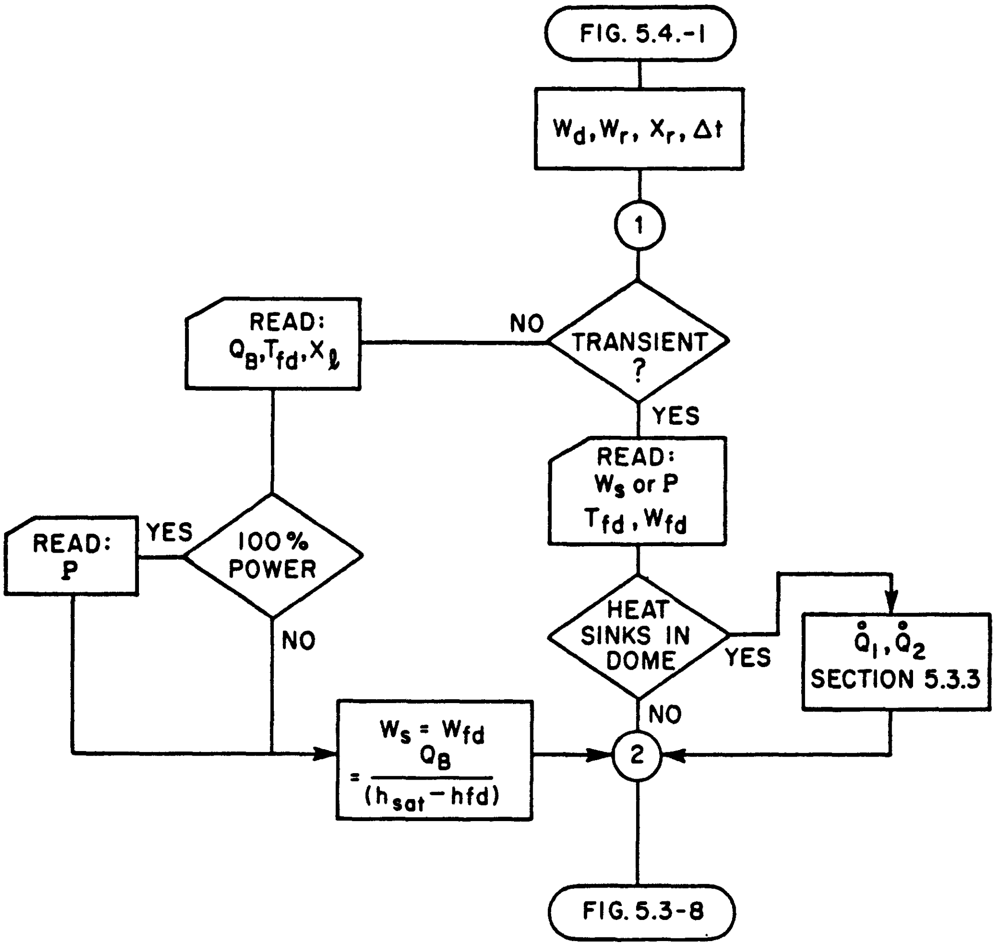


Figure 5.3-5.

Input selection logic.

This is equivalent to supplying the feed flow because in steady-state,

$$Q_B = w_s(h_s - h_{fd}),$$

and

$$w_{fd} = w_s$$

where

w_s = outlet steam flow rate

h_s = enthalpy of outlet steam flow (saturated)

h_{fd} = enthalpy of feed flow

A steady-state calculation at a power level other than 100 percent differs from the full power calculation in that the steam dome pressure is a calculated quantity rather than an input. This is discussed in detail in Section 4.5 and in Section 6.4.

5.3.4.2 Selection of the Control Volumes

The vapor control volume, V_1, in Fig. 5.2-1 is set at each time step equal to the volume occupied by the steam. This includes the free dome volume as well as that of the main steam line.

The liquid control volume, V_2, in Fig. 5.2-1 is treated in two ways depending on the relative position of the water level with respect to the feedwater ring.

Normally the level is above the ring. In this case, temperature stratification can occur in the downcomer: the water above the ring is nearer saturation, while that beneath the ring is usually subcooled. Upward mixing is unlikely primarily because the feedwater is sprayed downward and also because its temperature is lower than the average downcomer temperature.

This situation is illustrated in Fig. 5.3-6(a) where the small volume, V_{2b}, is added below V_2. The lower volume, V_{2b}, is chosen fixed in size and including the feedwater ring. Thus, w_{dl} represents the flow rate at the interface for this condition to be satisfied. For this choice, the mass balance over V_{2b} can be written:

$$\rho_{2b} \frac{dV_{2b}}{dt} + V_{2b} \frac{d\rho_{2b}}{dt} = (1-mfd).w_{fd} + w_{dl} - w_d \ .$$

where by hypothesis:

$$\frac{dV_{2b}}{dt} = 0$$

and (mfd) represents a mixing parameter for the feedwater. It is zero for no upward mixing and one for complete mixing of the feedwater in the upper volume.

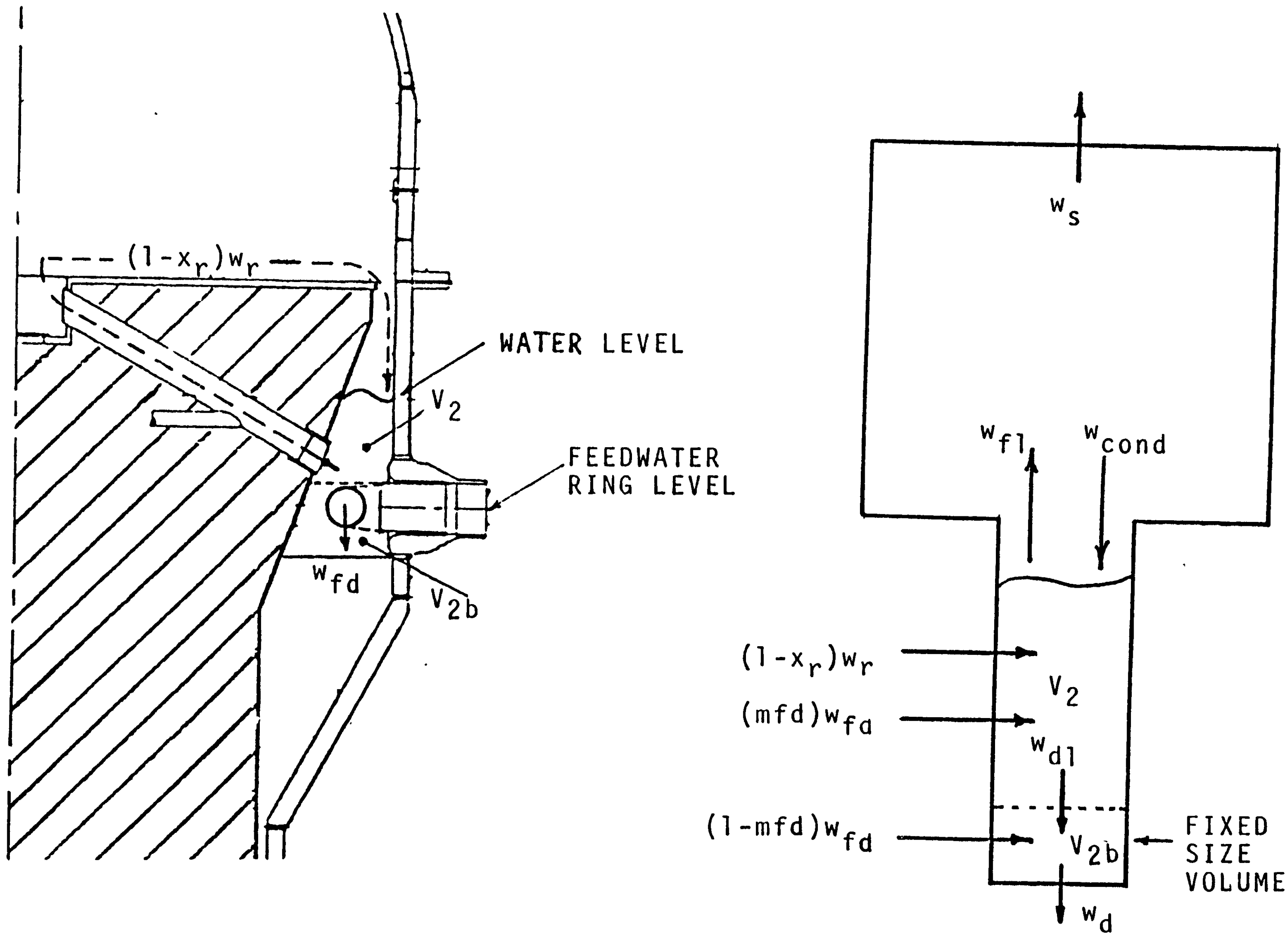


Figure 5.3-6(a).

Illustration of upper downcomer temperature stratification analysis.
Water level is above feed ring.

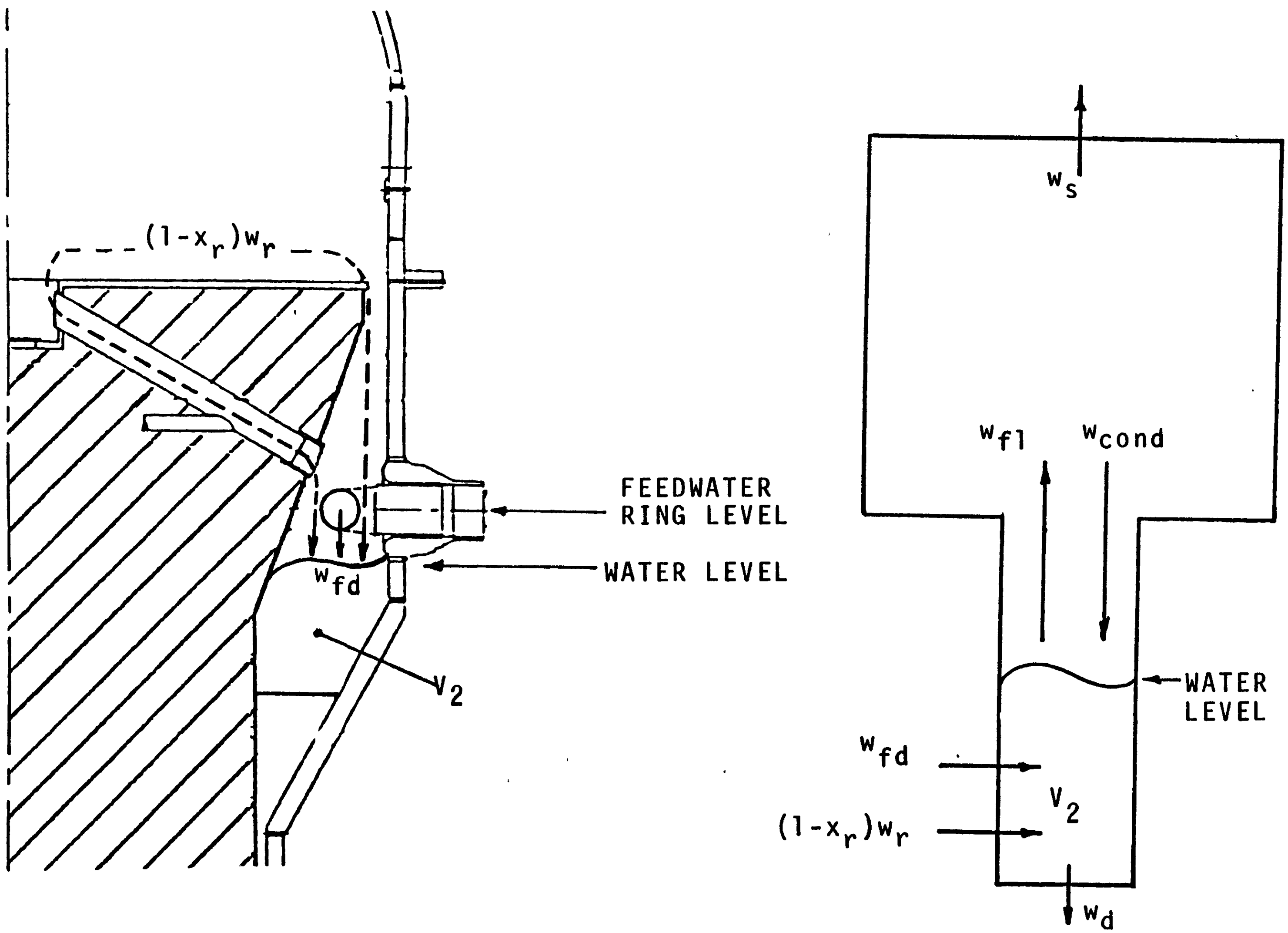


Figure 5.3-6(b).

Illustration of upper downcomer temperature stratification analysis. Water level is below feed ring.

With w_{2b} as a small volume, the mass balance gives:

$$w_{d1} = w_d - (1\text{-mfd})\; w_{fd} \;.$$

With 2_{d1} identified as above, it is possible to solve the recirculation model equations, as previously derived for the volume V_2, defined by the water level and the feedwater ring level, with a fictitious feed flow set at (mfd.w_{fd}), as is evident from comparing Figs. 5.3-6(a) and 5.3-6(b).

To obtain the enthalpy in the lower volume consider the energy conservation equation written for V_{2b},

$$\frac{d}{dt}\left[h_{2b}M_{2b}\right] = (1\text{-mfd})\; w_{fd}h_{fd} + w_{d1}h_2 - w_d h_{2b} + V_{2b}\dot{p} \;,$$

which is analogous to that written for V_2 in the previous section.

Having in mind that:

$$\frac{d}{dt}(Mh) = \dot{M}h + M\dot{h} = \dot{\rho}Vh + \rho\dot{V}h + \rho V\dot{h}$$

and that V_{2b} is constant and can be chosen as arbitrarily small, then:

$$\frac{d}{dt}(M_{2b}h_{2b}) \cong 0$$

and the energy balance can be solved for h_{2b} yielding:

$$h_{2b} = \frac{(1-mfd)W_{fd}h_{fd} + W_{dl}h_2}{W_d}$$

or alternatively:

$$h_{2b} = \frac{h_2 - (1-mfd)\, W_{fd}\, (h_2-h_{fd})}{W_d} \, .$$

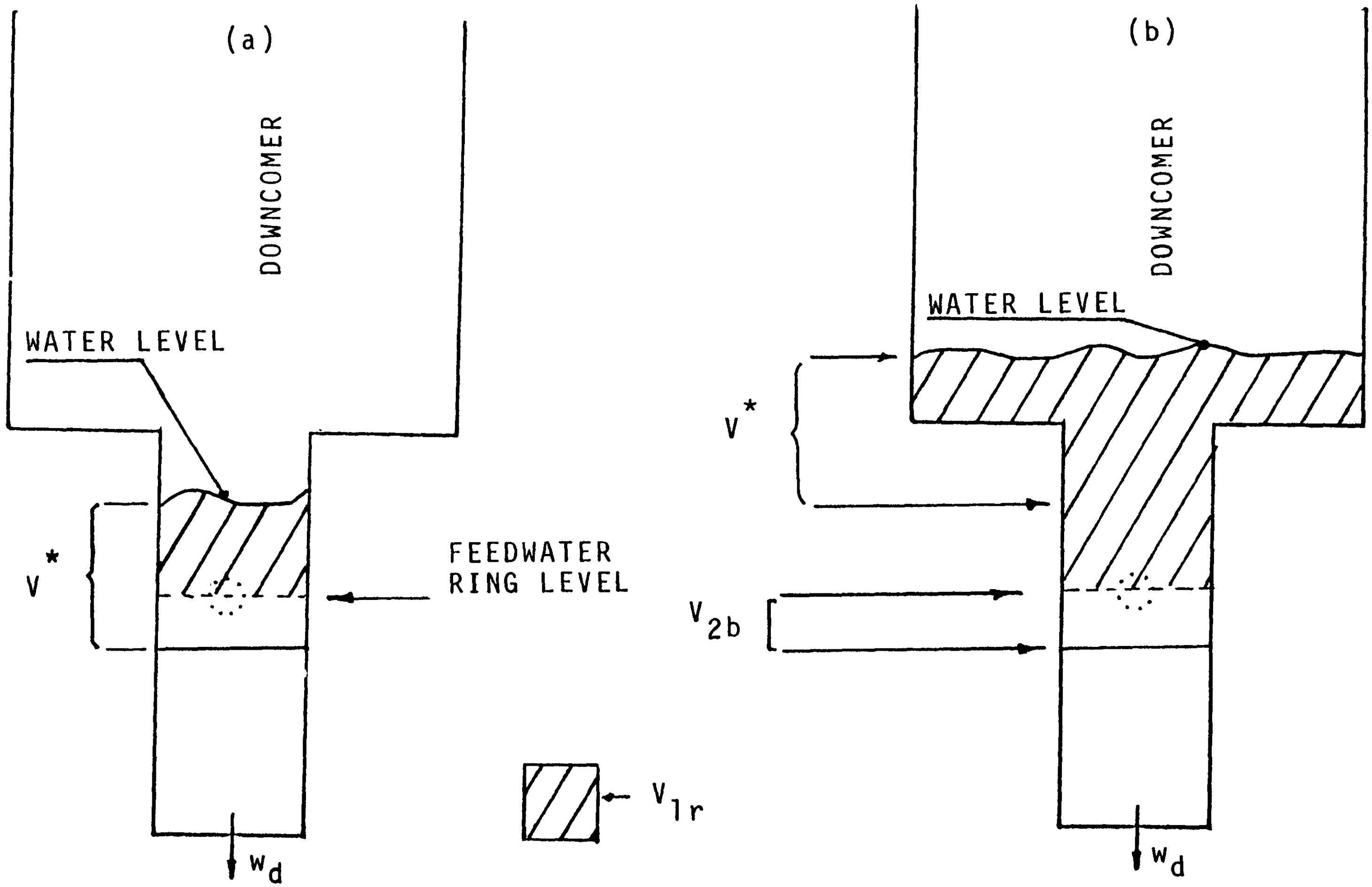
Clearly, if mfd = 1, the feedwater is assumed to mix into V_2, there is no stratification in the liquid control volume and $h_{2b} = h_2$ above, i.e. the problem automatically reduces itself to the appropriate case, as shown in Fig. 5.3-6(b).

Figures 5.3-7(a) and (b) illustrate how the liquid control volume is selected with respect to the relative position of ring and water level. The quantity V^* is a fixed volume arbitrarily chosen as 25 percent of the downcomer liquid volume in steady state. Results are insensitive to values between 10 and 30 percent. The residual volume between the water level and the feedwater ring is labeled V_{1r}. When

$$V_{1r} > V^*$$

as shown in Fig. 5.3-7(b), the liquid control volume is set at:

$$V_2 = V_{1r}$$



Figures 5.3-7 (a) and (b).

Liquid control volume, V_2, selection criterion.

and the feedwater distribution parameter is set at:

$$mfd = 0 ,$$

which means that all the feed flow is mixed into V_{2b}. Conversely, when

$$V_{1r} \leq V^*$$

as shown in Fig. 5.3-7(a), the liquid control volume is set at:

$$V_2 = V^* ,$$

and the feedwater distribution parameter becomes:

$$mfd = 1 ,$$

which means that all the feedwater is mixed into V_2 and the analysis of volume V_{2b} can be dropped in this case.

Obviously, if the level is below the ring, $V_{1r} < V^*$ and the appropriate conditions are automatically applied by use of the logic above.

Figure 5.3-8 summarizes the logic for choosing V_2 and dealing with the stratification issues just described. Basically, the parameter mfd specifies whether stratification

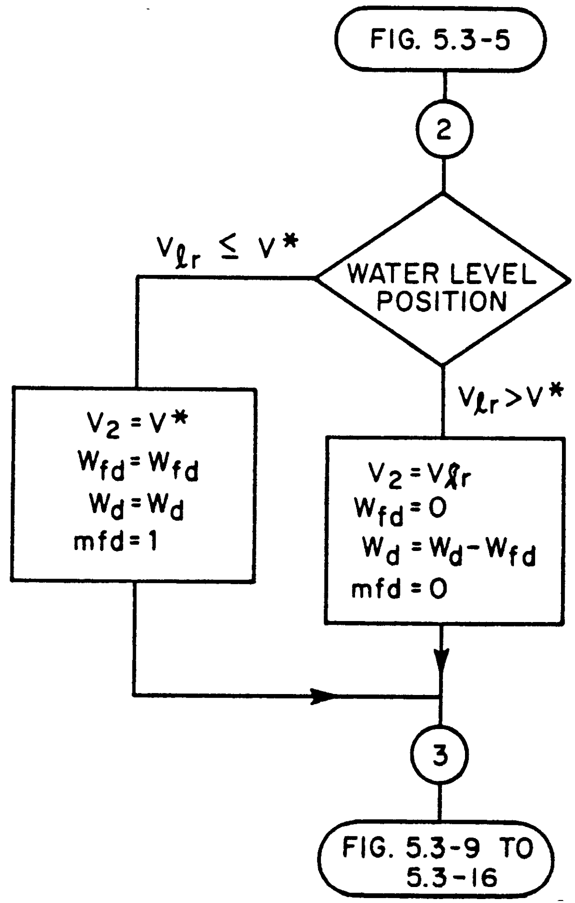


Figure 5.3-8.

Logic for selection of liquid control volume.

within the liquid volume is to be considered. The appropriate flows in and out of V_2 are obtained adjusting w_{fd} and w_d via $\dot{m}fd$. The model equations are then solved as shown in the next section. The liquid temperatures corresponding to h_{2b} calculated above for the appropriate values of mfd are set in the downcomer, as described in Section 5.4.

5.3.4.3 Case 1 and Case 5

The condition for Case 1 is rising pressure and the condition for Case 5 is superheated vapor and subcooled liquid. When any of these two situations prevails in the steam dome and upper downcomer there is no flashing of liquid into steam nor vapor condensation. Thus, by applying the constraints:

$$w_{f1} = w_{cond} = 0$$

in Eqs. (5.3-1), (5.3-2), (5.3-5), and (5.3-6) it is possible, as shown in Appendix A, to obtain the following set of equations:

$$\dot{p} = (\frac{\dot{M}_1}{\rho_1} + \frac{\dot{M}_2}{\rho_2} - p_{num1})/p_{den1} \qquad (5.3\text{-}19a)$$

$$w_s = w_r x_r - \rho_1(p_{den1} \cdot \dot{p} + p_{num1} - \frac{\dot{M}_2}{\rho_2}) \quad (5.3\text{-}19b)$$

$$\dot{M}_1 = w_r x_r - w_s \quad (5.3\text{-}20)$$

$$\dot{h}_1 = [w_r x_r(h_g - h_1) + \dot{Q}_1 + (\frac{M_1}{\rho_1})\ \dot{p}]/M_1 \quad (5.3\text{-}21)$$

$$\dot{T}_1 = [\dot{h}_1 - (\frac{\partial h_1}{\partial p})_T \dot{p}]/(\frac{\partial h_1}{\partial T})_p \quad (5.3\text{-}22)$$

$$\dot{T}_2 = [\dot{h}_2 - (\frac{\partial h_2}{\partial p})_T \dot{p}]/(\frac{\partial h_2}{\partial T})_p \quad (5.3\text{-}23)$$

$$\dot{M}_2 = (1-x_r)w_r + w_{fd} - w_d \quad (5.3\text{-}24)$$

$$\dot{h}_2 = [(1-x_r)w_r(h_f - h_2) + w_d(h_{fd} - h_2) + \dot{Q}_2 + (\frac{M_2}{\rho_2})\dot{p}]/M_2 \quad (5.3\text{-}25)$$

$$\dot{Q}_2 = - \dot{Q}_{1-w} \quad (5.3\text{-}26)$$

$$\dot{Q}_1 = \dot{Q}_{1-s} + \dot{Q}_{1-w} \quad (5.3\text{-}27)$$

where,

$$p_{num1} = \left(\frac{1}{\rho_1}\right)\left[\frac{\left(\frac{\partial\rho_1}{\partial T}\right)_p}{\left(\frac{\partial h_1}{\partial T}\right)_p}\right]\left[w_r x_r (h_g - h_1) + \dot{Q}_1\right] -$$

$$- \left(\frac{1}{\rho_2}\right)^2\left[\frac{\left(\frac{\partial\rho_2}{\partial T}\right)_p}{\left(\frac{\partial h_2}{\partial T}\right)_p}\right]\left[(1-x_r)w_r(h_f-h_2) + w_{fd}(h_{fd}-h_2) + \dot{Q}_2\right] \tag{5.3-28}$$

$$p_{den1} = \frac{M_1}{\rho_1}\left(\frac{1}{\rho_1}\right)^2\left(\frac{\partial\rho_1}{\partial T}\right)_p \Big/ \left(\frac{\partial h_1}{\partial T}\right)_p +$$

$$+ M_1\left(\frac{1}{\rho_1}\right)^2\left[\left(\frac{\partial\rho_1}{\partial p}\right)_T - \left(\frac{\partial\rho_1}{\partial T}\right)_p \cdot \left(\frac{\partial h_1}{\partial p}\right)_T \Big/ \left(\frac{\partial h_1}{\partial T}\right)_p\right] +$$

$$+ \left(\frac{M_2}{\rho_2}\right)\left(\frac{1}{\rho_2}\right)^2\left(\frac{\partial\rho_2}{\partial T}\right)_p \Big/ \left(\frac{\partial h_2}{\partial T}\right)_p \tag{5.3-29}$$

The procedure for calculating the heat sink terms $\dot{Q}_{1-w}$ and $\dot{Q}_{1-s}$ can be decoupled from this set of equations and calculated as shown in Section 5.3.3.

The set of Eqs. (5.3-19) to (5.3-21) is solved iteratively following the procedure illustrated by Fig. 5.3-9. The control variable, idome, of Fig. 5.3-9 is a user-specified parameter

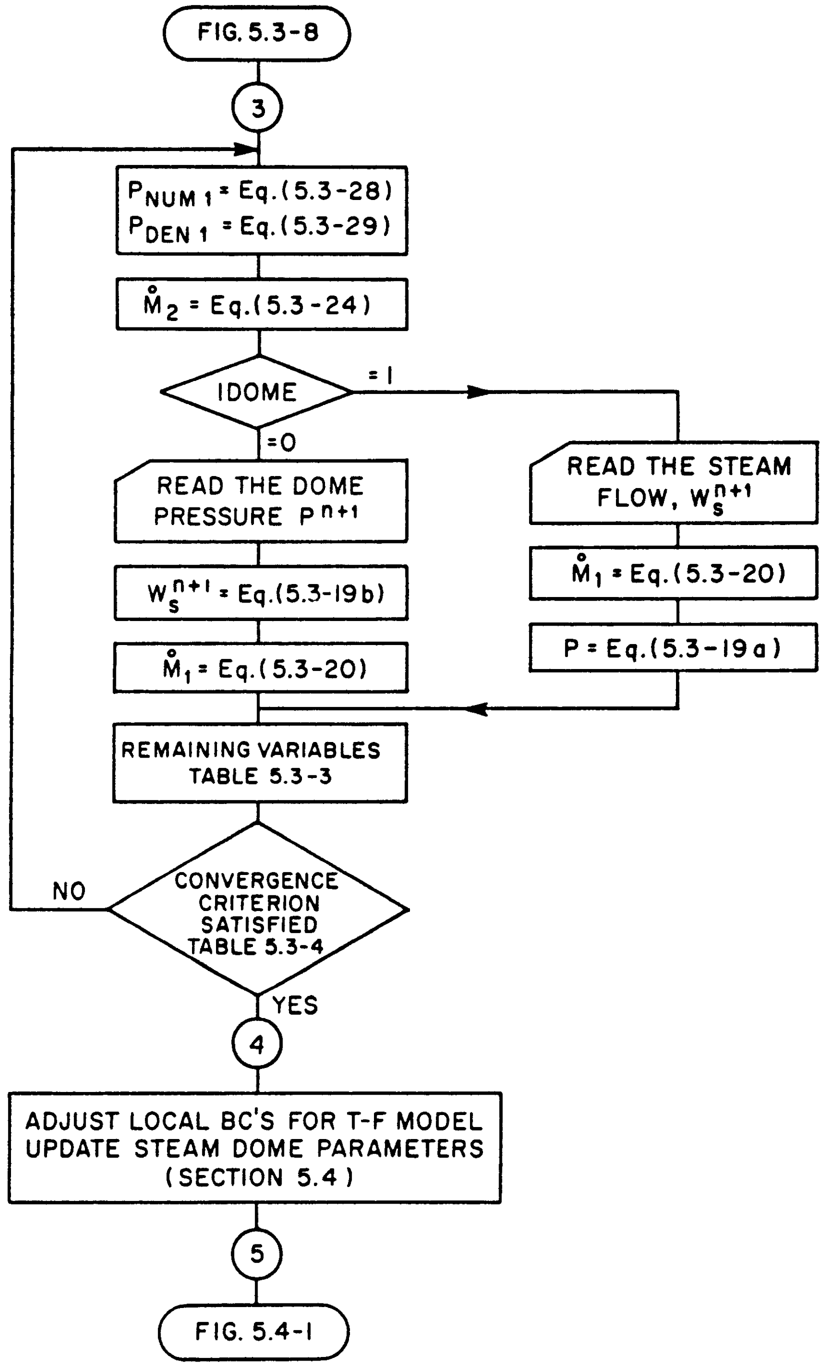


Figure 5.3-9.

Solution of recirculation model
Case 1 equations for transient analysis.

which signals whether steam flow or pressure is to be calculated.

The iterative procedure is straightforward. The key to the solution is in choosing the most favorable order for updating the variables. When the idome = 0 path is taken, convergence is usually achieved in two to three iterations.

Although conditions prevailing in Case 1 and Case 5 can exist only during transients, the steady state solution procedure is also included for these conditions and shown in Fig. 5.3-10. This is because in THERMIT-UTSG, the steady state is arrived at through a transient. The possibility of Case 1 or Case 5 being entered during a quest for steady state cannot be precluded.

5.3.4.4 Case 2

In Case 2 the vapor is saturated while the liquid is subcooled. These conditions lead to the constraints,

$$w_{f1} = 0$$

$$T_1 = Tsat$$

which, inserted into Eqs. (5.3-1), (5.3-2), and (5.3-5) to (5.3-11), as shown in Appendix A, results in the following set of equations:

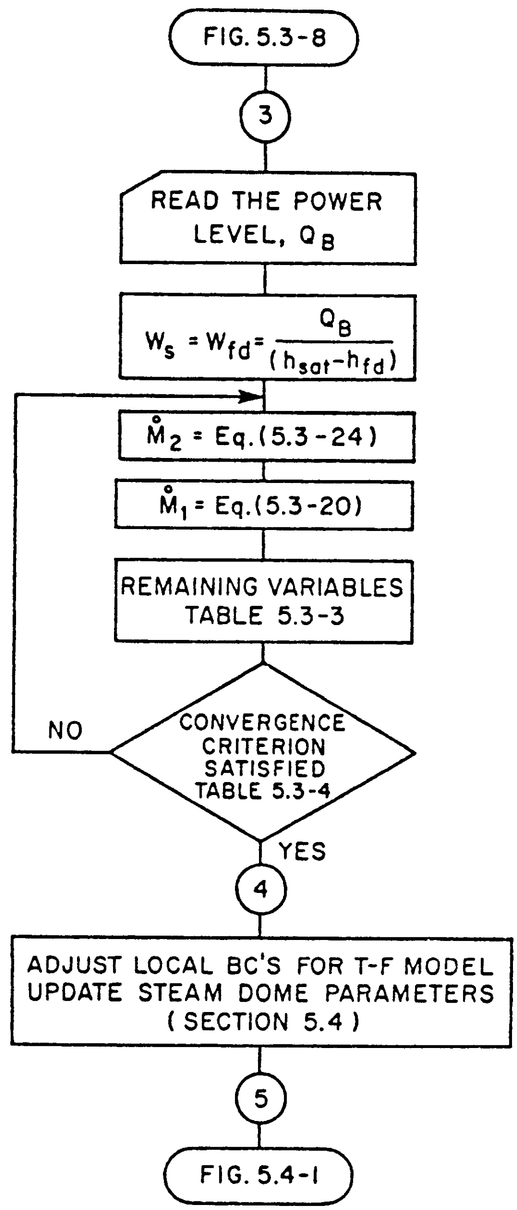


Figure 5.3-10.

Solution of recirculation model
Case 1 equations for steady state analysis.

REMAINING VARIABLES
$M_2^{n+1} = M_2^n + \dot{M}_2 \Delta t$
$M_1^{n+1} = M_1^n + \dot{M}_1 \Delta t$
$p^{n+1} = p^n + \dot{p} \Delta t$
$\dot{h}_1$ = Eq. (5.3-21)
$\dot{h}_2$ = Eq. (5.3-25)
$\dot{T}_2$ = Eq. (5.3-23)
$\dot{T}_1$ = Eq. (5.3-22)
$T_1^{n+1} = T_1^n + \dot{T}_1 \Delta t$
$T_2^{n+1} = T_2^n + \dot{T}_2 \Delta t$
$h_2^{n+1} = h_2(p_1, T_2)$
$h_1^{n+1} = h_1(p_1, T_1)$
$h_1^* = h_1^n + \dot{h}_1 \Delta t$
$h_2^* = h_2^n + \dot{h}_2 \Delta t$
$\rho_2^{n+1} = \rho_2(p_1, T_2)$
$\rho_1^{n+1} = \rho_1(p_1, T_1)$
$V_2^{n+1} = M_2^{n+1} / \rho_2^{n+1}$
$V_1^{n+1} = M_1^{n+1} / \rho_1^{n+1}$

Table 5.3.3.

Case 1 variables outstanding in Fig. 5.3-9.

Table 5.3-4. Case 1 iterative solution convergence criterion.

CONVERGENCE CRITERION
$(V_1+V_2)/V_T - 1 \leq \varepsilon_V$
$h_1^* - h_1(p_1,T_1) \leq \varepsilon_h$
$h_2^* - h_2(p_1,T_2) \leq \varepsilon_h$

$$\dot{p} = [(\frac{1}{\rho_g})(w_r x_r - w_s) + p_{num2}]/p_{den2} \tag{5.3-30a}$$

$$w_s = w_r x_r + \rho_g(p_{num2} - p_{den2} \cdot \dot{p}) \tag{5.3-30b}$$

$$\dot{M}_1 = w_r x_r - w_s - w_{cond} \tag{5.3-31}$$

$$h_1 = h_g \tag{5.3-32}$$

$$T_1 = T_{sat} \tag{5.3-33}$$

$$\dot{M}_2 = (1 - x_r)w_r + W_{fd} + w_{cond} - w_d \tag{5.3-34}$$

$$\dot{h}_2 = [(1-x_r)w_r(h_f-h_2) + w_{fd}(h_{fd}-h_2) + \frac{M_2}{\rho_2}\dot{p} +$$

$$+ w_{cond}(h_f-h_2)]/M_2 \tag{5.3-35}$$

$$\dot{T}_2 = [\dot{h}_2 - (\frac{\partial h_2}{\partial p})_T \dot{p}] / (\frac{\partial h_2}{\partial T})_p \tag{5.3-36}$$

$$w_{cond} = - \dot{p} \left\{ \frac{M_1}{h_{fg}} [\frac{1}{\rho_g} - (\frac{\partial h_g}{\partial p})_{sat}] \right\} \tag{5.3-37}$$

$$p_{den2} = M_1\left\{\frac{1}{h_{fg}}[(\frac{\partial h_g}{\partial p})_{sat} - \frac{1}{\rho_g}][\frac{1}{\rho_g} - \frac{1}{\rho_2} + (h_f - h_2)(\frac{1}{\rho_2})^2(\frac{\partial \rho_2}{\partial T_2})_p/(\frac{\partial h_2}{\partial T_2})_p + (\frac{1}{\rho_g})^2(\frac{\partial \rho_g}{\partial p})_{sat}]\right\} + (\frac{M_2}{\rho_2})(\frac{1}{\rho_2})^2(\frac{\partial \rho_2}{\partial T})_p/(\frac{\partial h_2}{\partial T})_p \quad (5.3\text{-}38)$$

$$p_{num2} = \frac{1}{\rho_2}[(1-x_r)w_r + w_{fd} - w_d - \frac{\dot{Q}_1}{h_{fg}}] - (\frac{1}{\rho_2})^2 \frac{(\frac{\partial \rho_2}{\partial T_2})_p}{(\frac{\partial h_2}{\partial T})_p}[(1-x_r)w_r(h_f-h_2) + w_{fd}(h_{fd}-h_2) - \dot{Q}_1 \frac{(h_f - h_2)}{h_{fg}} + \dot{Q}_2] \quad (5.3\text{-}39)$$

A Case 2 situation can occur not only during transients, but also in steady state. For example, in steady state operation, with the water level just below the feedwater ring, the liquid in V_2, shown in Fig. 5.2-1, will be subcooled, while the vapor in the dome remains saturated. The transient solution follows similarly to that described for Case 1 and is presented in block diagram form in Fig. 5.3-11. The solution of the Case 2 model equations when a steady-state is sought is illustrated in Fig. 5.3-12.

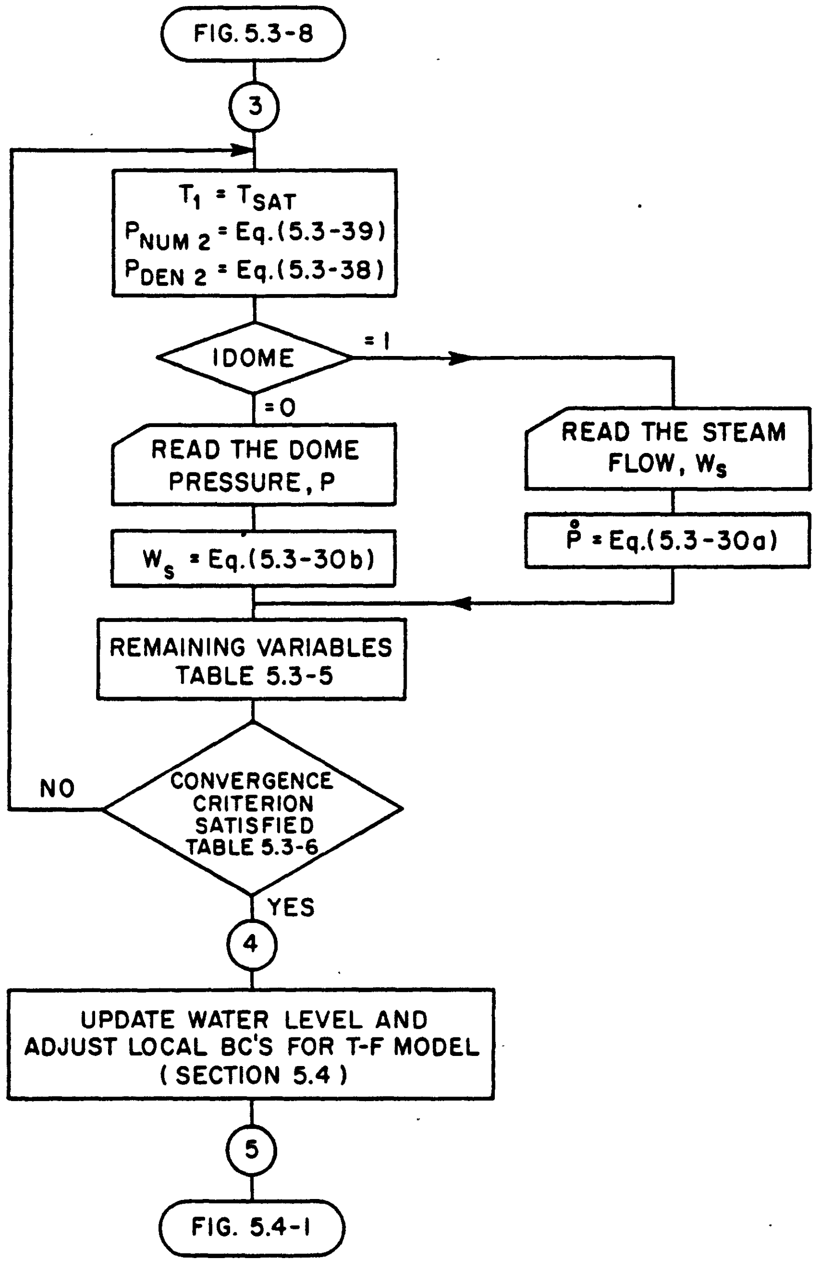


Figure 5.3-11.

Solution of recirculation model
Case 2 equations for transient analysis.

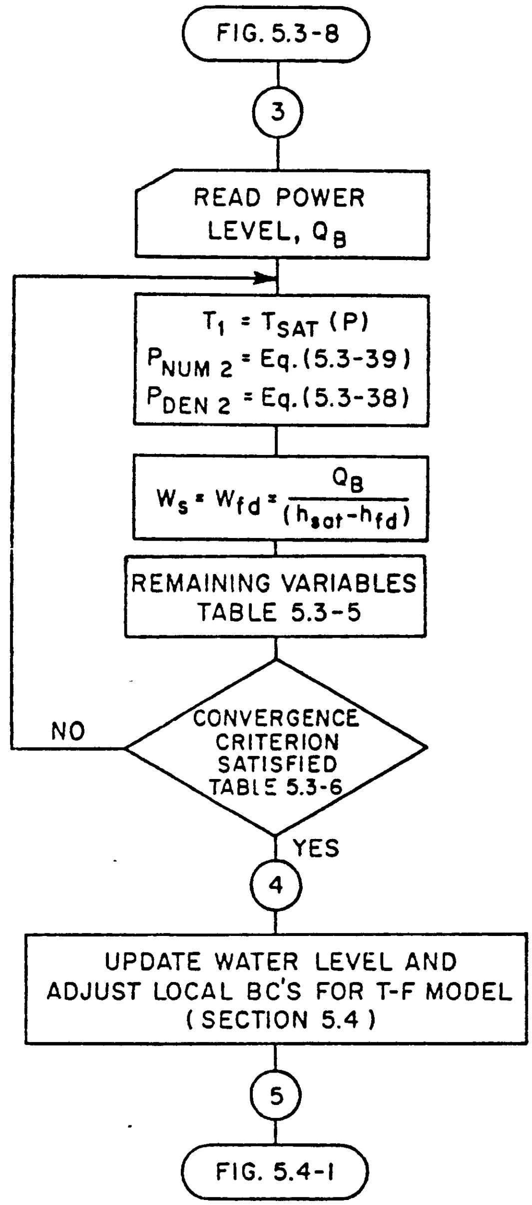


Figure 5.3-12.

Solution of recirculation model
Case 2 equations for steady-state analysis.

Table 5.3-5. Case 2 variables outstanding in Fig. 5.3-11.

REMAINING VARIABLES
w_{cond} = Eq. (5.3-37)
$\dot{M}_2$ = Eq. (5.3-34)
$\dot{M}_1$ = Eq. (5.3-31)
$M_1^{n+1} = M_1^n + \dot{M}_1 \Delta t$
$M_2^{n+1} = M_2^n + \dot{M}_2 \Delta t$
$\dot{h}_2$ = Eq. (5.3-35)
$\dot{T}_2$ = Eq. (5.3-36)
$T_2^{n+1} = T_2 + \dot{T}_2 \Delta t$
$h_1 = h_1\ (p,\ T_{sat})$
$h_2 = h_2\ (p,\ T_2)$
$h_2^* = h_2^n + h_2 \Delta t$
$\rho_2^{n+1} = \rho_2(p,\ T_2)$
$V_2^{n+1} = M_2^{n+1}/\rho_2^{n+1}$
$V_1^{n+1} = M_1^{n+1}/\rho_1^{n+1}$

Table 5.3-6. Case 2 iterative solution convergence criterion.

CONVERGENCE CRITERION
$\left\| h_2^* - h_2(p, T_2) \right\| \leq \varepsilon_H$
$\left\| (V_1 + V_2)/V_T - 1 \right\| \leq \varepsilon_V$

5.3.4.5 Case 3

It is shown in Appendix A that when the constraints listed in Table 5.3-1 for Case 3, viz.:

$$w_{cond} = 0$$

$$T_2 = T_{sat}$$

are applied to Eqs. (5.3-1), (5.3-2), and (5.3-5) to (5.3-11), the following set of equations can be derived:

$$\dot{p} = (p_{num3} - w_s/\rho_1)/p_{den3} \qquad (5.3\text{-}40a)$$

$$w_s = (p_{num3} - \dot{p} \cdot p_{den3})\rho_1 \qquad (5.3\text{-}40b)$$

$$\dot{M}_1 = w_r x_r - w_s + w_{f1} \qquad (5.3\text{-}41)$$

$$\dot{h}_1 = [(w_r x_r + w_{f1})(h_g - h_1) + (\frac{M_1}{\rho_1})\dot{p} + \dot{Q}_1]/M_1 \qquad (5.3.42)$$

$$\dot{T}_1 = [\dot{h}_1 - (\frac{\partial h_1}{\partial p})_T \dot{p}]/(\frac{\partial h_1}{\partial T})_p \qquad (5.3\text{-}43)$$

$$\dot{M}_2 = (1 - x_r)w_r + w_{fd} - w_d - w_{f1} \tag{5.3-44}$$

$$h_2 = h_f \tag{5.3-45}$$

$$T_2 = T_{sat} \tag{5.3-46}$$

$$p_{num3} = (w_r x_r + w_{fd}\frac{h_{fd}-h_f}{h_{fg}} + \frac{\dot{Q}_2}{h_{fg}})\frac{1}{\rho_1} + \left[(1 - x_r)w_r + w_{fd} - w_d - w_{fd}\frac{h_{fd}-h_f}{h_{fg}} - \frac{\dot{Q}_2}{h_{fg}}\right]/\rho_f - - (\frac{1}{\rho_1})^2 \frac{(\frac{\partial\rho_1}{\partial T})_p}{(\frac{\partial h_1}{\partial T})_p}\left[(w_r x_r + w_{fd}\frac{h_{fd}-h_f}{h_{fg}} + \frac{\dot{Q}_2}{h_{fg}})(h_g-h_1) - \dot{Q}_1\right] \tag{5.3-47}$$

$$p_{den3} = M_2\left\{\frac{1}{h_{fg}}[(\frac{\partial h_f}{\partial p})_{sat} - \frac{1}{\rho_f}][\frac{1}{\rho_1}-\frac{1}{\rho_f}-(\frac{1}{\rho_1})^2\frac{(\frac{\partial\rho_1}{\partial T})_p}{(\frac{\partial h_1}{\partial T})_p}(h_g-h_1)] + (\frac{1}{\rho_f})^2(\frac{\partial\rho_f}{\partial p})_{sat}\right\} + M_1\left\{(\frac{1}{\rho_1})\left[(\frac{\partial\rho_1}{\partial p})_T - \frac{(\frac{\partial\rho_1}{\partial T})_p}{(\frac{\partial h_1}{\partial T})_p}(\frac{\partial h}{\partial p})_T\right] + (\frac{1}{\rho_1})^3\frac{(\frac{\partial\rho_1}{\partial T})_p}{(\frac{\partial h_1}{\partial T})_p}\right\} \tag{5.3-48}$$

$$w_{f1} = \dot{p}\left[\frac{M_2}{h_{fg}}\left(\frac{1}{\rho_f} - \left(\frac{\partial h_f}{\partial p}\right)_{sat}\right)\right] + w_{fd}\left[\frac{h_{fd}-h_f}{h_{fg}}\right] + \frac{\dot{Q}_2}{h_{fg}} \quad (5.3\text{-}49)$$

The solution scheme for Case 3 is illustrated in the block diagram of Fig. 5.3-13 for transient analysis. Figure 5.3-14 shows the solution method when a steady state is sought.

5.3.4.6 Case 4

The condition for Case 4 is that both liquid and vapor be at saturation, i.e.:

$$T_1 = T_{sat}$$

and

$$T_2 = T_{sat} \quad .$$

When these constraints are applied to Eqs. (5.3-1), (5.3-2), and (5.3-5) to (5.3-11), it is shown in Appendix A that the following set of equations can be derived:

$$\dot{p} = (p_{num4} - w_s/\rho_g)/p_{den4} \quad (5.3\text{-}50a)$$

$$w_s = \rho_g(p_{num4} - \dot{p} \cdot p_{den4}) \quad (5.3\text{-}50b)$$

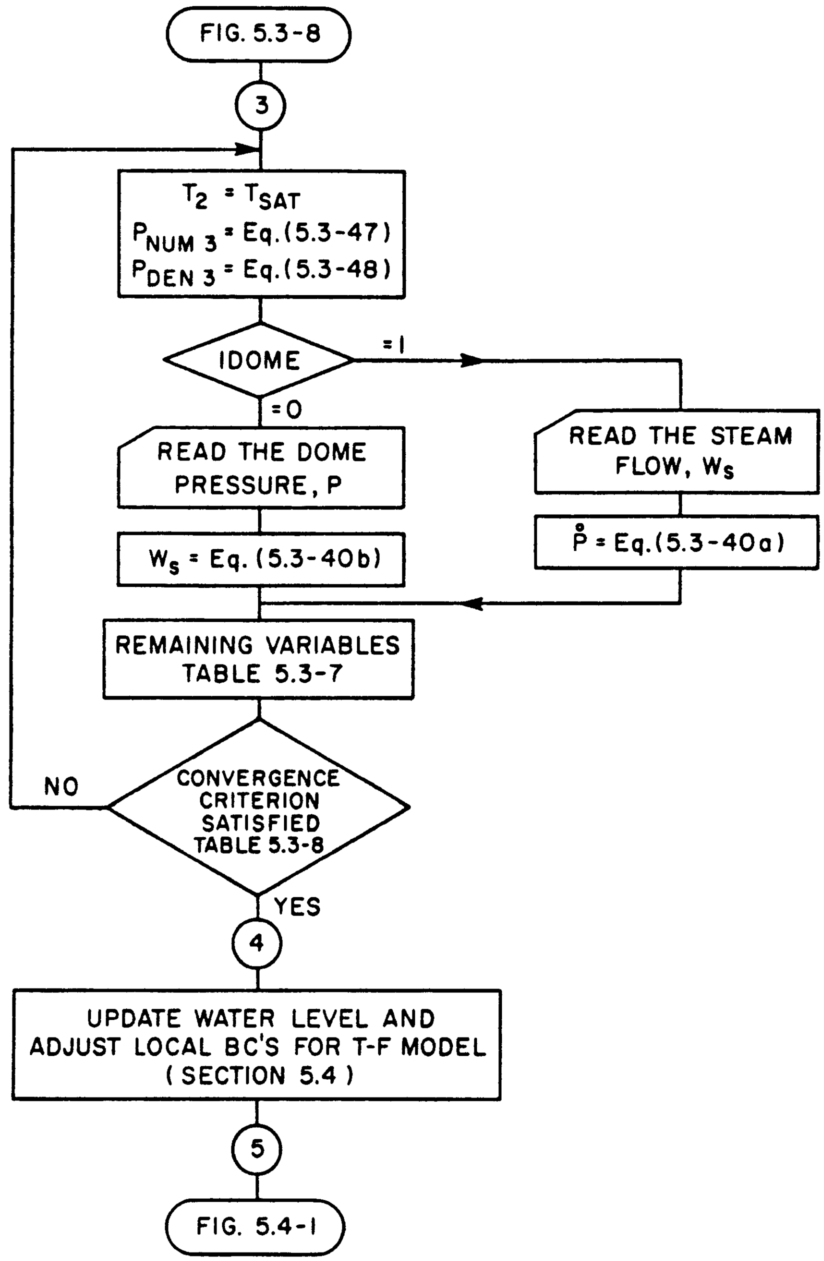


Figure 5.3-13.

Solution of recirculation model
Case 3 equations for transient analysis.

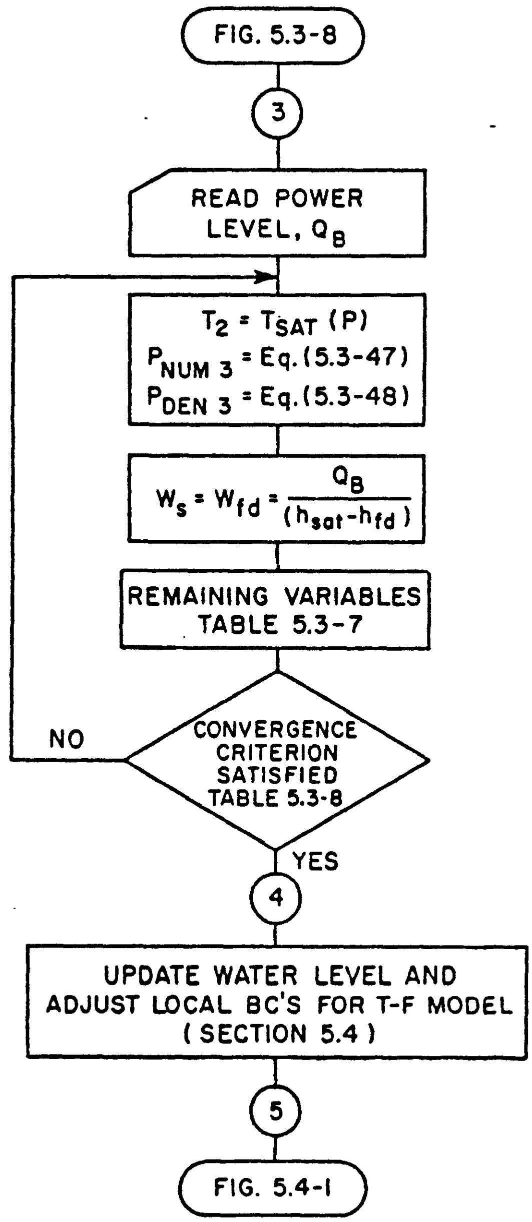


Figure 5.3-14.

Solution of recirculation model
Case 3 equations for steady-state analysis.

Table 5.3-7. Case 3 variables outstanding in Fig. 5.3-13

REMAINING VARIABLES
w_{f1} = Eq. (5.3-49)
$\dot{M}_1$ = Eq. (5.3-41)
$\dot{M}_2$ = Eq. (5.3-44)
$M_1^{n+1} = M_1^n + \dot{M}_1 \Delta t$
$M_2^{n+1} = M_2^n + \dot{M}_2 \Delta t$
$\dot{h}_1$ = Eq. (5.3-42)
$\dot{T}_1$ = Eq. (5.3-43)
$T_1^{n+1} = T_1^n + \dot{T}_1 \Delta t$
$p^{n+1} = p^n + \dot{p} \Delta t$
$h_1^{n+1} = h_1(p, T_1)$
$h_1^* = h_1^n + \dot{h}_1 \Delta t$
$\rho_1^{n+1} = \rho_1(p, T_1)$
$\rho_2^{n+1} = \rho_2(p, T_{sat})$
$h_2^{n+1} = h_2(p, T_{sat})$
$V_2^{n+1} = M_2^{n+1}/\rho_2^{n+1}$
$V_1^{n+1} = M_2^{n+1}/\rho_1^{n+1}$

Table 5.3-8. Case 3 iterative solution convergence criterion.

CONVERGENCE CRITERION
$\left\| h_1^* - h_1(p, T_1) \right\| \leq \varepsilon_H$
$\left\| (V_1 + V_2)/V_T - 1 \right\| \leq \varepsilon_V$

$$\dot{M}_1 = w_r x_r - w_s + w_{f1} - w_{cond} \tag{5.3-51}$$

$$h_1 = h_g \tag{5.3-52}$$

$$T_1 = T_{sat} \tag{5.3-53}$$

$$w_{f1} = \dot{p}\left\{\left(\frac{M_2}{h_{fg}}\right)\left[\frac{1}{\rho_f} - \left(\frac{\partial h_f}{\partial p}\right)_{sat}\right] + \frac{\dot{Q}_2}{h_{fg}}\right\} \tag{5.3-54}$$

$$\dot{M}_2 = (1 - x_r)w_r + w_{fd} - w_d + w_{cond} - w_{f1} \tag{5.3-55}$$

$$w_{cond} = \dot{p}\left\{\left(\frac{M_1}{h_{fg}}\right)\left[\left(\frac{\partial h_g}{\partial p}\right)_{sat} - \left(\frac{1}{\rho_g}\right)\right] - \frac{\dot{Q}_1}{h_{fg}}\right\} \tag{5.3-56}$$

$$p_{num4} = \left(\frac{1}{\rho_g}\right)\left[w_r x_r + \frac{\dot{Q}_1 + \dot{Q}_2}{h_{fg}}\right] + \frac{1}{\rho_f}\left[(1-x_r)w_r + \right.$$

$$\left. + w_{fd} - w_d - \frac{\dot{Q}_1 + \dot{Q}_2}{h_{fg}}\right] \tag{5.3-57}$$

$$p_{den4} = M_1 \left\{ \frac{\frac{1}{\rho_g} + \frac{1}{\rho_f}}{h_{fg}} \left[\left(\frac{\partial h_g}{\partial p}\right)_{sat} - \frac{1}{\rho_g} \right] + \left(\frac{1}{\rho_g}\right)^2 \frac{\partial \rho_g}{\partial p} \right\} +$$

$$+ M_2 \left\{ \frac{\frac{1}{\rho_g} + \frac{1}{\rho_f}}{h_{fg}} \left[\left(\frac{\partial h_f}{\partial p}\right)_{sat} - \frac{1}{\rho_f} \right] + \left(\frac{1}{\rho_f}\right)^2 \left(\frac{\partial \rho_f}{\partial p}\right)_{sat} \right\} \qquad (5.3\text{-}58)$$

Figure 5.3-15 illustrates the solution procedure in a transient case while Fig. 5.3-16 shows the procedure for steady state.

5.4 Coupling with the Two-Fluid Calculation

5.4.1 Description of the Procedure

Figure 5.4-1 is a block diagram showing the main stages in steam generator analysis with THERMIT-UTSG. The first step consists of reading in initial conditions. Next, the nature of the analysis with the respective input are read, following the logic of Fig. 5.3-5. The case number is then selected according to the discussion of Section 5.3-2 and, when appropriate, the heat sink terms are determined as shown in Section 5.3.3. The water level position with respect to the feedwater ring is assessed to decide whether stratification within the liquid control volume, V_2, is to be considered and the appropriate corrections are made as summarized in

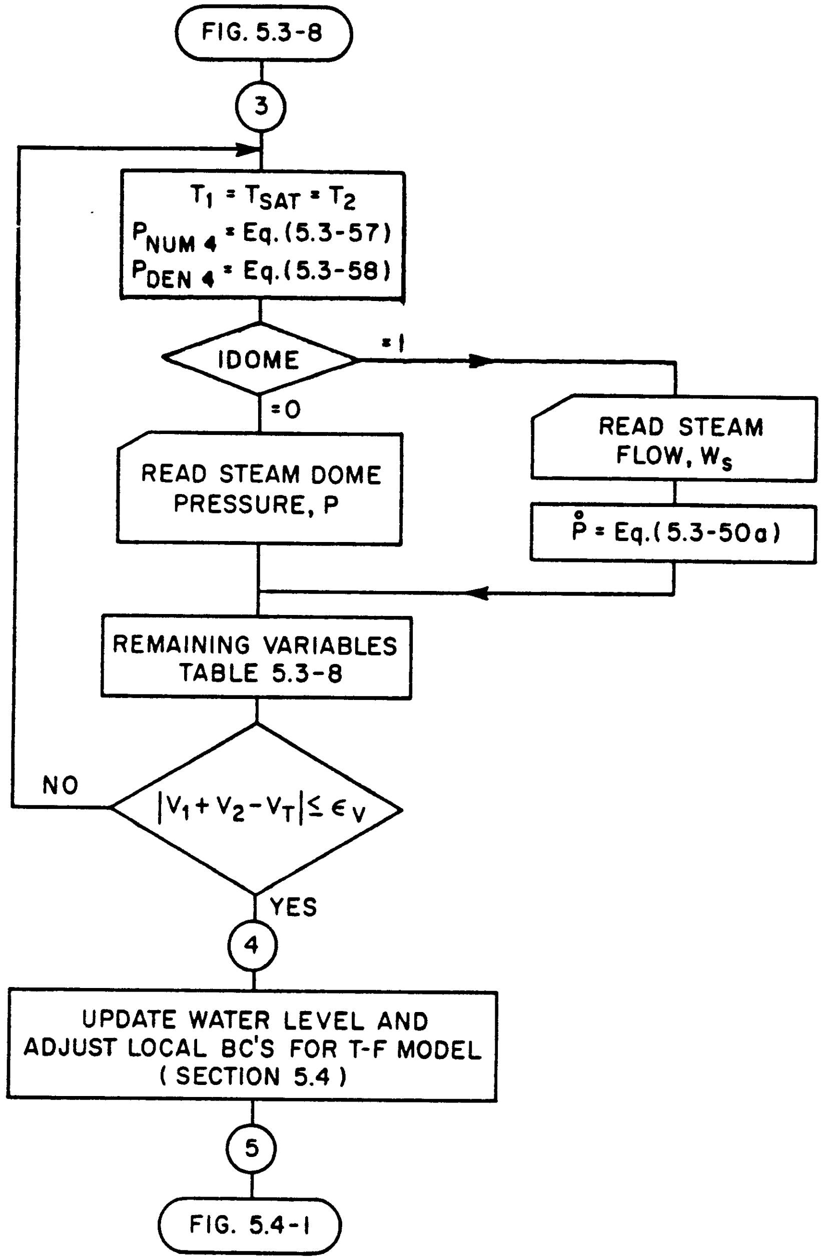


Figure 5.3-15.

Solution of recirculation model
Case 4 equations for transient analysis.

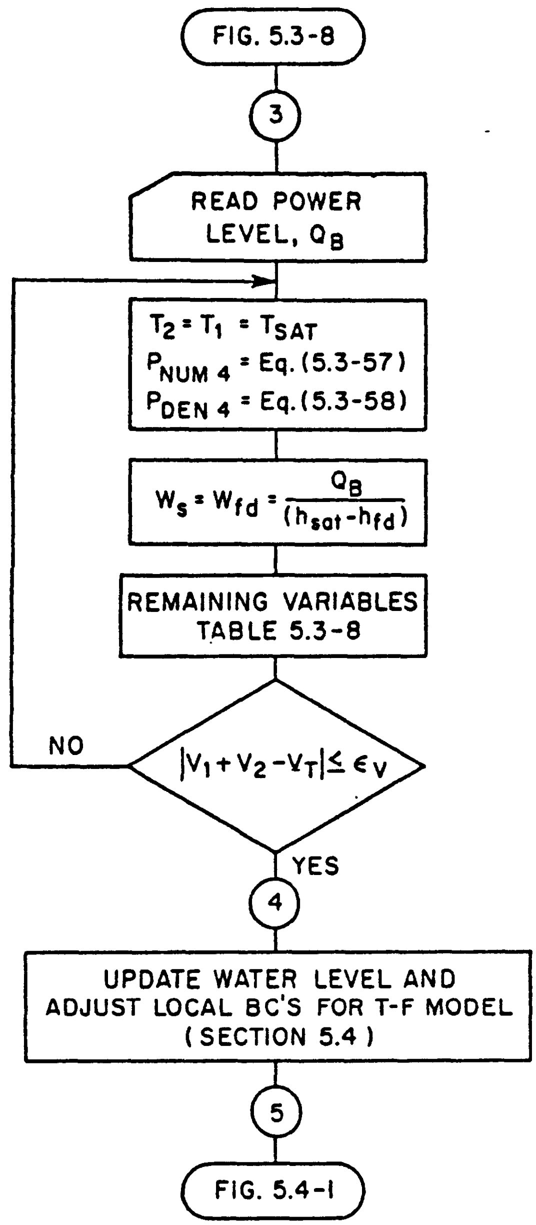


Figure 5.3-16.

Solution of recirculation model
Case 4 equations for steady-state analysis.

Table 5.3-9. Case 4 variables outstanding in Fig. 5.3-15.

REMAINING VARIABLES
w_{f1} = Eq. (5.3-54)
w_{cond} = Eq. 5.3-56)
$\dot{M}_1$ = Eq. (5.3-51)
$\dot{M}_2$ = Eq. (5.3-55)
$M_1^{n+1} = M_1^n + \dot{M}_1 \Delta t$
$M_2^{n+1} = M_2^n + \dot{M}_2 \Delta t$
$p^{n+1} = p^n + \dot{p} \Delta t$
$T_1^{n+1} = T_{sat}$
$T_2^{n+1} = T_{sat}$
$h_1^{n+1} = h_g(p, T_{sat})$
$h_2^{n+1} = h_f(p, T_{sat})$
$\rho_1^{n+1} = \rho_g(p, T_{sat})$
$\rho_2^{n+1} = \rho_f(p, T_{sat})$
$V_2^{n+1} = M_2^{n+1}/\rho_2^{n+1}$
$V_1^{n+1} = V_1^{n+1}/\rho_1^{n+1}$

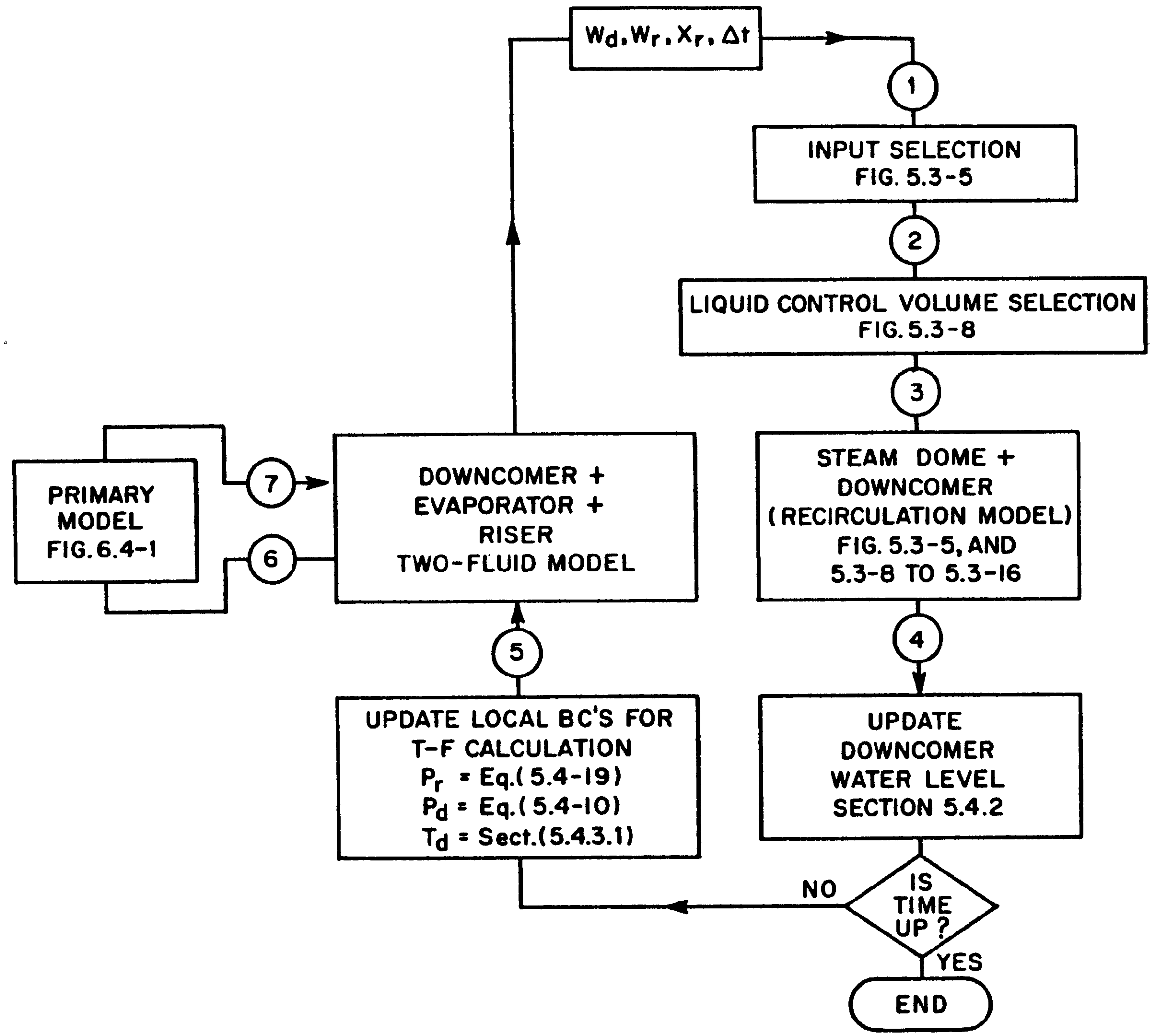


Figure 5.4-1.

Coupling two-fluid and recirculation models.

Fig. 5.3-8. Next, the recirculation model equations are solved following the iterative technique of section 5.3.4. The result of this calculation is used to update steam dome quantities and to calculate the new downcomer water level. The local boundary conditions for the two-fluid calculation are updated using these results and followig the procedures to be described in this section. The primary model and its coupling to the two-fluid calculation are discussed in Chapter 6.

5.4.2 Determination of the Downcomer Water Level

After the recirculation model equations have converged, as described in Section 5.3.4, and the model parameters given in Tables 5.3-3, 5.3-5, 5.3-7, and 5.3-9 resume their final values, the water level is updated by initially computing the change in the liquid volume,

$$\Delta V_2^{n+1} = V_2^{n+1} - V_2^{n} \qquad (5.4.1)$$

Then, advancing the downcomer volume by the same increment,

$$V_d^{n+1} = V_d^{n} + \Delta V_2^{n+1} \qquad (5.4\text{-}2)$$

the downcomer water level is determined through a geometrical relation linking volume to level, namely:

$$X_1^{n+1} = f(V_d^{n+1}) \quad (5.4\text{-}3)$$

Appendix H describes this function, which is generalized to enable analysis of most steam generator geometries.

5.4.3 Determination of Local Boundary Conditions for the Two-Fluid Calculation

5.4.3.1 Downcomer Top Cells

As discussed in Section 4.3, the top of the downcomer is a boundary of flow into the two-fluid domain and, as such, requires the assignment of values for: (a) pressure and (b) temperature. The values result from the calculations described throughout Section 5.3.4. Additional considerations for assigning each property follow. Other relevant thermodynamic properties are obtained via state equations.

(a) Pressure

The pressure, p_d, must be prescribed in the topmost cell of each downcomer channel as indicated in Fig. 4.2-2. This must be done in such a way that the resulting pressure at the water level is the new time steam dome pressure, p. In order to accomplish this, it must be remembered that the space between the water level, x_l, and the topmost cell level, x_{ref}, is filled with fictitious water, for the reason given in Section 5.1. For the given situation the pressure difference,

$$\Delta p = p_d - p \tag{5.4-4}$$

can be partioned into the components:

$$\Delta p = \Delta p_{fr} + \Delta p_a + \Delta p_{fo} + \Delta p_i + \Delta p_g \tag{5.4-5}$$

where the subscripts stand for

fr = friction
a = acceleration
fo = form loss or drag
i = inertia
g = gravity.

The non-gravity terms can be grouped together into a single term via

$$\Delta p_{ng} \doteq \Delta p - \Delta p_g \tag{5.4-6}$$

which is estimated through

$$\Delta p_{ng}^{n+1} \cong \left(\frac{dp}{dz}\right)_{ng}^{n} \left(x_1^{n+1} - x_{ref}\right) \tag{5.4-7}$$

where the superscript n stands for old time value and

$$(\frac{dp}{dz})^n_{ng} = (\frac{dp}{dz})^n_{total} + \rho_2^n \cdot g \qquad (5.4\text{-}8)$$

The gravitational head is:

$$\Delta p_g^{n+1} \doteq \rho_2^{n+1} g(x_1^{n+1} - x_{ref}). \qquad (5.4\text{-}9)$$

The pressure to be prescribed at the top of each downcomer channel is calculated by combining Eqs. (5.4-4) to (5.4-9). This yields

$$p_d^{n+1} = p^{n+1} + \left[(\frac{dp}{dz})^n + (\rho_2^{n+1} - \rho_2^n)\right] (x_1^{n+1} - x_{ref}) \qquad (5.4\text{-}10)$$

(b) Temperature

As discussed in Section 5.3.4.2 the liquid control volume for the recirculation model is treated differently according to its position relative to the feedwater distribution ring.

When,

$$V_{1r} > V^*$$

corresponding to Fig. 5.3-7(a), the feedwater mixing is taken to occur in V_{2b}. The resulting enthalpy within that volume is

$$h_{2b} = h_2 - w_{fd}(h_2 - h_{fd})/W_d$$

with mfd = 0 for no upward mixing, as discussed in that section. The temperature, T_{2b}, is calculated from the state equations using the Newton-Raphson (H3) method. The application is as follows. A value of T_{2b} is guessed, say,

$$T_{2b} = T_{2b}^0 \quad (5.4\text{-}11)$$

The function,

$$F(T_{2b}^0) = h_{2b} - f(T_{2b}^0) \quad (5.4\text{-}12)$$

and its derivative

$$F'(T_{2b}^0) = - \left[\frac{df(T_{2b})}{dT}\right]_p^0 = - C_p(T_{2b}^0) \quad (5.4\text{-}13)$$

are calculated using the themmodynamic property subroutine, represented above by the function $f(T_{2b})$. The specific heat,

$$C_p = (\frac{\partial h}{\partial T})_p = (\frac{\partial u}{\partial T})_p - (p/\rho^2)(\frac{\partial \rho}{\partial T})_p \quad (5.4\text{-}14)$$

is calculated as above, also via property subroutine and used in Eq. (5.4-13). The iteration procedure consists of updating the guess through

$$T_{2b}^{n+1} = T_{2b}^{n} - F(T_{2b}^{n})/F'(T_{2b}^{n}) \quad (5.4\text{-}15)$$

where all the quantities above are available in one call of the property subroutine.

Convergence in this method can only be guaranteed if (H3):

1) $F'(T_{2b})$ and $F''(T_{2b})$ have constant sign in the interval

$[T_{2b}^{0}, T_{2b}^{*}]$ where T_{2b}^{*} is such that

$F(T_2^{*}) = 0$ and

2) $$f(T_{2b}^{0}) \cdot F''(T_{2b}^{0}) > 0 \quad . \quad (5.4\text{-}16)$$

Condition (1) is true since the specific heat as well as its derivative with respect to temperature are always positive. Thus, condition (2) can be insured by prescribing an initial guess, such that,

$$F(T_{2b}^{0}) > 0 \quad . \quad (5.4\text{-}17)$$

This can be guaranteed by guessing,

$$T_{2b}^{0} > T_{2b}^{*} \quad , \quad (5.4\text{-}18)$$

since enthalpy increases monotonically with temperature for subcooled liquid.

With T_{2b} determined as described, all downcomer cells above the feedwater ring level are filled with water set to this temperature. The procedure guarantees that the calculated mixed upper downcomer temperature, T_{2b}, is convected downstream by the two-fluid calculation, in correct transport times from feedwater ring to evaporator inlet.

When the liquid control volume, V_2, is positioned with respect to the feedwater ring as shown in Fig. 5.3-7(b), or as long as $V_{1r} \leq V^*$, the liquid temperature in V_2 is given directly in the solution of the recirculation model equations, as described in Section 5.3.4: no further computations are required. This temperature, T_2, is then set in all downcomer cells above the lower boundary of V^*. The purpose of this is to obtain correct transport times for the mixed liquid in V_2 down to the evaporator inlet.

5.4.3.2 Evaporator Top Cells

At the beginning of each time step, the pressure in the top cell of each evaporator and riser channel is a local boundary condition for the two-fluid calculation. Because the separators are not included in the two-fluid domain, the assigned pressure, p_r shown in Fig. 4.2-2 is calculated as

$$p_r = \bar{p} + \Delta p_{sep} \qquad (5.4\text{-}19)$$

where

p = current steam dome pressure

and

Δp_{sep} = separator pressure drop

is given by (B4)

$$\Delta p_{sep} = \bar{\rho}\,(C_1 Q^2 + C_2) \tag{5.4-20}$$

where $\bar{\rho}$ is the average density and Q the volumetric flow rate, all in SI units. The constants can be used to adjust the overall circulation ratio if this quantity is available. If not, Burley (B4) recommends:

$$C_1 = 4494.5 \tag{5.4-21}$$

$$C_2 = 30.5 \tag{5.4-22}$$

for the separators used in this study.

Chapter 6

PRIMARY-SIDE MODEL

6.1 Introduction

The primary-side of a UTSG commonly designates the reactor coolant circulating within the U-tubes. In this work, the term is extended to include the tubes as well. The purpose of the present model is to calculate:

1. primary coolant temperature
2. primary-side wall temperature
3. intermediate wall temperature
4. secondary-side wall temperature

in as many locations as desired, within each computational cell of the secondary-side two-fluid model.

The starting point toward this goal is the one-dimensional energy equation written for the primary coolant within one tube. This equation is summed over a group of tubes resulting in the primary coolant temperature model. This model describes the average coolant temperatures of the groups of tubes and incorporates effects of tube to tube differences in length. Although all the tubes in the group are represented by only one coolant temperature, there can be as many groups as desired, provided certain geometrical restrictions are met. This method allows a good measure of flexibility. If a coarse secondary-side mesh is used and yet primary-side temperature distribution detail is desired, several groups of tubes can be placed within each cell. If less detail is required from the primary-side, few or even one group of tubes can be placed in a cell. Even when only one group is used, effects of tube to tube differences in length are retained in the average temperature equation of the group. Each tube representative of a group is called a tube bank. The concept is illustrated in Fig. 4.4-1, where the cell marked (J,I) coincides with Cell (J,I) in Fig. 4.2-2.

The tube bank metal thermal behavior is described by the one- dimensional heat conduction equation in cylindrical coordinates. The differential equation is integrated over two zones arbitrarily defined by an intermediate radius. Pre-definition of the number of zones allows the equations to be solved a priori, avoiding time consuming matrix forward elimination routines.

The heat transfer coupling between primary- and secondary-sides involves different procedures in full power steady-state analyses, in transients and other than full power steady-state calculations. This is because a fouling parameter is determined at 100 percent power.

6.2 Primary Coolant Temperature Model

The energy equation for the primary coolant within one tube is derived in Appendix C and can be written in the form:

$$\rho c \frac{\partial T}{\partial t} + cG \frac{\partial T}{\partial s} = - \frac{2}{r_p} q_p'' \qquad (6.2\text{-}1)$$

where

ρ = primary coolant density

c = primary coolant heat capacity

r_p = inner radius of a primary tube

s = path along a tube

G = primary coolant mass flux

q_p'' = heat flux (positive when leaving the primary coolant).

The use of Eq. (6.2-1) to analyze the primary coolant and calculate the heat transfer to the secondary requires further development. This is because for a given mesh scheme laid over the steam generator, e.g. Fig. 4.2-3, there will be several primary tubes within a given cell. Since the lengths of these tubes are different, they will carry different flow rates and have different primary temperatures, even when in the same secondary-side cell where secondary-side parameters are uniform. However, to solve Eq. (6.2-1) for each primary tube is unfeasible, since there are about 8000 tubes in an average size UTSG. Thus, a number of tubes must be used to represent the totality. In the present work this number of representative tubes is left open to user discretion, since there can be no a priori criterion to select an optimum number for all possible transients of interest. The loss of feedwater transient is an example of a situation in which more than one representative tube is desirable. If only one tube is used, and the level at which the tube bends boils dry, the total heat transfer from the primary is underestimated. This follows because lower-bending tubes, which are really in nucleate boiling, would be lumped into a tube exchanging heat in forced convection to vapor. On the other hand, if the tubes are not expected to uncover, or if primary temperature radial distribution is not of interest, then code input can be simplified and computer time saved by using only one tube bank.

The method for lumping primary tubes is as follows: one representative tube is utilized per secondary-side cell level in the U-bend region of the evaporator. This method allows for any number of tubes to be represented, provided there be as many mesh levels in the U-bend region as representative tubes. Each representative tube is called a tube bank. An example of an arrangement with three tube banks is given in Fig. 4.4-1. The assessment of the number of tubes and average horizontal length of each tube bank is discussed later in this section, in connection with the calculation of flow split parameters.

The energy equation for a tube bank is obtained as follows. Let ϕ represent an average primary-side variable for a given tube bank within a given secondary-side cell. For tube j then,

$$\phi_j = \overline{\phi}_i + \phi_j' \qquad (6.2\text{-}2)$$

where ϕ_j' represents the deviation in the local tube j variable from the average tube bank i value of that variable within a given secondary-side cell.

Utilizing this definition and summing Eq. (6.2-1) over the Ntb_i primary tubes contained in tube bank i, as shown in Appendix D, results in an equation of the form:

$$(Ntb_i)\bar{\rho}_i\bar{c}_i\left(\frac{\partial\bar{T}}{\partial t}\right)_i + \bar{c}_i\sum_{j=1}^{Ntb_i} G_j\left(\frac{\partial T}{\partial s}\right)_j =$$

$$= \frac{2}{r_p}\left[\sum_{j=1}^{Ntb_i}\left(\bar{h}_i + \frac{\partial h}{\partial G}.G_j'\right)\right](\bar{T}_{wp} - \bar{T}) + \frac{2}{r_p}\,q''(T') \quad (6.2\text{-}3)$$

where,

$$q''(T') = \left[\frac{\partial\bar{\rho}}{\partial T}\cdot\frac{\partial\bar{c}}{\partial T}\cdot\frac{\partial\bar{T}}{\partial t} + \frac{1}{2}\frac{\partial}{\partial T}(\overline{\rho c})\cdot\frac{\partial}{\partial t}\right]\sum_{j=1}^{Ntb_i}(T')^2 -$$

$$-\sum_{j=1}^{Ntb_i} h_j'(T_{wpj}' - T_j') - \frac{\partial\bar{c}}{\partial T}\sum_{j=1}^{Ntb_i} G_jT_j'\left(\frac{\partial T}{\partial s}\right)_j \quad (6.2\text{-}4)$$

and

T	=	coolant temperature
T_{wp}	=	primary wall temperature
h	=	heat transfer coefficient

The basic assumption utilized to make Eq. 6.2-3 tractable is that the extra heat flux, q"(T') where the single prime is to be interpreted as in Eq. (6.2-2), can be neglected by comparison to the average heat flux for the tube bank, i.e.

$$q''(T') \ll \bar{q}_p'' \tag{6.2-5}$$

With this assumption Eq. (6.2-4) vanishes identically. Two terms in Eq. (6.2-3) retain tube to tube differences within a tube bank. They are the convective and the heat flux terms. It is shown in Appendix D, Eq. (D-43), that these terms can be written so that Eq. (6.2-3) incorporating the assumption of Eq. (6.2-5) now reads:

$$\bar{\rho}_i \bar{c}_i \left(\frac{\partial \bar{T}}{\partial t}\right)_i + \bar{c}_i \bar{G} \left(\frac{\partial \bar{T}}{\partial s}\right)_i \left(\frac{1}{Ntb_i} \sum_{j=1}^{Ntb_i} f_j\right) =$$

$$= \frac{2}{r_p} \bar{h} \left[(1 - \lambda) + \frac{\lambda}{Ntb_i} \sum_{j=1}^{Ntb_i} f_j\right] (T_{wpi} - \bar{T}_i) \tag{6.2-6}$$

where

$$f_j = \frac{G_j}{\bar{G}} = \frac{\left(N \quad L_j^{\frac{1}{m-2}}\right)}{\sum_{k=1}^{N} L_k^{\frac{1}{m-2}}} \tag{6.2-7}$$

is a flow split parameter for each tube with respect to the average primary mass flux and

L_j = length of tube j

N = total number of U-tubes

Ntb_i = number of tubes in tube bank i, calculated via Eq. (6.2-14)

m = exponent of the Reynolds number in the friction factor correlation for the primary

λ = exponent of the Reynolds number in the heat transfer coefficient correlation for the primary fluid

The discretization of Eq. (6.2-6) is done in a fully implicit fashion with the convective term discretized according to the procedure known as "upwind differencing." The resulting finite difference equation is:

$$at(T - T^n) + aew(T - Tew) + ans(T - Tns) =$$

$$= \frac{2}{r_p}\left(\frac{\Delta sew_1 + \Delta sns_1}{\Delta sew_1 + \Delta sew_2 + \Delta sns_1 + \Delta sns_2}\right)[h(T_{wp} - T)] +$$

$$+ \frac{2}{r_p}\left(\frac{\Delta sew_2}{\Delta sew_1 + \Delta sew_2}\right)[h(T_{wp} - T)]_{ew} +$$

$$+ \frac{2}{r_p}\left(\frac{\Delta sns_1}{\Delta sns_1 + \Delta sns_2}\right)[h(T_{wp} - T)]_{nw} \qquad (6.2\text{-}8)$$

In this equation, all the temperatures are at the advanced time (except T^n) and all the coefficients are calculated for each cell. The primary temperature cells coincide with those for the secondary side. The coefficients as well as the heat transfer coefficient are calculated using old time step temperatures as input but advanced time flow rates. Also,

$$at = \overline{\rho}_i \overline{c}_i / \Delta t \ . \tag{6.2-9}$$

$$aew = \frac{1}{\Delta sew} \frac{1}{Ntb_i} \sum_{j=1}^{Ntb_i} f_j \ , \quad \text{or zero if ans} \neq 0 \ , \tag{6.2-10}$$

$$ans = \frac{1}{\Delta sns} \frac{1}{Ntb} \sum_{j=1}^{Ntb_i} f_j \ , \quad \text{or zero if aew} \neq 0, \tag{6.2-11}$$

$$h = [(1 - \lambda) + \frac{\lambda}{Ntb_i} \sum_{j=1}^{Ntb_i} f_j] \cdot \bar{h}_i \ , \tag{6.2-12}$$

Δsew_1, Δsew_2, Δsns_1, Δsns_2 are defined in Fig. 6.2-1, for tube bank number 2, aew is zero for all cells except those which the tube bank enters along a horizontal (<u>e</u>ast-<u>w</u>est) path, ans is zero except for those cells which the tube bank enters along a vertical (<u>n</u>orth-<u>s</u>outh) path. Whenever aew is zero, so are Δsew_1 and Δsew_2; whenever ans is zero, so are Δsns_1 and Δsns_2. A discussion on the terms which make up

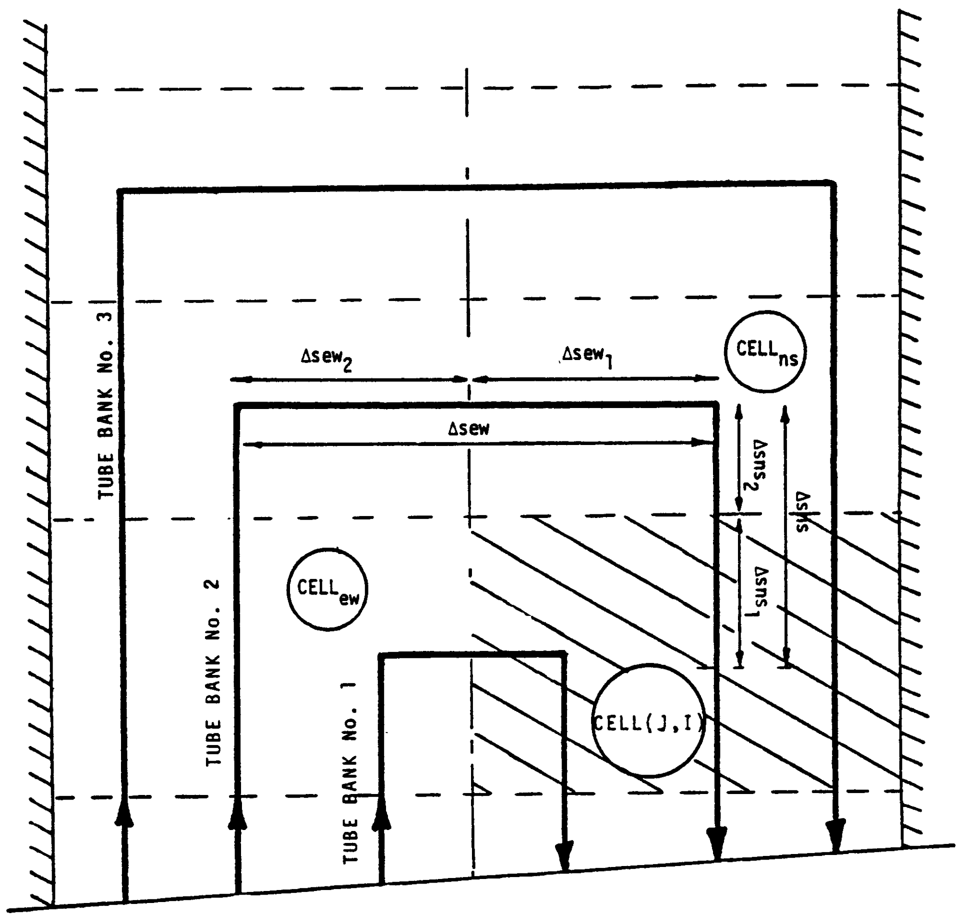


Figure 6.2-1.

Definition of notation for primary temperature equation for a tube bank.

the coefficients given in Eqs. (6.2-10) and (6.2-11) follows.

It should be noted that the terms constituting the convective and heat transfer coefficients of Eq. (6.2-8) depend only on geometrical factors. For this reason they need to be determined only once for a given steam generator and mesh layout. The flow split parameter,

$$F_i = \frac{1}{Ntb_i} \sum_{j=1}^{Ntb_i} f_j \tag{6.2-13}$$

is calculated as follows. Initially an assumption is made regarding the U-tube distribution. As a consequence of the square geometry depicted in Fig. 4.2-3 the tubes are assumed to be uniformly distributed in the straight tube region in rows of constant tube lengths and number. Thus, the number of tubes in tube bank i, Ntb_i, is given by:

$$Ntb_i = \sum_{j=iew_{min}(i)}^{iew_{max}(i)} n_j \tag{6.2-14}$$

where n_j designates the number of tubes in a row and

$iew_{min}(i)$ = order of number of the row of shortest tubes in bank i

$iew_{max}(i)$ = idem. for the longest tubes.

Thus, consistently with these definitions, the order number of the longest of all tube rows is $iew_{max}(i_{max})$, where i_{max} designates the last tube bank.

Given these definitions, the flow split parameter is calculated using the expression:

$$F_i = \frac{N}{Nbt_i} \cdot \frac{\sum_{j=iew_{min}(i)}^{iew_{max}(i)} \left(n_j L_j^{\frac{1}{m-2}}\right)}{\sum_{j=1}^{iew_{max}(i_{max})} \left(n_j L_j^{\frac{1}{m-2}}\right)} , \qquad (6.2.15)$$

where N, the total number of U-tubes, is simply:

$$N = \sum_{j=1}^{iew_{max}(i_{max})} n_j \qquad (6.2\text{-}16)$$

The numerical determination of the flow split parameters is done in a Fortran routine, which is presented at the end of Appendix D. The routine is independent from the code because the F_i as described above are purely geometrical factors, depending only on the particular steam generator and mesh layout. In addition to F_i, the routine also numerically computes the average length of a tube bank via:

$$\bar{L}_i = \frac{1}{Ntb_i} \sum_{j=iew_{min}(i)}^{iew_{max}(i)} n_j L_j \quad . \tag{6.2-17}$$

This formula insures that the appropriate heat transfer area results when the tube bank area is multiplied by the number of tubes in the bank, Ntb_i.

Another term of the convective coefficient, the vertical length Δsns, is the distance between a cell center and the center of the cell upstream of it, primary flow wise. Hence,

$$\Delta sns(j) = \begin{cases} [dz(j) + dz(j-1)]/2 & \text{(hot-side)} \\ \\ [dz(j) + dz(j+1)]/2 & \text{(cold-side)} \end{cases} \tag{6.2-18}$$

The remaining coefficient, Δsew, is the horizontal length of the tube bank, obtained by subtracting the total average length, $\bar{L}_i$ given in Eq. (6.2-17), from the total vertical length measured up to the midplane of the cell where the tube bends.

6.3 Wall Conduction Model

The wall conduction model is based on the energy equation for the tube wall,

$$\rho c \frac{\partial T}{\partial t} - \frac{1}{r}\frac{\partial}{\partial r} rk \frac{\partial T}{\partial r} = 0 \qquad (6.3\text{-}1)$$

in which axial conduction has been neglected.

This equation is also to be solved by a finite difference technique. In the interest of speed and simplicity only two zones are considered in the tube walls, one of which can be used to account for uncertainty in heat transfer capability.* Fig. 6.3-1 shows the radial mesh scheme and the relevant variables for the discretization of Eq. (6.3-1).

The discretization procedure is given in Appendix E. Eq. (6.3-1) is integrated three times: once between r_p and r_p', then between r_p' and r_m' and finally from r_m' to r_s. The resulting set of three algebraic equations can be written in the form:

$$\begin{bmatrix} (xp + yp) & -yp & 0 \\ -yp & (xm + ym + yp) & (-ym) \\ 0 & -ym & (xs + ym) \end{bmatrix} \begin{bmatrix} T_{wp} \\ T_m \\ T_{ws} \end{bmatrix} = \begin{bmatrix} xpT_{wp}^n + r_p q_p'' \\ xm\, T_m^n \\ xsT_{ws}^n - r_s q_s'' \end{bmatrix} \qquad (6.3\text{-}2)$$

The coefficients to these equations are given in Tables 6.3-1(a) and 6.3-1(b).

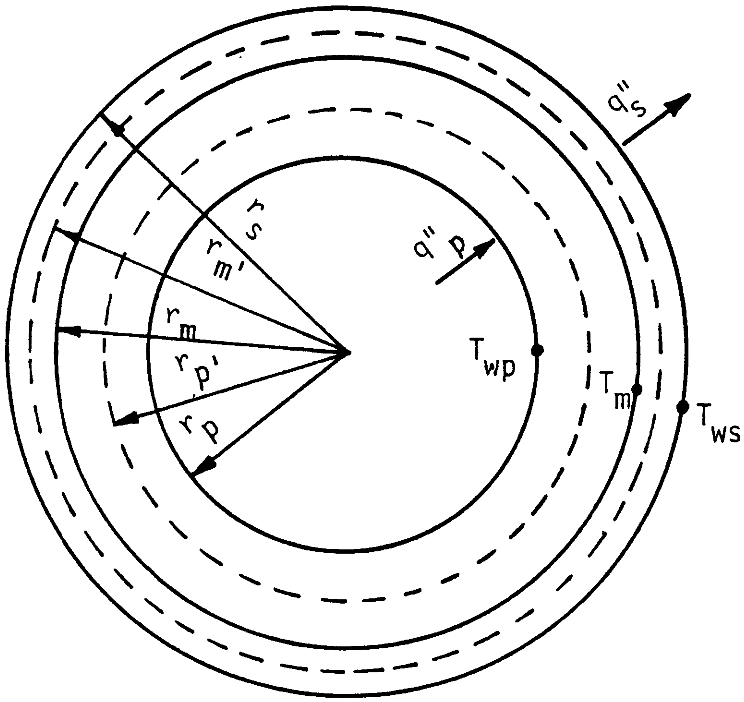


Figure 6.3-1.

Definition of notation for heat conduction finite difference equation.

Table 6.3-1(a). Coefficients for the wall conduction model.

Coefficient	Expression*	Eq.
A_p	$[(xs + ym)(xm + yp) + (xs \cdot ym)\, r_p \cdot D^{-1}$	6.3-3
B_p	$- yp \cdot ym \cdot r_s \cdot D^{-1}$	6.3-4
C_p^n	$[xs + ym)(xm + yp) + (xs \cdot ym)]xp \cdot T_{wp}^n +$	
	$+ (yp.ym.xs)T_{ws}^n + yp \cdot xm(xs + ym)T_m^n\, D^{-1}$	6.3-5
A_s	$yp \cdot ym \cdot r_p \cdot D^{-1}$	6.3-6
B_s	$- [(xp + yp)(xm + ym) + (xp \cdot yp)]r_s \cdot D^{-1}$	6.3-7
C_s^n	$[xp + yp)(xm + ym) + (xp \cdot yp)]xs \cdot T_{ws}^n +$	
	$+ (yp.ym.xp)T_{wp}^n + ym \cdot xm(xp + yp)T_m^n\, D^{-1}$	6.3-8
D	$(xs + ym)[(xp + yp)xm + xp \cdot yp] + (xp + yp)ym \cdot as$	6.3-9
xp	$(\rho c)_{p'}^n (r_{p'}^2 - r_p^2)(2\Delta t)^{-1}$	6.3-10
yp	$r_{p'}^2\, K_{p'}^n (r_m - r_p)^{-1}$	6.3-11
xm	$[(\rho c)_p^{n'}(r_m^{2'} - r_m^2) + (\rho c)_m^{n'}\,(r_m^2 - r_p^{2'})](2\Delta t)^{-1}$	6.3-12
ym	$r_{m'} K_{m'}^n (r_s - r_m)^{-1}$	6.3-13
xs	$(\rho c)_m^{n'}(r_s^2 - r_m^{2'})(2\Delta t)^{-1}$	6.3-14

*Physical and Geometrical Quantities Described in Table 6.3-1(b)

Table 6.3-1(b). Description of physical and geometrical U-tube parameters

SYMBOL*	DESCRIPTION
r_p	Inner U-tube Radius
r_s	Outer U-tube Radius
r_m	Intermediate U-tube Radius
$r_{p'}, r_{m'}$	Idem
T^n_{wp}	Primary-side Wall Temperature
T^n_{ws}	Secondary-side Wall Temperature
T^n_m	Tube Metal Temperature
$(\rho c)^n_{p'}$	Volumetric Heat Capacity at $r_{p'}$
$(\rho c)^n_{m'}$	Idem at $r_{m'}$
$K^n_{p'}$	Thermal Conductivity at $r_{p'}$
$K^n_{m'}$	Thermal Conductivity at $r_{m'}$ Divided by Fouling Coefficient ($K^n_{m'}$ = Thermal Conductivity/FOUL)
FOUL	Determined as Shown in Fig. 6.4-1 (Initially Set to Unity)
Δt	Current Time Step Size

*Subscripts refer to radial position in primary tube wall and are defined in Fig. 6.3-1. Superscript n implies previous time step value.

6.4 Primary- to Secondary-Side Coupling

The primary- and secondary-side energy equations are linked through the heat flux, q_s'', at the outer wall of the U-tubes, which is taken explicitly in both equations. This allows primary- and secondary-side solution procedures to be carried out independently. The use of the explicit heat flux can be justified in two ways. First, the quantities not appearing in time derivatives may be taken at any time between the old and new time, in the finite difference form of differential equations, provided the finite difference equation reduces to the differential equation, as the time step tends to zero. This restriction is satisfied by the explicit method.

Second, heat transfer and energy transport perturbations propagate at the flow velocity for convective heat transfer and at tenths of a second for conduction through the tube walls. This is at least one order of magnitude slower than flow disturbance propagation time, which is on the order of the length scale divided by the sonic velocity in the medium. Thus, the use of the explicit heat flux is not expected to introduce any detectable error in the calculation. It is recognized that an explicit term can introduce an instability in the calculation if the time step is too long. As with almost every new development in THERMIT, however, the key to success is in the validation process. In all cases studied, there has been no evidence of instability. This is probably because the Courant condition, which is the stability upper boundary on the time step for the two-fluid calculation, keeps the time step size small enough not to jeopardize the stability of the heat transfer problem.

In order to link primary- and secondary-side energy equations though the heat flux at the outer wall of the U-tubes, q_s'', it is necessary to eliminate the primary wall temperatures, T_{wp}, in Eq. (6.2-8), in favor of q_s''. This is done making use of the tube wall conduction model represented by Eq. (6.3-2). Forward elimination permits Eq. (6.3-2) to be solved for the wall temperatures in terms of the heat fluxes yielding:

$$T_{wp} = A_p q_p'' + B_p q_s'' + C_p^n \tag{6.4-1}$$

and

$$T_{ws} = A_s q_p'' + B_s q_s'' + C_s^n \tag{6.4-2}$$

where the coefficients are given in Table 6.3-1.

Since the primary wall heat flux can be written

$$q_p'' = h_p(T - T_{wp}), \tag{6.4-3}$$

this relation can be used to eliminate q_p'' from Eq. (6.4-1) which can then be written as

$$T_{wp} = T_{wp}^{(o)} + \frac{\partial}{\partial T} T_{wp} (T - T^n) + \frac{\partial T_{wp}}{\partial q_s''} (q_s'' - q_s''^n) \qquad (6.4\text{-}4)$$

where

$$T_{wp}^{(o)} = (A_p \cdot h_p \cdot bwpo)T^n + (b_p \cdot bwpo)q_s''^n + C_p^n \cdot bwpo \qquad (6.4\text{-}5)$$

$$\frac{\partial}{\partial T} T_{wp} = (A_p \cdot h_p)bwpo \qquad (6.4\text{-}6)$$

$$\frac{\partial}{\partial q_s''} T_{wp} = b_p \cdot bwpo \qquad (6.4\text{-}7)$$

$$bwpo = (1 + Ap^{-1} \cdot h_p) \qquad (6.4\text{-}8)$$

Similarly, the secondary-side wall temperature can be written in the form:

$$T_{ws} = T_{ws}^{(o)} + bwso[h_{lfc} (T_l - T_l^n) + h_{lnb} (T_{sat} - T_{sat}^n) +$$

$$+ h_{vfc} (T_v - T_v^n)] + awso(q_p'' - q_p''^n) \qquad (6.4\text{-}9)$$

where

$$bwso = \frac{1}{h_i}\frac{\partial Tws}{\partial T_i} = -\frac{B_s}{1 - B_s hs} \tag{6.4-10}$$

$i = l, v, sat$

$$awso = \frac{\partial Tws}{\partial q''_{s_p}} = \frac{A_s}{1 - B_s \cdot hs} \tag{6.4-11}$$

$$hs = h_{lnb} + h_{lfc} + h_{vfc} \tag{6.4-12}$$

$$T_{ws}^{(o)} = bwso[h_{lfc}T_l^n + h_{lnb}T_{sat}^n + h_{vfc}T_v^n - q_{lo}^n - q_{vo}^n] +$$

$$+ awso\ q_p''^n + C_s^n \tag{6.4-13}$$

q_{lo}^n and q_{vo}^n are defined in connection with Eq. (6.4-23).

Introducing Eq. (6.4-4) into Eq. (6.2-7) yields the definitive form of the primary temperature equation used in this work, for coupling to the secondary-side energy equation, viz.:

$$T = vpo + vt.T^n + vew.Tew + vns.Tns + vs(q_s'' - q_s''^n) \tag{6.4.14}$$

where the coefficients are given in Table 6.4-1 and the explicit

Coefficient	Expression	Eq. No.
vt	$[at - A_p(h_p)^2(bwpo)(Ads)]/v$	6.4-15
Ads	$\dfrac{(2/rp)(\Delta sew_1 + \Delta sns_1)}{\Delta sew_1 + \Delta sew_2 + \Delta sns_1 + \Delta sns_2}$	6.4-16
v	$at + aew + ans + (Ads \cdot h_p \cdot bwpo)$	6.4-17
vew	aew/v	6.4-18
vns	ans/v	6.4-19
vs	$Ads \cdot h_p \cdot bp \cdot bwpo/v$	6.4-20
vpo	$(Ads \cdot h_p \cdot Twp^{(o)} + C)/v$	6.4-21
C	$\left(\dfrac{\Delta sew_1}{\Delta sew_1 + \Delta sew_2}\right) \dfrac{2}{r_p} \cdot [h_p(Twp - T)]_{ew} +$	
	$+ \left(\dfrac{\Delta sns_1}{\Delta sns_1 + \Delta sns_2}\right) \dfrac{2}{r_p} \cdot [h_p(Twp - T)]_{ns}$	6.4-22

Table 6.4-1. Coefficients for primary temperature equation.

procedure cancels out the last term on the right hand side identically. Hence, Eq. (6.4-14) can be solved for each of the primary tube banks, over the whole solution domain, independently of the secondary-side two-fluid calculation.

The solution procedure linking primary- and secondary-sides is illustrated in the block diagram of Fig. 6.4-1. It is initiated with the recirculation model calculations described in Chapter 5. Then, for each tube bank, the heat transfer coefficient is determined according to the logic of the BEEST (B3) model as summarized in Fig. 3.2-1. The explicit secondary side heat flux is computed via:

$$q''_s = h^n_{lnb}(T^n_{ws} - T^n_{sat}) + h^n_{lfc}(T^n_{ws} - T^n_l) + h^n_{vfc}(T^n_{ws} - T^n_v) + q^n_{lo} + q^n_{vo} \qquad (6.4\text{-}23)$$

where q^n_{lo} and q^n_{vo} are zero if the heat transfer regime is not film boiling and the forced convection to vapor heat transfer coefficient h^n_{vfc} is zero if h^n_{lfn} is not and vice-versa. The nucleate boiling heat transfer coefficient h^n_{lnb} is nil when that is not the prevailing regime.

The solution of the primary model itself differs somewhat for transient and steady-state full power analyses, as evidenced by Fig. 6.4-2 and Fig. 6.4-3. respectively.

In the steady-state full power analysis, the primary model calculation is initiated with the determination of the current explicit power level, Q_B, which is given by:

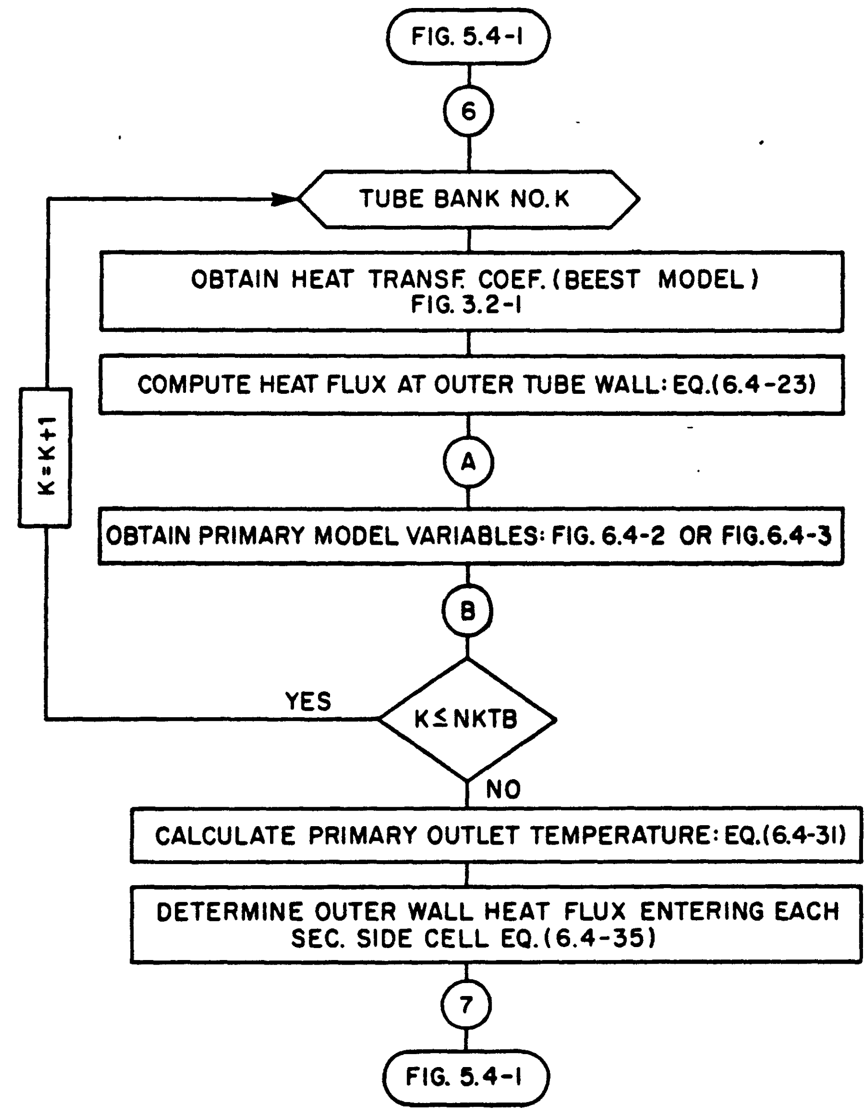


Figure 6.4-1.

Block diagram for primary- to secondary-side heat transfer coupling.

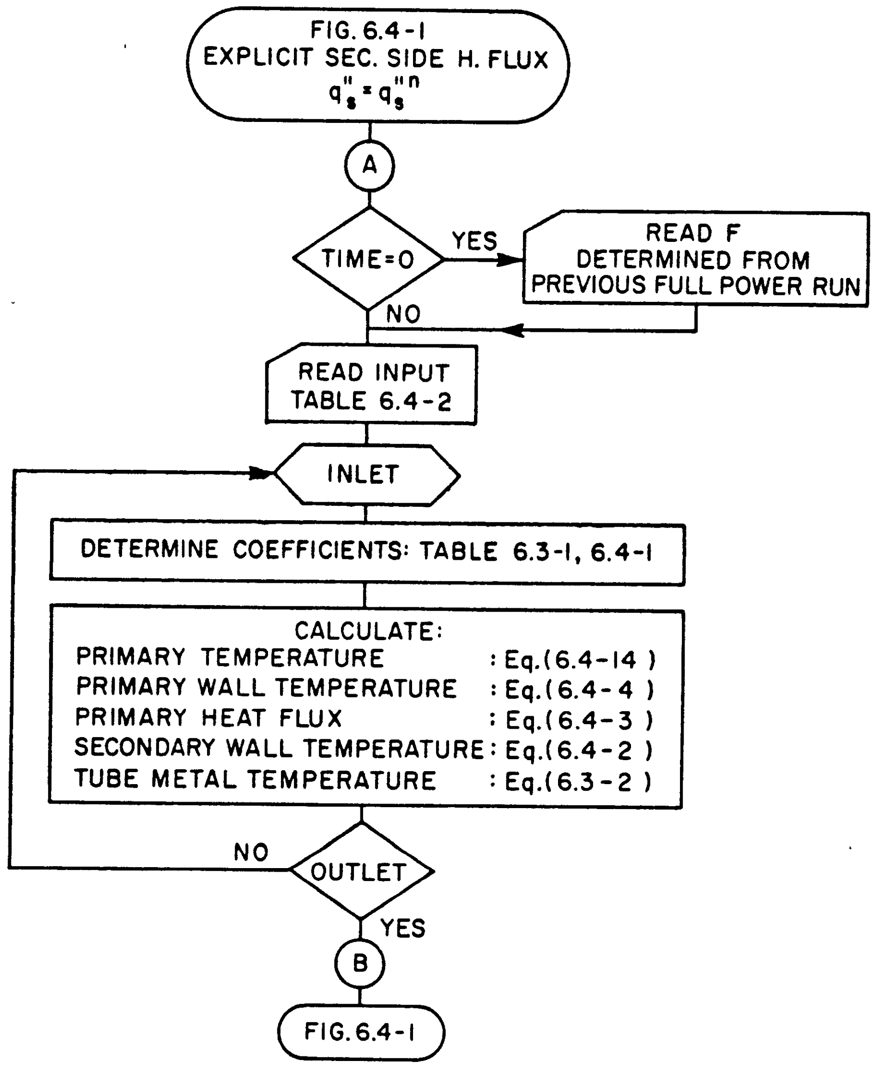


Figure 6.4-2.

Solution procedure for primary model in transient and other than full power steady-state analyses.

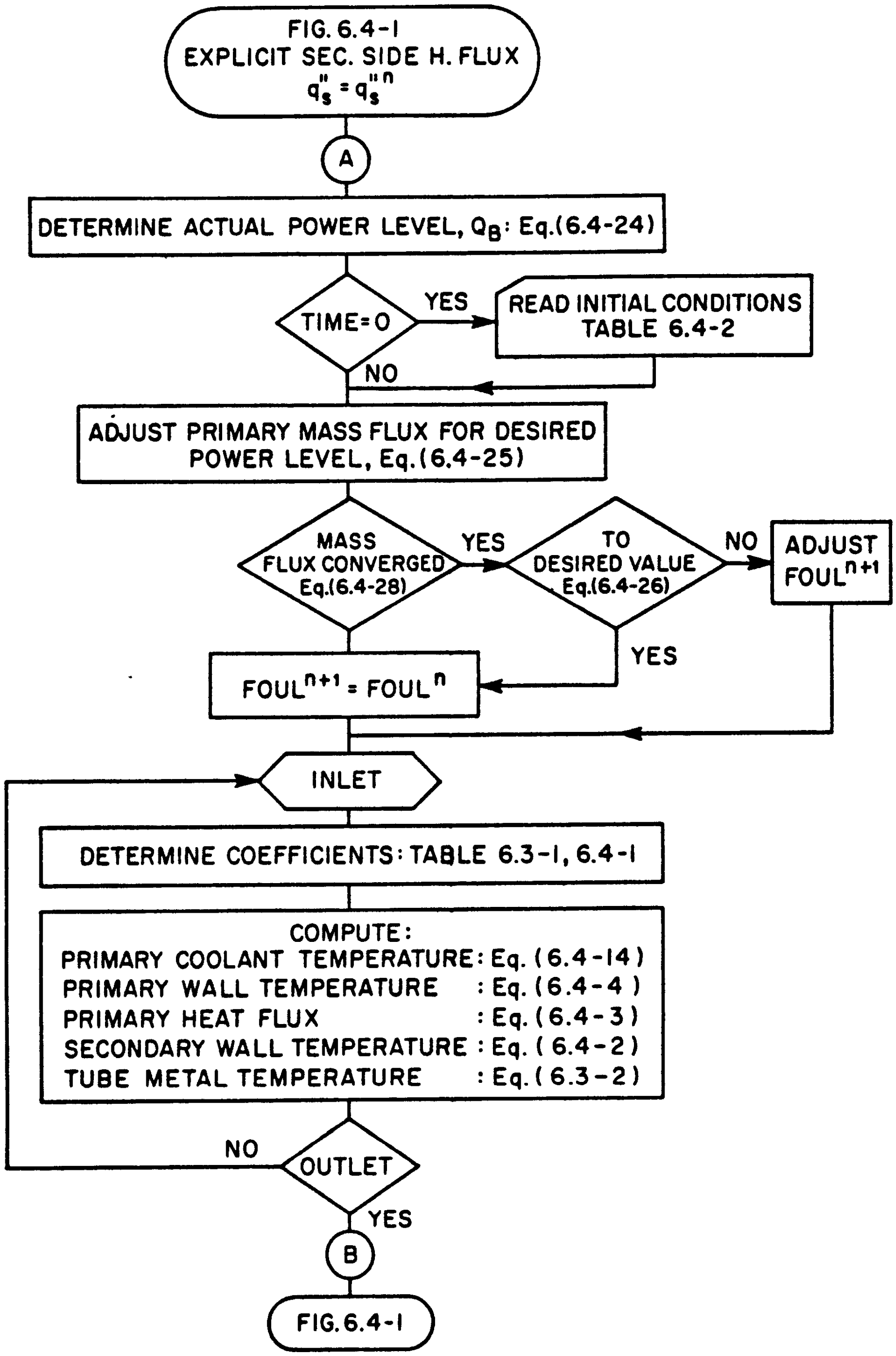


Figure 6.4-3.

Solution procedure for primary model in steady-state full power analysis.

$$Q_B = \sum_i \sum_j \sum_k q_s''(k,j,i).aht(k,j,i) \qquad (6.4\text{-}24)$$

where

q_s''	=	outer U-tube wall heat flux
aht	=	outer U-tube wall heat transfer area of a tube bank (includes all tubes in a tube bank)
NKTB	=	total number of tube banks
j	=	axial level designator of secondary-side cell
i	=	x-y plane designator of secondary-side cell
k	=	tube bank designator

In the first time interval, the input given in Table 6.4-2 is read.

The primary mass flux is adjusted via

$$G^{n+1} = Q_B^* \; G^n/Q_B \qquad (6.4\text{-}25)$$

so as to yield the prescribed power level. This procedure is consistent with the discussion presented in Section 4.5. The steady-state is only considered achieved when the mass flux, given by Eq. (6.4-25), is at G^*, the nominal input level, i.e.

$$\left| G^{n+1} - G^* \right| \leq \varepsilon \qquad (6.4\text{-}26)$$

Table 6.4-2. Input for transient and steady-state primary model analysis

SYMBOL	FORTRAN	NAME	TYPE OF ANALYSIS
P	PPRIM	PRIMARY SYSTEM PRESSURE	TRANSIENT
T_{in}	TPIN	PRIMARY INLET TEMPERATURE	
G	PMFLX	PRIMARY MASS FLUX	
F	FOUL	FOULING COEFFICIENT	
P	PPRIM	PRIMARY SYSTEM PRESSURE	STEADY-STATE
T_{in}	TPIN	PRIMARY INLET TEMPERATURE	
G^*	GSTR	NOMINAL PRIMARY MASS FLUX	
F	FOUL	FOULING COEFFICIENT	
Q_B^*	QB	NOMINAL POWER LEVEL	

where ε is an acceptable error. Nevertheless, it is essential to the solution procedure that the primary mass flux be adjusted at each time step according to Eq. (6.4-25). To illustrate why this is the case, the simplified analysis of Section 4.5 is useful. Under those assumptions, the following relationship

$$T_{sat} = \frac{\left(T_{in} - \frac{Q_B}{w_p \bar{C}_p}\right) \exp \frac{UA}{w_p C_p} - T_{in}}{\exp \left(\frac{UA}{w_p C_p}\right) - 1} \qquad (6.4\text{-}27)$$

involving

T_{sat} = secondary-side saturation temperature

T_{in} = primary-side inlet temperature

Q_B = power level

w_p = primary-side mass flow rate

U = Overall primary- to secondary-side heat transfer coefficient

can easily be demonstrated making use of Eqs. (4.5-2) to (4.5-6). For the full power steady-state calculation, all quantities above are given except for the fouling coefficient, FOUL, which affects the overall heat transfer coefficient. However, since FOUL must be guessed before the primary-model equations are solved, as seen in Table 6.3-1b, the resulting primary-side power will not match the nominal power level Q_B^*. Because Q_B^* is input to the secondary-side calculation, the incorrect fouling coefficient leads to a mismatch of primary- and secondary- side power levels, which prevents convergence. The power mismatch is avoided by adjusting the mass flux through Eq. (6.4-25) at each time step. This avoids the instability. The current value of FOUL is maintained until:

$$\left| G^{n+1} - G^{n} \right| \leq \varepsilon \qquad (6.4\text{-}28)$$

when this condition is met, the perturbation introduced by a previous change in FOUL is assumed to have dampened out. At this point Eq. (6.4-26) is inspected to verify whether the current mass flux value is acceptable and the fouling coefficient is (or isn't) adjusted accordingly.

The various primary and wall temperatures are then computed at each position and the tube bank is swept from inlet to outlet as shown in Fig. 6.4-3.

In steady-state analyses at powers other than full power, the fouling coefficient is already known from a previous full power calculation. In these types of analyses the primary mass flux remains a floating parameter much in the way just described. The only difference in procedure here is that the fouling parameter is now held constant, while the steam dome pressure is periodically adjusted with the objective of forcing the floating primary mass flux to converge to the known system value. In terms of the simplified model exemplified in Eq. (6.4-27) this procedure is equivalent to satisfying the equality by adjusting T_{sat}.

Returning to Fig. 6.4-1, once the primary model variables have been computed for each tube bank, it is then necessary to determine the primary outlet temperature. This is done by invoking the perfect mixing assumption. The assumption is only used to mix primary coolant from different tube banks. The primary temperature in each tube bank is computed all the way to the outlet plenum inclusive by Eq. (6.4-14). This means that intra-tube bank mixing with "previous time" primary coolant in the plenum is done by the temperature equation itself.

The average primary outlet temperature assuming perfect mixing is determined from each of the n outlet temperatures of the i tube banks starting with the definition

$$Cp = \left(\frac{\partial h}{\partial T} \right)_p$$

which is integrated between conditions for tube bank i and the average outlet condition, viz.:

$$\int_{T_i}^{\bar{T}} C_p \, \partial T = \int_{h_i}^{\bar{h}} \partial h$$

yielding

$$h_i = \bar{h} + Cp_i(T - T_i) \qquad (6.4\text{-}29)$$

assuming a constant heat capacity, Cp_i, computed at T_i over the small temperature range T_i to T. Defining the mixed enthalpy as

$$\bar{h} = \sum_{i=1}^{n} m_i h_i / \sum_{i=1}^{n} m_i \qquad (6.4\text{-}30)$$

and making use of Eq. (6.4-29) yields

$$\bar{T} = \sum_{i=1}^{n} m_i Cp_i T_i \; / \sum_{i=1}^{n} m_i Cp_i \qquad (6.4\text{-}31)$$

which is the expression used to compute the primary outlet temperature as a function of the outlet temperatures of each tube bank.

An important concern when comparing model calculations with data is instrument response time. This issue is mentioned at this point because commonly used temperature sensors, namely resistance temperature detectors (RTD) can have non-negligible response times in comparison with time span for transients of interest. Therefore, primary plena temperature predictions which are potentially instrumented via large time constant RTDs should be adjusted for data comparison. The relevance of this issue has been demonstrated in Ref (S2) and need not be reiterated here. Thus, the primary outlet temperature calculated in Eq. (6.4-31) is corrected by the sensor model given in Refs. (S8) and (S2) via the following first order lag differential equation:

$$\frac{d\phi(t)}{dt} = \frac{I(t) - \phi(t)}{\tau} \qquad (6.4\text{-}32)$$

where

$\phi(t)$ = sensor output

$I(t)$ = sensor input

τ = sensor time constant

Discretization of Eq. (6.4-32) using an implicit scheme yields:

$$\phi = \frac{I\Delta t + \phi^n \tau}{\Delta t + \tau} \qquad (6.4\text{-}33)$$

where the superscript, n, refers to the old time and advanced time superscripts, n+1, are omitted for simplicity of notation. Equation (6.4-33) is incorporated into the primary model and is used to adjust the outlet temperature of Eq. (6.4-31) by identifying T_{out} with I. This procedure adds a response time to the predicted temperature thereby allowing it to be compared to the data. The primary inlet temperature must be treated analogously when the same circumstances apply. However, in this case, the sensor input must be input to the calculation. Hence, the primary inlet temperature read in block 2 of Fig. 6.4-2 is identified with ϕ and the solution should proceed using

$$I = \phi + \frac{\tau}{\Delta t}(\phi - \phi^n) \tag{6.4-34}$$

as input. The sensor lag model is bypased whenever the time constant, τ, is set to zero.

After the primary outlet temperature is computed and adjusted if required, a final step is required prior to the initiation of the two-fluid model calculations. This step consists of determining the average heat flux per secondary side cell by weight averaging the heat fluxes of each tube bank in each cell via:

$$q_s''(j,i) = \frac{\sum_{k=1}^{NKTB} q_s''(k,j,i).aht(k,j,i)}{\sum_{k=1}^{NKTB} q_s''\ aht(k,j,i)} \tag{6.4-35}$$

where the quantities on the right hand side above have been defined in connection with Eq. (6.4-24).

Chapter 7

ASSESSMENT

7.1 Scope and Objective

Testing is a fundamental step in the development of any computer model. The purpose of the present assessment study is to verify the overall integrated model and, as possible, the various regional models and their coupling, as described in Chapters 4 to 6. The verification is accomplished by comparing model calculations to appropriate experimental data, as well as to calculations by another computer model.

In order to assess the present model, two distinct but equally important perspectives are adopted:

1. local parameter perspective, and
2. global parameter perspective.

Table 7.1-1 summarizes these viewpoints and characterizes the steam generator facilities to which the present model is applied in each case.

The local parameter viewpoint incorporates the assessment of model calculations such as:

1. primary temperature distributions,
2. secondary temperature distributions,
3. riser quality,
4. riser void fraction,

Table 7.1-1

Model assessment perspectives and steam generator facilities.

PERSPECTIVE	NATURE OF ANALYSIS (NUMBER OF TESTS)	THIS WORK COMPARED TO	FACILITY	POWER PER STEAM GENERATOR (MWt)	OPERATING PRESSURE (MPa)
Assessment of local parameter calculations	Steady-state (2)	Experiment Ref.(H5) computer code Ref. (H5)	Westinghouse Model Boiler-2 (MB-2)	6.67	6.9
Assessment of global parameter calculations	Transient (5)	Measured plant data Ref.(S8)	Arkansas Nuclear One - Unit 2 (ANO-2)	1408	6.2

5. various key pressure drops, and

6. downcomer flow and circulation ratio.

These quantities are compared to steady-state measurements (H5) and to the calculations of another computer model (S3)(H5). Part of the importance of the steady-state study lies in the fact that several of the various regional models can be validated in this way. For example, item 1 above evaluates the primary temperature model as well as its thermal coupling to the secondary-side two-fluid model. Item 2 assesses the energy balance part of the recirculation model by checking the temperature at which downcomer fluid enters the evaporator. The various heat transfer models in the subcooled liquid section of the evaporator are also evaluated via item 2. The remaining items test the overall performance of the model. The main objective of the local parameter steady-state tests, however, and the reason why they are conducted at full and half power, is to demonstrate the physical soundness of the integrated model. Different physical processes characterize operation at full and half power. For example, the gravity pressure loss component in the evaporator is more significant at lower power levels, where both mass velocities and void fractions are lower. The ability to correctly calculate circulation ratios, at both high and low power levels, indicates that the different key pressure drops are correctly apportioned in all regions. The correct calculation of the tube bundle pressure drop at full and half power can only be accomplished if the hydrodynamics and heat transfer of the region are modeled correctly. The correct determination of the separator pressure drop indicates that the volumetric flow rate at the exit is correctly predicted, thus enhancing the credibility of the integrated model. Although the separator loss coefficients given in Eqs. (5.4-21) and (5.4-22) are adjusted at full power to give the appropriate separator loss, the same full power constants are used at half power. Hence, the correct separator pressure drop can only be predicted at the lower power level if the volumetric flow rate through it is adequately calculated.

The global parameter perspective, as suggested in Table 7.1-1, is adopted with the objective of assessing the overall steam generator model response to actual plant transients. These transients are chosen such that model response can be evaluated for small, as well as substantial simultaneous perturbations of the system boundary conditions. Tests cover the full range of input disturbances that could be expected during operational transients. The calculated quantities in all cases are the time dependent responses of:

1. downcomer water level,
2. primary outlet temperature, and
3. outlet nozzle steam flow rate.

The present model computes the distributed parameters in transients as well as in steady-state. However, experimental data is only available for the global parameters listed above.

The measurements were performed during start up tests at the Arkansas Nuclear One power plant (S8). Table 7.1-2 shows qualitatively the type of variation of each of the system boundary conditions for each of the transients selected for the present study. Horizontal inspection of the table shows there is always a case for which any given system boundary condition undergoes substantial perturbation. The term "substantial" is qualitative and no attempt at defining it is made, other than to say it refers to perturbation range as well as rate-of-change. The term is used when these characteristics are as substantial as could be expected for that parameter, outside of accident conditions. The small perturbation studies are as significant as the more substantial disturbance studies because they demonstrate model stability as well as appropriate sensitivity to small changes in system boundary conditions. It will be noted in Table 7.1-2 that two of the assessment studies, the turbine trip and the full length power control element assembly (CEA) drop, are carried out for Steam Generator One (SG-1) and for Steam Generator Two (SG-2). Both SG-1 and SG-2 are included because they respond in a significantly different manner. This is because system boundary conditions, which are input to the calculations, also vary significantly for both units. There are differences in the initial conditions as well. Thus, both steam generators are simulated separately for each test, effectively doubling

Table 7.1-2.

Qualitative perturbation of each system boundary condition for global parameter tests.

SYSTEM BOUNDARY CONDITION	PERTURBATION				
	CEA DROP SG-1	CEA DROP SG-2	TURBINE TRIP SG-1	TURBINE TRIP SG-2	LOSS OF PRIMARY FLOW SG-1
Primary Inlet Temperature	Small	Small	Substantial	Substantial	Substantial
Average Primary Flux	None	None	None	None	Substantial
Primary System Pressure	Small	Small	Substantial	Substantial	Small
Steam Dome Pressure	Small	Small	Substantial	Substantial	Substantial
Feedwater Temperature	None	None	Substantial	Substantial	None
Feedwater Flow Rate	Small	Small	Substantial	Substantial	Substantial

the number of transient validation cases. The loss of primary flow was only carried out for SG-1 because the complete data set is not available for SG-2 for that event.

The power level at which tests were initiated ranges from 49 percent for the control element assembly drop to 82 percent for the loss of primary flow to 98 percent for the turbine trip. Other selected initial conditions are given in Table 7.3-1. Detailed initial conditions are given in the pertinent sections.

7.2 Local Parameter Tests

7.2.1 Background

The local parameters calculated by the present model which are compared to measurements (H5) and to predictions of the ATHOS2 (H5)(S3) computer code, from now on referred to as ATHOS, are listed in Table 7.2-1, along with the text locations of the comparisons.

The detailed measurements of Table 7.2-1 have only been carried out for steady state conditions. Hence, the present calculations are also for steady-state. The comparisons are carried out at two power levels, 100 percent and 50 percent, for the reasons given in Section 7.1.

The present study is applied to the Westinghouse Model Boiler No. 2 test facility. Model Boiler No. 2 (MB-2) is a one percent power-scaled model of the Westinghouse Model F steam

Table 7.2-1.

Text location of local parameter comparisons between present model calculations, measurements (H5), and other code calculations (H5)(S3).

No.	Parameter	Text Reference for 100 Percent Power	Text Reference for 50 Percent Power
1 2 3 4 5 6 7	Circulation Ratio Feedwater Flow Rate Riser Quality Riser Void Fraction Total Pressure Drop Separator Pressure Drop Bundle Pressure Drop	Table 7.2-5	Table 7.2-7
8	Primary Fluid Temperature Distributions	Fig. 7.2-3 through Fig. 7.2-6	Fig. 7.2-10 through Fig. 7.2-13
9	Tube Wall Temperature Distributions	Fig. 7.2-7	Fig. 7.2-14
10	Secondary-Side Temperature Distributions	Figs. 7.2-8 and 7.2-9	Figs. 7.2-15 and 7.2-16

generator, a non-preheat-type unit, designed to be geometrically and thermohydraulically similar to the Model F in several important aspects (H5), viz.:

1. By making use of the full-scale tube, pitch, and support plate geometry.
2. By operating the test model in primary and secondary environments nearly identical to those of the Model F.
3. By appropriate scaling of the flow areas to achieve comparable flow velocities and mass fluxes.
4. As in the Model F, there is an upper downcomer fluid mass extending between the riser and the shell. However, the lower downcomer region of MB-2 is represented by two separate downcomer pipes, one of which discharges into the hot leg and the other into the cold leg sides of the tube bundle. Downcomer pipes are required to provide fluid velocities comparable to those present in the Model F downcomer annulus since the cross-sectional area between the warpper box and the shell is far in excess of that needed for satisfactory downcomer scaling.
5. The model utilizes a moisture separator package which provides an overall similarity to that used in the Model F. It includes a primary separator assembly, a gravity separation region, and a single-tier demister assembly (secondary separator). The overall power

scaling basis (0.78 percent) was applied to the sizing of the moisture separator components so as to duplicate the mass fluxes and velocities in the full-size unit.

Details concerning the geometry of MB-2 and the test loop are available in Ref. (H5). Figure 7.2-1 provides a schematic side view of the unit, while Fig. 7.2-2 shows the cross-section schematic. The nominal operating parameters are given in Table 7.2-2. The axial mesh scheme used for the application of the present model is that of Fig. 4.2-2 with seventeen axial levels where levels one and fifteen correspond to the boundary cells and are shown hatched in Fig. 4.2-2. The horizontal mesh scheme or channel layout is that of Figs. 4.2-3 and 7.2-2. The primary model arrangement is that of Fig. 4.4-1 with three tube banks distributed as shown. The geometric input for this case, as well as the initial guesses to be used in the steady-state calculations, are given in Appendix F.

The mesh layout for the application of the ATHOS computer model is described in detail in Ref. (H5). An illustration of the ATHOS mesh is given in Fig. 7.2-2. The simulations published in Ref. (H5) were carried out for several combinations of fluid flow model, secondary-side heat transfer correlation and friction factor correlation. Table 7.2-3 lists the available options. Details of the various correlations are given in Ref. (S3).

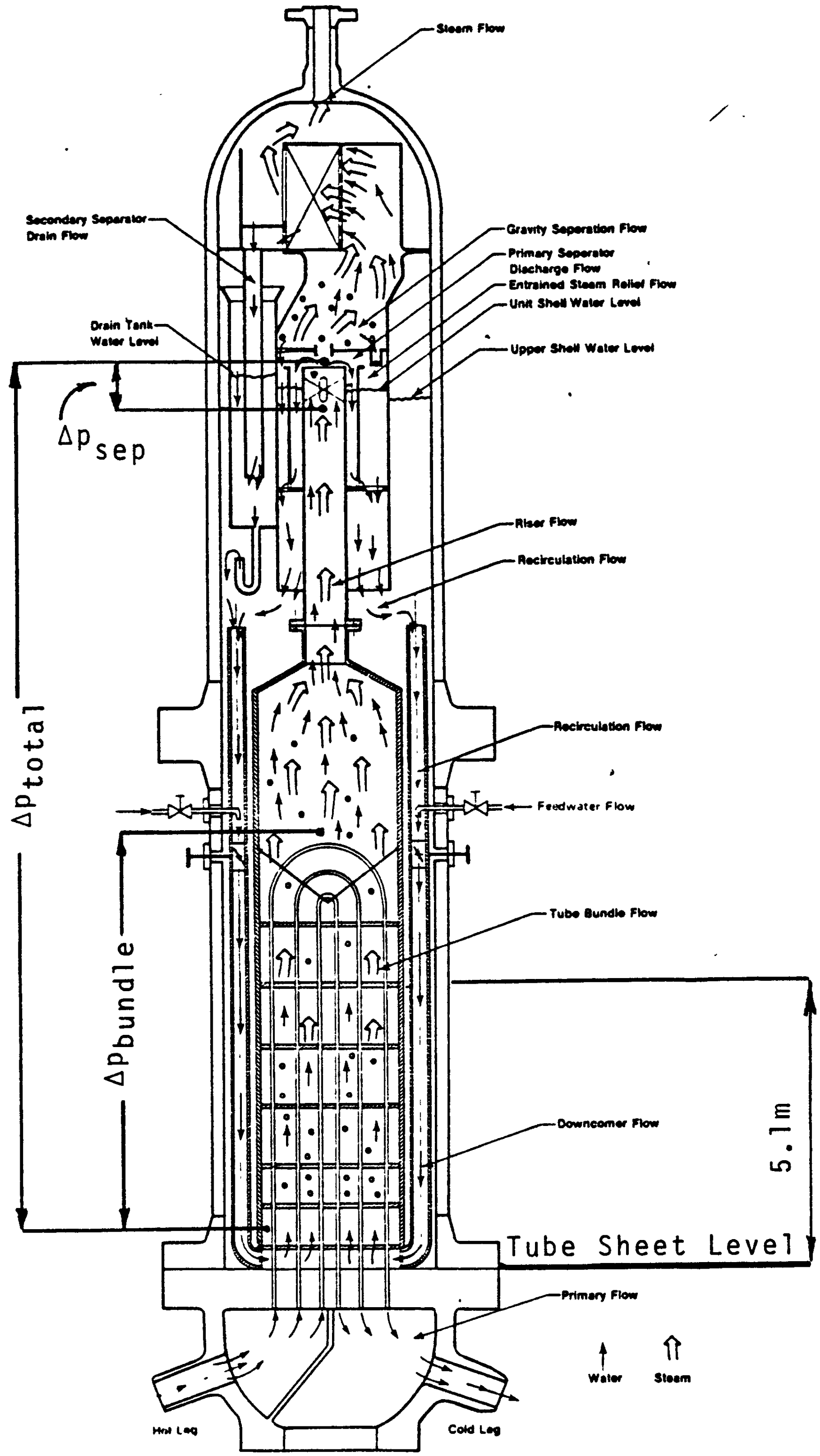


Figure 7.2-1. Side view of Model Boiler-2 (adapted from Ref.(H5)).

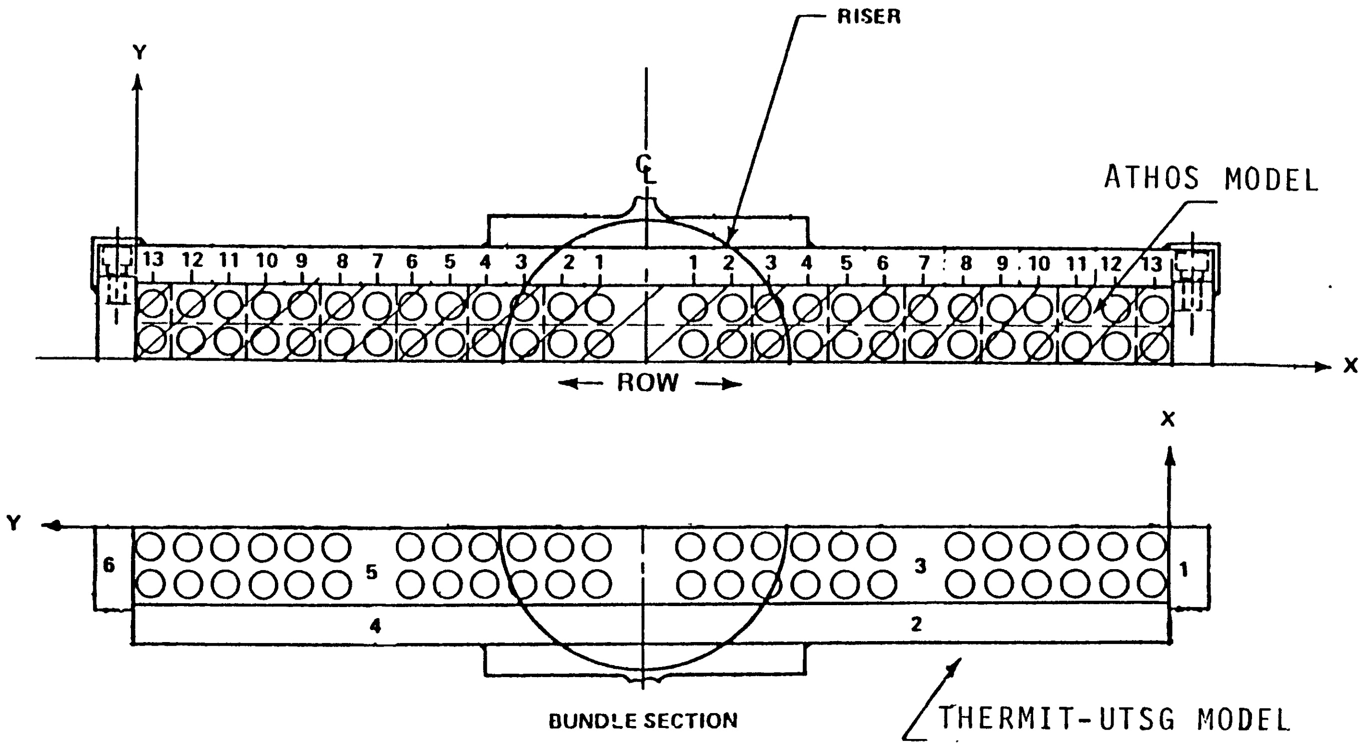


Figure 7.2-2.

Model Boiler-2 cross section showing ATHOS and THERMIT-UTSG channel layout scheme (adapted from Ref. (H5)).

Table 7.2-2.

MB-2 nominal operating parameters.

PARAMETER	VALUE
Primary mass flux	3764 kg/m^2/s
Primary mass flow rate	36.5 kg/s
Primary pressure	15.51 MPa
Steam dome pressure	6.89 MPa
Steam flow rate	3.7 kg/s
Feedwater temperature	499.7 K
Downcomer water level[(1)]	11.24 m
Primary inlet temperature	598.1 K
Primary outlet temperature	566.3 K
Circulation ratio	3.0
Power level	6.67 MWt

(1) above tubesheet surface.

Table 7.2-3.

ATHOS: models and correlations, Ref. (H5).

FLUID FLOW MODELS	HOMOGENEOUS(1),(3) ALGEBRAIC SLIP(2)
SECONDARY-SIDE HEAT TRANSFER CORRELATIONS	EPRI81(2),(3) LELE81 RCP81 CHEN(1)
FRICTION FACTOR CORRELATIONS	EPRI78(2) FLOW3(1),(3)

(1) This combination is referred to at ATHOS-1
(2) idem. ATHOS-2
(3) idem. ATHOS-3

7.2.2 Full Power Steady-State Test

The system boundary conditions for the calculation of the full power steady-state are listed in Table 4.5-1. These parameters are reproduced below for convenience:

1. Primary Inlet Temperature,
2. Average Primary Mass Flux,
3. Primary System Pressure,
4. Power Level,
5. Steam Dome Pressure,
6. Downcomer Water Level, and
7. Feedwater Temperature.

As described in Section 6.4, the primary mass flux is a floating parameter in the steady-state calculation. At full power, the fouling parameter is periodically adjusted so that the calculated primary mass flux converges to the desired value. In the present simulation it was not necessary to carry out this procedure. It was possible to obtain the desired primary mass flux, as shown in Table 7.2-4, with the fouling parameter at unity, i.e. without resorting to any artificial adjustment to match primary heat output to secondary enthalpy rise for the known primary flow rate and inlet temperature. This is a significant demonstration of the primary to secondary heat transfer calculation performance as indicated by the analysis of Section 4.5. Since MB-2 is a new unit, where actual tube fouling is not present, the use of a fouling parameter other than unity is equivalent to introducing a correction in the overall heat transfer coefficient, since the heat transfer area is well known in this case. When no correction is required, the implication is that the calculation of the overall heat transfer coefficient is very accurate. Since the present model does not use an overall heat transfer coefficient but, rather, the individual heat transfer mechanisms are represented at each computational cell, it is clear that the integrated effect of all these calculations has implied a correct overall heat transfer coefficient. This demonstrates the soundness of each of the steps of the heat transfer calculation from primary temperature model to tube wall heat conduction, to secondary-side heat transfer package.

The system boundary conditions for the full power tests are given in Table 7.2-4. The measurements were conducted on several occasions in order to demonstrate the reproducibility of results. The range shown in Tables 7.2-4 and 7.2-5 corresponds to the range observed in the various reproducibility tests.

Calculated quantities are shown in Table 7.2-5. In the column corresponding to ATHOS the parameter range corresponds to the variation observed bewteen results of the various models as given in Table 7.2-3.

The primary outlet temperature, the circulation ratio, the feedwater flow rate, and the riser quality are in excellent agreement with the measured values.

Table 7.2-4.

System boundary conditions for MB-2 full power tests.

PARAMETER	EXPERIMENT (H5)	THERMIT-UTSG
Power level (MWt)	6.65 - 6.67	6.67
Steam dome pressure (MPa)	6.92	6.92
Primary Inlet Temperature (K)	598.1	598.1
Downcomer Water Level (m)(1)	11.24	11.24
Primary Mass Flux ($kg/m^2/s$)(2)	3734 - 3771	3755

(1) Referenced to tube sheet.

(2) Range observed in reproducibility tests.

Table 7.2-5.

Comparison of calculated and measured parameters for MB-2 full power tests.

PARAMETER	EXPERIMENT(1),(H5)	ATHOS(2),(H5)	THERMIT-UTSG
Primary Outlet Temperature	566.2 - 566.8	Not given	566.6
Circulation Ratio	2.98 - 3.03	3.06 - 3.63	3.03
Feedwater flow Rate (kg/s)	3.7	3.7	3.7
Riser Quality	0.33 - 0.34	0.16 - 0.33	0.33
Riser Void Fraction	0.77	0.81 - 0.90	0.85
Bundle Pressure Drop (KPa)	32 - 33	28 - 36	28
Separator Pressure Drop (KPa)	33 - 34	25 - 35	34
Overall Pressure Drop (KPa)	72 - 74	63 - 73	69
Fouling Parameter	---	---	1.0

(1) Range observed in reproducibility tests.

(2) Range calculated by the models listed in Table 7.2-3.

The calculated riser void fraction is higher than the measured value. A lower calculated void fraction would have resulted if the MIT model (K1) for the interfacial exchange of momentum had been used in lieu of the Los Alamos model (K1). This is because the former is about one order of magnitude less (K1) than the latter. As argued in (K1) and suggested by the present results, the MIT model appears to be phenomenologically more accurate. However, numerical instability was observed when the option was activated from the initial conditions. It is possible that this difficulty is unavoidable, that the numerical instability is simply caused by the loose coupling between liquid and vapor momentum equations resulting from the smaller interfacial momentum exchange term given by the MIT model. The weak coupling in conditions of large (more than 7 percent in mass) cross flow would be expected to converge with more difficulty. The problem may disappear with a smaller time step. It is also possible that the stability problem may be avoided if the MIT model is activated using as initial condition the converged solution obtained with the Los Alamos model. Such an investigation is, however, beyond the scope of the present work. In any event, the values quoted as "measured void fraction" may also be low. The void fraction was inferred from measurements of the pressure drop between two taps placed h = 1.33m apart, where the uppermost tap was near the primary steam separator inlet, which can be identified in Fig. 7.2-1. The total pressure drop , $|\Delta p|_t$, was assumed to be due to gravity. With friction and acceleration neglected, the "measured" void fraction, whose value is quoted in Table 7.2-5 is given by:

$$\alpha_m = \frac{\rho_l gh - |\Delta p|_t}{(\rho_l - \rho_v)gh} \quad .$$

If the other components of the pressure drop, call their sum $|\Delta p|_{etc}$, had been included, the adjusted void fraction, α_a, would be:

$$\alpha_a = \alpha_m + \frac{|\Delta p|_{etc}}{gh(\rho_l - \rho_v)} \quad .$$

It is stated in Ref. (H5) that the magnitude of this correction is negligible. However, no quantitative argument is presented. Since all predictions are higher than the measurements and since the correction given above is clearly in the right direction, it seems that at least a small part of the small discrepancy can be attributed to the neglect of $|\Delta p|_{etc}$.

The separator pressure drop is in excellent agreement with the experimental value. The significance of this fact at this power level is limited because the two constants of Eq.(5.4-20) must be adjusted. The adjustment is necessary because the present separator model is quite different from that for which the values given in Eqs. (5.4-21) and (5.4-22) were obtained. The criterion for establishing the current values:

$c_1 = 561.8$ and

$c_2 = 3.8$

is to obtain the correct separator pressure drop. The ratio of C_1 to C_2 is assumed to be the same value determined by Burley (B4) and reflected in Eqs. (5.4-21) and (5.4-22). It is significant that the same constants are used at half power. In that case, good agreement with the measured values confirms that the volumetric flow rate is correctly calculated by the integrated steam generator model.

The calculated tube bundle pressure drop is low. The reason for this can be understood with the aid of Fig. 4.4-1. In that figure, the tube bend region occupies three axial mesh levels. The same applies to the mesh layout for MB-2. This tube bend arrangement is for the heat transfer calculation in the context of the tube bank concept as described in Chapter 6. For the fluid dynamics computations of the secondary-side flow in that region, the position of the rods in each level, whether vertical, horizontal, or at an angle, is only perceived by the model, through the wall-to-fluid interaction term. The calculation of this term is a significant problem in itself, and a detailed analysis is available in Ref. (E1). It can be said, however, that the pressure drop across an axial level in which the tubes are vertical is significantly lower than that associated with other angles. Given the channel layout of

Fig. 4.2-3, where there are only two channels representing the evaporator, the pressure drop at any given level in the U-bend region will be overestimated if the rod position given to the code is not vertical and underestimated if the vertical position is given. This is because the code "sees" all the rods in a given cell as having the same inclination. Clearly, the difficulty is avoided if a larger number of channels is used. However, at present this is prohibitively expensive in terms of computer time and prohibitively time-consuming with respect to the input-preparation effort. The way around this is to assume that all the rods are horizontal in a fraction of the U-bend level(s) and that they are vertical in the remaining U-bend level(s). This rod angle apportioning will overestimate the bundle pressure drop if two levels are given horizontal rods and underestimate it if two levels are given vertical rods, for the arrangement of three levels in the U-bend region. The calculated bundle pressure drop of Table 7.2-5 resulted from positioning the tubes vertically in the two lower U-bend levels and horizontally in the upper U-bend level.

The overall pressure drop calculation is low by the same amount as the tube bundle drop. This means that the riser pressure drop is correctly calculated.

Figures 7.2-3 through 7.2-6 show primary temperature distributions in the hot- and cold-sides; channels 3 and 5, respectively, for the channel layout of Fig. 7.2-2. Each figure in this series corresponds to a different elevation referenced to the tube sheet. Although the present model utilizes one channel for each hot- and cold-leg, respectively, there are three calculated primary temperatures in each channel by virtue of the tube bank method described in Chapter 6 and illustrated by Fig. 4.4-1. It is useful to observe that, as expected, the radial temperature variation increases along the primary tube path. The variation is small, though, and indicates that for the purpose of global parameter computations, the use of a single representative tube is expected to give good results in normal conditions. The different results by the ATHOS code correspond to the combinations of flow model, heat transfer, and friction factor correlations listed in Table 7.2-3. All calculated results for both ATHOS and THERMIT-UTSG are in excellent agreement with the measured temperatures with the exception of the combination labeled ATHOS-1. In fact, the ATHOS-1 combination gave the best predictions of the parameters listed in Table 7.2-5 and cannot be discarded as wrong, although the temperature prediction discrepancies have not been explained as yet.

Tube wall temperatures are compared at 0.51m and 6.6m elevations above the tube sheet in Fig. 7.2-7. The tube wall is 1.02mm thick and the thermocouple hot end was embedded in the wall 0.41mm from the outer surface. Again, calculations are in good agreement with the measurements with the exception of the ATHOS-1 combination. It should be noted that knowledge

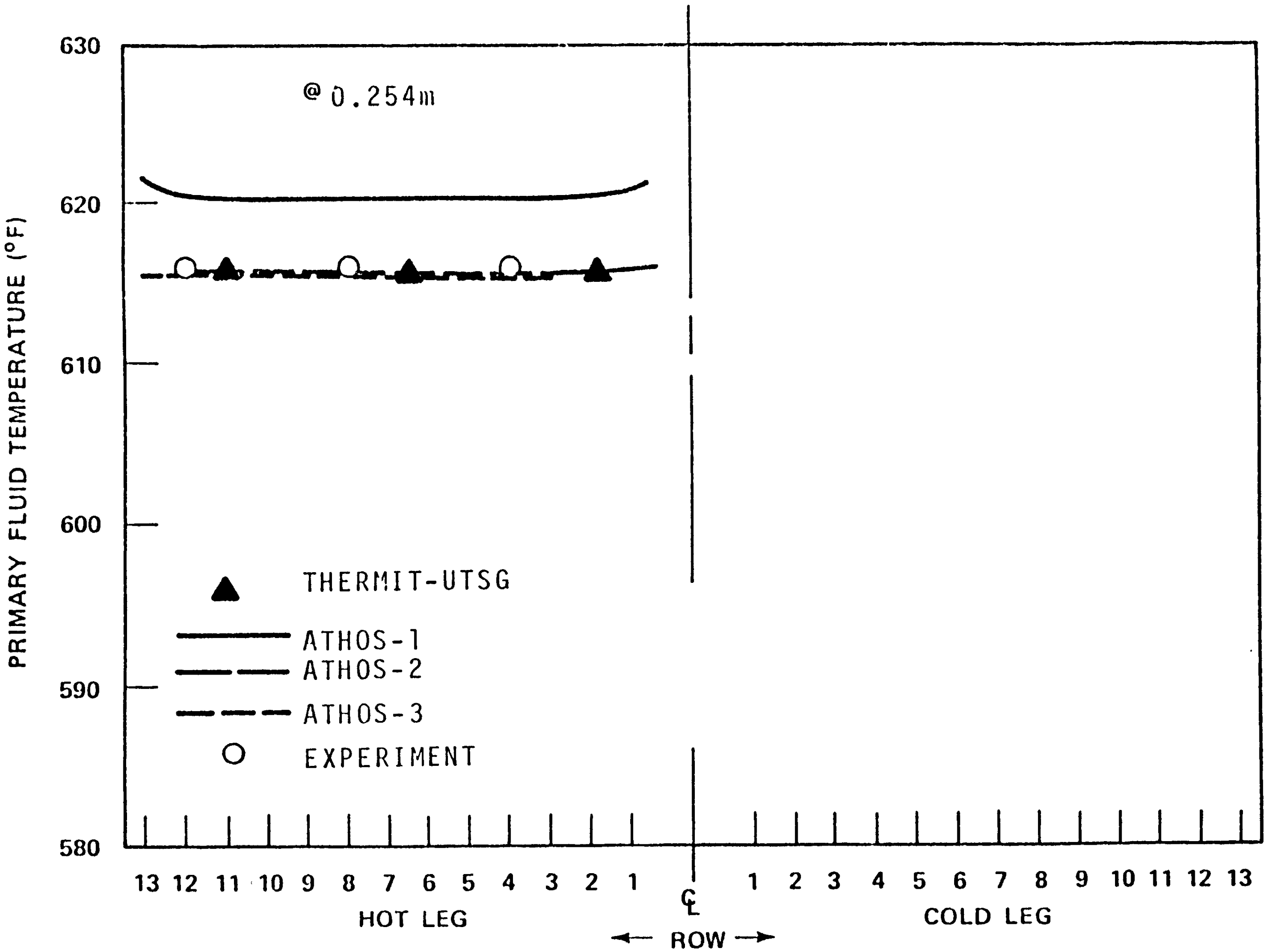


Figure 7.2-3.

Comparison of primary fluid temperatures at
100 percent power, 0.254 m elevation.

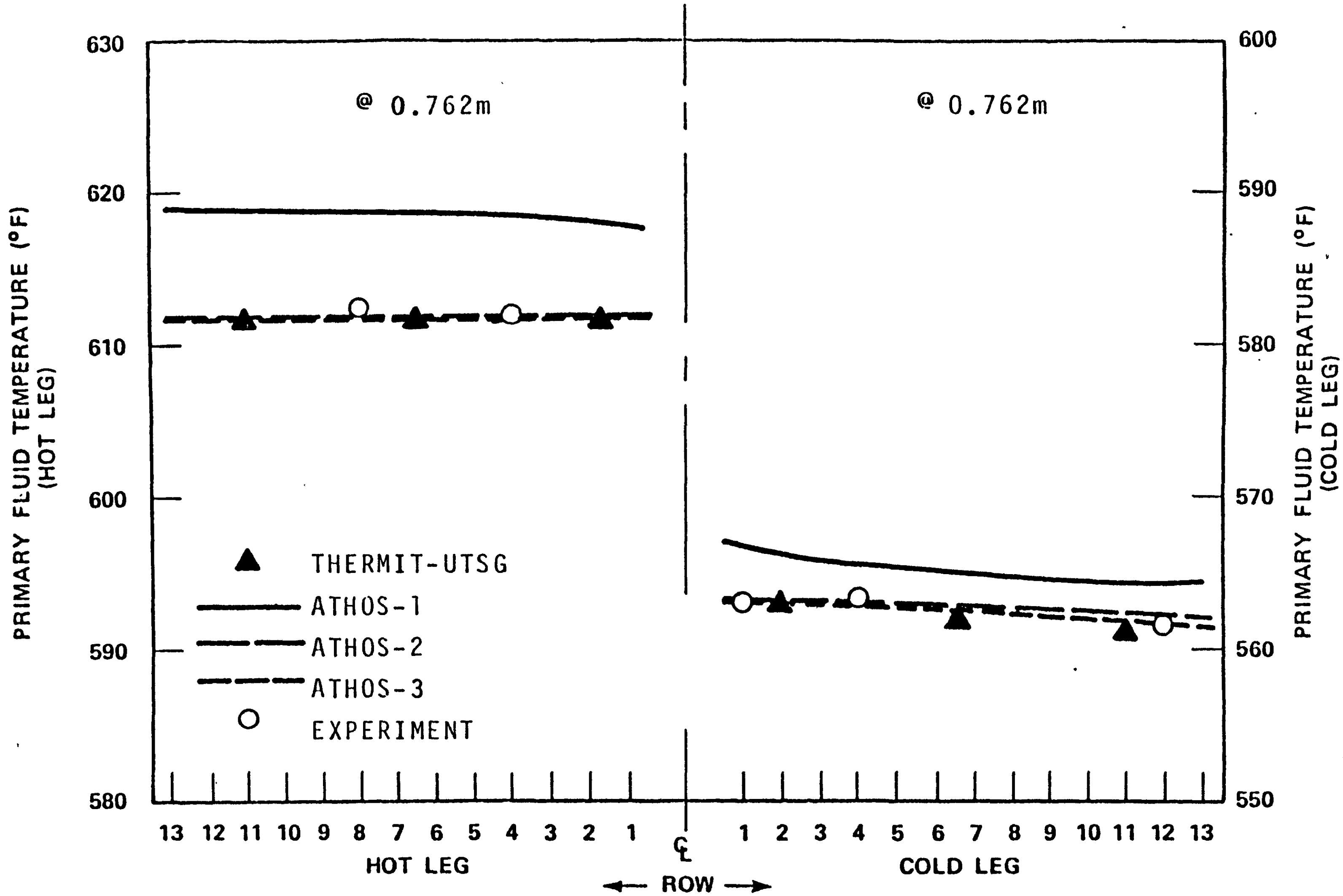


Figure 7.2-4.

Comparison of primary fluid temperatures at
100 percent power, 0.762 m elevation.

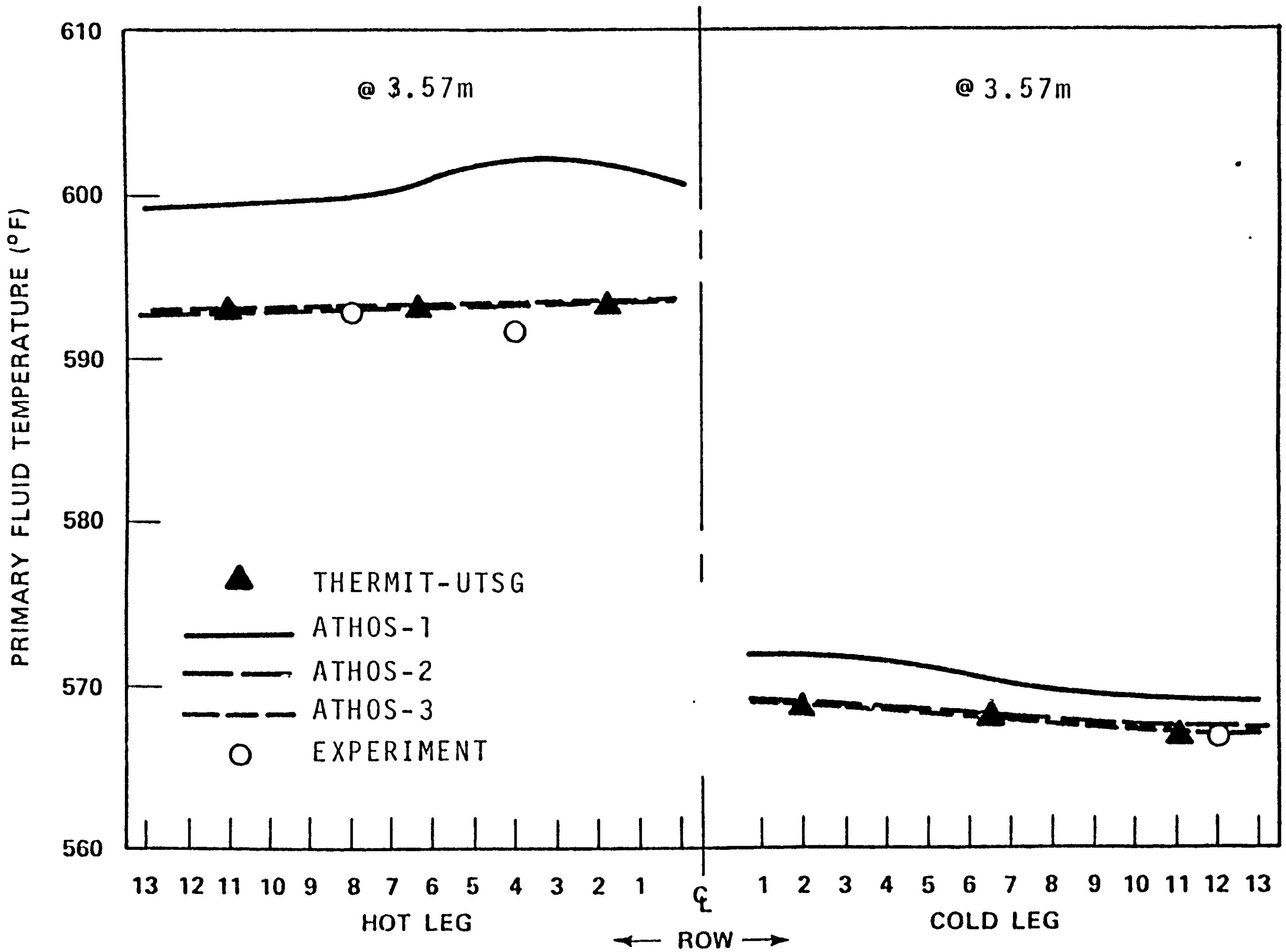


Figure 7.2-5.

Comparison of primary fluid temperatures at 100 percent power, 3.57 m elevation.

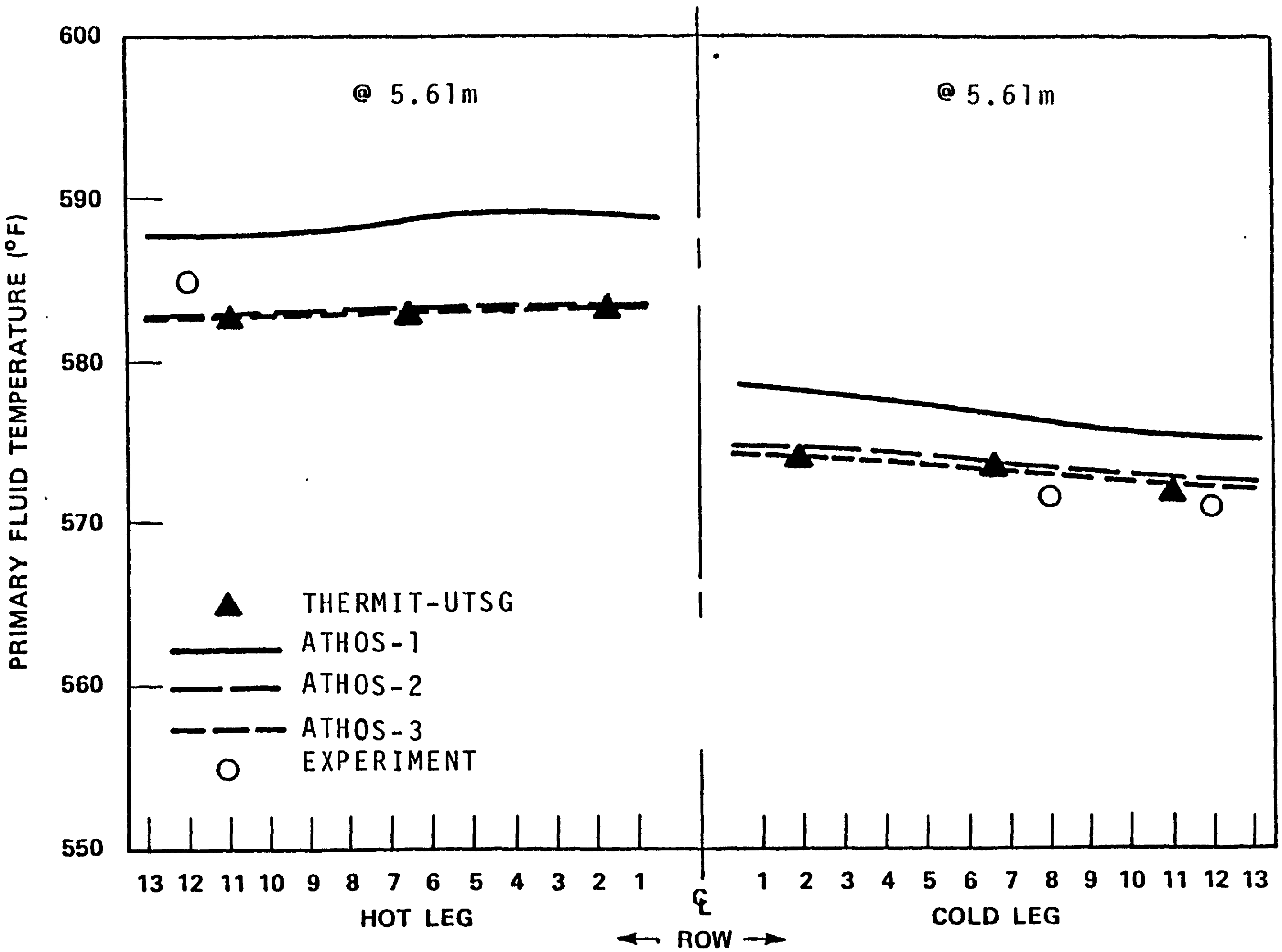


Figure 7.2-6.

Comparison of primary fluid temperatures at
100 percent power, 5.61 m elevation.

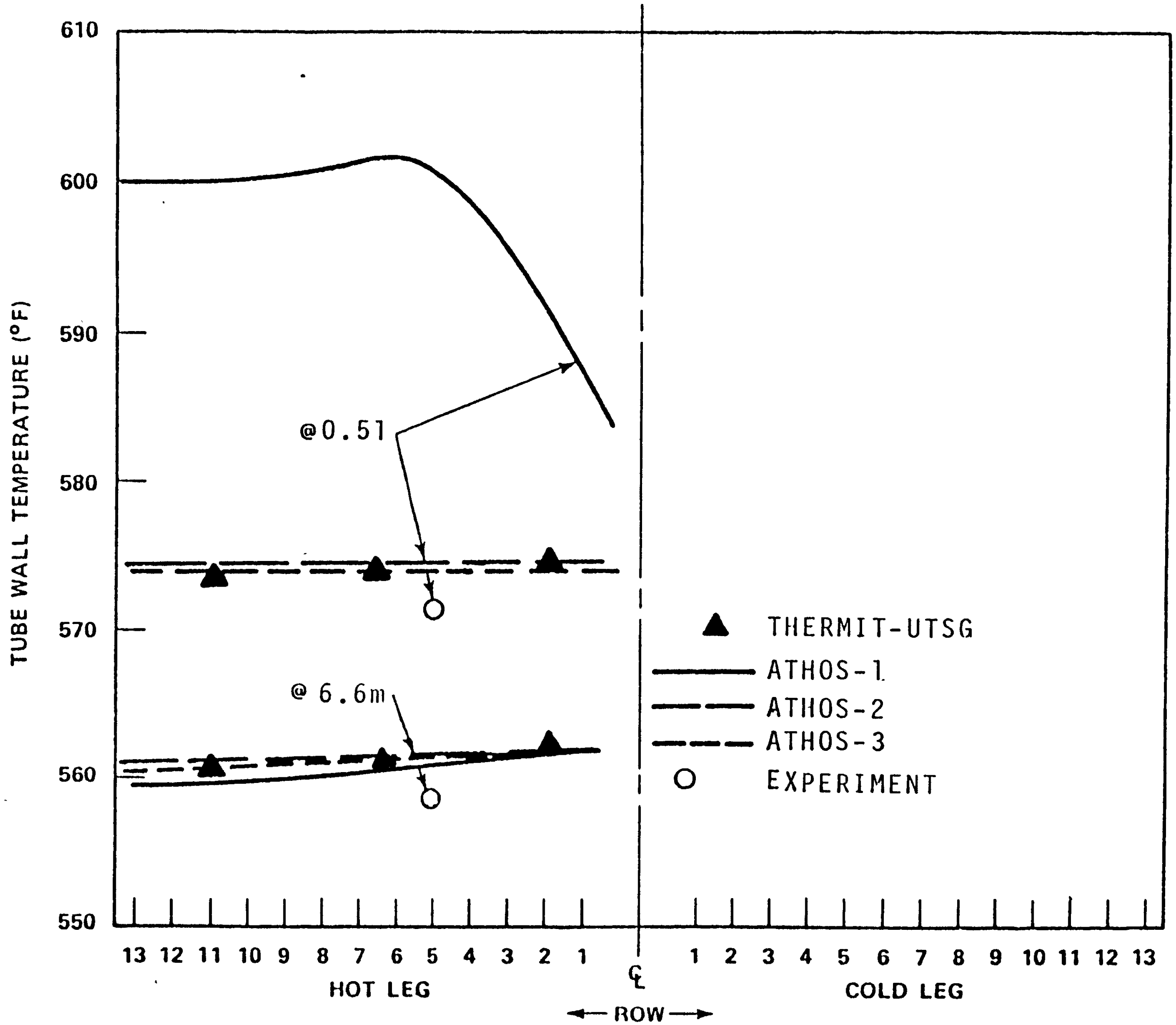


Figure 7.2-7.
Comparison of tube wall temperatures at
100 percent power, 0.51 m and 6.6 m elevations.

of the exact position of the thermocouple hot end is crucial for this calculation because the primary to secondary wall temperature difference at 0.51m is as large as 20 K, and still about half that at 6.6m.

The last set of results for the full power calculation is presented in Figs. 7.2-8 and 7.2-9 showing secondary-side axial temperature distributions in the hot- and cold-sides, respectively. Again, with the exception of ATHOS-1, all calculations are in excellent agreement with the measurements.

7.2.3 Half Power Steady-State Test

The system boundary conditions for the present calculation are listed in Table 4.5-1. These parameters are given again below for convenience:

1. Primary Inlet Temperature,
2. Average Primary Mass Flux,
3. Primary System Pressure,
4. Power Level,
5. Fouling Coefficient (determined from the full power calculation)
6. Downcomer Water Level,
7. Feedwater Temperature

As described in Section 6.4, the primary mass flux is a floating parameter in the steady-state calculation. At full power, the fouling parameter is periodically adjusted so that

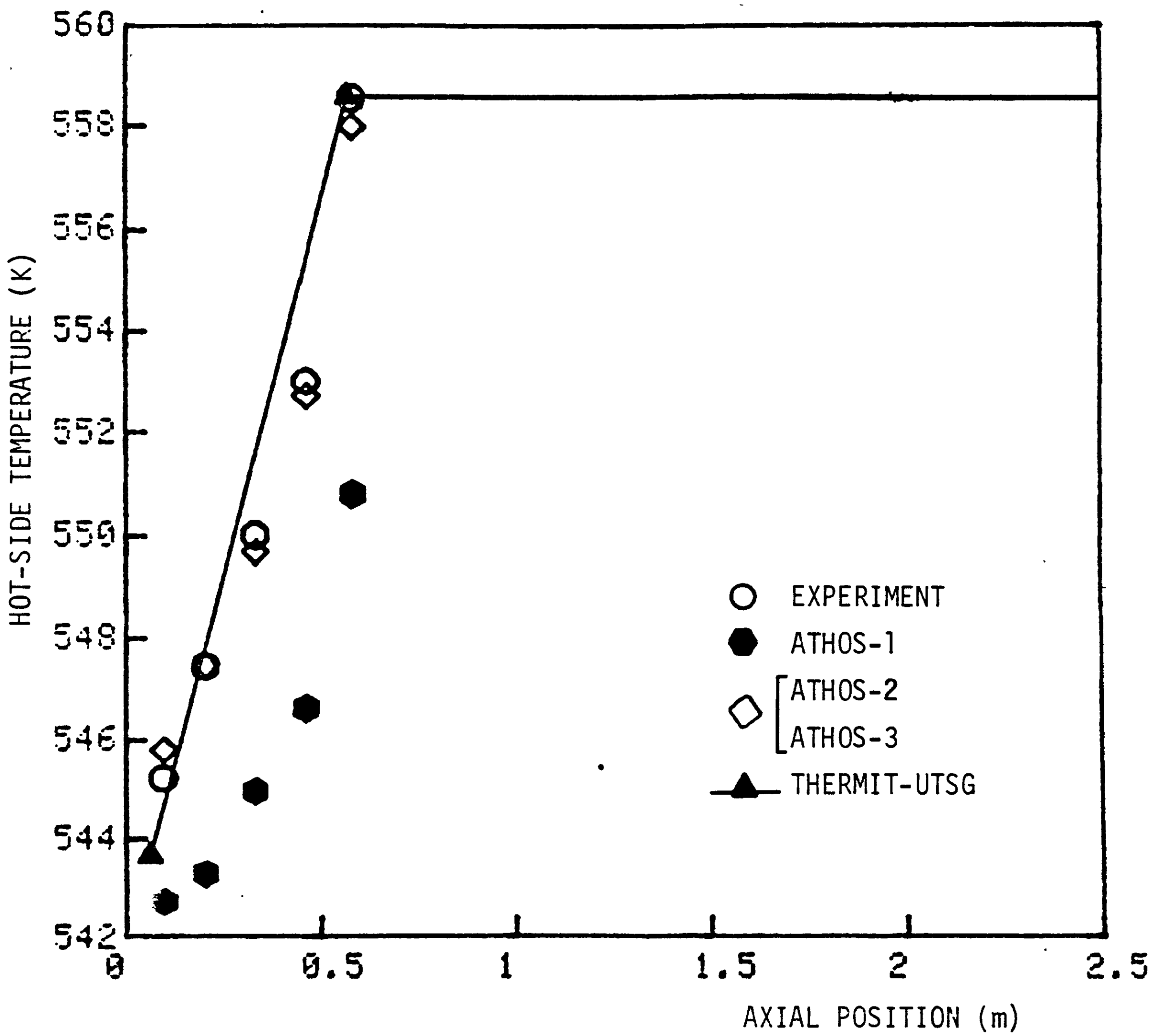


Figure 7.2-8.

Secondary-side axial temperature distribution in the hot-side at full power.

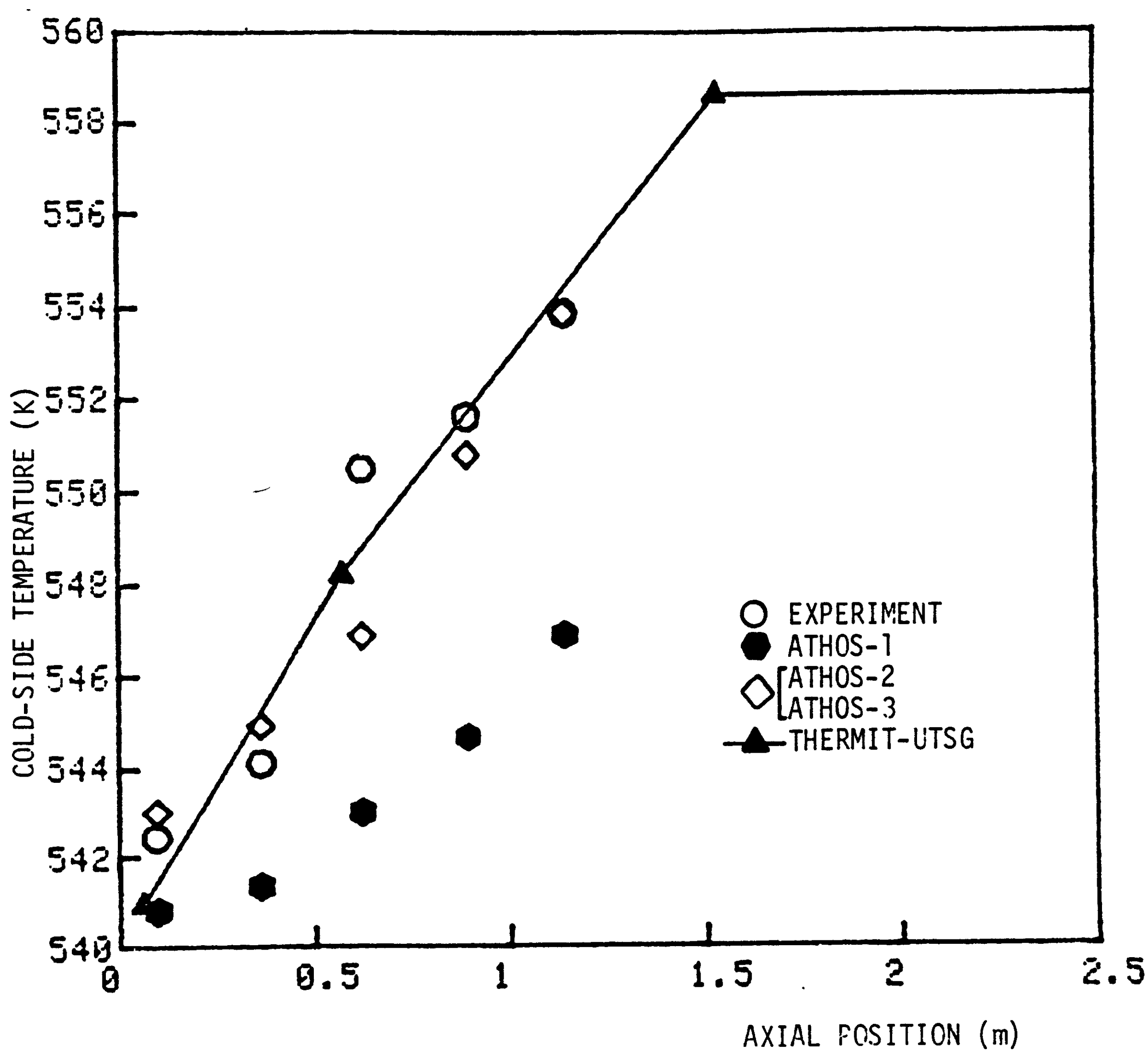


Figure 7.2-9.

Secondary-side axial temperature distribution in the cold-side at full power.

the calculated primary mass flux converges to the desired value. At power levels other than full power, the fouling parameter must be the same as that calculated at 100 percent power. For these cases, the steam dome pressure need not be specified and can be calculated. This is done in a way analogous to the fouling factor calculation previously described. The steam dome pressure is periodically adjusted until this parameter becomes the driving force which the floating mass flux converges to the system value. In the present calculation, the full power fouling parameter is unity, the mass flux converges to the measured value, within tolerance, as evidenced in Table 7.2-6, for the steam dome pressure at its measured value. The significance of this result is analogous to that discussed in Section 7.2.2 in connection with the full power fouling factor. The remaining system boundary conditions are given in Table 7.2-6.

Calculated quantities are compared to measurements in Table 7.2-7. All calculated parameters are in excellent agreement with measured values with the exception of the void fraction. The reasons for the discrepancy remain the same discussed in Section 7.2.2. The tube bundle pressure drop is not underpredicted in this case because at lower powers the gravitational pressure drop dominates and is insensitive to tube inclination. The separator pressure drop is in excellent agreement with the measured value. Since the constants were

Table 7.2-6.

System boundary conditions for MB-2,

50 percent power tests.

PARAMETER	EXPERIMENT (H5)	THERMIT-UTSG
Power Level (MWt)	3.30	3.33
Fouling Parameter	---	1.0
Primary Inlet Temperature (K)	581.3	581.3
Downcomer Water Level (m)(1)	11.24	11.24
Primary Mass Flux ($kg/m^2/s$)(2)	3914 - 3957	3954

(1) Referenced to tube sheet.

(2) Range observed in reproducibility tests.

Table 7.2-7.

Comparison of calculated and measured parameters for MB-2 50 percent power tests.

PARAMETER	EXPERIMENT(1),(H5)	ATHOS(2),(H5)	THERMIT-UTSG
Primary Outlet Temperature	565.4 - 566.8	NA	566.0
Circulation Ratio	6.88	7.20 - 7.80	6.85
Feedwater flow Rate (kg/s)	1.69	1.69	1.69
Riser Quality	0.15	0.08 - 0.13	0.15
Riser Void Fraction	0.61	0.62 - 0.74	0.70
Bundle Pressure Drop (KPa)	38	37 - 39	38
Separator Pressure Drop (KPa)	10 - 20	17 - 22	19
Overall Pressure Drop (KPa)	71	64 - 67	70
Steam Dome Pressure (MPa)	7.24	7.24	7.24

(1) Range observed in reproducibility tests.

(2) Range calculated by the different models given in Table 7.2-3.

only calibrated to give the correct pressure drop at full power, it is clear that the volumetric flow rate at the riser top is being correctly calculated, as indicated by Eq. (5.4-20).

Figures 7.2-10 through 7.2-13 show primary temperature distributions in the hot- and cold-sides. The agreement is excellent in all cases except for the ATHOS-1 combination repeating what was observed at full power.

All the tube wall temperature calculations are shown in Fig. 7.2-14 once again to be in excellent agreement with the measurements, except for ATHOS-1.

Finally, secondary-side axial temperature distributions are presented in Figs. 7.2-15 and 7.2-16 for hot- and cold-sides respectively. Once more, with the exception of the ATHOS-1 combination, all calculated values are in excellent agreement with the measurements.

7.3 Global Parameter Tests

7.3.1 Background

The time dependent calculated model response of global steam generator parameters includes:

1. downcomer water level,
2. primary outlet temperature, and
3. steam flow rate.

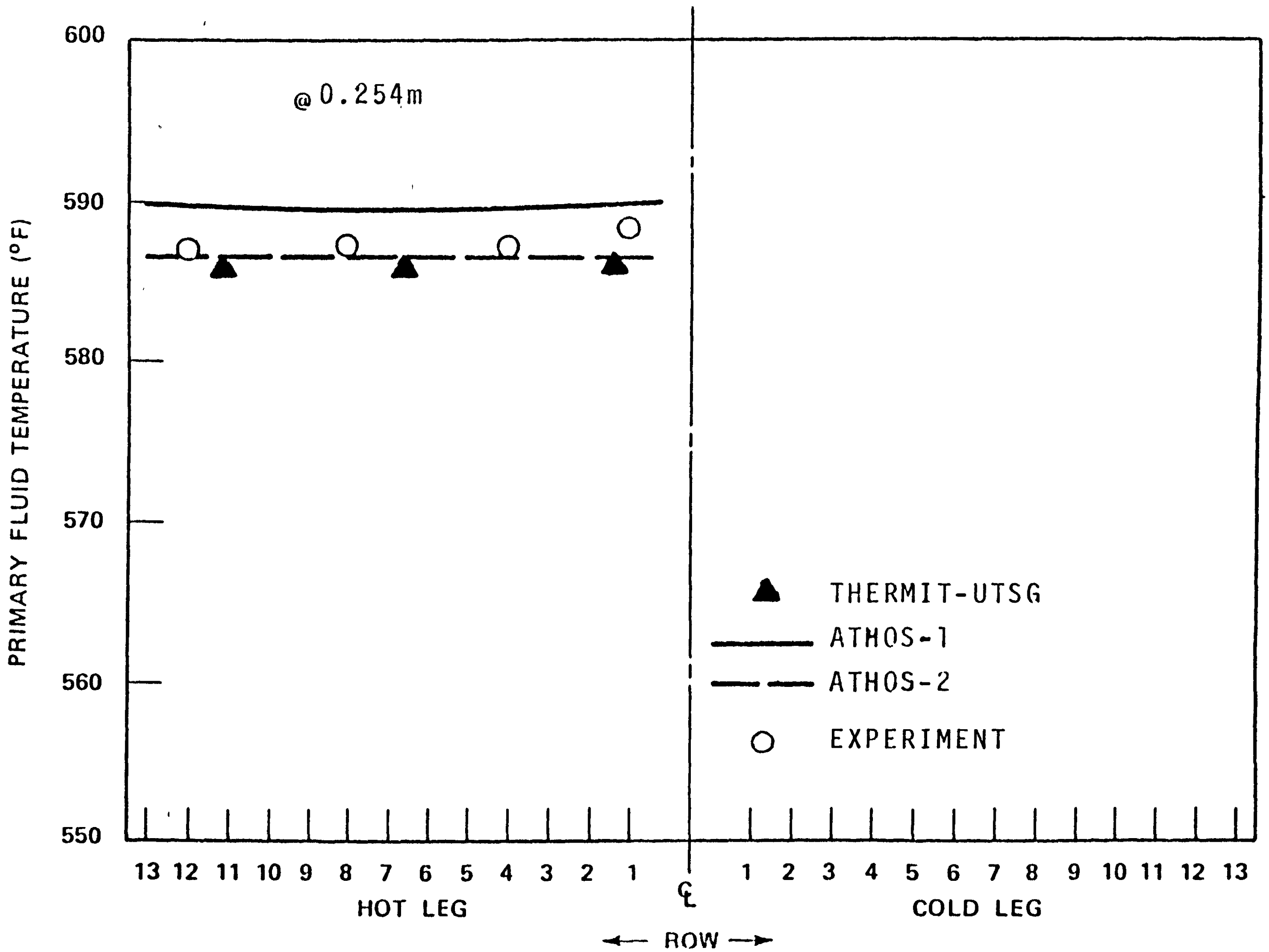


Figure 7.2-10.

Comparison of primary fluid temperatures at 50 percent power, 0.254 m elevation.

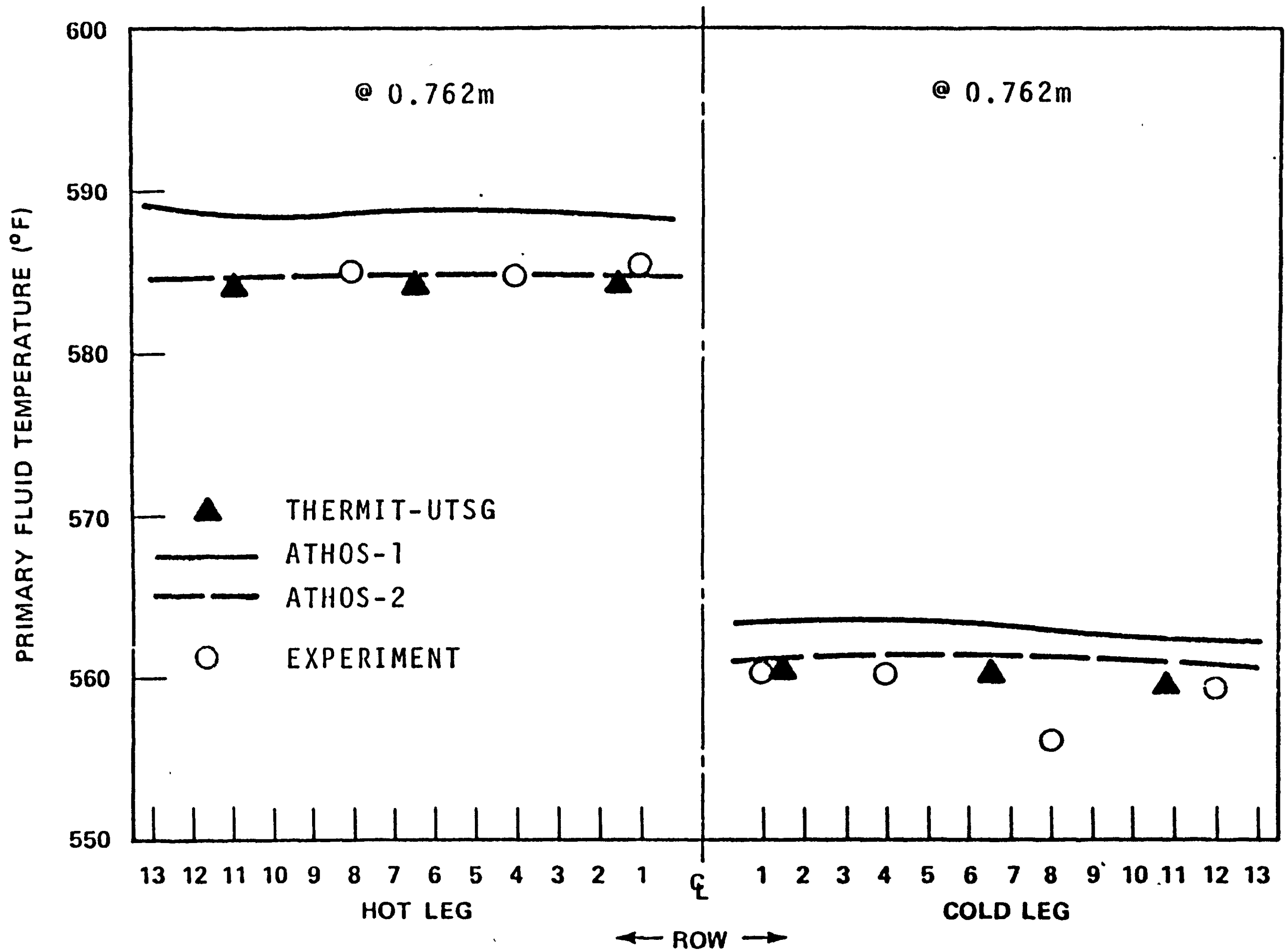


Figure 7.2-11.

Comparison of primary fluid temperatures at 50 percent power, 0.762 m elevation.

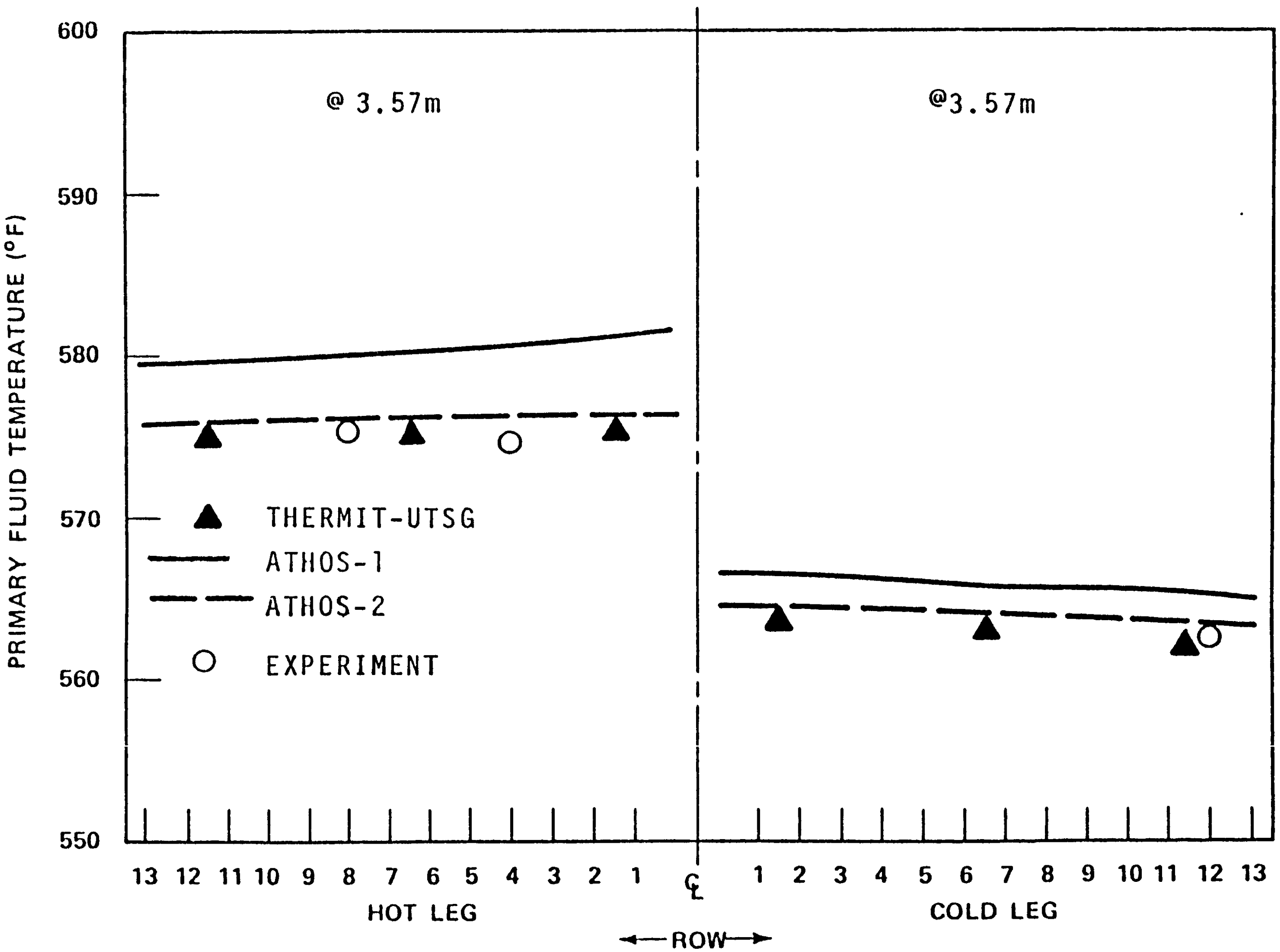


Figure 7.2-12.

Comparison of primary fluid temperatures at
50 percent power, 3.57 m elevation.

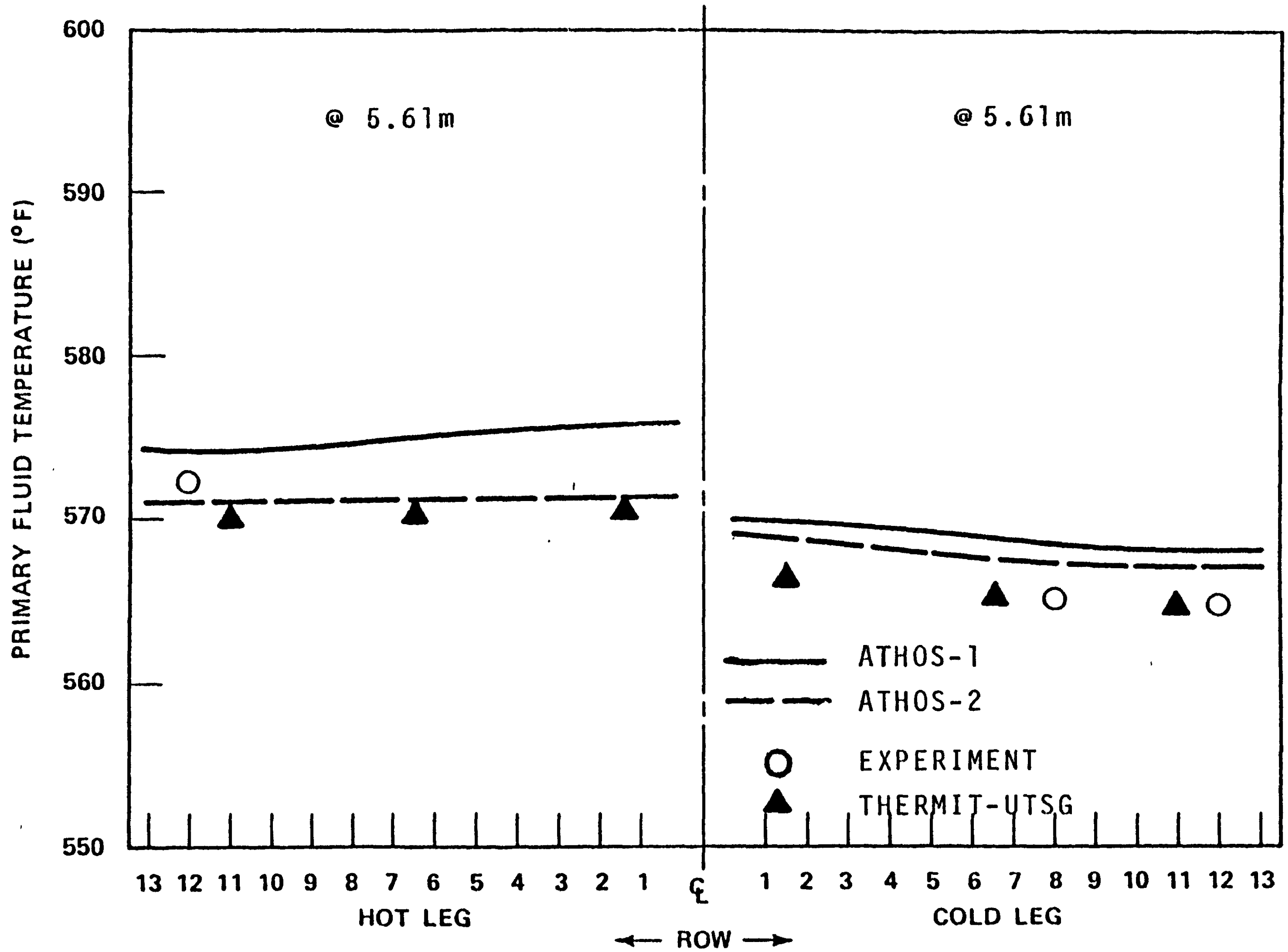


Figure 7.2-13.

Comparison of primary fluid temperatures at 50 percent power, 5.61 m elevation.

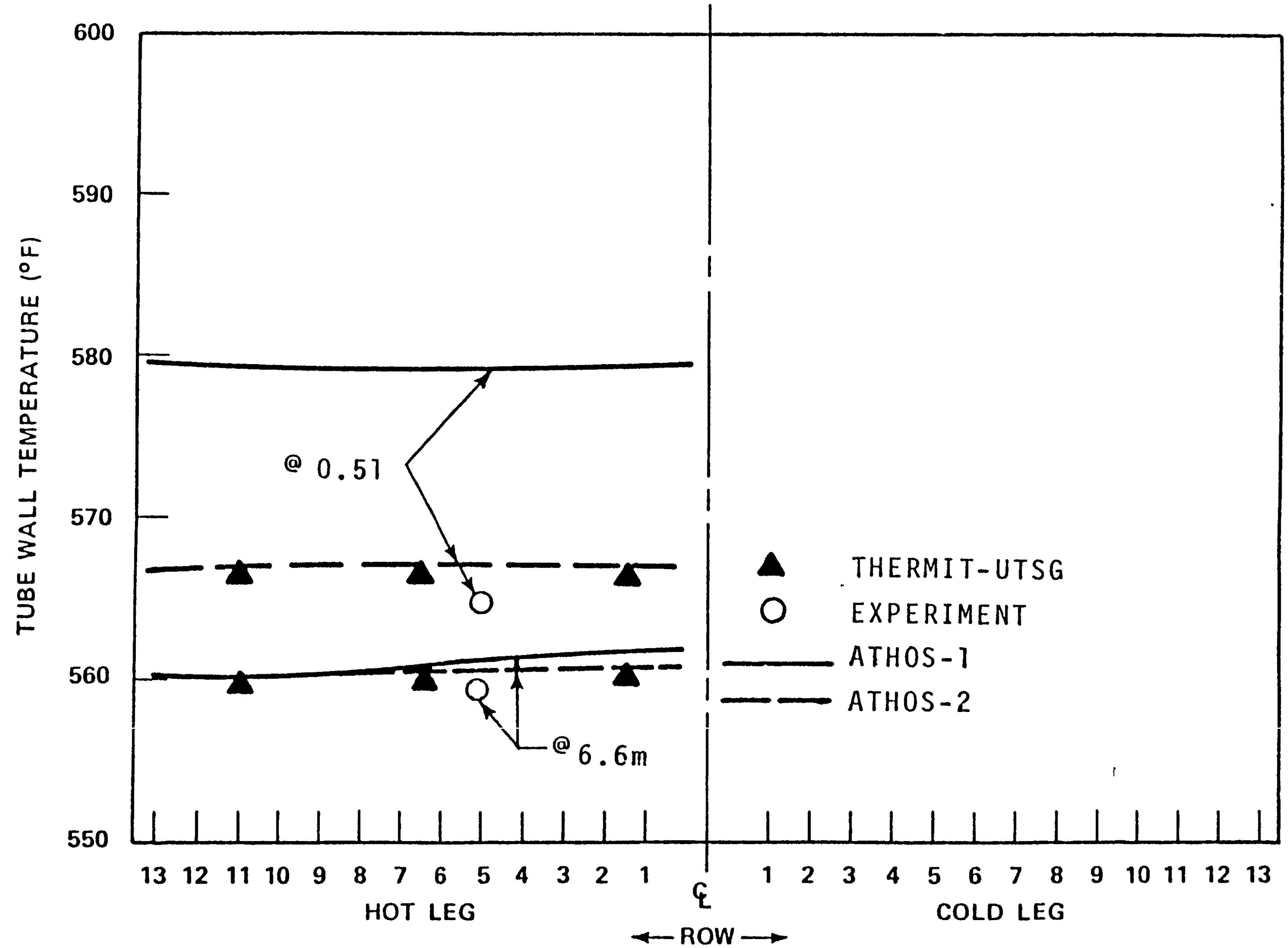


Figure 7.2-14.

Comparison of tube wall temperatures at
50 percent power, 0.51 m and 6.6 m elevation.

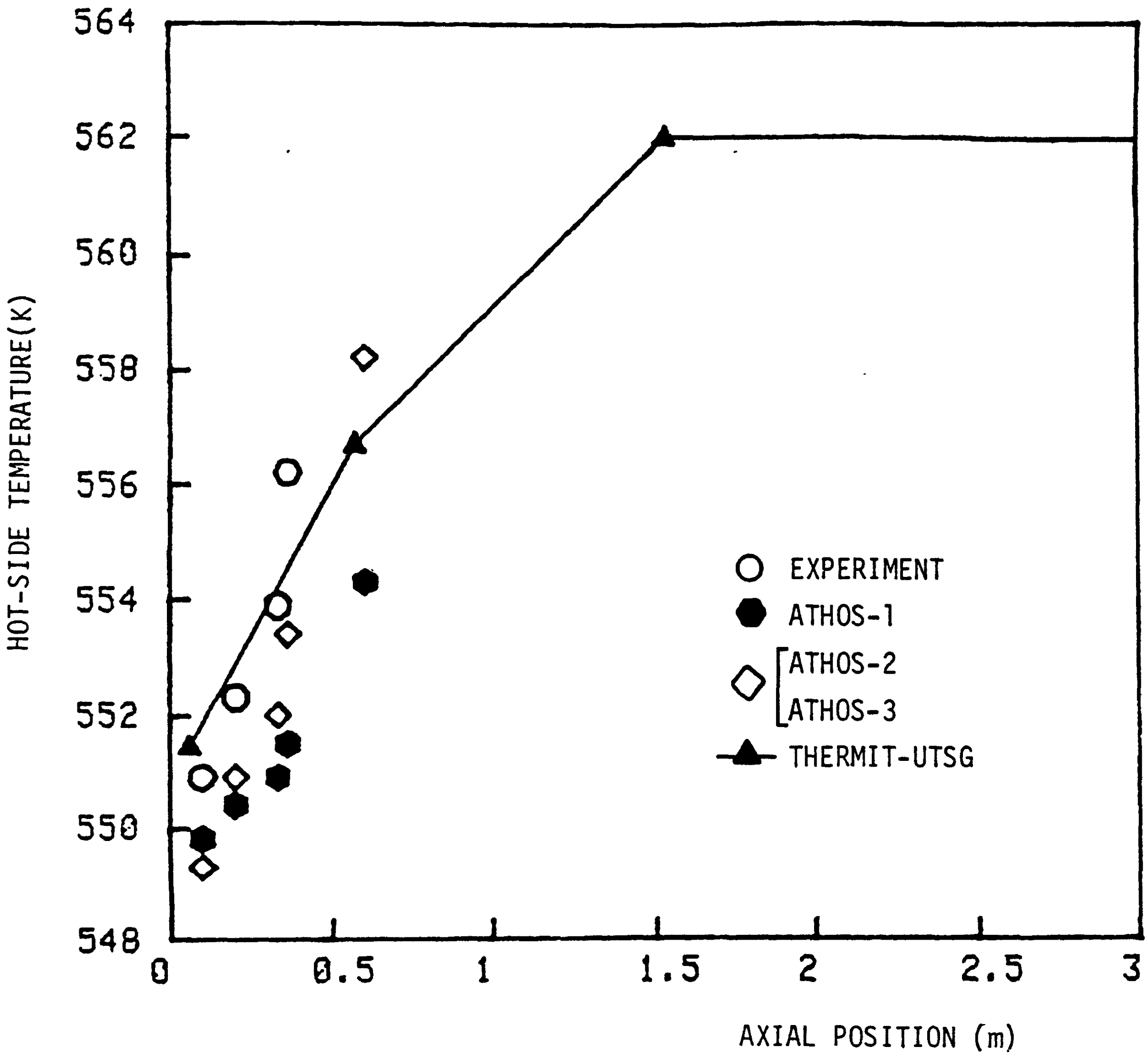


Figure 7.2-15.

Secondary-side axial temperature distribution in the hot-side at half power.

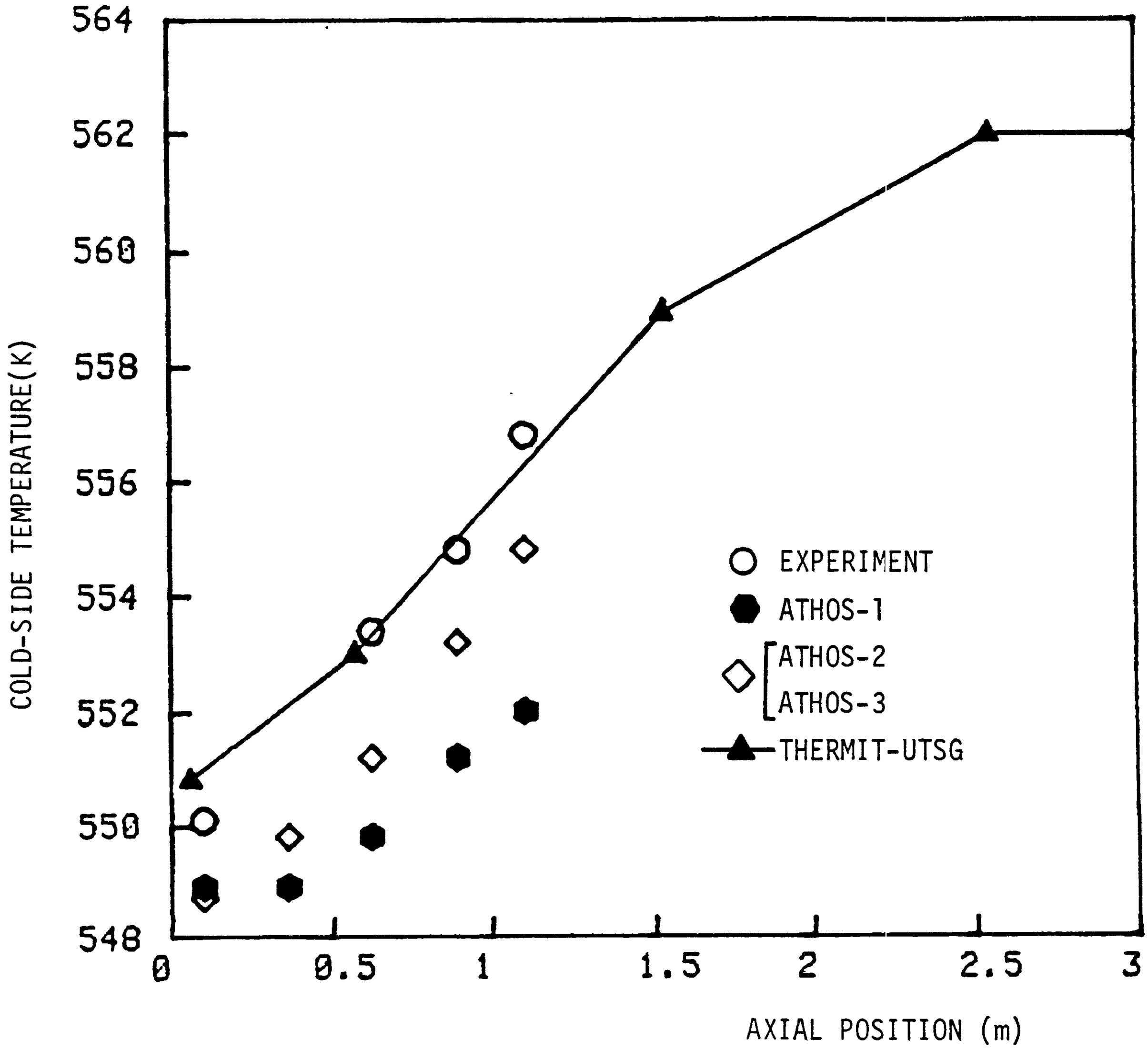


Figure 7.2-16.

Secondary-side axial temperature distribution in the cold-side at half power.

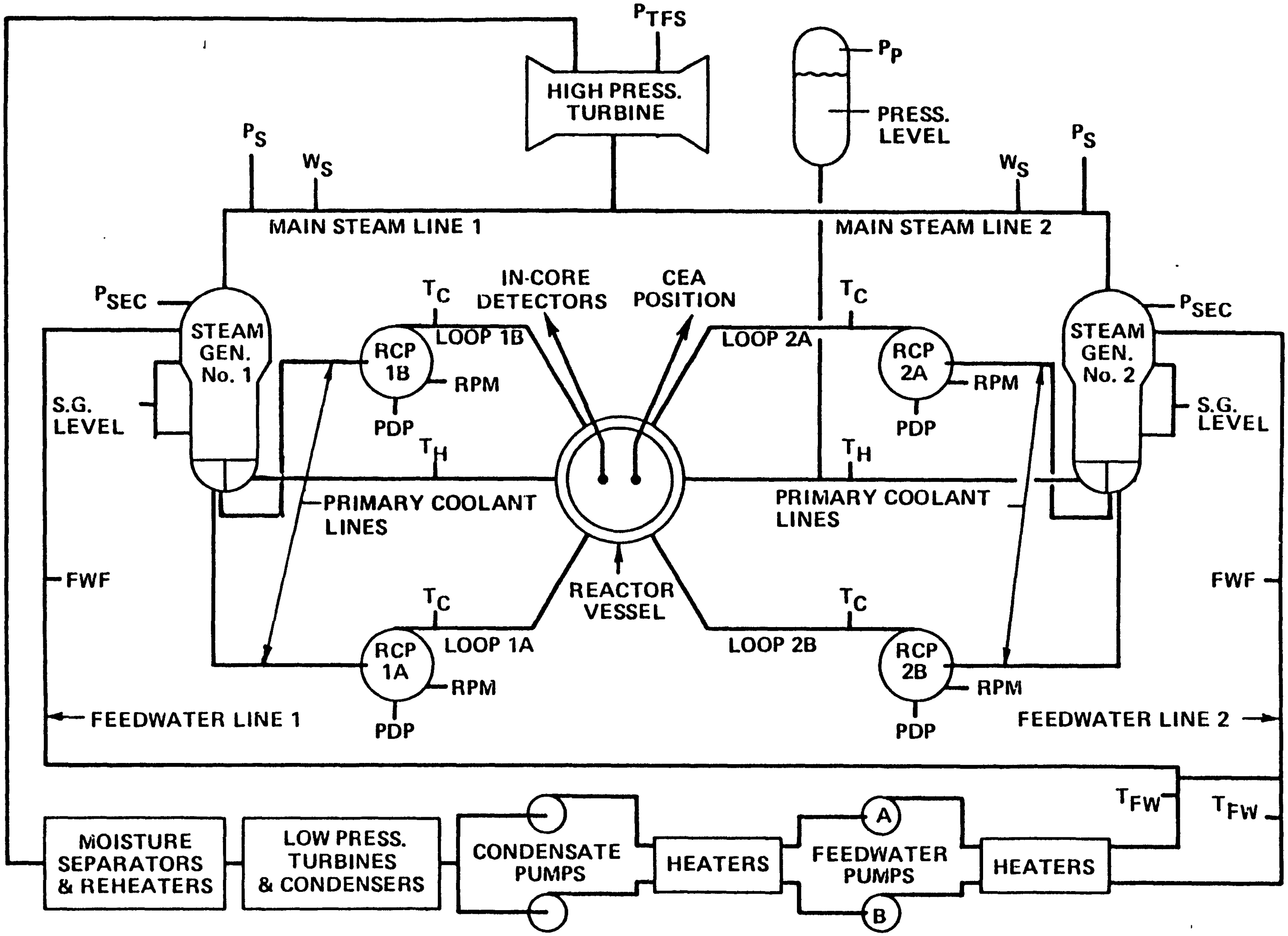


Figure 7.3-1. Arkansas Nuclear One - Unit 2. Schematic component layout (Ref. G2).

These quantities are compared in this section to measurements carried out at the Arkansas Nuclear One - Unit 2 (ANO-2) power plant for the tests listed in the first line of Table 7.1-2.

The input parameters for each test are the transient system boundary conditions of Table 4.5-1, viz.:

1. primary inlet temperature,
2. average primary mass flux,
3. primary system pressure,
4. steam dome pressure,
5. feedwater temperature, and
6. feedwater flow rate,

which are available from measurements of these quantities for the same tests.

Arkansas Nuclear One - Unit 2 (ANO-2) is a Combustion Engineering nuclear steam supply system (NSSS) with a rated thermal power of 2815 MWt. The primary loop consists of the reactor, pressurizer, two steam generators, and four reactor coolant pumps, with one hot-leg and two cold-legs per steam generator. The arrangement is shown in Fig. 7.3-1. The nominal operating conditions for the steam generators are given in Table 7.3-1.

The steam generators are Combustion Engineering twin units of the series 67 type. Their interior arrangement is that of Fig. 4.2-1 where the actual dimensions have been removed for

Table 7.3-1.

Arkansas Nuclear One - Unit 2 Steam Generator nominal design operating parameters.

Parameter	Value
Primary mass flux	4138 kg/m^2,s
Primary pressure	15.5 MPa
Steam dome pressure	6.2 MPa
Steam flow rate	797 kg/s
Feedwater temperature	506 K
Downcomer water level	10.4(*) m
Power level	1408 MWt

(*) Referenced to tube sheet.

clarity. The dimensions upon which the present calculations are based are quoted in Fig. 3-5 of Ref. (G2). The axial mesh scheme used for the application of the present model is that of Fig. 4.2-2 with fifteen axial levels where levels one and fifteen correspond to the boundary cells and are shown hatched in Fig. 4.2-2. The horizontal mesh scheme or channel layout is that of Fig. 4.2-3. Since the actual unit is cylindrical and the present model uses Cartesian coordinates, the main guideline in establishing x and y dimensions based on Fig. 4.2-3 is that axial flow areas be preserved. The primary model arrangement is that of Fig. 4.4-1 with three tube banks distributed as shown. The geometric input for this case as well as the initial guesses to be used in the quest for steady-state operating conditions are given in Appendix F.

A small but basic inconsistency in the data of Ref. (S8) had to be resolved for the determination of the steady-state conditions which preceded the initiation of each transient. The power calculated using the given primary mass flow rate and enthalpy drop did not match the stated power. This was resolved by postulating the problem was in the flow rate measurement. The measured values for power and enthalpy drop were assumed to be accurate. Justification for this course of action on the basis of the experiment is not necessary because the magnitude of the discrepancy is small. In reality, the inconsistency is most certainly due to errors in all parameters involved. However, when assuming all the error to be

Table 7.3-2.

Selected [1] initial conditions for the global parameter calculations.

CASE NO.	TEST	SELECTED INITIAL CONDITIONS (PERCENT NOMINAL)		
		POWER	PRIMARY MASS FLUX	DOME PRESSURE
1.	FLCEAD[2]: SG-1	49	98	104
2.	FLCEAD: SG-2	49	99	104
3.	TT[3]: SG-1	98	101	101
4.	TT: SG-2	98	99	101
5.	LOF[4]: SG-1	82	108	103

(1) Detailed initial conditions are given in the section dealing with each test.

(2) Full Length Control Element Assembly Drop

(3) Turbine Trip

(4) Loss of (Primary) Flow

concentrated in the flow rate, that parameter varied from 98 to 108 percent nominal, for all cases, as shown in Table 7.3-2. This variation about the nominal was felt to be small enough to allow the calculation to proceed as if all experimental error were concentrated in the flow rate. The advantage of this assumption is that the steady state calculation can be done in one run. This is because the primary flow rate is a floating parameter in the steady-state calculation, as pointed out in Chapter 6 and in the previous section. The values of Table 7.3-2 are those to which the mass flux converged for the given power and inlet temperature assuming no fouling i.e. a fouling parameter equal to one.

A small overprediction (less than 0.3 K for the worst case) of the measured outlet temperatures was observed. This discrepancy is well within the experimental error and was therefore ignored. Reference (S8) points out that calibration errors of plant temperature instrumentation are on the order of ± 1.1 K (±2 R) on the absolute temperature. It was thus possible to calculate all the initial conditions without utilizing a fouling coefficient. The statement should not be interpreted as a suggestion that fouling was not present, but merely that it was not necessary to activate the model's option to include the term in order to calculate the initial steady-state within experimental uncertainty. The remark is significant because in any computer model, a fouling factor can be used to accommodate other sources of discrepancy than tube fouling per se. This is a conclusion which can easily be drawn from the discussion presented in Section 4.5. In fact, it would have been perfectly possible and it was actually done in one unpublished case, to utilize the fouling parameter to eliminate the outlet temperature overprediction altogether. This course of action also led to lower primary mass fluxes, a tendency which would place the mean of the span on that parameter, which is 103 percent according to the values in Table 7.3-2, closer to the nominal value. This was not done for any of the cases presented in this work for three reasons.

First, it was felt to be an excessive refinement, particularly in view of the cost in computer time.

Second, the fouling parameter would have to be activated in the direction of enhancing the heat transfer. This means it would not be representing fouling at all but some other source of discrepancy. The culprit here is the heat transfer area. In the present model, only local tube bank heat transfer areas are required as input. These areas must be calculated based on approximate tube bank lengths, determined as described in Appendix D. Unfortunately the sum of all the calculated local areas (7965 m^2) was not checked against the actual total area (~8800 m^2) at the time of the ANO-2 runs. The code input description given in Appendix F now prompts the user to make minor adjustments in the local areas in such a way that their sum matches the known total area. This was done for the MB-2 tests.

Third, and most important, the observed ability to predict all the steady-state initial conditions to within experimental error without resorting to any artificial adjustment parameter is a significant demonstration of the soundness of the overall integrated model.

The feedwater temperature is not given in the test data of Ref. (S8). The steady state feedwater temperature was inferred from the heat balance on the secondary side of the steam generator given by Eq. (4.5-2) which can be written in the form:

$$h_{fd} = h_s - \frac{Q_B}{W_s} \qquad (7.3\text{-}1)$$

where the nomenclature is given in connection with Eq. (4.5-2). All the quantities on the right hand side of Eq. (7.3-1) are given so that h_{fd} can be calculated. The feedwater temperature is then easily obtained using a steam table.

Finally, in plots of primary outlet temperature versus time there are two curves corresponding to measurements and one to calculations. The two measured curves correspond to temperature data gathered for the two cold-legs which can be seen in Fig. 7.3-1. If the fluid in the steam generator outlet plenum is well mixed, then the two cold leg temperatures should be the same. Steady-state differences in the measured temperatures of the two cold legs are most certainly due to calibration errors which are on the order of ± 1.1 K(±2 R), (S8). Differences in transient measurements incorporate the effect of different resistance temperature detector (RTD) response times as well. This point is discussed in Section 6.4. Obviously, the calculated primary outlet plenum temperature is unique, i.e. only one temperature is calculated by the code.

It is pointed out in Ref. (S8) that time constants for the resistance temperature detectors (RTD) used to measure primary inlet (hot-leg) and outlet (cold-leg) temperatures are:

$$\tau_{hot} = 4.753 \text{ s}$$

and

$$\tau_{cold} = 4.898 \text{ s.}$$

Clearly these time constants are not negligible in comparison with the times over which significant transient changes are observed. Thus, the procedure described in Section 6.4 was activated in all the transient tests in this chapter in order to appropriately account for RTD lag.

7.3.2 Full Length Control Element Assembly Drop Test

This test was initiated from the steady-state conditions given in Table 7.3-3 for SG-1 and in Table 7.3-4 for SG-2. The power level was 49 percent of the nominal full power level in both cases. Immediately following the CEA drop, the turbine load limit was adjusted to match the new average core power level. The CEA dropped was the nearest aligned with the SG-2 hot-leg. That CEA was deliberately selected in order to maximize the asymmetry of the response of both steam generators. The hot- and cold-leg temperatures associated with SG-2 drop at a faster rate than those associated with SG-1. Similar asymmetries occur for the feedwater and main steam flow rates.

There is some discrepancy in the feedwater and steam flow rate data of Ref. (S8). The steady-state measured feedwater flow rate is greater than the measured steam flow rate by 2.7 percent in SG-1 and 3.7 percent in SG-2. This is within instrumentation calibration uncertainty. As pointed out in Ref. (S8), the overall uncertainty of the data acquisition system was primarily due to the delay times associated with plant instrumentation (as much as 2 percent) and was estimated to be on the order of 5 percent of the initial steady-state parameter values, all causes included. The uncertainty attributed to instrument delay time cannot cause the mismatch in the reading of steam and feed flowrate. However, there remains a 3 percent uncertainty per parameter when delay time errors are discounted. These errors are stated in Ref. (S8) to be due to:

1. sensor calibration (1 percent),
2. recording system calibration (1 percent), and
3. signal conditioning amplifier linearity and drift and analog to digital conversion (1 percent).

Since these uncertainties affect both steam and feedwater flow rate measurements, measured values of both quantities could deviate from each other by as much as 6 percent in steady state due to the above listed uncertainties alone. The observed inconsistencies were resolved by lowering the whole feedwater flow rate curve by the appropriate amount to match the steam flow value at time zero.

The primary system pressure varied little throughout this test. Its behavior, shown in Fig. 7.3-2, was input identically for both steam generators. The primary mass flux did not vary and the feedwater temperature remained constant as well.

7.3.2.1 Steam Generator 1

The conditions preceding the full length CEA drop for SG-1 are given in Table 7.3-3. The input used as forcing functions specifically for SG-1 are given in Fig. 7.3-3. For the reasons explained in the previous section, the feedwater flow rate was biased by - 10 kg/s (-2.7 percent of the steam flow) for the duration of the event.

Table 7.3-3.

Full length CEA drop:
Initial conditions for SG-1.

Parameter	Value
Bundle power	695.5 MWt
Steam dome pressure	6.42 MPa
Steam flow	365 kg/s
Feedwater temperature(1)	478.2 K
Primary inlet temperature	575.8 K
Primary outlet temperature	559.5 K
Primary system pressure	15.47 MPa
Primary mass flux	4054 $kg/m^2/s$
Downcomer water level(2)	10.4 m (70 percent)
Downcomer flow	2910 kg/s
Circulation ratio	8.0
Fouling parameter	1.0

(1) Calculated using Eq. (7.3-1).
(2) Distance is referenced to tube sheet. Percentage is of instrument span (4.20 m) where lower instrument tap is 7.45 m above tube sheet.

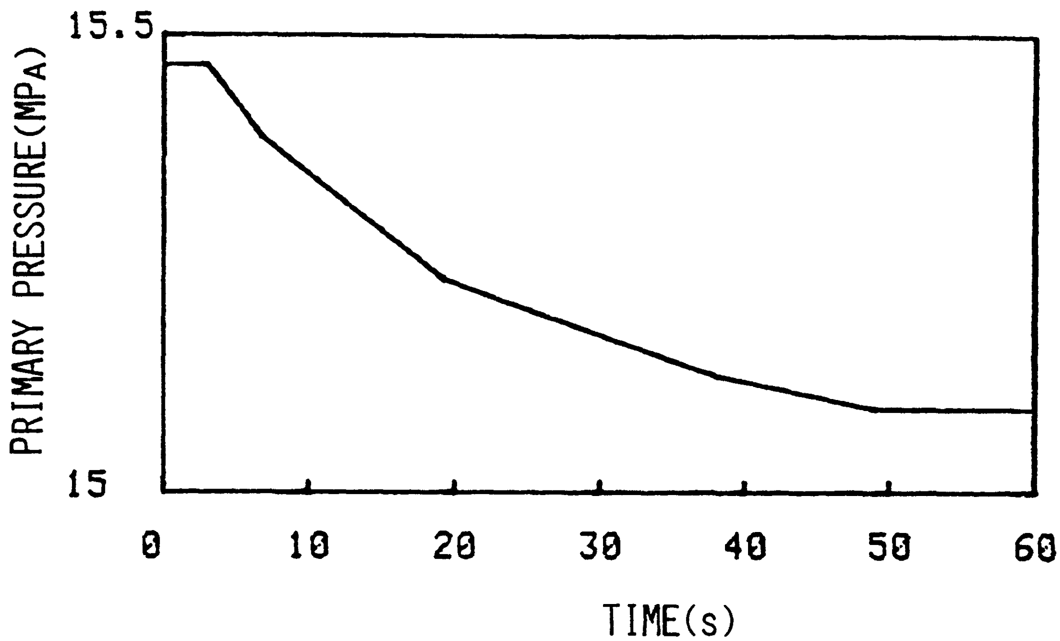


Figure 7.3-2

CEA drop test.
Input common to SG-1 and SG-2.

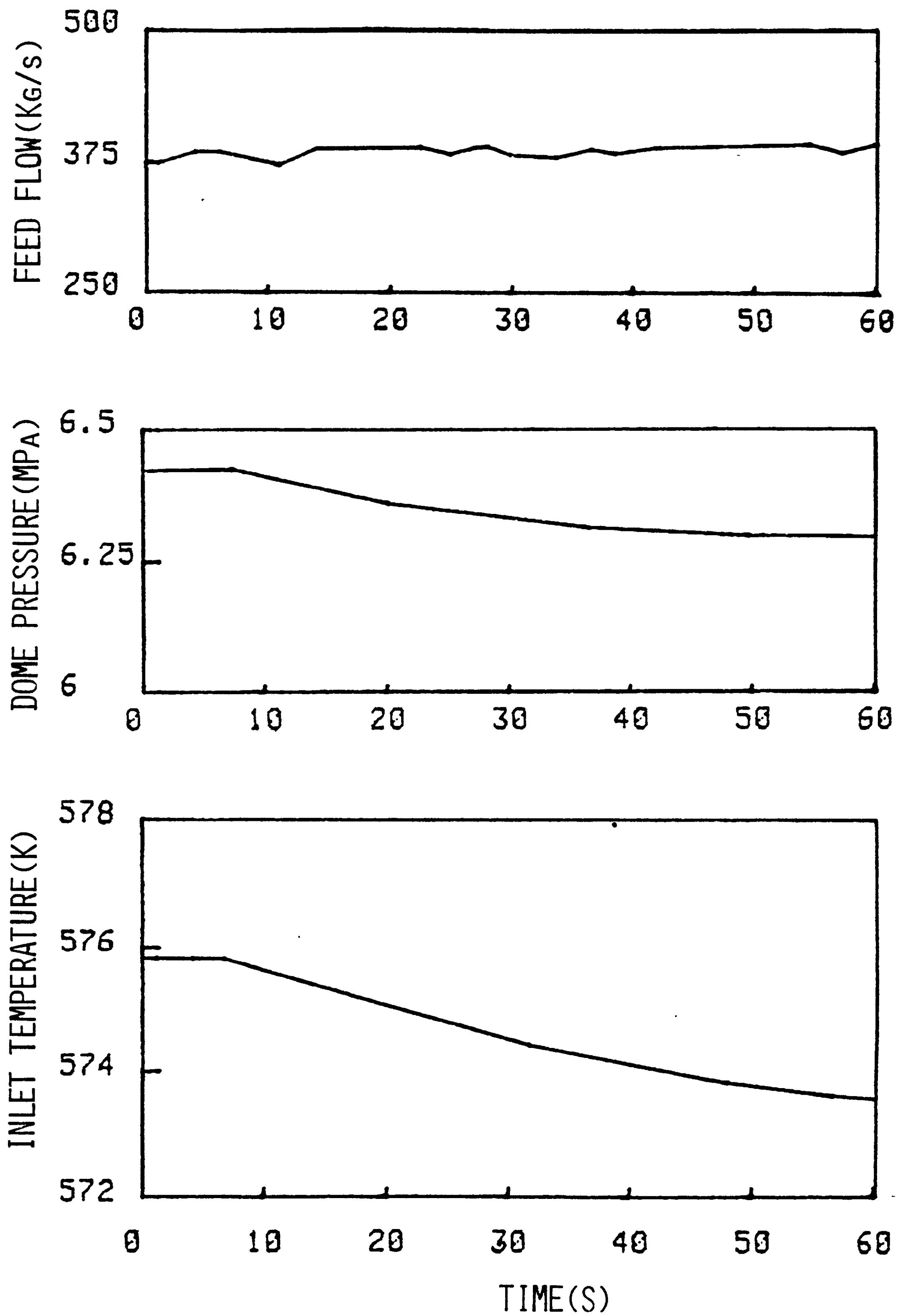


Figure 7.3-3.

CEA drop. Input for SG-1.

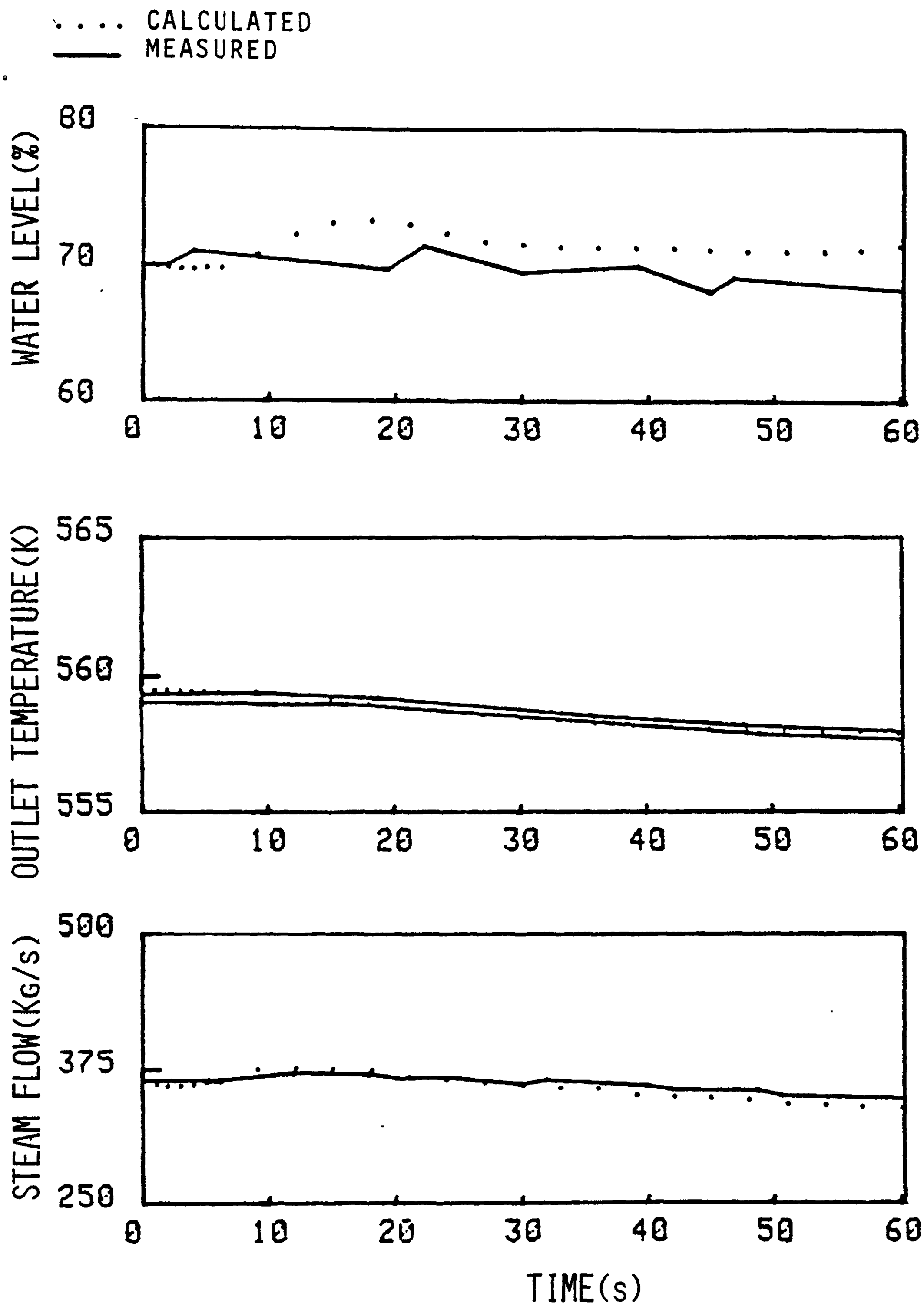


Figure 7.3-4.

CEA drop. Response of SG-1.

The response of SG-1 is shown in Fig. 7.3-4. Both the calculated outlet temperature and steam flow are in excellent agreement with the data. The water level is in good agreement as well.

7.3.2.2 Steam Generator 2

The initial conditions for SG-2 are given in Table 7.3-4. The input for SG-2 is given in Fig. 7.3-5. The feedwater flow rate is biased by -14 Kg/s (-3.7 percent of the steady-state steam flow value) throughout the transient for reasons given in Section 7.3.2.

The response of SG-2 is given in Fig. 7.3-6. The calculated primary outlet temperature is in excellent agreement with the data. The calculated steam flow is slightly above but in good agreement with the data. The calculated and measured downcomer water level are within 2 percent of each other (~0.09 m).

7.3.3 Turbine Trip Test

This test was initiated from the steady-state conditions given in Tables 7.3-5 and 7.3-6 for SG-1 and SG-2, respectively. The initial power level was 98 percent of the full power in both cases. The sequence of events for the turbine trip test is the following. At time zero the main turbine is manually tripped. At two seconds into the transient, the atmospheric dump valves and the turbine bypass

Table 7.3-4.

Full length CEA drop:
Initial conditions for SG-2.

Parameter	Value
Bundle power	695.5 MWt
Steam dome pressure	6.42 MPa
Steam flow	371 kg/s
Feedwater temperature(1)	485.3 K
Primary inlet temperature	576.4 K
Primary outlet temperature	559.2 K
Primary system pressure	15.47 MPa
Primary mass flux	4092 $kg/m^2/s$
Downcomer water level(2)	10.3 m (70 percent)
Downcomer flow	2915 Kg/s
Circulation ratio	7.9
Fouling parameter	1.0

(1) Calculated using Eq. (7.3-1).

(2) Distance is referenced to tube sheet. Percentage is of instrument span (4.20 m) where lower instrument tap is 7.45 m above tube sheet.

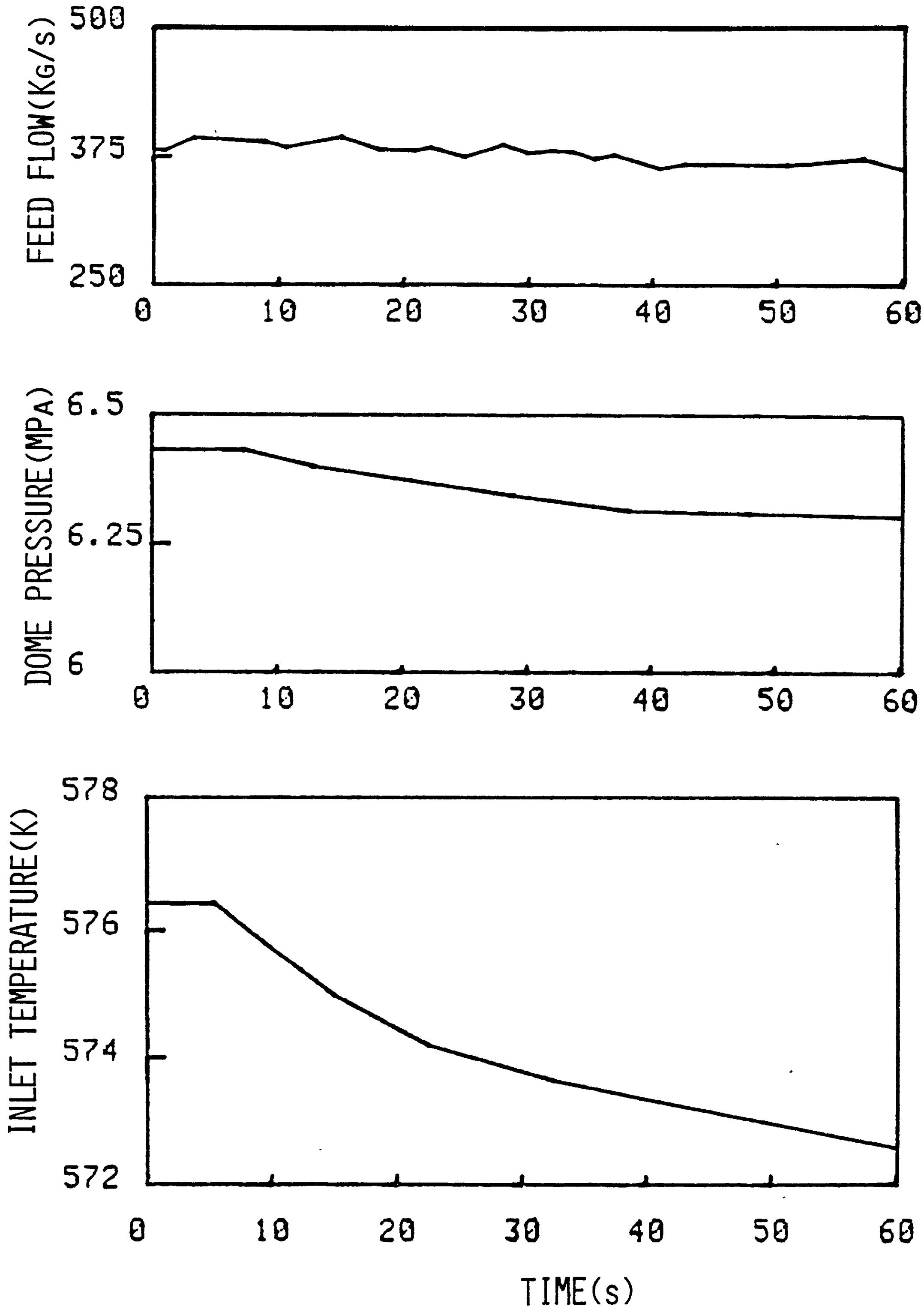


Figure 7.3-5.

CEA drop. Input for SG-2.

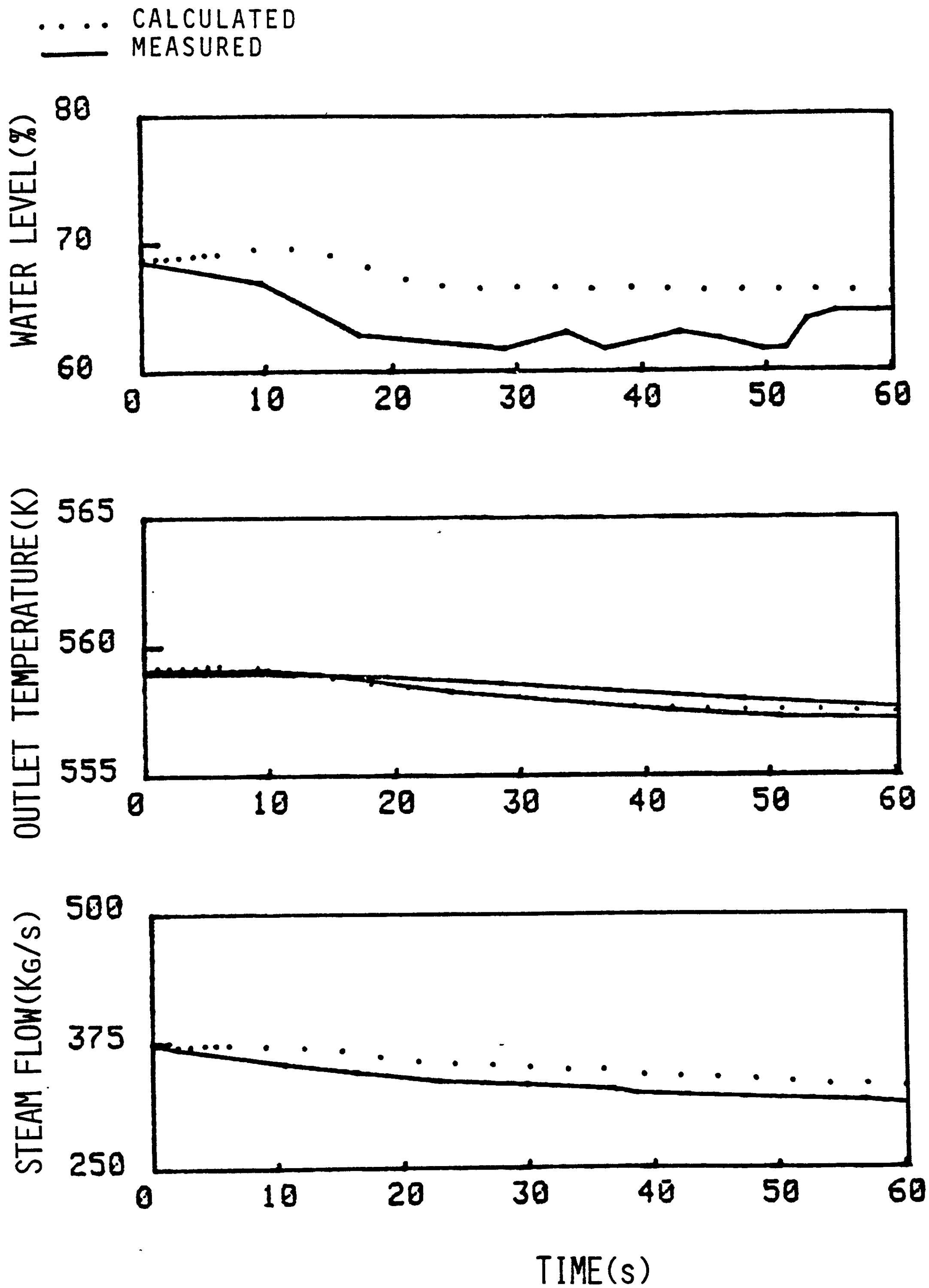


Figure 7.3-6.

CEA drop. Response of SG-2.

valves receive quick open signals and are fully open one second later. The pressurizer spray valve opens at two seconds. At 6.1 seconds the reactor trips on a lower steam generator water level signal and an emergency feedwater actuation signal is simultaneously generated. The emergency feedwater is drawn from the condensate storage tank, which is maintained at the ambient temperature of 300 K (Ref. (A2)). Unfortunately the behavior of the feedwater temperature with time was not measured. Thus, an assumed behavior had to be developed for input to the calculations. The sensitivity of calculated model response to this parameter was informally tested and found to be small. Qualitatively, the calculated model responses remained insensitive. The sensitivity test was carried out using as a base case no feedwater temperature variation at all. The calculated results shown in this section, however, are based on the more reasonable assumption that the feedwater temperature ramps down linearly from the normal operating value to the temperature of the emergency feedwater. The ramping is assumed to take place over a period of three seconds and to be initiated one second after the emergency feedwater starts to flow. Continuing with the sequence of events, at 21 seconds the bypass valves begin to close while an atmospheric dump valve remains open. The bypass valves are fully closed at 29 seconds. No other significant events occur over the time period of the simulation.

In this test, three input parameters are common to the calculation corresponding to the two steam generators. These parameters are: feedwater temperature, which had to be assumed as described above; the primary mass flux, which remained constant at the initial value; and the primary system pressure, whose behavior is shown in Fig. 7.3-7.

7.3.3.1 Steam Generator 1

The initial conditions for SG-1 are given in Table 7.3-5. The specific input used as forcing functions for the calculations pertaining to SG-1 are given in Fig. 7.3-8. As explained in Section 7.3.2, the feedwater flow rate input is adjusted to maintain consistency with the steam flow rate at time zero. In the present case, the correction corresponded to a positive bias of 24 Kg/s (3 percent of the steam flow), in effect for the duration of the transient.

Figure 7.3-9 exhibits the response of Steam Generator 1. All calculated parameters are in good agreement with the data.

7.3.3.2 Steam Generator 2

The conditions prevailing in steam Generator 2 prior to the initiation of the turbine trip test are given in Table 7.3-6. The forcing functions for this test specific for SG-2 are given in Fig. 7.3-10. As described in Section 7.3.2, the feedwater flow rate input is corrected so that it matches the

Table 7.3-5.

Turbine Trip: Initial conditions for SG-1.

Parameter	Value
Bundle power	1382 MWt
Steam dome pressure	6.27 MPa
Steam flow	806 kg/s
Feedwater temperature(1)	519.2 K
Primary inlet temperature	594.0 K
Primary outlet temperature	562.9 K
Primary system pressure	15.51 MPa
Primary mass flux	4188 $kg/m^2/s$
Downcomer water level(2)	10.5 m (72 percent)
Downcomer flow	3270 kg/s
Circulation ratio	4.1
Fouling parameter	1.0

(1) Calculated using Eq. (7.3-1).
(2) Distance is referenced to tube sheet. Percentage is of instrument span (4.20 m) where lower instrument tap is 7.45 m above tube sheet.

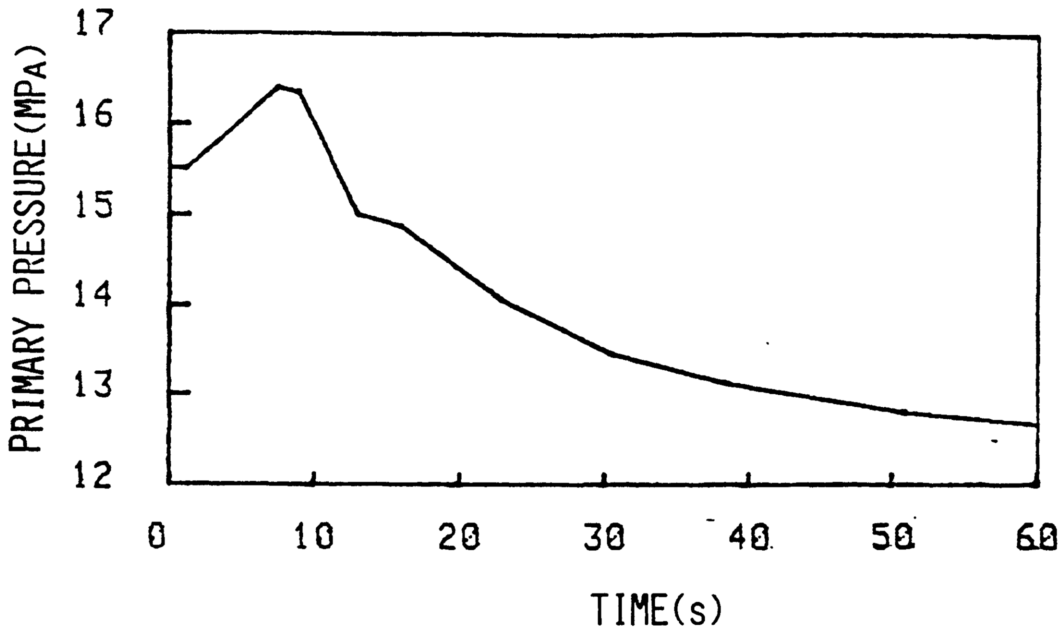


Figure 7.3-7.

Turbine trip test. Input common to SG-1 and SG-2.

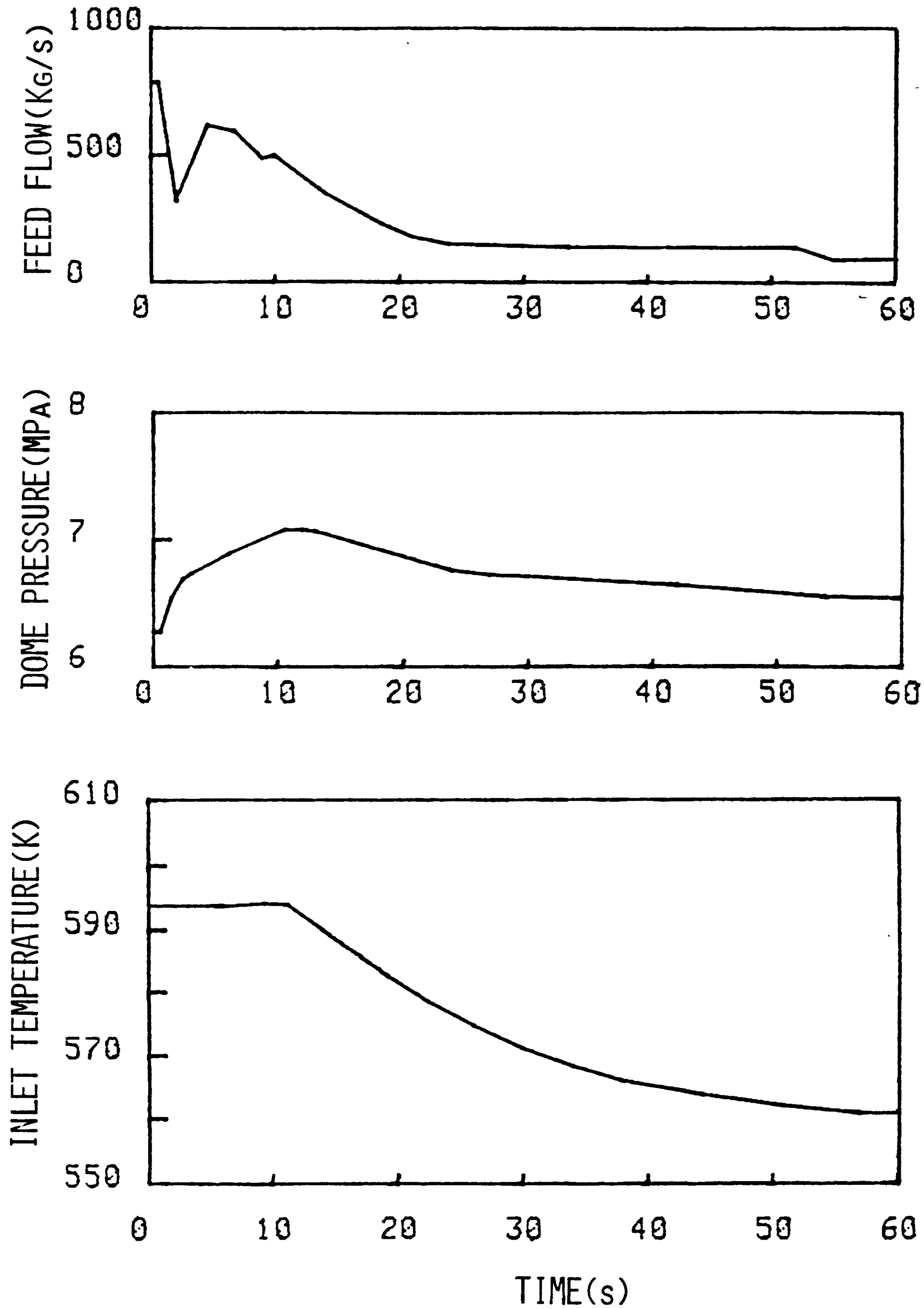


Figure 7.3-8.

Turbine trip. Input for SG-1.

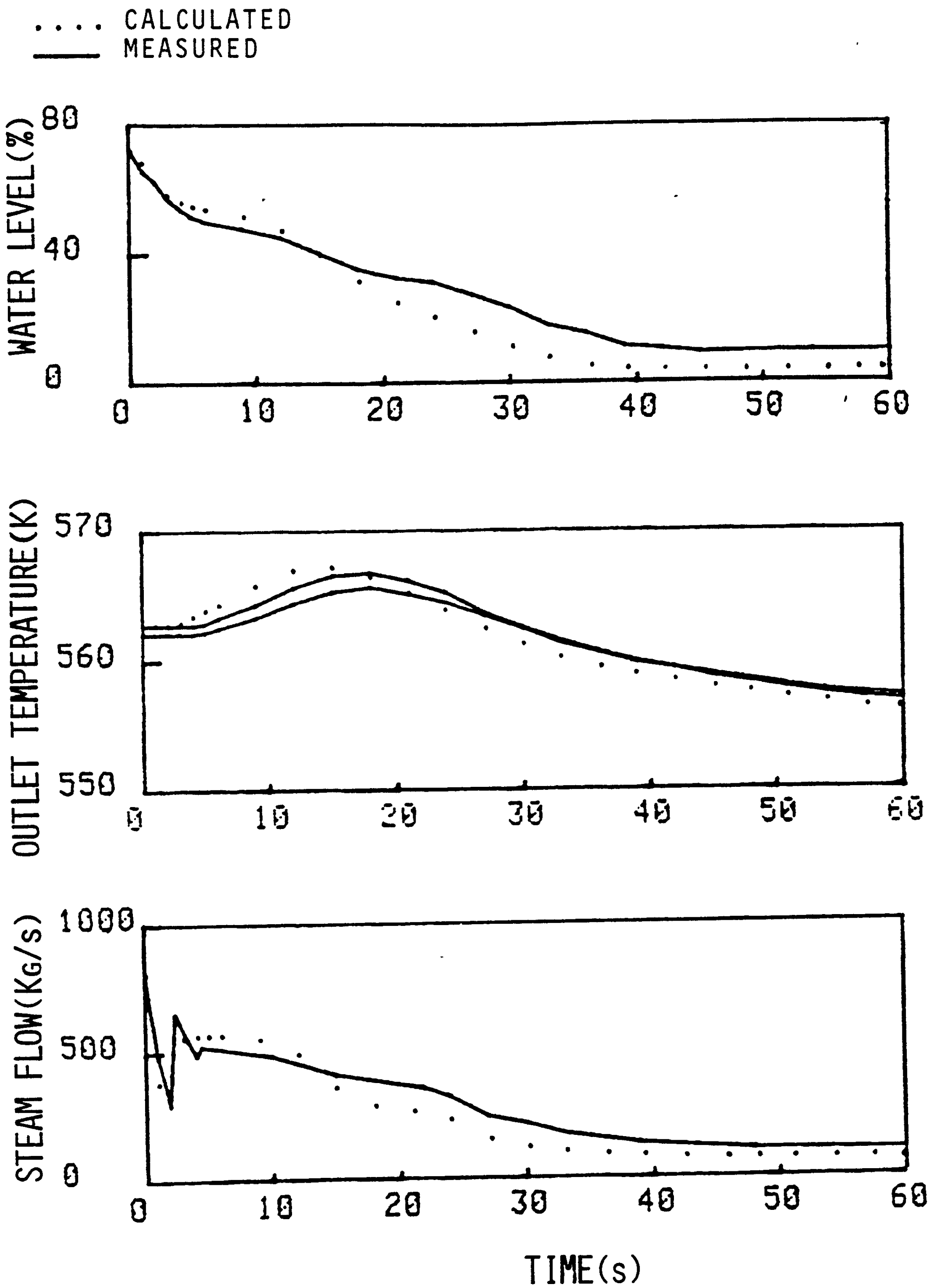


Figure 7.3-9.

Turbine trip. Response of SG-1.

Table 7.3-6.

Turbine Trip: Initial conditions for SG-2.

Parameter	Value
Bundle power	1382 MWt
Steam dome pressure	6.27 MPa
Steam flow	804 kg/s
Feedwater temperature(1)	519.4 K
Primary inlet temperature	594.5 K
Primary outlet temperature	562.8 K
Primary system pressure	15.51 MPa
Primary mass flux	4109 $kg/m^2/s$
Downcomer water level(2)	10.5 m (72 percent)
Downcomer flow	3266 kg/s
Circulation ratio	4.1
Fouling parameter	1.0

(1) Calculated using Eq. (7.3-1).
(2) Distance is referenced to tube sheet. Percentage is of instrument span (4.20 m) where lower instrument tap is 7.45 m above tube sheet.

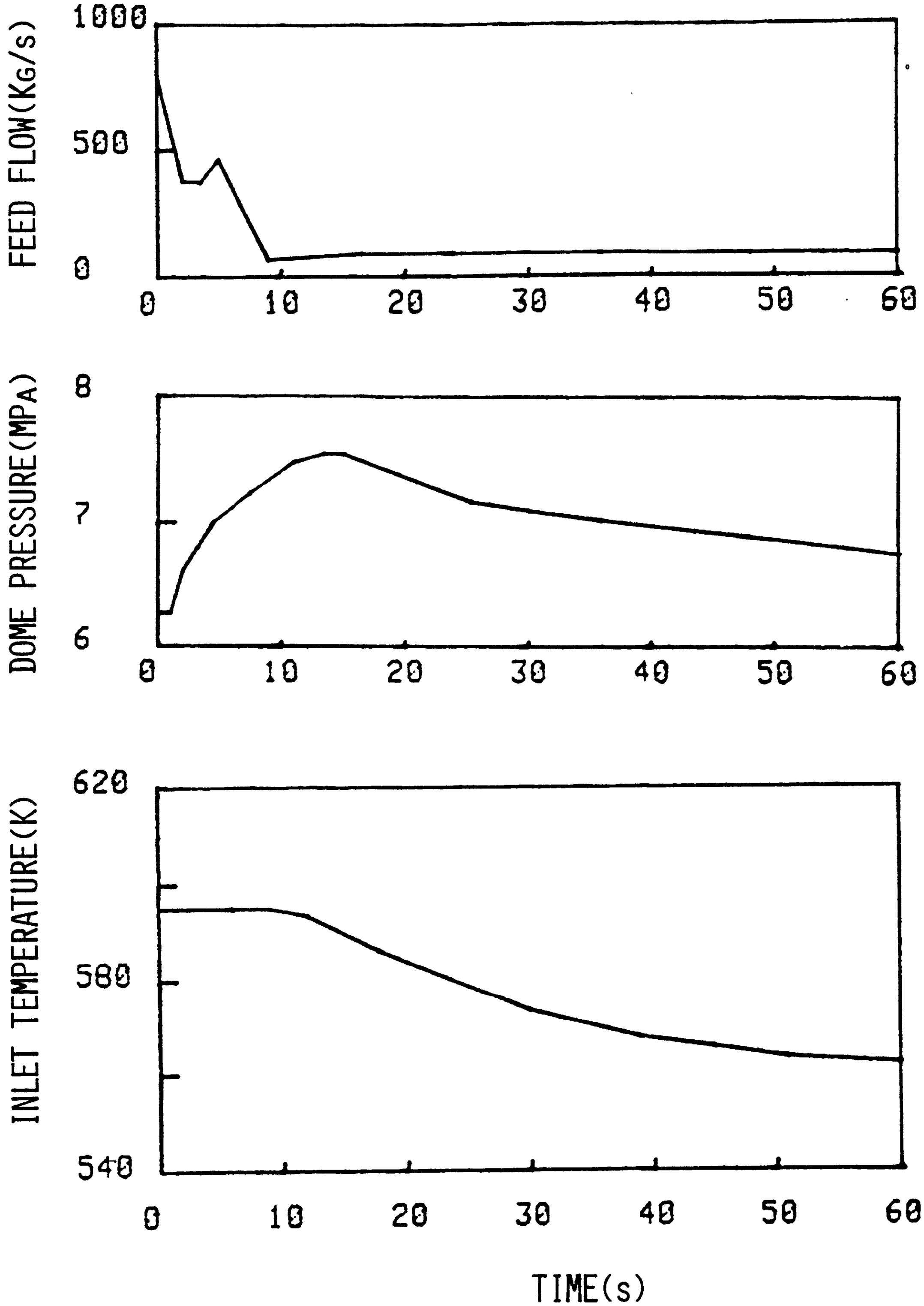


Figure 7.3-10.

Turbine trip. Input for SG-2.

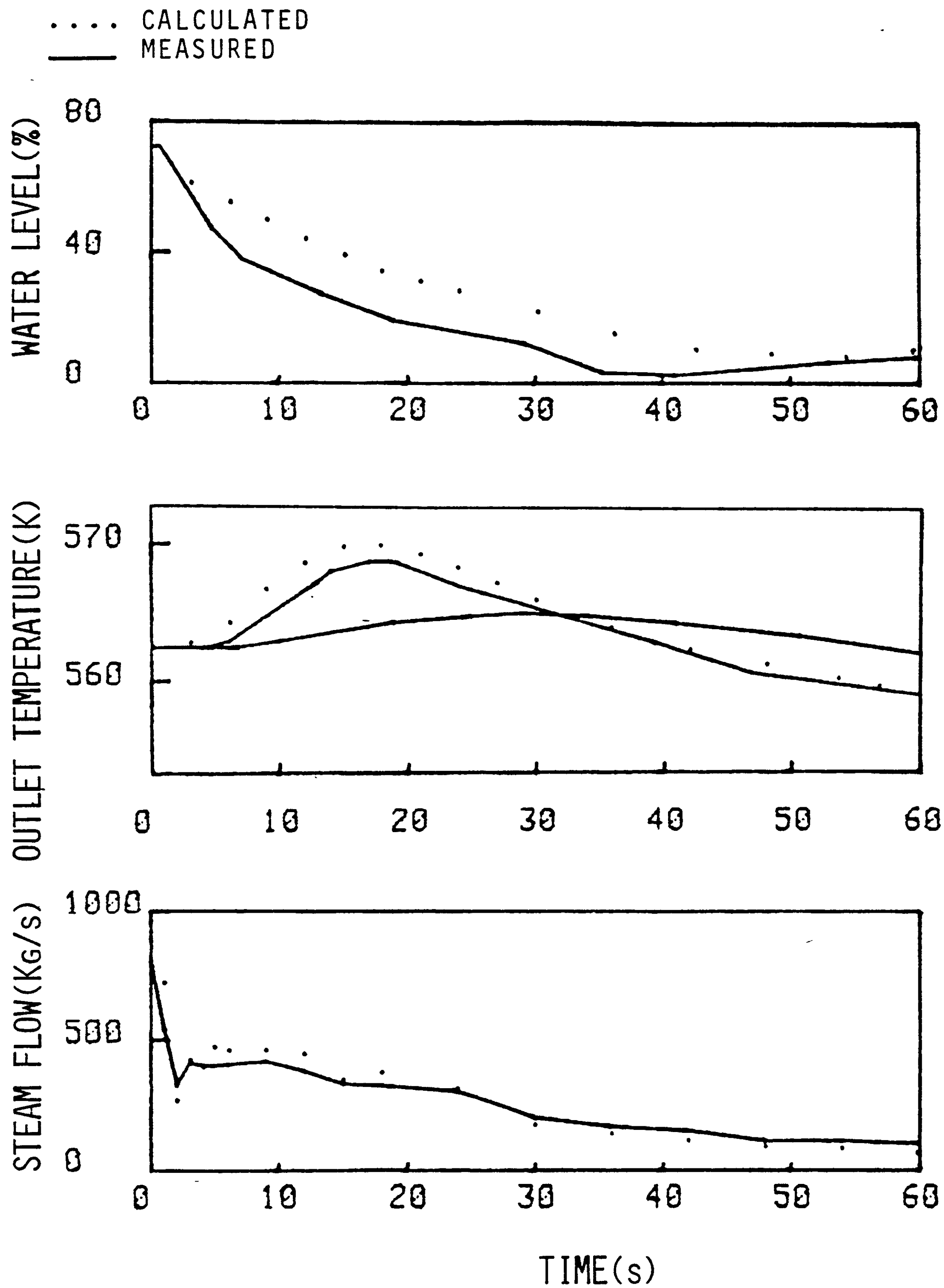


Figure 7.3-11.

Turbine trip. Response of SG-2.

steady-state steam flow rate prior to the event. In the present case, the adjustment corresponded to a 12 Kg/s (1.5 percent of the steam flow) maintained throughout the test.

The response of Steam Generator 2 is shown in Fig. 7.3-11. All calculated parameters are in good agreement with the data.

7.3.4 Loss of Primary Flow Test

This test is initiated at 82 percent power with a manual trip of all four reactor coolant pumps. This occurs at time zero. At 0.2 seconds the reactor is tripped, followed by a turbine trip at 0.4 seconds. At one second the turbine bypass valves begin to open and are fully open a second later. At two seconds the atmospheric dump valve opens and closes 4 seconds later. At 12 seconds the turbine bypass valves begin to close and are fully closed at 18 seconds. No other significant events occur over the time period of the simulation.

In this transient, only the response of Steam Generator 1 is calculated. This is because the measurements for SG-2 are incomplete and it was felt that the number of simulations carried so far was sufficient for the purposes given in Section 7.1.

The data presented in Ref. (S8) for the time behavior of the primary mass flux is not adequate for use with the present model. This is because there is data for the flow rates through pumps 1A and 1B which are shown in Fig. 7.3-1. Since the present model utilizes only one value of the primary mass flux, it was necessary to utilize a simple flow coastdown model (S2) and then fit the data using the model. Two mechanisms are considered for the flow coastdown calculation:

1. pump coast down, and
2. natural circulation.

The pump coastdown model is that of Ref. (T1) given by:

$$\frac{dG}{dt} + aG^2 = 0$$

which when solved for the initial condition:

$$G = G_i \quad \text{at} \quad t = 0$$

yields

$$G = \frac{G_i}{1 + bt}$$

where

$$b = 0.115$$

is determined by fitting the data for the two pumps.

After natural circulation is established, a different scheme is used to determine the time dependent behavior of the mass flux. The scheme is based on the flow-to-power ratio, which is defined in dimensionless form by expressing both quantities in percent of their respective full power values. This quantity is taken as 4.5, a typical value for another (S2) Combustion Engineering NSSS of the same size and similar design. The decay power is modeled as:

$$Q_d(t) = \sum_{i=1}^{4} Q^o_{di}(t)^{-\lambda_i t} \qquad (7.3\text{-}2)$$

where

$Q_d(t)$ = fraction of full power (dimensionless)

Q^o_{di} = contribution of decay group i (dimensionless)

λ_i = decay constant of group i (s^{-1}).

The pertinent numerical values are given in Table 7.3-7. The primary mass flux in natural circulation is obtained by multiplying the result of Eq. (7.3-2) by the flow-to-power ratio. The mechanism corresponding to the largest flow rate is the prevailing mechanism at any time for the duration of the

Table 7.3-7.

Decay power parameter (Ref. (S2)).

Group	Q_{di}^{0}	$\lambda_i(sec^{-1})$
1	0.0054	0.7600
2	0.0150	0.1309
3	0.0185	0.01299
4	0.0260	$3.4679 \cdot 10^{-4}$

transient. In the present case, utilizing the two models previously described, the transition from pump coastdown to natural circulation was observed to take place at 40 seconds into the transient. The time dependent behavior of the primary mass flux, calculated as described above and used as input to the calculations for the loss of flow test is given in Fig. 7.2-12. The behavior of the primary system pressure is shown in the same figure. The feedwater temperature does not vary throughout the event.

The initial conditions for this test are given in Table 7.3-8. The input used as forcing functions for the calculations is given in Figs. 7.3-12 and 7.3-13. For the reasons given in Section 7.3.2, the feedwater flow rate input is adjusted to give a value consistent with the steady-state steam flow immediately preceding the event. For this test, the correction was of 2 Kg/s (0.33 percent of the steam flow) maintained throughout the test.

The response of SG-1 is presented in Fig. 7.3-14. All calculated quantities are in good agreement with the data.

Table 7.3-8.

Loss of primary flow: Initial conditions for SG-1.

Parameter	Value
Bundle power	1148 MWt
Steam dome pressure	6.37 MPa
Steam flow	607 kg/s
Feedwater temperature(1)	482.1 K
Primary inlet temperature	586.9 K
Primary outlet temperature	562.6 K
Primary system pressure	15.46 MPa
Primary mass flux	4534 $kg/m^2/s$
Downcomer water level(2)	10.4 m (70 percent)
Downcomer flow	4460 kg/s
Circulation ratio	7.3
Fouling parameter	1.0

(1) Calculated using Eq. (7.3-1).
(2) Distance is referenced to tube sheet. Percentage is of instrument span (4.20 m) where lower instrument tap is 7.45 m above tube sheet.

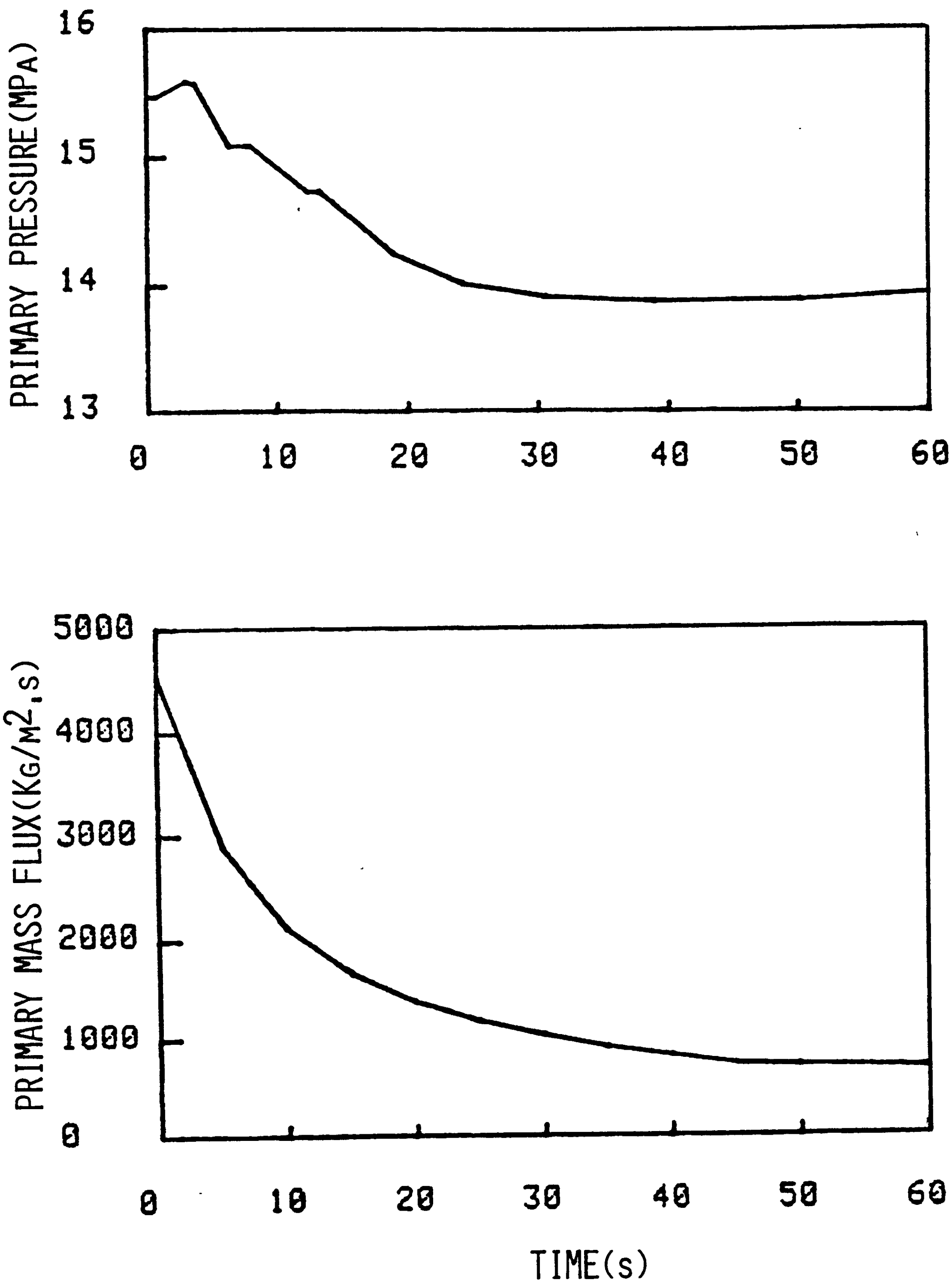


Figure 7.3-12.

Input for loss of primary flow.

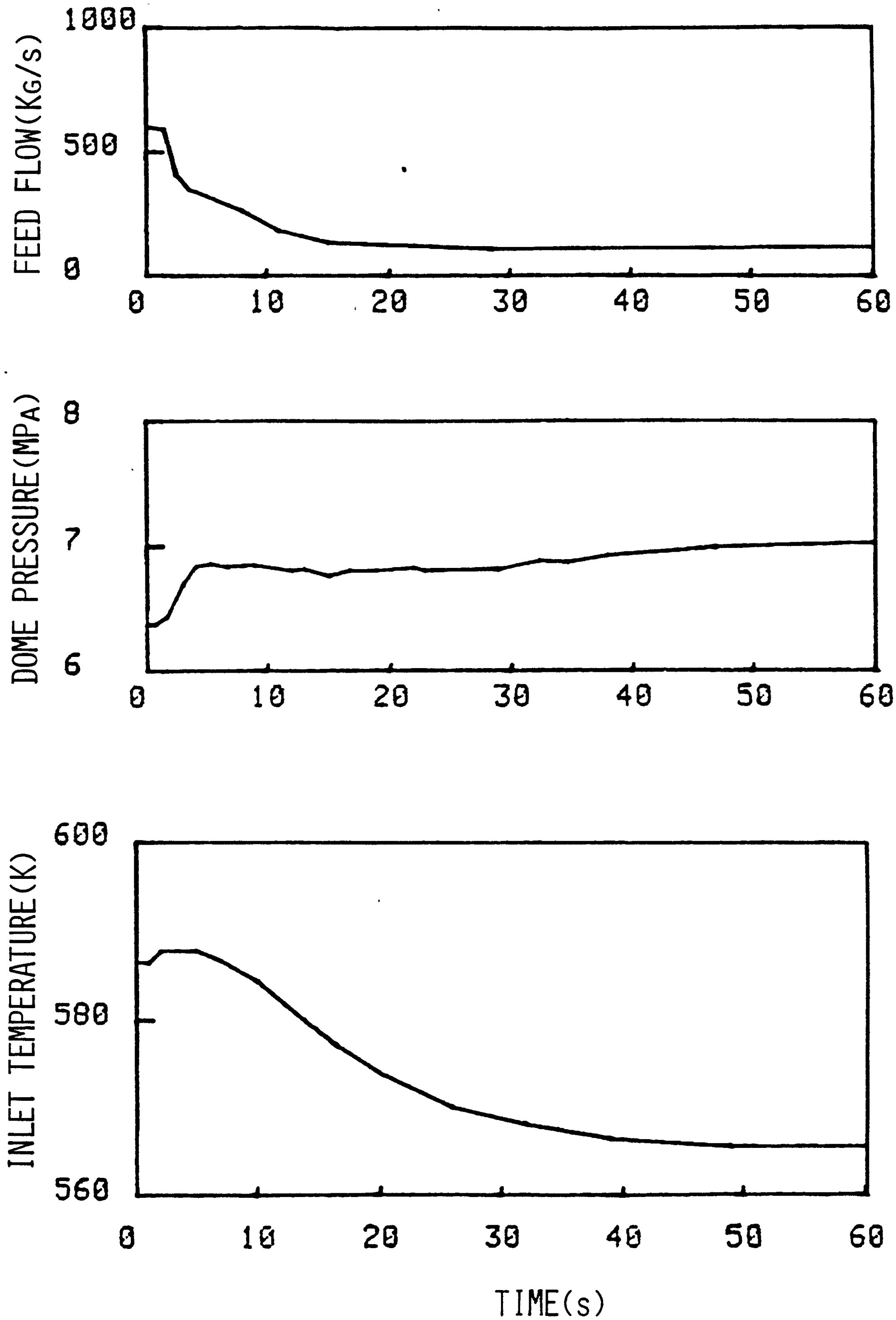


Figure 7.3-13.

Loss of primary flow. Input for SG-1.

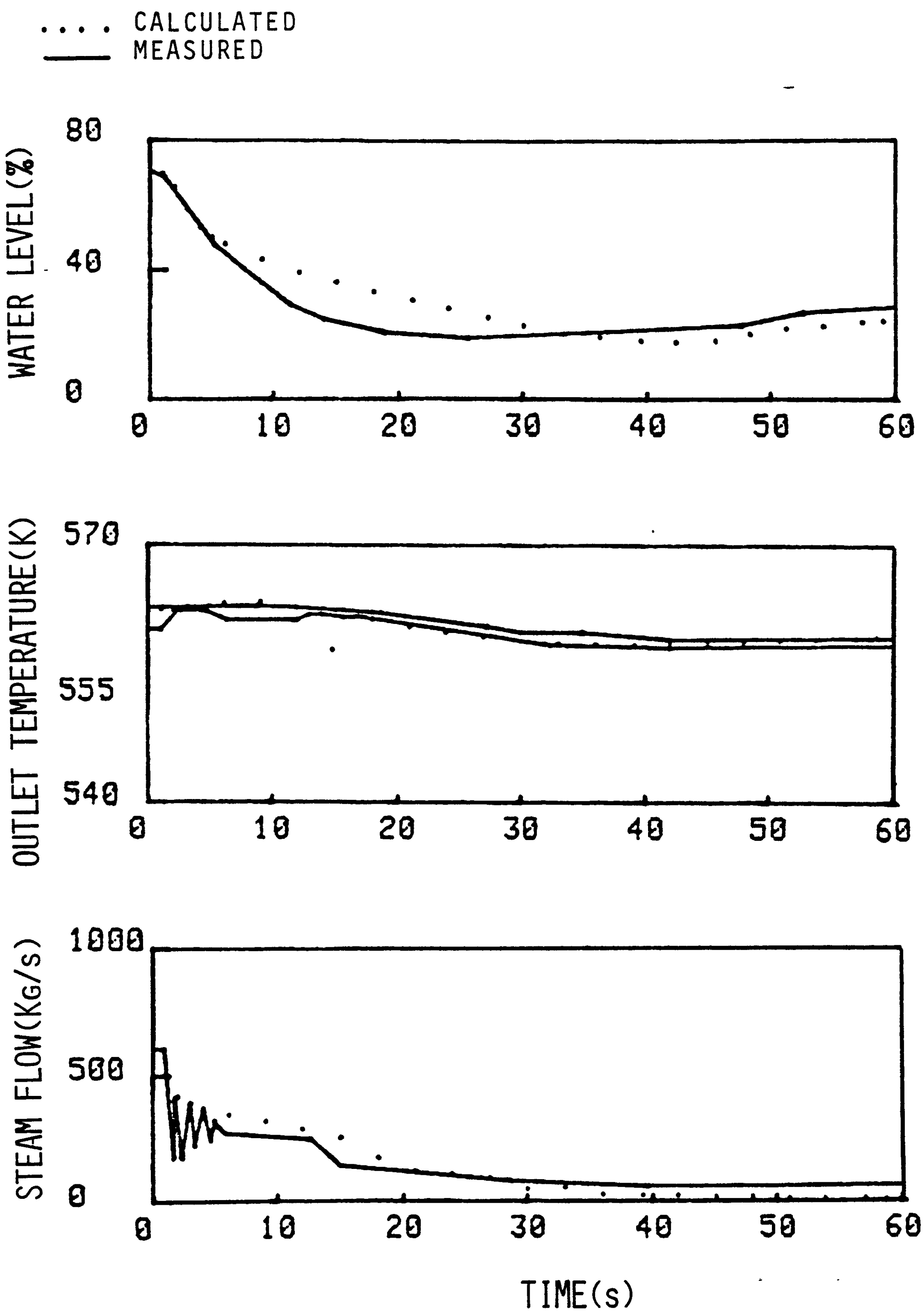


Figure 7.3-14.

Loss of primary flow. Response of SG-1.

Chapter 8

SUMMARY, CONCLUSIONS, AND RECOMMENDATIONS

8.1 Summary

This research has addressed the problem of developing a benchmark computer model for U-tube steam generator thermo-hydraulic analysis. The starting point has been the two-fluid model computer code THERMIT (R1)(K1). The final result is that the first integrated U-tube steam generator model utilizing the porous body, two-fluid formulation with the demonstrated capability of calculating (a) global parameters for both mild and severe operational transients and (b) steady-state parameter distributions with intermediate resolution has been developed.

The immediate applications of the present model include aiding in the development of computer codes for Safety Parameter Display Systems (SPDS) or in Fault Detection and Identification (FDI) systems by providing a test ground for the validation of assumptions that can help the requirements of great speed and reliability needed for those applications to be met simultaneously. Another immediate application is in conjunction with system codes such as RETRAN where optimal nodalization schemes and constants for the steam generators can be sought by benchmarking with a component model calculation such as the present. An example of this type of application is given in Ref. (H6).

The approach used in the present work has been to develop separate models for the steam dome, for the downcomer, evaporator and riser, and for the primary fluid and U-tubes, as well as to develop the procedures necessary to integrate the various regional models into the THERMIT framework. The steam dome compressibility effects are accounted for and integrated with the riser and upper downcomer calculations through a procedure called the recirculation model.

This model calculates the transient outlet nozzle steam flow rate (or steam dome pressure), downcomer water level, and upper downcomer temperature. It is linked to the downcomer-evaporator-riser two-fluid model calculation via riser outlet flow rate and quality and through the downcomer flow rate. This regional model incorporates separate equations for the liquid and the vapor with flashing and condensation allowed.

A one-dimensional heat sink model has been made available and can optionally be incorporated into this calculation. The model calculates the thermal power lost by the steam in the dome to structural material and to the downcomer liquid by heat conduction alone. This effect is demonstrated to be small for regular-size units even during fairly abrupt pressure surges such as those occurring immediately after main steam valves close and prior to the opening of dump and/or bypass valves in events such as a load rejection by the turbines. For smaller (model) units, the heat loss to structure could, however, be significant.

The downcomer, evaporator, and riser region are represented by the two-fluid model described in detail in Ref. (K1). The salient features of this model include:

1. Thermal and mechanical non-equilibrium between phases.
2. A complete boiling curve is constructed at each cell with an elaborate heat transfer logic (Fig. 3.2-1) to select an appropriate heat transfer correlation (Table 3.2-2).
3. The numerical method utilizes a semi-implicit technique particularly suitable for severe transient analysis.

The primary-side model incorporates a two-region wall heat conduction calculation and determines the temperatures of: primary coolant, primary-side wall, intermediate wall and secondary-side wall of the U-tubes. The salient feature of this regional model is that these temperatures are calculated in as many locations as desired within each computational cell of the secondary-side two-fluid model. This is done by lumping together tubes into groups called tube banks. The effects of tube length differences on the flow split and on primary fluid transport times are taken into consideration.

The final step in the present study is model assessment. Present model calculations of local parameter distributions are compared to measurements performed in a Westinghouse Model Boiler experimental facility (H5) and to calculations by another computer code (S3), (H5), developed under the sponsorship of the Electric Power Research Institute. Results are shown to be in excellent agreement. Calculations of actual plant transients for both small and substantial simultaneous disturbances of all system boundary conditions are compared to measurements (S8) performed at the Arkansas Nuclear One-Unit 2 facility. Results demonstrate the applicability of the present model to the calculation of both mild and severe operational transients.

8.2 Conclusions

8.2.1 System Boundary Conditions

Engineers working in the computer modeling of U-tube steam generators are often in need of criteria for selecting between the various possible combinations of system boundary conditions. While the combination adopted in this work is not original, nor is it the only one possible, it is hoped the discussions of Sections 4.5, 5.4, 6.4, 7.2.2, 7.2.3, and 7.3.1 will be helpful in that respect. Conclusions which may be drawn from those discussions include:

First, the list of candidates for system boundary conditions is:

1. Primary Inlet Temperature
2. Primary Average (or Outlet) Temperature
3. Primary Mass Flux
4. Primary System Pressure

5. Power Level
6. Steam Dome Pressure
7. Steam Flow
8. Downcomer Water Level
9. Feedwater Temperature
10. Feedwater Flowrate
11 Fouling Coefficient.

Second, for transient analysis, the combination of parameters universally preferred is: 1, 3, 4, 9, 10, 11, and either 6 or 7. The last two conditions are equivalent but the steam dome pressure will generally allow a more straightforward solution because properties can be calculated without iteration on the pressure. An excellent discussion on the practicality of using downcomer water level as a system boundary condition is given in Ref. (S2).

Third, in steady-state analysis, several combinations are possible. Equation (4.5-2) shows that 7 is not needed in this case and indicates that 5 and 10 are essentially equivalent. The fouling parameter is usually not necessary in a benchmark calculation where the heat transfer model is capable of accounting for different local regimes. This is in the sense that it can be set at its neutral value unless there is evidence of substantial tube fouling per se. However, for models with less spatial resolution, some sort of adjustment parameter should be used in order to accommodate the effect of local variations in heat transfer rates on the overall heat transfer coefficient. Still with respect to steady-state analysis, between 1, 2, 3, 5, 6, and 11, four parameters need be given. In this work, 1, 3, 5, and 6 (full power) or 11 (other powers) are given. Either 11 (full power) or 6 (other powers) are used (and hence calculated) to force 3 (which is a floating parameter) to converge to its given value. Alternatively, as in Ref. (S3), it is possible to fix both 6 and 11 and use 1 to force either 3 or 5 (one fixed, the other floating) to a prescribed value. There is an equivalence between 1 and 2. As pointed out in Section 4.5, these quantities are essentially interchangeable. Parameter 2 involves an additional iteration procedure, although it is more suitable for applications in conjunction with a systems code. Parameters 4 and 8 should also be given in a steady-state analysis.

8.2.2 Steam Superheat and Heat Sinks in the Dome

During the turbine trip tests described in Section 7.2, the steam was calculated to be in superheated condition for the first 15 seconds in SG-1 and 21 seconds in SG-2. Maximum superheat was calculated as 2.8 K at 3.0 seconds for SG-1 and 4.3 K at 5.0 seconds for SG-2. Unfortunately, a calculation with the steam forced to remain at saturation was not made for comparison of the overall predictions. It can be said, however, that agreement with steam flow and water level data is very good in the initial stages of both tests and that allowance for steam superheat has positively contributed to this by dampening the effect of the substantial pressure surge on both parameters.

In contrast, runs were performed for the turbine trip test in SG-1 with the heat sink calculation both on and off. As discussed in Chapter 5, their effect is small for regular power units and barely discernible only on the water level predictions. The saturated steam condition is reestablished primarily by the renewal of steam in the dome rather than by heat conduction losses. However, for small units or in case the steam generator becomes bottled up, calculation of the heat sink terms will lead to more realistic results, and a ready-to-use model for these calculations is available from this work.

8.2.3 Two-Phase Models and Degree of Spatial Resolution

In order to calculate the behavior of global parameters such as: steam flow, primary outlet temperature, and downcomer water level for operational transients, even when substantial simultaneous perturbations of all the system boundary conditions is involved, it is not necessary to utilize either high spatial resolution nor a detailed two-fluid model. Nevertheless, careful attention must be paid to slip. In addition, parameter distributions must be taken into account as well when computational nodes are large. The results of the present work, along with the results of Ref. (S2) where equivalent predictions are obtained for the same transients with a simpler model, substantiate this conclusion. The reason for this result seems to be that, for operational transients, the three parameters mentioned above are predominantly controlled by the downcomer and riser flow rates. These flow rates in turn are dominated by the steam dome pressure and three flow resistances: tube bundle, separator, and inlet to evaporator. These resistances can be adequately represented utilizing very few computational cells and with the application of global drift flux methodology. In order to expand the applicability range of simpler models, a model having a finer spatial resolution for benchmarking is helpful. A two-fluid model for this purpose is also useful in order to evaluate the applicability of a simpler model for non-equilibrium situations.

Hence, the two-fluid model is ideally suited for non-equilibrium situations, which occur in severe accident conditions such as in a rapid depressurization transient. Although no calculation of this sort has been attempted using the present model, there is no reason to believe it would not be possible to do so. This point is discussed further in the next section.

Another important area where the two-fluid formulation is indicated is in microscale studies such as those of Ref. (S1). In this type of application, an intermediate scale resolution mesh, such as the one used in the MB-2 calculations, must be set up first and a steady-state solution obtained. This solution is then used as a boundary condition for the microscale problem. The two-fluid formulation is useful in this case to point out potential dryout or flow starvation areas as well as local hot spots for design studies. This is also an area for which the present model establishes the groundwork for future developments. This point is to be discussed in the next section as well.

8.3 Recommendations for Future Work

The present model is well suited for two distinct types of future developments.

One possibility is application for microscale problems in steady-state. The version of THERMIT upon which the present integrated U-tube steam generator model was constructed has been demonstrated to be applicable to subchannel analysis (K1). Furthermore, in this work it is shown that parameter distributions are well calculated by the present model on an intermediate level of spatial resolution. These two facts indicate that steady-state calculations such as those in Section 7.2 can be used to obtain boundary conditions which can then be applied to the solution of problems involving spatial resolution of at least subchannel scale. Although this type of calculation is not new (see Ref. (S1)), it has important applications in designing against corrosion.

Another type of application which would be new involves applying the present model to calculations under severe accident conditions. Although there are several thermohydraulics codes with the capability of treating situations involving significant non-equilibrium, none incorporate an integrated steam generator model. If the transient involves a sudden depressurization of the secondary-side, there seems to be little difficulty with respect to the numerical method or the two-fluid model. The interfacial momentum exchange terms might require the inclusion of the virtual mass force and the Basset force (K1), (N2). As long as the primary-side remains single-phase, no changes seem to be necessary with respect to that model nor the primary to secondary-side heat transfer model. Furthermore, no changes seem necessary in the recirculation model other than to note that the liquid control volume, V_2, will now represent a "collapsed" rather than an actual volume if the downcomer becomes two-phase. However, in the analysis of a steam line break it is appropriate to utilize some critical flow model relating the steam dome pressure to the blowdown flow rate. Since the recirculation model incorporates an additional relationship between these quantities, it will no longer be necessary to input one of them when analyzing this type of transient. Nevertheless, the appropriate critical flow model and the best way to couple it with the recirculation model are still issues open to future investigation.

Application of the present model for depressurization transients on the primary-side requires the development of an entirely new primary-side model, allowing for two-phase and possible flow instabilities within the U-tubes. While this is no simple task, the integrated steam generator model with this added capability would be well within reach as soon as such a model became available in an appropriate form. This is because if sufficiently small time steps ($\Delta t \sim 0.01$ s) are taken, the primary- to secondary-side heat transfer coupling methodology developed in Chapter 6 remains valid. It appears feasible to develop a one-dimensional, two-phase flow model, possibly even based on THERMIT itself again, to calculate the flow within a U-tube for the desired conditions. This model then could be coupled to the other regional models utilizing the tube bank concept and the coupling strategy of Chapter 6.

From the perspective of this research, there are three areas open to future investigation. One is related to the previous discussion on steam superheat. It would be useful to verify, in depth, the importance of allowing for steam superheat on the calculated values of global parameters. This requires running at least the turbine trip transients again, while forcing a saturated steam condition calculation. Results would then be compared with those already available. Another useful continuation would be to verify whether any improvements can be obtained in the transient predictions by adjusting the local tube bank areas in such a way that their sum matches exactly the, as built, total heat transfer area. Thirdly, by pursuing steady-state calculations with the MIT interfacial momentum exchange model, it might be possible to determine satisfactorily whether indeed large and multiple cross flows, combined with loose momentum coupling, can in fact cause convergence difficulties in the two-fluid solution.

It is appropriate to end this discussion with some remarks concerning the computer program which is associated with the present study. Prospective users should be aware that the program is not user-oriented and substantial skill is required to run it successfully. Inspection of the input guide given in Appendix F will reveal that in spite of the great effort spent in constructing this part, much can still be done to simplify and minimize error in the input. The primary model is programmed in a way that channel 3 is the evaporator hot-side and channel 5 is the cold-side. While the theory developed in Chapter 6 is general for any channel layout, the coding of subroutine ptemp is not. The user interested in a different channel layout should examine that subroutine with some care. If application of the code to several geometries is expected, it is probably simpler to generalize the program in this respect. The simplest way of accomplishing this is probably to define the neighbors of the primary tube banks via input rather than attempt to generalize the presently used mathematical relation between secondary-side cell indices.

REFERENCES

(A1) S. Y. Ahmad, "Axial Distribution of Bulk Temperature and Void Fraction in a Heated Channel with Inlet Subcooling," Journal of Heat Transfer, 92, p. 595, 1970.

(A2) "Arkansas Nuclear One, Unit 2 License Application, Final Safety Analysis Report," Docket 50368-R1, Arkansas Power and Light Company, 1974.

(B1) J. M. Bates and C. W. Stewart, "Experimental Study of Single- and Two-Phase Flow Fields Around PWR Steam Generator Tube Support Plates," EPRI NP-1142, 1979.

(B2) N. W. S. Bruens, "U-Tube Steam Generator Dynamics Modelling and Verification," Proceedings of the Second International Conference on Boiler Dynamics and Control in Nuclear Power Stations, Bournemouth, October 23-25, 1979, The British Nuclear Energy Society, London, 1980.

(B3) T. A. Bjornard, "Blowdown Heat Transfer in a Pressurized Water Reactor," PhD thesis, MIT, 1977.

(B4) E. L. Burley, "Performance of Internal Steam Separation Systems in Boiling Water Reactors," ASME 69-WA/NE-24, 1969.

(B5) R. B. Bird, W. E. Steward, and E. N. Lightfoot, "Transport Phenomena," John Wiley and Sons, New York, 1960.

(C1) J. C. Chen, "A Correlation for Boiling Heat Transfer to Saturated Fluids in Convective Flow," ASME, 63-HT-34, 1963.

(C2) H. S. Carslaw and J. C. Jaeger, "Conduction of Heat in Solids," Oxford University, 2nd Ed., p. 103, 1959.

(C3) W. G. Clarke, "Transient Analysis of Steam Generators in Nuclear Power Plants Using Digital Computer Techniques," M.S. Thesis, University of Pittsburgh, 1965.

(C4) W. R. Carson and H. K. Williams, "Method of Reducing Carry-Over and Reducing Pressure Drop Through Steam Separators," EPRI-NP-1607, 1980.

(D1) R. S. Dougall, et al., "DUVAL: A Computer Program for the Numerical Solution of Two-Dimensional, Two-Phase Flow Problems," EPRI NP-2099, Volumes 1-3, April 1982.

(E1) D. Ebeling-Koning, "Hydrodynamic of Single- and Two-Phase Flow in Inclined Rod Arrays," PhD Thesis, MIT, 1983.

(G1) D. J. Gorman and E. E. Drucker, "A Method of Predicting Steam-Surge Tank Transients Based on One-Dimensional Heat Sinks," Nuc. Sci. Eng. 21, p. 473, 1965.

(G2) P. A. Gagne, "NSSS Design and Cycle 1 Operating History Data for Arkansas Nuclear One, Unit-2," EPRI NP-1707, 1981.

(H1) A. Hoeld, "A Theoretical Model for the Calculation of Large Transients in Nuclear Natural-Circulation U-Tube Steam Generators (Digital Code UTSG)," Nuc. Eng. Des. 47, pp. 1-23, 1978.

(H2) F. H. Harlow and A. A. Amsden, "A Numerical Fluid Dynamics Calculation for All Flow Speeds," J. Comp. Phys. 8, p. 197, 1971.

(H3) F. B. Hildebrand, "Advanced Caclulus for Applications," Prentice-Hall, p. 376, 1976.

(H4) J. P. Holman, "Heat Transfer," McGraw-Hill Kogakusha, LTD., International Student Edition, p. 395, 1976.

(H5) G. W. Hopkins, A. Y. Lee, and O. J. Mender, "Thermal and Hydraulic Code Verification: ATHOS2 and Model Boiler No. 2 Data," EPRI NP-2887, 1983.

(H6) F. B. Haghighi, "Thermal Hydraulic Analysis of a Pressurized Water Reactor During a Total Loss of AC Power Accident," PhD Thesis, MIT, 1983.

(J1) M. Jacob, "Heat Transfer," Vol. 1, John Wiley, New York, pp. 658-661, 1958.

(K1) J. E. Kelly, "Development of a Two-Fluid, Two-Phase Model for Light Water Reactor Core Analysis," PhD Thesis, MIT, 1980.

(L1) J. C. Lee, et al., "Review of Transient Modeling of Steam Generator Units in Nuclear Power Plants," EPRI-NP-1576, 1980.

(L2) J. C. Lee, et al., "Transient Modeling of Steam Generator Units in Nuclear Power Plants: Computer Code TRANSG-01," EPRI NP-1368, March 1980.

(M1) W. H. McAdams, "Heat Transmission," McGraw-Hill Book Company, New York, 1954.

(M2) E. D. Moeck and H. W. Hinds, "A Mathematical Model of Steam-Drum Dynamics," AECL-5057, 1976.

(N1) A. V. Nero, "A Guidebook to Nuclear Reactors," University of California Press, 1979.

(N2) H. C. No, "An Investigation of the Physical and Numerical Foundations of Two-Fluid Representation of Sodium Boiling," PhD Thesis, MIT, 1983.

(P1) S. V. Patankar, et al., "A Calculation Procedure for Heat, Mass, and Momentum Transfer in Three-Dimensional Parabolic Flows," Int. J. Heat Mass Transfer, Vol. 15, p. 1787, 1972.

(P2) S. V. Patankar, "Numerical Heat Transfer and Fluid Flow," Hemisphere Publishing Corporation, 1980.

(P3) M. Petrick, "A Study of Vapor Carry-Under and Associated Problems," ANL-6581, 1962.

(R1) W. H. Reed and H. B. Steward, "THERMIT: A Computer Program for Three-Dimensional Thermal Hydraulic Analysis of Light Water Reactor Cores," MIT Report prepared for EPRI, 1978.

(R2) B. W. Rhee, "Thermal Hydraulic Modeling of Pressurized Water Reactor Steam Generators," SM Thesis, MIT, 1981.

(R3) W. C. Rivard and M. D. Torrey, "Numerical Calculations of Flashing from Long Pipes Using a Two-Fluid Model," LA-6104-MS, 1975.

(R4) J. T. Rogers and R. G. Rosehart, "Mixing by Turbulent Interchange in Fuel Bundles, Correlations, and Inferences," ASME, 72-HT-53, 1972.

(R5) Z. Rouhani, "Steam-Water Separation," in "Two-Phase Flows and Heat Transfer," edited by J. J. Ginoux, Hemisphere Publishing Corporation, 1978.

(R6) W. M. Rohsenow and H. Choi, "Heat Mass and Momentum Transfer," Prentice-Hall, Inc., p. 308, 1961.

(R7) Z. Rouhani, "A Mathematical Model for Analysis of the Hydraulic Performance of Cyclone-Type Steam Separators," Preprint, presented at ANS/ASME International Topical Meeting on Nuclear Reactor Thermal-Hydraulics, Saratoga Springs, October 5-8, 1980.

(S1) C. W. Steward, et al., "Analysis of Single- and Two-Phase Flow Fields Around PWR Steam Generator Tube Support Plates," EPRI NP-1162, August 1979.

(S2) W. H. Strohmayer, "Dynamic Modeling of Vertical U-Tube Steam Generators For Operational Safety Systems," PhD Thesis, MIT, 1982.

(S3) A. K. Singhal, et al., "ATHOS: A Computer Program for Thermal-Hydraulic Analysis of Steam Generators," EPRI NP-2698, 1982.

(S4) A. L. Schor, "A Four-Equation Two-Phase Flow Model for Sodium Boiling Simulation of LMFBR Fuel Assemblies," PhD Thesis, MIT 1982.

(S5) P. Saha, "A Non-Equilibrium Heat Transfer Model for Dispersed Droplet Post-Dryout Regime," Int. J. Heat Mass Trans, 23, p. 481, 1980.

(S6) E. N. Sieder and C. E. Tate, "Heat Transfer and Pressure Drop of Liquids in Tubes," Ind. Eng. Chem., Vol. 28, p. 1429, 1936.

(S7) A. K. Singhal, et al., "The URSULA 2 Computer Program," Vols. 1-4, EPRI NP-1315, 1980.

(S8) D. P. Siska, "NSSS Transient Tests at ANO-2," EPRI NP-1708, May 1981.

(S9) C. W. Stewart, et al., "Improvements to the COBRA-TF (EPRI) Computer Code for Steam Generator Analysis," EPRI NP-1509, 1980.

(T1) L. S. Tong and J. Weisman, "Thermal Analysis of Pressurized Water Reactors," American Nuclear Society, p. 313, 1979.

(T2) J. R. S. Thom, Prediction of Pressure Drop During Forced Convection Circulation of Water," Int. J. Heat Mass Transfer 7, pp. 709-724, 1946.

(V1) M. J. V. Vorst and J. H. Stuhmiller, "Numerical Simulation of the Fluid Flow in a Centrifugal Steam Separator," paper presented at the ANS/ENS International Topical Meeting on Advances in Mathematical Methods for Nuclear Engineering Problems," Munich, April 20-27, 1981.

(W1) C. L. Williams and S. J. Green, "Thermal and Hydraulic Aspects of PWR Steam Generators," Preprint. Presented at ANS/ASME International Topical Meeting on Nuclear Reactor Thermal-Hydraulics, Saratoga Springs, October 5-8, 1980.

archive

THERMOHYDRAULIC ANALYSIS OF U-TUBE STEAM GENERATORS
by
Hugo Cardoso da Silva Jr.

Energy Laboratory Report No. MIT-EL 84-005

April 1984

VOLUME II

THERMOHYDRAULIC ANALYSIS OF U-TUBE STEAM GENERATORS
Volume II

by

Hugo Cardoso da Silva Jr.

Principal Investigators

Prof. David Lanning
Prof. John Meyer
Prof. Neil Todreas

Energy Laboratory
and
Department of Nuclear Engineering

Massachusetts Institute of Technology
Cambridge, Massachusetts 02139

Sponsored by:

Northeast Utilities Service Company
Yankee Atomic Electric Company

under the

MIT Energy Laboratory Electric Utility Program

MIT Energy Laboratory Report No. MIT-EL 84-005

April 1984

APPENDIX A

DERIVATION OF THE RECIRCULATION MODEL EQUATIONS

1. Introduction

As shown in Chapter 5, the conservation equations for the liquid and vapor in the upper downcomer and steam dome, respectively, are:

$$\frac{dM_2}{dt} = w_{fd} + (1-x_r)w_r + w_{cond} - w_{fl} - w_d \qquad \text{(A-1)}$$

$$\frac{dM_1}{dt} = w_r x_r - w_s - w_{cond} + w_{fl} \qquad \text{(A-2)}$$

$$\frac{d}{dt}(M_2 h_2) = w_{fd} h_{fd} - w_d h_2 + w_{cond} h_f - w_{fl} h_g + \frac{M_2}{\rho_2}\frac{dp}{dt} +$$

$$+ (1 - x_r) w_r h_f + \dot{Q}_2 \qquad \text{(A-3)}$$

$$\frac{d}{dt}(M_1 h_1) = w_r x_r h_g + w_{fl} h_g - w_{cond} h_f + \frac{M_1}{\rho_1}\frac{dp}{dt} - w_s h_1 + \dot{Q}_1 \qquad \text{(A-4)}$$

$$h_1 = h_1(T_1, p) \qquad \text{(A-5a)}$$

$$\rho_1 = \rho_1(T_1, p) \tag{A-5b}$$

$$h_2 = h_2(T_2, p) \tag{A-6a}$$

$$\rho_2 = \rho_2(T_2, p) \tag{A-6b}$$

$$\frac{d}{dt}(M_1/\rho_1 + M_2/\rho_2) = 0 \ . \tag{A-7}$$

The heat source/sink terms are obtained as indicated in Section 5.3.3 and are assumed known at this point. The unknowns are, therefore: M_2, ρ_2, T_2, h_2, T_1, M_1, ρ_1, h_1, w_{fl}, w_{cond}, p or w_s. These are eleven in number, while there are only nine equations. As pointed out in Section 5.3.2, two of these unknowns can be eliminated once the direction of the rate-of-change of dome pressure and state of steam and water are known, as shown in Table 5.3-1. The possibilities for the state of the steam and of the water are limited only by the assumption that subcooled vapor and superheated liquid do not occur. To each combination of allowable states for the vapor (saturated or superheated), liquid (saturated of subcooled), and direction of pressure change (rising or falling) a case number was attributed, as shown on Table 5.3-1 and according to the discussion presented in Section 5.2. To each case number there corresponds a set of nine unknowns and nine equations. The derivation of each of those equations in each case is the subject of this Appendix.

2. Case 1 - Rising Pressure and Case 5 - Superheated Vapor and Subcooled Liquid.

To simplify the derivation procedure it is noted that Eqs. (A-5a), (A-5b), (A-6a), (A-6b) imply:

$$\rho_1 = \rho_1(h_1,p)$$

and

$$\rho_2 = \rho_2(h_2,p) \ .$$

It can therefore be written:

$$\frac{d}{dt}\left(\frac{1}{\rho_1}\right) = -\left(\frac{1}{\rho_1}\right)\left[\left(\frac{\partial\rho_1}{\partial h}\right)_p / \frac{dh_1}{dt} + \left(\frac{\partial\rho_1}{\partial p}\right)_h \frac{dp}{dt}\right] \qquad \text{(A-8)}$$

$$\frac{d}{dt}\left(\frac{1}{\rho_2}\right) = -\left(\frac{1}{\rho_2}\right)^2\left[\left(\frac{\partial\rho_2}{\partial h}\right)_p \frac{dh_2}{dt}\right] \qquad \text{(A-9)}$$

where the liquid compressibility,

$$- (\frac{1}{\rho_2})^2 (\frac{\partial \rho_2}{\partial p})_h$$

has been neglected with respect to its thermal expansion,

$$- (\frac{1}{\rho_2})^2 (\frac{\partial \rho_2}{\partial h})_p \ .$$

These equations are to be solved along with Eqs. (A-1) to (A-4) and (A-7) with the flashing and condensation terms removed.

Substitution of Eqs. (A-8) and (A-9) into (A-7) yields:

$$\frac{dM_1}{dt} + \frac{dM_2}{dt} - \frac{M_1}{\rho_1} \left[(\frac{\partial \rho_1}{\partial h})_p \dot{h}_1 + (\frac{\partial \rho_1}{\partial p})_h \dot{p} \right] -$$

$$- \frac{M_2}{\rho_2} (\frac{\partial \rho_2}{\partial h})_p \dot{h}_2 = 0 \ . \qquad \text{(A-10)}$$

From Eq. (A-4) it can be written,

$$\dot{h}_1 = \frac{1}{M_1} \left[w_r x_r h_g - w_s h_1 + \frac{M_1}{\rho_1} \frac{dp}{dt} + \dot{Q}_1 - h_1 \dot{M}_1 \right] \qquad \text{(A-11)}$$

and when $\dot{M}_1$ is taken from (A-2), it is seen that,

$$\dot{h}_1 = \frac{1}{M_1}\left\{w_r x_r (h_g - h_1) + (\frac{M_1}{\rho_1})\,\dot{p} + \dot{Q}_1\right\}. \qquad \text{(A-12)}$$

Similarly, from Eq. (A-3),

$$\dot{h}_2 = \frac{1}{M_2}\left\{(1 - x_r)w_r h_f + w_{fd}h_{fd} - w_d h_2 + \right.$$

$$\left. + (\frac{M_2}{\rho_2})\dot{p} + Q_w - h_2\dot{M}_2\right\} \qquad \text{(A-13)}$$

where again elimination of $\dot{M}_2$ through the mass balance (A-1) yields:

$$\dot{h}_2 = \frac{1}{M_2}\left\{(1 - x_r)w_r(h_f - h_2) + w_{fd}(h_{fd} - h_2) + (\frac{M_2}{\rho_2})\dot{p} + Q_w\right\} \qquad \text{(A-14)}$$

Substituting Eqs. (A-12) and (A-14) for $\dot{h}_1$ and $\dot{h}_2$, respectively, into Eq. (A-10) and solving for $\dot{p}$ yields:

$$\dot{p} = (\dot{M}_1/\rho_1 + \dot{M}_2/\rho_2 - \text{pnum})/\text{pden} \qquad \text{(A-15)}$$

where,

$$\text{pnum} = \left[\frac{1}{\rho_1}\right]^2 \left[\frac{\partial \rho_1}{\partial h}\right]_p [w_r x_r (h_g - h_1) + \dot{Q}_1] -$$

$$- \left[\frac{1}{\rho_2}\right]^2 \left[\frac{\partial \rho_2}{\partial h}\right]_p [1 - x_r) w_r (h_f - h_2) + w_{fd}(h_{fd} - h_2) + \dot{Q}_2] \qquad \text{(A-16)}$$

and

$$\text{pden} = \frac{M_1}{\rho_1}\left[\frac{1}{\rho_1}\right]^2 \left[\frac{\partial \rho_1}{\partial h}\right]_p + M_1 \left[\frac{1}{\rho_1}\right]^2 \left[\frac{\partial \rho_1}{\partial p}\right]_h + \frac{M_2}{\rho_2}\left[\frac{1}{\rho_2}\right]^2 \left[\frac{\partial \rho_2}{\partial h}\right]_p \qquad \text{(A-17)}$$

The state equations must have pressure and temperature as dependent variables. Therefore the property derivatives in pnum and pden are written in the form

$$\left(\frac{\partial \rho}{\partial h}\right)_p = \left(\frac{\partial \rho}{\partial T}\right)_p \Big/ \left(\frac{\partial h}{\partial T}\right)_p$$

$$(\frac{\partial \rho}{\partial p})_h = (\frac{\partial \rho}{\partial p})_T - \frac{(\frac{\partial \rho}{\partial T})_p}{(\frac{\partial h}{\partial T})_p} \cdot (\frac{\partial h}{\partial p})_T \; .$$

When the pressure is assumed known and the exit steam flow rate, w_s, is the unknown, Eq. (A-15) is used to yield $\dot{M}_1$ which is substituted for by the right hand side of Eq. (A-2). After rearrangement this leads to

$$w_s = w_r x_r - \rho_1 \left[(pden)\dot{p} + pnum - \frac{\dot{M}_2}{\rho_2} \right] . \qquad \text{(A-18)}$$

The state equations in THERMIT-UTSG are of the form,

$$\phi = \phi(p,t) \; . \qquad \text{(A-19)}$$

Hence, the temperatures can be computed through the relation,

$$dh = \left[\frac{\partial h}{\partial T}\right]_p dT + \left[\frac{\partial h}{\partial p}\right]_T dp \; . \qquad \text{(A-20)}$$

That is,

$$\dot{T} = \left[\dot{h} - \left[\frac{\partial h}{\partial p}\right]_T \dot{p} \right] / \left[\frac{\partial h}{\partial T}\right]_p \; . \qquad \text{(A-21)}$$

The case 1 and case 5 equation set is made up of Eqs. (A-15) or (A-18), (A.1), (A-2), (A-14), (A-12), (A-21) for vapor and liquid, (A-5a), (A-5b), (A-6a), (A-6b). The method of solution is described in Chapter 5.

3. Case 2 - Falling Pressure, Saturated Steam, and Subcooled Liquid

Following the argumentation presented in Section 5.3.2, it is evident in this case that

$$w_{fl} = 0 \qquad \text{(A-22)}$$

and

$$h_1 = h_g \ . \qquad \text{(A-23)}$$

In a manner analogous to that used to derive Eq. (A-10), it can be written,

$$(\frac{1}{\rho_g}) \dot{M}_1 - (\frac{1}{\rho_2}) \dot{M}_2 (\frac{1}{\rho_g})^2 \left[\frac{d\rho_g}{dp}\right]_{sat} \dot{p} M_1^{-} (\frac{1}{\rho_2})^2 M_2 \left[\frac{\partial \rho_2}{\partial h}\right]_p \dot{h}_2 = 0 \ . \qquad \text{(A-24)}$$

From Eq. (A-4), $\dot{h}_g$ can be isolated as

$$\dot{h}_g = [w_{cond} h_{fg} + \frac{M_1}{\rho_g} \dot{p} + \dot{Q}_1] / M_1 \ . \qquad \text{(A-25)}$$

However, from the identity

$$\dot{h}_g = \left[\frac{dh_g}{dp} \right]_{sat} \left[\frac{dp}{dt} \right] \qquad \text{(A-26)}$$

and Eq. (A-25) it is possible to obtain:

$$w_{cond} = - \dot{p} \left\{ \frac{M_1}{h_{fg}} \left[\frac{1}{\rho_g} - \left[\frac{dh_g}{dp} \right]_{sat} \right] \right\} - \frac{\dot{Q}_1}{h_{fg}} \ . \qquad \text{(A-27)}$$

Combining Eqs. (A-1) and (A-3), it is possible to write

$$\dot{h}_2 = \frac{1}{M_2} [(1 - x_r) w_r (h_f - h_2) + (\frac{M_2}{\rho_2}) \dot{p} + \dot{Q}_2 -$$

$$- w_{cond} (h_f - h_2)] \ . \qquad \text{(A-28)}$$

Substituting (A-27) into (A-1) and (A-2) having (A-22) in mind, results in

$$\dot{M}_1 = w_r x_r - w_2 + \dot{p}\left\{\frac{M_1}{h_{fg}} \left[\frac{1}{\rho_g} - \frac{dh_g}{dp}\right] + \frac{\dot{Q}_1}{h_{fg}}\right\} \tag{A-29}$$

and

$$\dot{M}_2 = (1-x_r)w_r + w_{fd} - w_d - \dot{p}\left\{\frac{M_1}{h_{fg}} \left[\frac{1}{\rho_g} - \frac{dh_g}{dp}\right]\right\} - \frac{\dot{Q}_1}{h_{fg}} \tag{A-30}$$

respectively. These two equations for $\dot{M}_1$ and $\dot{M}_2$, along with Eq. (A-28) can be substituted into Eq. (A-24). The resulting expression, when solved for $\dot{p}$, is of the form

$$\dot{p} = [(w_r x_r - w_s) / \rho_g + \text{pnum}] / \text{pden} \tag{A-31}$$

where

$$\text{pnum} = \frac{1}{\rho_2} \left[(1-x_r)w_r + w_{fd} - w_d - \frac{\dot{Q}_1}{h_{fg}}\right] - \left[\frac{1}{\rho_2}\right]^2 \left[\frac{\partial \rho_2}{\partial h}\right]_p \cdot$$

$$\left[(1-x_r)w_r(h_f-h_2) + w_{fd}(h_f-h_2) - \dot{Q}_1 \frac{(h_f-h_2)}{h_{fg}} + \dot{Q}_2\right] \tag{A-32}$$

and

$$pden = M_1\left\{\frac{1}{h_{fg}}\left[\left[\frac{dh_g}{dp}\right]_{sat} - \frac{1}{\rho_g}\right]\left[\frac{1}{\rho_g} - \frac{1}{\rho_2} + (h_f-h_2)\left[\frac{1}{\rho_2}\right]^2\left[\frac{\partial\rho_2}{\partial h}\right]_p\right] + \left(\frac{1}{\rho_g}\right)^2\left[\frac{d\rho_g}{dp}\right]\right\} + \frac{M_2}{\rho_2}\left[\frac{1}{\rho_2}\right]^2\left[\frac{d\rho_2}{dh}\right]_p . \qquad (A-33)$$

In the event that $\dot{p}$ is given and w_s is the unknown, Eq. (A-31) can be solved for that variable yielding

$$w_s = [-\dot{p}\ pden + pnum]\rho_g + w_r x_r . \qquad (A-34)$$

The remaining equations are derived as described in the previous section for Case 1.

4. Case 3 - Falling Pressure, Superheated Steam, Saturated Water

According to the temperature-entropy diagram analysis of Section 5.3.2, in this case,

$$w_{cond} = 0 \qquad (A-35)$$

and

$$h_2 = h_f . \qquad (A-36)$$

Similarly to the procedure described in the previous cases, substitution of Eqs. (A-8) and (A-9) into (A-7) incorporating conditions (A-35) and (A-36) above leads to

$$(\frac{1}{\rho_1})\dot{M}_1 + (\frac{1}{\rho_f})\dot{M}_2 - (\frac{1}{\rho_1})^2 M_1\left[\left[\frac{\partial\rho_1}{\partial p}\right]_h \dot{p} + \left[\frac{\partial\rho_1}{\partial h}\right]_p \dot{h}_1\right] -$$

$$- (\frac{1}{\rho_f})^2 M_2 \left[\frac{d\rho_f}{dp}\right]_{sat} \dot{p} = 0 .$$

(A-37)

Substituting (A-1) into (A-3) for the present case yields

$$\dot{h}_f = \frac{1}{M_2}[w_{fd}(h_{fd}-h_f) + (1-x_r)w_r(h_f-h_2) - w_{f1}(h_g-h_f) +$$

$$+ (\frac{M_2}{\rho_f})\dot{p} + \dot{Q}_2] \qquad (A-38)$$

and

$$\dot{h}_f = \left[\frac{dh_f}{dp}\right]_{sat} \frac{dp}{dt} . \qquad (A-39)$$

Combining the immediately preceding two expressions and solving them for w_{f1} gives

$$w_{f1} = \dot{p}\left[\frac{M_2}{h_{fg}}\left[\frac{1}{\rho_f} - \left(\frac{dh}{dp}\right)_{sat}\right]\right] + w_{fd}\left[\frac{h_{fd}-h_f}{h_{fg}}\right] + \frac{\dot{Q}_2}{h_{fg}} . \qquad \text{(A-40)}$$

As in the previous case, this expression for w_{f1} is inserted into the two mass balances (A-1) and (A-2). The resulting expressions for $\dot{M}_1$ and $\dot{M}_2$ incorporating Eq. (A-22) are substituted into Eq. (A-24) which can then be solved for $\dot{p}$, yielding,

$$\dot{p} = (\text{pnum} - w_s/\rho_1) \ / \ \text{pden} \qquad \text{(A-41)}$$

where in this instance

$$\text{pnum} = \left[w_r x_r + w_{fd}\frac{h_{fd}-h_f}{h_{fg}} + \frac{\dot{Q}_2}{h_{fg}} \ / \ \rho_1\right] +$$

$$+ \left[(1-x_r)w_r + w_{fd} - w_d - w_{fd}\frac{h_{fd}-h_f}{h_{fg}} - \frac{\dot{Q}_2}{h_{fg}}\right] / \ \rho_f \ -$$

$$- \left[\frac{1}{\rho_1}\right]^2 \left[\frac{\partial \rho_1}{\partial h}\right]_p \left[\left[w_r x_r + w_{fd}\frac{h_{fd}-h_f}{h_{fg}} + \frac{\dot{Q}_2}{h_{fg}}\right](h_g - h_1) - \dot{Q}_1\right]$$

(A-42)

and

$$\text{pden} = M_2\left\{\frac{1}{h_{fg}}\left[\left[\frac{dh_f}{dp}\right]_{sat} - \frac{1}{\rho_f}\right]\left[\frac{1}{\rho_1} - \frac{1}{\rho_f} - \left[\frac{1}{\rho_1}\right]^2\left[\frac{\partial\rho_1}{\partial h}\right]_p(h_g - h_1)\right] + \left[\frac{1}{\rho_f}\right]^2\left[\frac{d\rho_f}{dp}\right]_{sat}\right\} + M_1\left\{\left[\frac{1}{\rho_1}\right]^2\left[\frac{\partial\rho_1}{\partial p}\right]_h + \left[\frac{1}{\rho_1}\right]^3\left[\frac{\partial\rho_1}{\partial h}\right]_p\right\}. \quad \text{(A-43)}$$

When it is the case that $\dot{p}$ is given and the exit steam flow rate is wanted, this quantity can be obtained from Eq. (A-41) written in the form

$$w_s = \rho_1(\text{pnum} - \dot{p} \cdot \text{pden}) . \quad \text{(A-44)}$$

The remaining equations are analogous to those for Case 2.

5. <u>Case 4 - Falling Pressure, Saturated Steam, Saturated Water</u>

The starting point for this case, as shown in Table 5.3.1 and discussed in Chapter 5 is

$$h_1 = h_g \quad \text{(A-45)}$$

$$h_2 = h_f . \quad \text{(A-46)}$$

With these pre-conditions, inserting Eqs. (A-8) and (A-9) into (A-7) yields:

$$(\frac{1}{\rho_g})\dot{M}_1 + (\frac{1}{\rho_f})\dot{M}_2 - (\frac{1}{\rho_g})^2 M_1 \left[\frac{d\rho_g}{dp}\right]_{sat} \dot{p} - (\frac{1}{\rho_f})^2 M_2 \left[\frac{d\rho_f}{dp}\right]_{sat} \dot{p} = 0 \; . \tag{A-47}$$

Following the same procedure described in the two previous cases, it can be obtained

$$w_{fl} = \dot{p}\left\{\frac{M_2}{h_{fg}}\left[\frac{1}{\rho_f} - (\frac{dh_f}{dp})_{sat}\right]\right\} + \frac{\dot{Q}_2}{h_{fg}} \tag{A-48}$$

and

$$w_{cond} = -\dot{p}\left\{\frac{M_1}{h_{fg}}\left[\frac{1}{\rho_g} - (\frac{dh_g}{dp})_{sat}\right]\right\} - \frac{\dot{Q}_1}{h_{fg}} \; . \tag{A-49}$$

When the above expressions are substituted into the mass balances yielding expressions for $\dot{M}_1$ and $\dot{M}_2$, those formulae can, in turn, be inserted into Eq. (A-47), which when solved for $\dot{p}$ becomes of the form:

$$\dot{p} = [pnum - w_s/\rho_g] \; / \; pden \tag{A-50}$$

where

$$pnum = w_r x_r + \left[\left(\frac{\dot{Q}_1 + \dot{Q}_2}{h_{fg}} \right) \right] / \rho_g + \left[(1-x_r) w_r + w_{fd} - w_d - \frac{\dot{Q}_1 + \dot{Q}_2}{h_{fg}} \right] / \rho_f \tag{A-51}$$

and

$$pden = M_1 \left\{ \frac{\frac{1}{\rho_g} + \frac{1}{\rho_f}}{h_{fg}} \left[\left(\frac{dh_g}{dp} \right)_{sat} - \frac{1}{\rho_g} \right] + \left(\frac{1}{\rho_g} \right)^2 \left(\frac{d\rho_g}{dp} \right)_{sat} \right\} +$$

$$+ M_2 \left\{ \frac{\frac{1}{\rho_g} + \frac{1}{\rho_f}}{h_{fg}} \left[\left(\frac{dh_f}{dp} \right)_{sat} - \frac{1}{\rho_f} \right] + \left(\frac{1}{\rho_f} \right)^2 \left(\frac{d\rho_f}{dp} \right)_{sat} \right\} . \tag{A-52}$$

When the pressure is given, the exit steam flow rate is determined from Eq. (A-50), written in the form,

$$w_s = \rho_g (pnum - \dot{p} \cdot pden) . \tag{A-53}$$

In this case the set of equations is completed with

$$T_1 = T_{sat} \tag{A-54}$$

and

$$T_2 = T_{sat} \,. \tag{A-55}$$

APPENDIX B

DERIVATION OF THE EXPRESSIONS FOR STRUCTURE AND LIQUID HEAT CONTENT

1. Structure Modeled as a One-Dimensional Heat Sink

For an infinite slab of thickness l, as the one shown in Fig. 5.3-2(a), with boundary conditions:

1) no heat flow at x=0 (insulated), and
2) temperature $T_b(t)$ at x=l,

the temperature distribution along x at any time t is given by (C2):

$$(T(x,t) = \frac{2}{l} \sum_{k=0}^{\infty} \exp\left[-\alpha(2k+1)^2 \pi^2 t/4l^2\right] \cos \frac{(2k+1)\pi x}{2l} \cdot$$

$$\left\{\frac{(2k+1)\pi\alpha(-1)^k}{4l}\int_0^t \exp[\alpha(2k+1)^2\pi^2 t'/4l^2] T_b(t')dt'\right\} \quad \text{(B-1)}$$

where T(x,t) is the temperature above base steady-state temperature and α is the thermal diffusivity of the slab.

The heat content of the slab at any time will be given by:

$$Q_1(t) = \int_0^l \rho C_p T(x,t) A \, dx = \rho A C_p \int_0^l T(x,t) \, dx \quad \text{(B-2)}$$

where ρ and C_p are slab density and specific heat respectively, assumed constant with temperature.

Evidently, the problem is to integrate the temperature distribution, i.e.,

$$I(t) = \int_0^1 T(x,t)dx = \frac{2}{T} \sum_{k=0}^{\infty} e^{-a_k t} \left[\sqrt{a_k \alpha}\ (-1)^k \int_0^t e^{a_k t'} T_b(t')dt' \right] .$$

$$\int_0^1 \cos \left[\sqrt{a_k / \alpha}\ x \right] dx \qquad \text{(B-3)}$$

where,

$$a_k = \left[\frac{(2k+1)}{21} \pi \right]^2 \alpha \ . \qquad \text{(B-4)}$$

Integration of the cosine in (B-3) yields:

$$\int_0^1 \cos(\sqrt{a_k / \alpha}\ x)dx = \sin\left[(2k+1)\frac{\pi}{2}\right] / \sqrt{a_k / \alpha} = (-1)^k / \sqrt{a_k / \alpha}\ . \qquad \text{(B-5)}$$

Therefore,

$$I(t) = \frac{2}{T} \sum_{k=0}^{\infty} e^{-a_k t} \alpha \int_0^t e^{a_k t'} T_b(t')dt' . \qquad \text{(B-6)}$$

The next difficulty in line involves the solution of the integral in the expression above. In the present approach, the time span from zero to t is broken into equal subintervals of magnitude Δt such that:

$$(t-0) = N.\Delta t . \qquad \text{(B-7)}$$

Thus, N is the total number of time subintervals within the time span of interest.

Next in line, the boundary temperature is assumed to vary linearly within each time interval. For example, for $(n-1)\Delta t < t < n\Delta t$, it is assumed that:

$$T_b(t) = T_b[(n-1)\Delta t] + \frac{T_b(n\Delta t) - T_b[(n-1)\Delta t]}{\Delta t} [t - (n-1)\Delta t] . \qquad \text{(B-8)}$$

For the same interval indicated above, the integral of (B-6) can be solved by parts:

$$\int_{(n-1)\Delta t}^{n\Delta t} e^{a_k t} T_b(t)\, dt = \frac{1}{a_k}\left\{\left[T_b(t)e^{a_k t}\right]_{(n-1)\Delta t}^{n\Delta t} - \right.$$

$$\left. -\int_{(n-1)\Delta t}^{n\Delta t} \frac{dT_b(t)}{dt} \cdot e^{a_k t}\, dt\right\} \qquad \text{(B-9)}$$

and substituting Eq. (B-8) into (B-9) results in:

$$\int_{(n-1)\Delta t}^{n\Delta t} e^{a_k t} T_b(t)dt = \frac{1}{a_k}\left\{T_b(n\Delta t)e^{a_k n\Delta t} - T_b[(n-1)\Delta t]e^{a_k(n-1)\Delta t}\right\} -$$

$$- \frac{T_b[n\Delta t] - T_b[(n-1)\Delta t]}{a_k^2 \Delta t}\left[e^{a_k n\Delta t} - e^{-a_k(n-1)\Delta t}\right] =$$

$$= T_b[n\Delta t]\left\{\frac{e^{a_k n\Delta t}}{a_k^2 \Delta t}\left[(a_k\Delta t-1) + e^{-a_k\Delta t}\right]\right\} -$$

$$- T_b[(n-1)\Delta t]\left\{\frac{e^{a_k(n-1)\Delta t}}{a_k^2 \Delta t}\left[(a_k\Delta t+1) - e^{a_k\Delta t}\right]\right\} \qquad \text{(B-10)}$$

At this point it is convenient to define:

$$A_n = \frac{e^{a_k n\Delta t}}{a_k^2 \Delta t}\left[(a_k\Delta t-1) + e^{-a_k\Delta t}\right] \tag{B-11}$$

and

$$B_{n-1} = \frac{e^{a_k(n-1)\Delta t}}{a_k^2 \Delta t}\left[(a_k\Delta t+1) - e^{a_k\Delta t}\right]. \tag{B-12}$$

From Eqs. (B-2), (B-3), and (B-6) it can be written

$$Q_1(t) = \alpha \cdot \frac{2}{T} \cdot \rho \cdot A \cdot C_p \sum_{k=0}^{\infty} e^{-a_k t} \int_o^t e^{a_k t'} T_b(t')\, dt' \tag{B-13}$$

which, when incorporating the result of (B-10) with the definitions of (B-11) and (B-12), can be cast in the form:

$$Q_1(N\Delta t) = K\sum_{k=o}^{\infty} e^{-a_k N\Delta t}\left\{\sum_{n=0}^{N} T_b(n\Delta t)A_n - T_b[(n-1)\Delta t]B_{n-1}\right\} \tag{B-14}$$

where evidently,

$$K = \frac{2}{l} \cdot \alpha \cdot \rho \cdot A \cdot C_p = \frac{2A}{l} K' \tag{B-15}$$

with K' the thermal conductivity of the solid.

Expanding the summation of Eq. (B-14), it can be seen that:

$$Q_1(N\Delta t) = K\sum_{k=0}^{\infty} e^{-a_k N\Delta t}[T_b(\Delta t)A_1 - T_b(0)B_o + T_b(2\Delta t)A_2 - T_b(\Delta t)B_1 +$$

$$+ \ldots + T_b(N\Delta t)A_N - T_b[(N-1)\Delta t]B_{N-1} =$$

$$= \left\{K\sum_{k=0}^{\infty} e^{-a_k N\Delta t}\right\}\left\{T_b(\Delta t)(A_1-B_1) + T_b(2\Delta t)(A_2-B_2) + \right.$$

$$\left. + \ldots + T_b[(N-1)\Delta t]\ (A_{N-1}-b_{N-1}) + T_b(N\Delta t)A_N\right\}. \tag{B-16}$$

The result

$$A_n - B_n = \frac{e^{na_k\Delta t}}{a_k^2\Delta t}\left[e^{-a_k\Delta t} + e^{a_k\Delta t} - 2\right] \tag{B-17}$$

from Eqs. (B-11) and (B-12), along with the fact that

$$T_b(0) = 0 \tag{B-18}$$

can be inserted into Eq. (B-16), which, after inverting the order of the summations can be made to read:

$$Q_1(N\Delta t) = \frac{K}{\Delta t}\left\{\sum_{n=0}^{N-1} T_b(n\Delta t) \sum_{k=0}^{\infty} e^{-(N-n)a_k\Delta t}\left[\frac{e^{a_k\Delta t} + e^{-a_k\Delta t} - 2}{a_k^2}\right] + \right.$$

$$\left. + T_b(N\Delta t) \sum_{k=0}^{\infty}\left[e^{-a_k\Delta t} + a_k\Delta t - 1\right]\right\} \tag{B-19}$$

which is the expression used to compute wall heat content at $t = N\Delta t$.

2. <u>The Liquid Modeled as a Semi-Infinite Heat Sink</u>

As shown in Fig. 5.3.2(b), the temperature distribution in the semi-infinite body is given by (C2):

$$T(x,t) = \frac{x}{2\sqrt{\pi\alpha}} \int_0^t T_b(t') \frac{e^{\frac{-x^2}{4\alpha(t-t')}}}{(t-t')^{3/2}} dt' . \tag{B-20}$$

In keeping with the same procedure of the previous section, the term of interest at this point is

$$\frac{x}{2\sqrt{\pi\alpha}} \int_{(n-1)\Delta t}^{n\Delta t} T_b(t') \frac{e^{\frac{-x^2}{4\alpha(t-t')}}}{(t-t')^{3/2}} dt' = \Delta T_N . \qquad \text{(B-21)}$$

A classical change of variable to the above integral is (C2):

$$\mu = \frac{x}{2\sqrt{\alpha(t-t')}} . \qquad \text{(B-22)}$$

Implementation of this change leads to:

$$\Delta T_N = \frac{2}{\sqrt{\pi}} \int_{x/2\sqrt{\alpha(N-n-1)\Delta t}}^{x/2\sqrt{\alpha(N-n-1)\Delta t}} T_b\left[N\Delta t - \frac{x^2}{4\alpha\mu^2}\right] e^{-\mu^2} d\mu . \qquad \text{(B-23)}$$

From Eq. (B-8) it can be written

$$T_b(t) = T_b[(n-1)\Delta t] - \frac{T_b(n\Delta t) - T_b[(n-1)\Delta t]}{\Delta t} \cdot (n-1)\Delta t +$$

$$+ \frac{T_b(n\Delta t) - T_b[(n-1)\Delta t]}{\Delta t} t . \qquad \text{(B-24)}$$

Therefore,

$$T_b\left[N\Delta t - \frac{x^2}{4\alpha\mu^2}\right] = C_1 + C_2 x^2 \mu^{-2} \quad \text{(B-25)}$$

where

$$C_1 = T_b[(n-1)\Delta t] + \frac{T_b(n\Delta t) - T_b[(n-1)\Delta t]}{\Delta t}(N-n-1)\Delta t \quad \text{(B-26)}$$

and

$$C_2 = \frac{T_b(n\Delta t) - T_b[(n-1)\Delta t]}{4\alpha\Delta t} . \quad \text{(B-27)}$$

In order to determine the stored energy in the liquid it is necessary to compute

$$\int_{X=0}^{X=\infty} \Delta T_N \, dx \quad \text{(B-28)}$$

where the integrand is given in Eq. (B-21).

To perform this integration consider first the term originated when Eq. (B-25) is inserted into (B-23) and which is multiplied by C_1, i.e.,

$$\int_0^{\infty} dx \int_{x/2\sqrt{(N-n-1)\Delta t\alpha}}^{x/2\sqrt{(N-1)\Delta t\alpha}} \exp(-\mu)^2 \, d\mu \qquad \text{(B-29)}$$

which upon integration by parts equals

$$\left[x \cdot \int_{x/2\sqrt{(N-n-1)\Delta t\alpha}}^{x/2\sqrt{(N-1)\Delta t\alpha}} e^{-\mu^2} d\mu \right]_0^{\infty} - \int_0^{\infty} x \left[\frac{e^{\frac{-x^2}{4\alpha(N-n)\Delta t}}}{2\sqrt{\alpha(N-n)\Delta t}} - \frac{e^{\frac{-x^2}{4\alpha(N-n-1)\Delta t}}}{2\sqrt{\alpha(N-n-1)\Delta t}} \right] dx.$$

(B-30)

The first term clearly goes to zero as $x \to 0$. As to the upper limit,

$$\lim_{x \to \infty} x \cdot \int_{x/2\sqrt{(N-n-1)\Delta t\alpha}}^{x/2\sqrt{(N-1)\Delta t\alpha}} e^{-\mu^2} \, d\mu = 0 \qquad \text{(B-31)}$$

a result which can be demonstrated by use of l'Hopital's rule,* and thus Eq. (B-29) becomes simply

---

*See NOTE at end of this Appendix.

$$\sqrt{\alpha(N-n-1)\Delta t} - \sqrt{\alpha(N-n)\Delta t} \qquad \text{(B-32)}$$

i.e. (B-29) = (B-32).

The next step in solving (B-28) is to compute

$$\int_{x/2\sqrt{(N-n+1)\Delta t\alpha}}^{x/2\sqrt{(N-n)\Delta t\alpha}} e^{-\mu^2} \, d\mu \qquad \text{(B-33)}$$

which upon integration by parts equals,

$$\frac{2\sqrt{\alpha(N-n+1)\Delta t}}{x} e^{\frac{-x^2}{4\alpha(N-n+1)\Delta t}} - \frac{2\sqrt{\alpha(N-n)\Delta t}}{x} e^{\frac{-x^2}{4\alpha(N-n)\Delta t}} -$$

$$- 2\int_{x/2\sqrt{\alpha(N-n+1)\Delta t}}^{x/2\sqrt{\alpha(N-n)\Delta t}} e^{-\mu^2} \, d\mu \, . \qquad \text{(B-34)}$$

From Eqs. (B-23), (B-25), and (B-28) it is clear that it is necessary to compute the integral of zero to infinity in dx of the product of each of the terms above by x^2. Consider initially,

$$-2\int_0^{\infty} x^2\left[\int_{x/2\sqrt{\alpha(N-n+1)\Delta t}}^{x/2\sqrt{\alpha(N-n)\Delta t}} e^{-\mu^2}\,d\mu\right]dx \qquad \text{(B-35)}$$

and break it up by parts into

$$-2\left\{\left(\frac{x^3}{3}\left[\int_{x/2\sqrt{\alpha(N-n+1)\Delta t}}^{x/2\sqrt{\alpha(N-n)\Delta t}} e^{-\mu^2}\,d\mu\right]\right)\Bigg|_0^{\infty} - \int_0^{\infty}\left[\frac{e^{\frac{-x^2}{4\alpha(N-n)\Delta t}}}{2\sqrt{\alpha(N-n)\Delta t}} - \frac{e^{\frac{-x^2}{4\alpha(N-n+1)\Delta t}}}{2\sqrt{\alpha(N-n+1)\Delta t}}\right]\left[\frac{x^3}{3}\right]dx\right\}. \qquad \text{(B-36)}$$

The first term of Eq. (B-36) is reduced to

$$\lim_{x\to\infty} \frac{x^3}{3}\int_{x/2\sqrt{\alpha(N-n+1)\Delta t}}^{x/2\sqrt{\alpha(N-n)\Delta t}} e^{-\mu^2}\,d\mu = 0 \qquad \text{(B-37)}$$

where the equality above can be demonstrated by l'Hopital's rule.* The second term of (B-36) can be integrated by parts yielding

*See NOTE at end of this Appendix.

$$\frac{8}{3}\,[\alpha(N-n)\Delta t]^{3/2} - \frac{8}{3}\,[(N-n+1)\Delta t\alpha]^{3/2}\;. \tag{B-38}$$

After some rearrangement it is possible to write Eq. (B-28) as

$$\int_0^\infty \Delta T_N\, dx = T_b[(n-1)\Delta t]D_{n-1} + T_b(n\Delta t)E_n \tag{B-39}$$

where

$$D_{n-1} = -\frac{2}{\sqrt{\pi}}\left[\left[\sqrt{\alpha(N-n+1)\Delta t} - \sqrt{\alpha(N-n)\Delta t}\right](N-n) + \right.$$

$$\left. + \frac{[\alpha(N-n+1)\Delta t]^{3/2} - [\alpha(N-n)]^{3/2}}{3\alpha\Delta t}\right] \tag{B-40}$$

$$E_n = \frac{2}{\sqrt{\pi}}\left[(N-n+1)\sqrt{\alpha(N-n+1)\Delta t} - \sqrt{\alpha(N-n)\Delta t} - \right.$$

$$\left. - \frac{[\alpha(N-n+1)\Delta t]^{3/2} - [\alpha(N-n)]^{3/2}}{3\alpha\Delta t}\right]. \tag{B-41}$$

The heat content of the liquid at time $t = N\Delta t$ will be given by

$$Q_2(N\Delta t) = \int_0^\infty \rho AC_p \Delta T_N dx = \rho AC_p \int_0^\infty \Delta T_N dx =$$

$$= \rho AC_p \Big[T_b(0)D_o + T_b(\Delta t)E_1 + T_b(\Delta t)D_1 + T_b(2\Delta t)E_2 +$$

$$+ \ldots + T_b[(N-1)\Delta t]D_{N-1} + T_b[N\Delta t]E_N\Big] =$$

$$= \rho AC_p \Big[T_b(\Delta t)(D_1+E_1) + T_b(2\Delta t)(D_2+E_2) +$$

$$+ \ldots + T_b[(N-1)\Delta t](D_{n-1}+E_{n-1}) + T_b[N\Delta t]E_n\Big]. \quad \text{(B-42)}$$

Clearly, it is interesting at this point to compute

$$D_n+E_n = \frac{4\sqrt{\alpha\Delta t}}{3\sqrt{\pi}} [(N-n+1)^{3/2} + (N-n-1)^{3/2} - 2(N-n)^{3/2}] \quad \text{(B-43)}$$

and to introduce this result into Eq. (B-42) to obtain

$$Q_2(N\Delta t) = \frac{4}{3}\sqrt{\frac{\alpha\Delta t}{\pi}}\,\rho A C_p \left\{ \sum_{n=1}^{N-1} T_b(N\Delta t)\left[(N-n+1)^{3/2} + (N-n-1)^{3/2} - 2(N-n)^{3/2}\right] + T_b(N\Delta t) \right\} \qquad \text{(B-44)}$$

which is the expression used in the text for the computation of the stored heat in the liquid at a time $t = N\Delta t$.

NOTE: The limit below has appeared twice in the text with p=1 and p=3, respectively,

$$\lim_{x \to \infty} x^p \int_{a_1 x}^{a_2 x} e^{-u^2}\,du \overset{①}{=} \lim_{x \to \infty} \frac{e^{-(a_2 x)^2} - e^{-(a_1 x)^2}}{(-p)x^{-(p+1)}} =$$

$$\overset{②}{=} A_1 \lim_{x \to \infty} \frac{x^q}{e^{+(a_2 x)^2}} \overset{③}{=} A_2 \lim_{x \to \infty} \frac{x^{q-2}}{e^{+(a_2 x)^2}} = 0$$

The equality labeled ① is an application of l'Hopital's rule. Equality ② is the separation of the numerator into each of its components. The component involving a_1 has been omitted since it is of the same form as the one under discussion. Stage ③ is one more application of the rule to show the point, namely that numerator will eventually go to 1 while denominator goes to infinity regardless of the number of derivations.

APPENDIX C

DERIVATION OF THE PRIMARY COOLANT TEMPERATURE EQUATION

From Ref. (B5), the energy equation is available in the form:

$$\frac{\partial}{\partial t}(\rho c T) = -\vec{\nabla} \cdot \rho c T\vec{v} - \vec{\nabla} \cdot \vec{q} - (\tau : \nabla\vec{v}) + \left(\frac{\partial \ln V}{\partial \ln T}\right)\frac{Dp}{Dt} + \rho T\frac{Dc}{Dt} \quad \text{(C-1)}$$

In applying this equation to the flow of pressurized water, within a steam generator primary tube, several simplifications can be made.

Viscous dissipation and compression work are small compared to the convective term, i.e.:

$$\vec{\tau} : \vec{\nabla} \cdot \vec{v} \ll \vec{\nabla} \cdot \rho c T\vec{v}$$

$$\left(\frac{\partial \ln V}{\partial \ln T}\right)\frac{Dp}{Dt} \ll \vec{\nabla} \cdot \rho c T\vec{v} \; .$$

These terms are accordingly dropped from Eq. (C-1).

Specializing for one-dimensional flow along the coordinate s yields:

$$\frac{\partial\ \rho cT}{\partial t} + \frac{\partial\ GcT}{\partial s} - \rho T\frac{\partial c}{\partial t} - GT\frac{\partial c}{\partial s} = - \vec{\nabla} . \vec{q} \qquad \text{(C-2)}$$

where the mass flux

$$\vec{G} = \rho\vec{v}$$

has been introduced and the vectorial notation abandoned for simplicity. The power per unit volume leaving the fluid,

$$\vec{\nabla} . \vec{q}$$

can be expressed in terms of the local heat flux, q", leaving the primary coolant, by use of the divergence theorem, i.e.

$$\int_0^{r_p} rdr \int_0^{2\pi} d\theta \int_{z_1}^{z_2} dz\ \vec{\nabla} . \vec{q} = \int_{z_1}^{z_2} dz \int_0^{r_p} dr \int_0^{2\pi} d\theta\ q''$$

which gives:

$$\frac{\int \vec{\nabla} . \vec{q}\ dV}{\int dV} = \frac{2}{r_p}\ q_p'' , \qquad \text{(C-3)}$$

where q_p'' is the average heat flux leaving the primary coolant at the inner wall, i.e.,

$$q_p'' = \frac{1}{2\pi r_\rho} \int_0^{r_p} dr \int_0^{2\pi} d\theta \, q''$$

Eqs. (C-2) and (C-3) can be combined to read:

$$\frac{\partial \, \rho c T}{\partial t} + \frac{\partial \, G c T}{\partial s} - \rho T \frac{\partial c}{\partial t} - G T \frac{\partial c}{\partial s} = - \frac{\partial}{r_p} q_p'' \qquad \text{(C-4)}$$

with the understanding that the quantities on the left hand side are assumed to be uniform throughout the cross section of the tube and the volume integration step is omitted for the sake of brevity.

Eq. (C-4) can be further simplified by noting that:

$$\frac{\partial \, \rho c t}{\partial t} - \rho T \frac{\partial c}{\partial t} = c \frac{\partial \, \rho T}{\partial t} ,$$

$$\frac{\partial \, G c t}{\partial s} - G T \frac{\partial c}{\partial s} = c \frac{\partial \, G T}{\partial s}$$

and

$$\frac{\partial \, G T}{\partial s} + \frac{\partial \, \rho T}{\partial t} = \rho \frac{\partial T}{\partial t} + G \frac{\partial T}{\partial s}$$

where the last equation incorporates continuity, i.e.:

$$\frac{\partial \rho}{\partial t} + \frac{\partial G}{\partial s} = 0 \ .$$

Making use of the above expressions, the equation for the primary temperature within one tube is written in form

$$\rho c\frac{\partial T}{\partial t} + cG\frac{\partial T}{\partial s} = - \frac{2}{r_p} q_p'' \qquad \text{(C-5)}$$

which is the form used throughout this work.

APPENDIX D

DERIVATION OF THE TEMPERATURE EQUATION FOR A TUBE BANK

In summing the terms in Eq. (6.2-1) over the Ntb_i tubes in tube bank i it is convenient to define:

$$\textcircled{I} = \sum_j \rho_j c_j \left(\frac{\partial T}{\partial t}\right)_j \tag{D-1}$$

$$\textcircled{II} = \sum_j \left(Gc\frac{\partial T}{\partial s}\right)_j \tag{D-2}$$

$$\textcircled{III} = - \sum_j (q_p'')_j \tag{D-3}$$

where the summation limits, j=1 to $j=Ntb_i$, are omitted to simplify notation.

Applying Eq. (6.2-2) to the three summations above yields for the first term:

$$\textcircled{I} = \sum (\bar{\rho}+\rho')(\bar{c}+c') \frac{\partial}{\partial t} (\bar{T}+T') =$$

$$= Ntb_i \cdot \bar{\rho} \cdot \bar{c} \cdot \frac{\partial \bar{T}}{\partial t} + \bar{c} \frac{\partial \bar{T}}{\partial t} \sum \rho' + \bar{\rho} \frac{\partial \bar{T}}{\partial t} \sum c' +$$

$$+ \bar{\rho}\, \bar{c} \frac{\partial}{\partial t} \sum T' + \frac{\partial \bar{T}}{\partial t} \sum \rho' c' + \bar{c} \sum \rho' \frac{\partial T'}{\partial t} +$$

$$+ \bar{\rho} \sum c' \frac{\partial T'}{\partial t} + \sum \rho' c' \frac{\partial T'}{\partial t} \qquad \text{(D-4)}$$

But, by definition,

$$\sum_{i=1}^{Ntb_i} \phi' = 0 \; . \qquad \text{(D-5)}$$

Furthermore, it can be written:

$$\phi' = \frac{\partial \bar{\phi}}{\partial T} T' + \frac{\partial \bar{\phi}}{\partial p} p' \qquad \text{(D-6)}$$

The pressure term above is neglected for two reasons. First, the pressure dependence of ϕ = c or ρ is considerably smaller than the temperature dependence. Second, the pressure differences themselves between primary tubes within a given secondary-side cell (and consequently at the same axial level) are also minimal. Thus,

$$\phi' \cong \frac{\partial \bar{\phi}}{\partial T} \cdot T' \ . \qquad \text{(D-7)}$$

Introducing (D-7) and (D-5) into (D-4) yields,

$$\text{(I)} \quad = Ntb_i \cdot \bar{\rho} \cdot \bar{c} \cdot \frac{\partial T}{\partial t} + \frac{\partial \rho}{\partial T} \cdot \frac{\partial c}{\partial T} \cdot \frac{\partial T}{\partial t} \sum (T')^2 +$$

$$+ (\bar{c}\frac{\partial \rho}{\partial T} + \bar{\rho}\frac{\partial c}{\partial T}) \sum T' \frac{\partial T'}{\partial t} \qquad \text{(D-8)}$$

Similarly, for the second term,

$$\text{(II)} \quad = \sum G(\bar{c}+c')\frac{\partial T}{\partial s} = \sum G(\bar{c} + \frac{\partial c T'}{\partial T})\frac{\partial T}{\partial s} =$$

$$= \bar{c}\sum G\frac{\partial T}{\partial s} + \frac{\partial \bar{c}}{\partial T}\sum GT'(\frac{\partial T}{\partial s}) \qquad \text{(D-9)}$$

And finally, for the last term,

$$\textcircled{III} = -\sum q_p'' = \sum h(T_w - T) = \sum(\bar{h}+h')[(\bar{T}_w + T_w') - (\bar{T}+T')] =$$

$$= (\bar{T}_w - \bar{T})\sum(\bar{h}+h') + \sum(T_w' - T')(\bar{h}+h') \ .$$

However, since

$$h' = \frac{\partial h}{\partial G} \cdot G' + \frac{\partial h}{\partial T} \cdot T' \ ,$$

it can be written:

$$\textcircled{III} = (\bar{T}_w - \bar{T})\sum(\bar{h} + \frac{\partial h}{\partial G} \cdot G') + (\bar{T}_w - \bar{T})\frac{\partial h}{\partial T}\sum T' +$$

$$+ \bar{h}\sum(T_w') - \bar{h}\sum(T') + \sum(T_w' - T')h' =$$

$$= [\sum(\bar{h} + \frac{\partial h}{\partial G} G')]\,(\bar{T}_w - \bar{T}) + \sum h'(T_w' - T') \qquad \text{(D-10)}$$

Combining Eqs. (D-8), (D-9), and (D-10) yields:

$$(Ntb_i)\ \bar{\rho}_i \bar{c}_i \left(\frac{\partial \bar{T}}{\partial t}\right)_i + \bar{c}_i \sum_{j=1}^{Ntb_i} G_j \left(\frac{\partial T}{\partial s}\right)_j =$$

$$= \frac{2}{r_p} \left[\sum_{j=1}^{Ntb_i} \left(\bar{h}_i + \frac{\partial h}{\partial G} \cdot G_j^{'}\right)\right] (\bar{T}_{wp} - \bar{T}) + \frac{2}{r_p} q^{''}(T') \tag{D-11}$$

where

$$q''(T') = \left[\frac{\partial \bar{\rho}}{\partial \bar{T}} \cdot \frac{\partial \bar{c}}{\partial \bar{T}} \cdot \frac{\partial \bar{T}}{\partial t} + \frac{1}{2} \frac{\partial}{\partial \bar{T}} (\bar{\rho}\ \bar{c}) \cdot \frac{\partial}{\partial t}\right] \sum_{j=1}^{Ntb_i} (T')^2 -$$

$$- \sum_{j=1}^{Ntb_i} h_j^{'}(T_{wpj}^{'} - T_j^{'}) - \left(\frac{\partial \bar{c}}{\partial \bar{T}}\right) \sum_{j=1}^{Ntb_i} G_j T_j^{'} \left(\frac{\partial T}{\partial s}\right)_j \tag{D-12}$$

Consider the convective term in Eq. (D-11)

$$S_c = \bar{c}_i \sum_j G_j \left(\frac{\partial T}{\partial s}\right)_j \tag{D-13}$$

where the summation limits are omitted to simplify the notation. The mass flux and the temperature gradient are still individual tube parameters. In order to simplify this expression it is assumed that

$$(\frac{\partial T}{\partial s})_j \sim (\frac{\partial \overline{T}}{\partial s})_i \qquad \text{(D-14)}$$

namely, the temperature gradient along each tube can be approximated by the average temperature gradient along the tube bank path. This would be strictly true if the overall heat transfer coefficient from primary to secondary were the same for all tubes within a tube bank. Although this condition is sufficient because secondary-side parameters are identical for tubes in the same bank, the restriction is not a necessary one, since Eq. (D-14) is taken as an approximation rather than an identity. With this approximation the convective term can now be written:

$$S_c = \overline{c}_i (\frac{\partial \overline{T}}{\partial s})_i \sum_j G_j \qquad \text{(D.15)}$$

The summation above must now be evaluated in terms of the average mass flux for the primary, $\overline{G}$. To accomplish this the pressure gradient along a primary tube is written in the form:

$$\frac{dp}{ds} = (\frac{dp}{ds})_f + (\frac{dp}{ds})_a + (\frac{ds}{dp})_b \qquad (D-16)$$

where f stands for friction, a for acceleration, and b for buoyancy. Because the length to diameter ratio for the primary tubes is high, the friction component is taken to dominate. When this is the case, Eq. (D-16) reduces to:

$$\frac{dp}{ds} \sim (\frac{dp}{ds})_f = K(\frac{GD}{\mu})^{-m} \; \frac{1}{D} \frac{\nu}{\mu} G^2 = K' \frac{\nu}{\mu^{1-m}} G^{2-m} \qquad (D-17)$$

where K and K' are constants.

The total pressure drop across a primary tube, j, of length, L_j, can be obtained by integration of Eq. (D-17), viz.,

$$\Delta p_j = K' \int_0^{L_j} \frac{\nu}{\mu^{1-m}} \; G^{2-m} \, ds \; . \qquad (D-18)$$

Defining an average mass flux along the tube length as

$$G_j^{2-m} = \frac{\int_o^{L_j} \left(\frac{\nu}{\mu^{1-m}}\right) G^{2-m}\, ds}{\int_o^{L_j} \left(\frac{\nu}{\mu^{1-m}}\right) ds} \qquad \text{(D-19)}$$

and defining:

$$\left[\frac{\nu}{\mu^{1-m}}\right]_j = \frac{\int_o^{L_j} \left(\frac{\nu}{\mu^{1-m}}\right) ds}{\int_o^{L_j} ds} \qquad \text{(D-20)}$$

it can be written:

$$\Delta p_j = K' \left[\frac{\nu}{\mu^{1-m}}\right]_j G_j^{2-m} L_j \;. \qquad \text{(D-21)}$$

Assuming that:

$$\nu = \nu_{in} + \lambda_\nu (T - T_{in}) = \lambda_\nu T + (\nu_{in} - \lambda_\nu T_{in}) = \lambda_\nu T + \nu_o \qquad \text{(D-22)}$$

and

$$\mu = \lambda_\mu T + \mu_o \qquad \text{(D-23)}$$

with $\lambda_\nu = \frac{\partial \nu}{\partial T}$ and $\lambda_\mu = \frac{\partial \mu}{\partial T}$ approximately constant in the range $T = T_{in}$ to $T = T_{in} + \Delta T$ where ΔT is the temperature drop of a primary tube. Given these assumptions, it can be written:

$$\int_0^{L_j} \left(\frac{\nu}{\mu^{1-m}}\right) \partial s = \int_{T_{in}}^{T_{out}} \frac{\lambda_\nu T + \nu_o}{(\lambda_\mu T + \mu_o)^{1-m}} \frac{\partial s}{\partial T} \cdot dT =$$

$$= \left(\overline{\frac{\partial s}{\partial T}}\right) \int_{T_{in}}^{T_{out}} \frac{\lambda_\nu T + \nu_o}{(\lambda_\mu T + \mu_o)^{1-m}} \, dT =$$

$$= \left(\overline{\frac{\partial s}{\partial T}}\right) \left\{ \frac{\lambda_\nu}{(m+1)\lambda_\mu^2} \left[\mu_{out}^{m+1} - \mu_{in}^{m+1}\right] + \frac{\lambda_\mu \nu_{in} - \lambda_\nu \mu_{in}}{m\lambda_\mu^2} \left[\mu_{out}^{m} - \mu_{in}^{m}\right] \right\} =$$

$$= \left(\overline{\frac{\partial s}{\partial T}}\right) F(\Delta T) \qquad \text{(D-24)}$$

where F(ΔT) is unambiguously defined above. Numerical calculations for typical temperatures shown in Fig. (D-1) reveal that:

$$F(\Delta T) \cong C \,.\, \Delta T \qquad \text{(D-25)}$$

where C is, for all practical purposes, constant.

Assuming,

$$\overline{\left(\frac{\partial s}{\partial T}\right)} \sim \frac{\Delta s}{\Delta T} = \frac{L_j}{\Delta T} \qquad \text{(D-26)}$$

it can be seen that

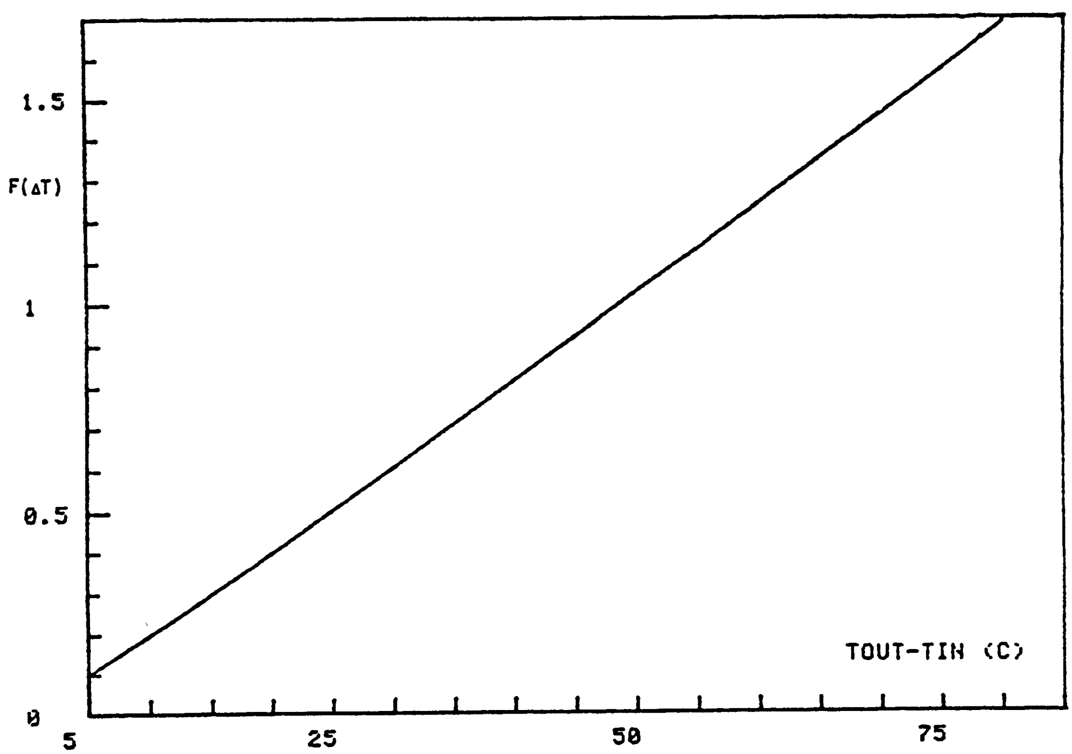
$$\int_o^{L_j} \left(\frac{\nu}{\mu^{1-m}}\right) \partial s = \frac{L_j}{\Delta T} \,.\, C \,.\, \Delta T = C \quad L_j \,. \qquad \text{(D-27)}$$

Thus,

$$\left[\frac{\nu}{\mu^{1-m}}\right]_j = \frac{\int_o^{L_j} \left(\frac{\nu}{\mu^{1-m}}\right) \partial s}{\int_o^{L_j} \partial s} = \frac{c \;\, L_j}{L_j} = c \qquad \text{(D-28)}$$

Figure D-1

Numerical demonstration of Eq. (D-25)
where F(ΔT) is in SI units.



Therefore,

$$\frac{\Delta p_k}{\Delta p_j} = \frac{G_k^{2-m}\ L_k}{G_j^{2-m}\ L_j} \ . \qquad \text{(D-29)}$$

Since the pressure drop must be the same for all the N tubes in the UTSG, it can be written:

$$L_1\ G_1^{2-m} = L_2\ G_2^{2-m} = \dots = L_N G_N^{2-m} \ . \qquad \text{(D-30)}$$

Defining,

$$f_j = \frac{G_j}{\overline{G}} \ , \qquad \text{(D-31)}$$

where $\overline{G}$ is the average mass flux through the primary, i.e.,

$$\overline{G} = \frac{\text{PRIMARY MASS FLOW RATE}}{(\pi r_p^2)(\text{TOTAL NO. OF U-TUBES})} \ . \qquad \text{(D-32)}$$

It can be written:

$$L_1 f_1^{2-m} = L_2 f_2^{2-m} = \dots = L_N f_N^{2-m} \ , \qquad \text{(D-33)}$$

or, since $f_1 + f_2 + \ldots + f_N = N$,

$$f_1 \left[\left(\frac{L_1}{L_1}\right)^{\frac{1}{2-m}} + \left(\frac{L_1}{L_2}\right)^{\frac{1}{2-m}} + \ldots + \left(\frac{L_1}{L_N}\right)^{\frac{1}{2-m}} \right] = N , \qquad \text{(D-34)}$$

leading to:

$$f_j = N \left[\sum_{k=1}^{N} \left(\frac{L_j}{L_k}\right)^{\frac{1}{2-m}} \right]^{-1} . \qquad \text{(D-35)}$$

Utilizing the definition of Eq. (D-31), the convective term of Eq. (D-15) can now be written in the form

$$S_c = \bar{c}_i \left(\frac{\partial T}{\partial s}\right)_i \bar{G} \sum_j f_j \qquad \text{(D-36)}$$

where f_j is given in Eq. (D-35).

Next consider the remaining heat transfer term of Eq. (D-11):

$$S_{ht} = \frac{2}{r_p} \sum_j \left[\bar{h}_i + \left(\frac{\partial h}{\partial G}\right) G_j' \right] (T_{wp} - T) \qquad \text{(D-37)}$$

where the summation limits are again omitted for simplicity.

Since the heat transfer coefficient is given by an expression of the form

$$\overline{h} = c \cdot \left(\frac{\overline{G}r_p}{2\mu}\right)^{\lambda} (P_r)^{\lambda'} \left(\frac{2K}{r_p}\right) \tag{D-38}$$

it is easy to see that:

$$\frac{\partial \overline{h}}{\partial G} = \overline{h}\ \lambda/\overline{G}\ . \tag{D-39}$$

Furthermore, since

$$G_j^{'} = G_j - \overline{G}_i \tag{D-40}$$

and considering Eq. (D-31), it can be written:

$$G_j^{'} = \overline{G}_i\ (f_j - 1)\ . \tag{D-41}$$

Introducing Eqs. (D-41) and (D-39) into Eq. (D-37) results in:

$$S_{ht} = \frac{2}{rp} (T_{wp} - T)\, \bar{h} \left[(1-\lambda) Ntb_i + \lambda \sum_j f_j \right] . \qquad (D.42)$$

Combining Eqs. (D-42) and (D-36) into Eq. (D-11) and dividing throughout by the number of tubes in bank i, Ntb_i yields

$$\bar{\rho}_i \bar{c}_i \frac{\partial \bar{T}}{\partial t} + \bar{c}_i \left(\frac{\partial \bar{T}}{\partial s}\right)_i F_i \bar{G} = \frac{2}{r_p} \bar{h} (\bar{T}_{wp} - \bar{T})_i\, H_i \qquad (D-43)$$

where,

$$F_i = \frac{1}{Ntb_i} \sum_{j=1}^{Ntb_i} f_j \qquad (D-44)$$

and

$$H_i = 1 + \lambda (F_i - 1) . \qquad (D-45).$$

The parameter F_i distributes the flow among the tube banks and takes into consideration tube to tube flow differences based on an equal frictional pressure drop from inlet to outlet plenum. The parameter H_i accounts for variations in the primary heat transfer coefficient of the tube bank because of inter-tube differences in flow. Eq. (6.2-6) corresponds to Eq. (D-43) with the definitions of Eq. (D-44) and Eq. (D-45).

The flow split parameter F_i is numerically determined through the expression:

$$F_i = \frac{N}{Ntb_i} \frac{\sum_{j=iew_{min}(i)}^{iew_{max}(i)} (n_j \cdot L_j^{\frac{1}{m-2}})}{\sum_{j=1}^{iew_{max}(i_{max})} (n_j . L_j^{\frac{1}{m-2}})} \qquad \text{(D-46)}$$

where

$$N = \sum_{j=1}^{iew_{max}(i_{max})} n_j \qquad \text{(D-47)}$$

$$Ntb_i = \sum_{iew_{min}(i)}^{iew_{max}(i)} n_j \qquad \text{(D-48)}$$

and the iew's are defined in connection to Eq. (6.2-13) and in

Table D-1. The numerical computation of Eq. (D-46) is done in a Fortran routine, whose listing is given at the end of this Appendix. The input required for each computation is:

nc =	number of channels
nr =	number of rows of channels
j1 =	last axial level in the straight tube region
j2 =	last axial level in the bent tube region
nzp2 =	total number of axial levels
da =	tube alley width
p_i=	tube pitch direction i
ncr(i) =	number of channels in row i
dx, dy, dz	

Figure D-2 illustrates selected input parameters for the case of Arkansas Nuclear One which is given in the computer listing.

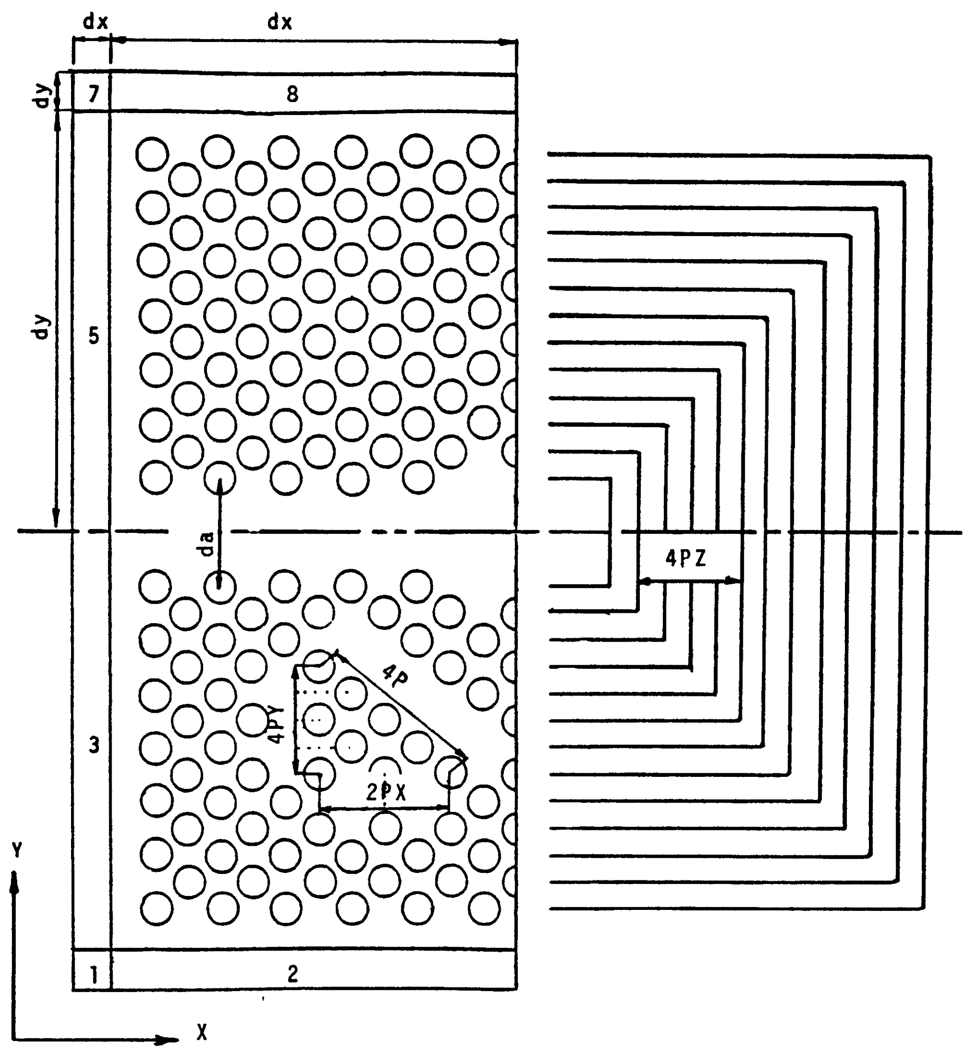
As shown in Fig. D-2, the routine assumes the tubes bend at 90° angles. This is a good approximation for Combustion Engineering designed units. For the case of Westinghouse units, in spite of the circular bend, it is believed that a reasonable approximation for the parameters given in Table D-1 can be obtained as well, provided the number of tubes is large. In any case, the total heat transfer area must be checked with design values and small adjustments can be made using good judgment. The routine is not recommended for use

Table D-1. Flow split routine output description.

PARAMETER	FORTRAN	DESCRIPTION
N	ntot	total number of U-tubes Eq. (D-47)
Ntb_i	new	number of tubes in tube bank i. Eq. (D-48)
$iew_{min}(i)$	i3	order number of first tube row to bend in a given tube bank
$iew_{max}(i)$	i4	idem of last tube row
F_i	Fi	flow split parameter Eq. (D-46)
Δsew	Dsew	horizontal tube bank length

Figure D-2.

Illustration of input for routine calculating flow split parameters.



with few tubes, particularly if the U-tubes bend along a circular path.

The program listing follows. The instructions up to label 10 define the input and should be updated for each different use. The routine output displays the relevant geometrical parameters defined in Table D-1. These quantities are then used to prepare the geometrical input for THERMIT-UTSG as described in Appendix F.

```
c (c) Calculation of The Flow Split Parameters

cTHIS PROGRAM DETERMINES THE NUMBER OF RODS PER SECONDARY SIDE CHANNEL
cAND THE NUMBER OF THESE RODS WHICH ENTER EACH CELL FROM THE EAST-WEST
cDIRECTION. THE FLOW SPLIT PARAMETER, Fi, AND THE HORIZONTAL ROD
cLENGTH, XLEW, ARE COMPUTED AS WELL.
c
c    THE INPUT ARE THE VARIABLES DEFINED UP TO LABEL 10
c
c    RESTRICTIONS;
c     1) DELY(NR/2) SHOULD NOT BE LESS THAN DA
c     2) THE NUMBER OF CELLS IN A ROW SHOULD BE EQUAL FOR ALL ROWS
cA SYMMETRICAL NUMBER FOR HOT AND COLD SIDES MAY WORK (CHECK)
c     3)MAKE SURE IZT(J2)=IYT(IROW(NR-1))
c  NOTE; CHANNEL NUMBERS DON'T NECESSARILY NEED TO
c  COINCIDE WITH THOSE TO BE USED IN THERMIT-UTSG.IN THIS
c  CALCULATION THE CHANNEL LAYOUT IS THAT OF FIG. D-2
c  WHILE IN THERMIT-UTSG THE ARRANGEMENT IS THAT OF FIG. 4.2-3.
c***********************************************************************
      data nc,nr,j1,j2,nzp2/8,4,7,10,15/
      data da,p,pz,xm/0.064,0.0254,0.030,0.2/
      dimension ncr(10),irow(30),i1(20,30),i2(20,30),i3(20,30),
     1dz(20),dy(30),izt(20),iyt(30),dely(30),i4(20,30),
     2n(20,30),new(20,30),f(20,30),few(20,30),rew(20,30),rns(20,30),
     3dx(30),nrx(100),xl(100),iside(30),rht(20,30),
     4dsew(20,30),xlew(20,30),xlavg(20,30),dsns(20,30),fl(20,30)
      data icase/1/
      data xlamb/0.8/
      x=1./(xm-2.)
      py=p/sqrt(2.)
      px=2.*py
      do 5 i=1,nr
5     ncr(i)=2
      do 7 i=1,nc-1,2
7     dx(i)=.9052
      do 71 i=2,nc,2
71    dx(i)=1.6911
      dz(1)=3.75
      dz(2)=.4064
      x10=dz(2)
      do 8 j=3,j1+1
      dz(j)=1.05225
8     x10=x10+dz(j)
      x10=x10-dz(8)
      do 83 j=j1+2,j2
83     dz(j)=.81863
      do 82 j=j2+1,nzp2
82     dz(j)=.60973
      dely(1)=.0952
      dely(2)=1.6911
      do 10 i=1,ncr(1)
10     irow(i)=1
```

```
      i6=ncr(1)
      do 25 j=2,nr
      i5=i6+1
      i6=i5+ncr(j)-1
      do 20 i=i5,i6
      irow(i)=j
20    continue
25    continue
      do 6 iii=1,nr/2
      i=nr-iii+1
6     dely(i)=dely(iii)
      dely(nr-1)=dely(nr-1)-py
      izt(j1)=1
      do 50 j=j1+1,j2
50    izt(j)=int(dz(j)/pz)+izt(j-1)
      xk=(dely(nr/2+1)-da)/py
      iyt(nr/2+1)=max(int(xk),0)+1
      if(xk.lt.0.) iyt(nr/2+1)=0
      k11=0
      if(xk.lt.0.) k11=1
      do 60 k=nr/2+2,nr
      iyt(k)=iyt(k-1)+int(dely(k)/py)+k11
60    k11=0
      do 30 i=nc/2+1,nc
      i1(j1+1,i)=1
      if(i.gt.(nc/2+ncr(nr/2+1))) i1(j1+1,i)=iyt(irow(i)-1)+1
30    continue
      do 36 i=nc/2+1,nc
      do 36 j=1,j1
      i2(j,i)=iyt(irow(i))
      if(irow(i).eq.irow(nc/2+1)) go to 37
      i1(j,i)=iyt(irow(i)-1)+1
      go to 36
37    i1(j,i)=1
36    i3(j,i)=0
      do 40 i=nc/2+1,nc
      do 40 j=j1+1,j2
      i2(j,i)=max(izt(j),iyt(irow(i)))
      if(j.ne.j1+1) i1(j,i)=max(izt(j-1),iyt(irow(i)-1))+1
      i3(j,i)=max(izt(j-1)+1,iyt(irow(i))+1)
      if(i3(j,i).gt.i2(j,i)) i3(j,i)=0
40    continue
      do 70 j=1,j2
      do 70 i=nc/2+1,nc
      isncr=0
      jsncr=0
      jjj=irow(i)-nr/2-1
      if(jjj)76,76,77
77    continue
      do 75 jj=1,jjj
75    jsncr=jsncr+ncr(irow(jj))
76    isncr=jsncr+ncr(irow(jjj+1))
      ii=i-jsncr-isncr
      if(i1(j,i).gt.i2(j,i)) i1(j,i)=0
```

```
      if(i1(j,i).eq.0) i2(j,i)=0
      i1(j,ii)=i1(j,i)
      i2(j,ii)=i2(j,i)
      i3(j,ii)=i3(j,i)
      iyt(irow(ii))=iyt(irow(i))
      k1=0
      if(irow(i)-1.gt.0) k1=iyt(irow(i)-1)
      k2=0
      if(j.gt.j1) k2=izt(j-1)
      i3(j,i)=max(k1,k2)+1
      if(irow(i).eq.nr/2+1) i3(j,i)=izt(j-1)+1
      if(j.eq.j1+1.and.irow(i).eq.nr/2+1) i3(j,i)=izt(j-1)
      i4(j,i)=izt(j)
      if(izt(j).lt.k1.and.irow(i).ne.nr/2+1) i4(j,i)=0
      if(i3(j,i).eq.0) i4(j,i)=0
      if(i.le.nc/2+ncr(nr/2+1)) i3(j,i)=i1(j,i)
      if(i3(j,i).gt.i4(j,i)) i4(j,i)=0
      if(j.le.j1) i4(j,i)=0
      if(i4(j,i).eq.0) i3(j,i)=0
      i4(j,ii)=i2(j,i)
      if(i3(j,ii).eq.0)i4(j,ii)=0
70    continue
      write(7,150)
      dely(nr-1)=dely(nr-1)+py
      do 80 i=ncr(1)+1,nc-ncr(nr)
      do 80 j=2,j2
      n(j,i)=0
      new(j,i)=0
      f(j,i)=0
      few(j,i)=0
      fl(j,i)=0.
      rew(j,i)=0
      rns(j,i)=0
      if(irow(i).ne.irow(i-1)) go to 80
      iside(i)=1
      if(irow(i).le.nr/2) iside(i)=0
      dsns(j,i)=iside(i)*(dz(j+1)+dz(j))/2+(1-iside(i))*(dz(j)+
     1dz(j-1))/2.
      nrx(i)=int((dx(i))/px)
      if(i1(j,i).eq.0) go to 351
      do 350 ii=i1(j,i),i2(j,i)
      xl(ii)=(xl0+da)*2.+2.*(py+pz)*(ii-1)
      if(icase.eq.2.or.icase.eq.4) xl(ii)=15.42
      n(j,i)=n(j,i)+nrx(i)
      f(j,i)=f(j,i)+nrx(i)*xl(ii)**x
350   continue
351   continue
      if(i1(j,i).eq.0)      write(6,210)
      if(i3(j,i).eq.0) go to 401
      do 400 iii=i3(j,i),i4(j,i)
      xl(iii)=(xl0+da)*2.+2.*(py+pz)*(iii-1)
      if(icase.eq.2.or.icase.eq.4) xl(iii)=15.42
      new(j,i)=new(j,i)+nrx(i)
      few(j,i)=few(j,i)+nrx(i)*xl(iii)**x
```

```
      fl(j,i)=fl(j,i)+nrx(i)*xl(iii)
400   continue
401   continue
80    continue
      do 312 i=nc/2+2,nc-ncr(nr)
      xlew(j1,i)=0.
     xlavg(j1,i)=2.*x10-dz(j1)
     do 312 j=j1+1,j2
      if(new(j,i).eq.0.) go to 313
      xlavg(j,i)=fl(j,i)/new(j,i)
      xlew(j,i)=xlew(j-1,i)-xlavg(j-1,i)+xlavg(j,i)-dz(j-1)-dz(j)
      dsew(j,i)=xlew(j,i)
      if(icase.eq.2.or.icase.eq.4) dsew(j,i)=(dely(irow(i))+dely(
     1irow(i)-1))/2.
312   continue
313   continue
      ntot=0
      ftot=0.
      do 402 i=ncr(1)+2,nc/2
      if(irow(i).ne.irow(i-1)) go to 402
      ntot=ntot+n(j1,i)
      ftot=ftot+f(j1,i)
402   continue
       write(7,153) ntot,ftot
      do 700 i=ncr(1)+2,nc-ncr(nr)
      do 700 j=2,j2
      if(irow(i).ne.irow(i-1).and.j.eq.2) go to 701
      if(irow(i).ne.irow(i-1).and.j.gt.2) go to 700
      if(ftot.eq.0.0.or.new(j,i).eq.0) go to 90
      rew(j,i)=ntot*few(j,i)/(ftot*new(j,i))
90    continue
      write(7,200)i,j,i1(j,i),i2(j,i),i3(j,i),i4(j,i),n(j,i),
     1new(j,i),ftot,few(j,i),rew(j,i),xlew(j,i)
      go to 700
  701  write(7,151)
  700  continue
100   format(9i10)
150   format(t7,'i',t17,'j',t23,'i1',t31,'i2',t39,'i3',t47,'i4',
     1t55,'n',t61,'new',t71,'f',t81,'few',t91,'Fi',t101,'Dsew')
151   format('**************************************************DOWNCOMER CE
     1LL(S)****************************************************************
     2***')
200   format(8i8,1pe10.3,1pe10.3,1pe10.3,1pe10.3,1pe10.3,1pe10.3)
210   format(1x,'da>iyt(irow(nr/2+1)): this can lead to error')
900   continue
153   format(6x,'ntot=',i5,93x,'ftot=',1pe10.3)
      end
```

i	j	i1	i2	i3	i4	n	new	f	few	Fi	Dsew
ntot= 4230											
4	2	1	90	0	0	4230	0	9.254E+02	0.000E+00	0.000E+00	0.000E+00
4	3	1	90	0	0	4230	0	9.254E+02	0.000E+00	0.000E+00	0.000E+00
4	4	1	90	0	0	4230	0	9.254E+02	0.000E+00	0.000E+00	0.000E+00
4	5	1	90	0	0	4230	0	9.254E+02	0.000E+00	0.000E+00	0.000E+00
4	6	1	90	0	0	4230	0	9.254E+02	0.000E+00	0.000E+00	0.000E+00
4	7	1	90	0	0	4230	0	9.254E+02	0.000E+00	0.000E+00	0.000E+00
4	8	1	90	0	0	4230	0	9.254E+02	0.000E+00	0.000E+00	0.000E+00
4	9	37	90	0	0	2538	0	9.254E+02	0.000E+00	0.000E+00	0.000E+00
4	10	64	90	0	0	1269	0	9.254E+02	0.000E+00	0.000E+00	0.000E+00
DOWNCOMER CELL(S)											
6	2	1	90	0	0	4230	0	9.254E+02	0.000E+00	0.000E+00	0.000E+00
6	3	1	90	0	0	4230	0	9.254E+02	0.000E+00	0.000E+00	0.000E+00
6	4	1	90	0	0	4230	0	9.254E+02	0.000E+00	0.000E+00	0.000E+00
6	5	1	90	0	0	4230	0	9.254E+02	0.000E+00	0.000E+00	0.000E+00
6	6	1	90	0	0	4230	0	9.254E+02	0.000E+00	0.000E+00	0.000E+00
6	7	1	90	0	0	4230	0	9.254E+02	0.000E+00	0.000E+00	0.000E+00
6	8	1	90	1	36	4230	1692	9.254E+02	4.055E+02	1.096E+00	7.544E-01
6	9	37	90	37	63	2538	1269	9.254E+02	2.707E+02	9.750E-01	1.905E+00
6	10	64	90	64	90	1269	1269	9.254E+02	2.492E+02	8.976E-01	2.858E+00

APPENDIX E

DISCRETIZATION OF THE HEAT CONDUCTION EQUATION FOR THE TUBE MODEL

The heat conduction equation for the tubes, in cylindrical coordinates, is:

$$\rho c \frac{\partial T}{\partial t} - \frac{1}{r} \frac{\partial}{\partial r} rK \frac{\partial T}{\partial r} = 0 \qquad \text{(E-1)}$$

where axial conduction is neglected and ρ, c, and K are density, heat capacity, and thermal conductivity of the tube metal respectively.

The discretization procedure consists of integrating Eq. (E-1) three times:

(1) from r_p to $r_{p'}$

(2) from $r_{p'}$ to $r_{m'}$

(3) from $r_{m'}$ to r_s.

The radii above are shown in Fig. 6.3-1.

Integration of the first term in Eq. (E-1) from r_p to $r_{p'}$ yields

$$\int_{r_p}^{r_{p'}} dr \cdot r \cdot \rho c \frac{\partial T}{\partial t} = (\rho c)^n_{p'} \frac{1}{2}(r^2_{p'} - r^2_p) \frac{1}{\Delta t} (T_{wp} - T^n_{wp}) \ . \qquad \text{(E-2)}$$

Integration of the second term yields:

$$\int_{r_p}^{r_{p'}} \partial[rK\frac{\partial T}{\partial r}] = r_{p'}K_{p'}^{n} \left(\frac{T_m - T_{wp}}{r_m - r_p}\right) - r_p \left(K\frac{\partial T}{\partial r}\right)_p . \tag{E-3}$$

However,

$$\left(K\frac{\partial T}{\partial r}\right)_p = - q_p'' . \tag{E-4}$$

Defining,

$$xp = \frac{1}{2\Delta t} (\rho c)_{p'}^{n} (r_{p'}^2 - r_p^2) \tag{E-5}$$

$$yp = r_{p'} K_{p'}^{n} (r_m - r_p)^{-1} \tag{E-6}$$

and substituting Eqs. (E-6), (E-5), (E-4), (E-3), and (E-2) into (E-1) leads to:

$$xp(T_{wp} - T_{wp}^{n}) - yp(T_m - T_{wp}) - r_p q_p'' = 0 \tag{E-7}$$

Following the same procedure for $r_{p'} < r < r_{m'}$ it can be written

$$\int_{r_{p'}}^{r_{m'}} r dr\ \rho c \frac{\partial T}{\partial t} = (\overline{\rho c})^n_{p',m'} \cdot \frac{1}{\Delta t} \cdot (T_m - T^n_m) \qquad \text{(E-8)}$$

with

$$(\overline{\rho c})^n_{p',m'} = (\rho c)^n_{p'} \cdot \frac{1}{2} \cdot (r^2_{m'} - r^2_m) + (\rho c)^n_{m'} \cdot \frac{1}{2} \cdot (r^2_m - r^2_{p'}) \qquad \text{(E-9)}$$

and

$$\int_{r_{p'}}^{r_{m'}} \partial [rK \frac{\partial T}{\partial r}] = (r_{m'} K_{m'})^n (\frac{T_{ws} - T_m}{r_s - r_m}) - (r_{p'} K_{p'})^n (\frac{T_m - T_{wp}}{r_m - r_p}) \ . \qquad \text{(E-10)}$$

Defining

$$xm = \frac{1}{\Delta t} (\overline{\rho c})^n_{p',m'} \qquad \text{(E-11)}$$

$$ym = (r_{m'} K_{m'})^n (r_s - r_m)^{-1} \qquad \text{(E-12)}$$

and substituting Eqs. (E-12), (E-11), (E-10), (E-9), (E-8), and (E-6) into (E-1) yields:

$$xm(T_m - T_m^n) - ym(T_{ws} - T_m) + yp(T_m - T_{wp}) = 0 \; . \qquad \text{(E-13)}$$

Again, repeating the procedure for $r_{m'} < r < r_s$, it can be written,

$$\int_{r_{m'}}^{r_s} r \, . \, dr \, . \, \rho c \, \frac{\partial T}{\partial t} = (\rho c)^n_{m'} \, . \, \frac{(r_s^2 - r_{m'}^2)}{2\Delta t} \, . \, (T_{ws} - T^n_{ws}) \qquad \text{(E-14)}$$

and

$$\int_{r_{m'}}^{r_s} \partial \, [rK \frac{\partial T}{\partial r}] = r_s (\frac{K \partial T}{\partial r})_s - (r_{m'} K_{m'})^n \, (\frac{T_{ws} - T_m}{r_s - r_m}) \qquad \text{(E-15)}$$

with,

$$(\frac{K \partial T}{\partial r})_s = - q_s'' \; . \qquad \text{(E-16)}$$

Defining,

$$xs = \frac{1}{2\Delta t}\,(r_s^2 - r_{m'}^2)\,\cdot\,(\rho_{m'} c_{m'})^n \qquad \text{(E-17)}$$

and substituting Eqs. (E-11), (E-16), (E-15), and (E-12) into (E-1) leads to,

$$xs(T_{ws} - T_{ws}^n) + r_s q_s'' + ym\,(T_{ws} - T_m) = 0\ . \qquad \text{(E-18)}$$

Equations (E-7), (E-13), and (E-18) can be combined and written together in the form:

$$\begin{bmatrix} (xp + yp) & -yp & 0 \\ -yp & (xm + ym + yp) & (-ym) \\ 0 & -ym & (xs + ym) \end{bmatrix} \begin{bmatrix} T_{wp} \\ T_m \\ T_{ws} \end{bmatrix} = \begin{bmatrix} xpT_{wp}^n + r_p q_p'' \\ xm\ T_m^n \\ xsT_{ws}^n - r_s q_s'' \end{bmatrix}$$

(E-20)

which constitutes the model used for wall conduction in this work.

APPENDIX F

THERMIT-UTSG INPUT DESCRIPTION AND EXAMPLES OF INPUT AND OUTPUT

The input required to run THERMIT-UTSG is divided into five parts. Each part contains one or more groups which are described in more detail below. It should be noted that the data for each group must begin on a new card. However, more than one card may be used for a particular group's data.

PART I - Overall Problem Description

The following cards are read via list-directed input (*-FORMAT). Fields are separated by one or more blanks or by commas. A null field can be specified by the occurrence of consecutive commas. Basically a constant (entered as a field) is assigned to the corresponding list element as if the constant were the right side of an assignment statement whose left side was the list element.

For additional details on the use of list-directed input, the user is referred to the Multics Fortran manual.

Group No.	Format	Contents
1	*	Job Control Indicator NTC NTC > 0 Number of title cards to be read (see Group 2). NTC = 0 The job is ended. NTC = -2 The job is a restart from a previously created dump file. NTC = -3 The job is a restart from a previously created dump file. Additionally, the real time and time step number are set to zero and the values for the boundary conditions are saved.
2	20A4	Title Cards (these are the only cards in fixed format)
3	*	Array Dimensions NC, NR, NRODS, NZ, NCF, NCC, NKTB, PMFLX, PPRIM, JHEM NC = Number of cells in X-Y plane. NR = Number of rows of cells in X-Y plane.

NRODS	=	Set equal to NC.
NZ	=	Number of axial cells.
NCF	=	1 (Not used in UTSG version.)*
NCC	=	1 (Not used in UTSG version.)
NKTB	=	Number of primary tube banks.
PMFLX	=	Initial value for primary mass flux.
PPRIM	=	Primary system pressure.
JHEM	=	Hem option is invoked above this axial cell level.

4 * Thermal-Hydraulic Indicators and Data
ITB, IBB, IFLASH, IFINTR, IHT, ISS, IQSS, ICHF, IWFT, IVEC, ITAM, IMIXM, IMIXE, IAFM, ITFM, IGFM, GRAVX, GRAVY, GRAVZ, VELX

ITB	=	0 (Always in UTSG version.)
IBB	=	1 (Always in UTSG version.)
IFLASH	=	Phase change model indicator: (0/1/2) (Nigmatulin Model/Supressed/Subcooled Model).
IFINTR	=	Interfacial momentum exchange model: (0/1) (MIT/LASL).
IHT	=	Heat transfer indicator: (0/1) (no heat transfer/heat transfer included).
ISS	=	Heat transfer calculation type: (1/2) (normal/normal with critical heat flux check suppressed).
IQSS	=	0 (No function in UTSG version.)
ICHF	=	Critical heat flux indicator: (1/2/3) (Biasi/W-3/Cise).
IWFT	=	Transverse friction model indicator: (0/1) (no friction/Gunter-Shaw correlation).
IVEC	=	Transverse velocity indicator: (0/1) (normal/magnitude of velocity used).
ITAM	=	Fluid dynamics indicator: (0/1/2) (no transverse flow allowed/normal/inclined rods).
IMIXM	=	Momentum turbulent mixing indicator: (0/1) (no mixing/fixing included) set to 0 in UTSG version.
IMIXE	=	Energy turbulent mixing indicator: (0/1) (no mixing/mixing included) set to 0 in UTSG version.
IAFM	=	Axial friction model indicator: (0/1) (default/user supplied; see below).

(*) Some of the variables listed are not used in the UTSG version. However, these variables are still in the input list so they must be input. Values given in those cases are dummy variables known to be safe.

ITFM = Transverse friction model indicator: (0/1) (default/user supplied; see below).
IGFM = Grid friction model indicator: (0/1) (default/user supplied; see below).
GRAVX = Gravitational constant in X direction.
GRAVY = Idem in Y direction.
GRAVZ = Idem in Z direction (normally = -9.81 M/S^2).
VELX = Velocity multiplier for transverse friction.

5 * Friction Model Correlations

5A If IAFM=1 then specify AO, REX, A, B
where in laminar flow F = AO/RE
in turbulent flow F = A*RE**B
REX is transition Reynolds number
Default values are AO = 64., REX = 1502.11, A = .184, B = -.2.

5B If ITFM=1 then specify AO, RET, A, B
where in laminar flow F = AO/RE
in turbulent flow F = A*RE**B
RET is transition Reynolds number
Default values are AO = 180., RET = 202.5, A = 1.92, B = -.145.

5C If IGFM=1 then specify A, B
where F = A*RE**B
Default values are A = 3., B = -.1.

6 * Iteration Control and Dump Indicator
IDUMP, NITMAX, IITMAX, EPSN, EPSI

IDUMP = Dump file request indicator: (0/1) (no/yes).
NITMAX = Maximum number of Newton iterations.
IITMAX = Maximum number if inner iterations.
EPSN = Newton iteration convergence criterion.
EPSI = Inner iteration convergence criterion.

7 * Overall Steam Generator Characteristics
QO, HRIS, VT, NSEP

QO = Power.
HRIS = Length of riser region.
VT = Volume of steam dome and main steam line and downcomer.
NSEP = Number of steam separators.

PART II - Array Input Data

In order to simplify the procedure for entering the data, the following groups 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. No commas.

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

Group No.	Contents	
	A. Geometrical Data	
1	IDWN(NC)	= Channel type indicator: (-1/1) (downcomer/evaporator).
2	JMTB(NKTB)	= Level corresponding to bending of each tube bank.
3	RAD(4)	= RAD(2) inner tube radius RAD(3) intermediate tube radius RAD(4) outer tube radius RAD(1) = RAD(2)
4	FG(NKTB)	= Flow split parameter for each tube bank. They are the Fi of Table D-1.
5	DSEW(NKTB*NC*NZP2)	= Horizontal length of tube bank. (Set only in cell primary flow; enters horizontally. Otherwise set to zero.) (Fig. F-2 and note.)
6	DSNS(NKTB*NC*NZP2)	= Height of the upstream cell primary temperature-wise (Fig. F-2 and Note).
7	AHT(NKTB*NC*NZP2)	= Heat transfer area for each tube bank.
8	NCR(NR)	= Number of cells in each row.
9	INDENT(NR)	= Indentation for each row.

10	ARX(NZ,NC)	= Mesh cell areas in the X-direction (M**2).
11	ARY(NZ,NC)	= Mesh cell areas in the Y-direction (M**2).
12	ARZ(NZ+1,NC)	= Mesh cell areas in the Z-direction (M**2).
13	VOL(NZ,NC)	= Mesh cell volumes (M**3).
14	DX(NC)	= Mesh spacing in the X-direction (M).
15	DY(NC)	= Mesh spacing in the Y-direction (M).
16	DZ(NZ+2)	= Mesh spacing in the Z-direction (M).
17	HDZ(NC,NZ+2)	= Hydraulic diameter for each cell (M).
18	HDT(NC,NZ+2)	= Transverse hydraulic diameter for each cell (M).
19	SIJ(4,NC)	= 0 (Not used in UTSG version.)

B. Axial Friction Model

20 IFWZ(NZ+1) = Indicator for axial friction model:

Ten Digit = 0 > Axial friction only
1 > Axial friction + form loss
2 > As 1 + funnel effect

Units Digit = 1 Martinelli model
2 Martinelli and Jones model
3 Levy model
4 rough tube correlation

IFWZ = 10 > form loss without axial friction

21	PH(NC,NZ+2)	= Rod angle with Z axis for each cell (RAD) (see Fig. F-1).
22	TH(NC,NZ+2)	= Angle between X axis and the line formed at the intersection of (X-Y) and rod bundle planes (see Fig. F-1).

C. Initial Conditions

23	P(NZ+2,NC)	= Initial pressures (PA).
24	ALP(NZ+2,NC)	= Initial vapor volume fractions.
25	TV(NZ+2,NC)	= Initial vapor temperature (DEG.K) (note: initial liquid temperature set equal to TV).
26	VVZ(NZ+1,NC)	= Initial vapor axial velocity (M/S) (note: initial liquid velocity set equal to VVZ).

D. Heat Transfer Data

The following data is required only when the heat transfer calculation is requested (i.e., IHT not equal to 0).

27	ICR(NRODS)	= Adjacent channel number for given rod.
28	HDH(NC)	= Equivalent heated diameter for given channel.
29	TW(NZ,NRODS)	= Outer wall surface temperature (DEG.K).
30	QZ(NZ)	= Axial power shape (initial guess).
31	QT(NRODS)	= Transverse power shape (initial guess).
32	QR(NCF+1+NCC)	= 1.0 2(0.0) (Not used in UTSG version.)
33	RN(NRODS)	= Number of tubes in each channel.

34	FRACP(NRODS)	=	1.0 (Not used in UTSG version.)
35	TR(4*NKTB*NZP2*NC)	=	Initial values of primary, primary-wall, tube metal, and secondary-wall temperatures (see note).

PART III - Transient Forcing Function Data
(See PART I for *-FORMAT Description)

Group No.	Format	Contents
1	*	Transient Forcing Function Indicators NT, NTEMP, NB, NQ, NPT, NPG NT = Number of entries in feedwater temperature table (<= 30). NT must be greater than 2 for R-model to be activated. NTEMP = Number of entries in steam dome pressure (or flow rate) table (<=30). NB = Number of entries in feedwater flow rate (or power) Table (≤30). NQ = Number of entries in primary pressure table (≤30). NPT = Number of entries in primary inlet temperature table (≤30). NPG = Number of entries in primary mass flux table (≤30).
2	*	Feedwater Temperature Tabular Values (°K) (TOPFAC(I), YT(I)), I = 1,NT TOPFAC(I)= Feedwater temperature value (°K). YT(I) = Time corresponding to temperature. N.B.: If recirculation model is to be activated at least 3 values must be given (they may be identical).
3	*	Steam dome pressure (Psia) tabular values (if idome = 0). Steam flow (KG/S) tabular values (if idome = 1) (TINFAC(I),YTEMP(I)), I=1,NTEMP TINFAC(I)= Steam dome pressure (Psia) or steam flow (KG/S) value. YTEMP(I) = Time corresponding to pressure or flow rate. N.B.: Pressure version is preferable. If steady-state is being sought, give only two values (identical).
4	*	Steady-State Power Level (W) or Feedwater Flow Rate (KG/S) Tabular Values (BOTFAC(I)), YB(I)), I = 1,NB

BOTFAC(I)= Feedwater flow rate (KG/S) values (if NTEMP < 2 or idome = 1). Steady-state power level (W) (if NTEMP < 2 and idome = 0).
YB(I) = Time corresponding to feed flow or power level.

5	*	Primary System Pressure (Psia) Tabular Values (QFAC(I), YQ(I)), I = 1,NQ QFAC(I) = Primary system pressure (Psia) value. YQ(I) = Time corresponding to pressure.
6	*	Primary Inlet Temperature (°K) Tabular Values (PTFAC(I), YPT(I)), I = 1,NPT PTFAC(I) = Primary temperature (°K) value. YPT(I) = Time corresponding to temperature.
7	*	Primary Mass Flux (KG/M**2/S) Tabular Values (PGFAC(I), YPG(I)), I = 1,NPG PGFAC(I) = Primary mass flux value. YPG(I) = Time corresponding to mass flux.

PART IV - Time Cards
(See Part I for *-FORMAT description)

Group No.	Format	Contents
1	*	Time Zone Control Cards TEND, DTMIN, DTMAX, DTSP, DTLP, CLM, IREDMX

TEND = End of time zone (S).
DTMIN = Minimum time step (S).
DTMAX = Maximum time step (S).
DTSP = Short print time interval (S).
DTLP = Long print time interval (S).
CLM = Convective limit multiplier.
IREDMX = Maximum number of time step reductions.

Note: As many time cards as needed may be input; if DTMIN>=DTMAX, then this will be the time step used throughout the current time zone; if TEND = 0.0, the case is ended; if TEND < 0.0, then subsequent time cards will be read from the terminal; if this has already been signaled by a previous time card with

TEND < 0.0 then the restart option is requested. In the UTSG version this option is used to initiate transients and to continue a search for steady-state from a dump file.

PART V - Initiating a Transient

The following items (previously defined) are read via "RESTART" namelist when RESTART option is invoked:

NITMAX, IITMAX, IFLASH, IWFT, IHT, ISS, EPSN, EPSI, GRAVZ, ITAM, ICHF, IDUMP, IVEC, and (defined below);

TAUH	=	Sensor lag for primary inlet temperature (S).
TAUC	=	Sensor lag for primary outlet temperature (S).
PSDOM	=	Initializes steam dome pressure in first pass ever through the recirculation model. This should be set corresponding to TINFAC(1) but units are Pa.
VD	=	Downcomer volume in a steady-state calculation. This indirectly initializes the downcomer water level since the two match biunivocally.

Also read via restart are:
VT, NSEP, HRIS, IFINTR (all previously defined).

The following items (previously defined) are read via "TFF DATA" namelist.

BOTFAC(I), TOPFAC(I), TINFAC(I), QFAC(I), PTFAC(I), PGFAC(I), YB(I), YT(I), YTEMP(I), YQ(I), YPT(I), YPG(I), NB, NT, NTEMP, NQ, NPT, NPG

Note that immediately after the above information is supplied a time card is required.

The input should look like

```
  $RESTRT  F1,F2,F3,...FN $   for namelist RESTRT and
  $TFFDATA  F1,F2,F3,...FN $     for namelist TFDATA
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 $-sign signifying the end of the namelist input should appear only on the last card.

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

* * * * *

NOTES ON UTSG INPUT

PART I

Group No.	Variable	Comment
3	NKTB	Number of representative primary tubes (tube banks). This number should equal the number of axial levels used in the U-bend region. For example, in Fig. 6.2-1, NKTB = 3.
3	PMFLX	Initial value for primary mass flux.
3	PPRIM	Initial value for primary system pressure.
3	JHEM	This parameter allows the Homogeneous Equilibrium Model (HEM) to be invoked automatically above and including the cells at level iz=JHEM. If JHEM $\geq$ NZ+2, the option is waived.
7	QO	Power level. This input must correspond to the fraction of the unit being represented. (1/2 SG Symmetry=>1/2 Nominal Power)
7	HRIS	Length of the riser measured from top of last cell in the U-bend region to top of last active cell.
7	VT	Volume of Steam Dome + Main Steam Line + Downcomer. This volume must correspond to the whole unit. The volume is the sum of that occupied by steam and liquid above the level of the evaporator entry port.
7	NSEP	Number of Steam Separators.

PART II

Group No.	Variable	Comment
1	IDWN(NC)	For each channel a value of IDWN is specified here (+1=>evaporator) (-1=>downcomer)
2	JMTB(NKTB)	For each tube bank the level (iz) at which it is horizontal (U-bend) is specified here.
3	RAD(4)	Primary Tube radii. RAD(2) = inner radius RAD(3) = intermediate radius RAD(4) = outer radius RAD(1) = RAD(2) Four values must always be given.
4	FG(NKTB)	Flow Split Parameter for each tube bank, e.g. for tube bank i FG(i) = Gi/$\bar{G}$ where $\bar{G}$ is the average primary mass flux. This parameter is calculated in the routine given at the end of Appendix D where it is called Fi.
5	DSEW(KNTB*NC*NZP2	Horizontal length of each tube bank in each cell. A tube bank is said to have a horizontal length in a cell if and only if the primary coolant in the tube bank <u>enters</u> that cell horizontally. The variable is also a control parameter and must be set to zero in all other cells. The order of input is: increment first the tube bank number from 1 to NKTB, then the axial level from 1 to NZP2, then the channel from 1 to NC. (See Fig. F-2.) The non-zero values are calculated in the routine given at the end of Appendix D.

6	DSNS(NKTB*NC*NZP2)	Height of the cell's tube bank upstream neighbor primary temperature wise. A tube bank is said to have an upstream neighboring cell primary temperature-wise if, and only if, the primary coolant in that tube bank enters that cell from the vertical (above or below) direction. Thus

DSNS(KTB, J, I) = dz(J-1)(or dz(j+1)) if I is a hot (or cold) side channel and tube bank KTB exists at level J and is horizontal there. Otherwise DSNS should be set to zero. Exceptionally, in the hot side:

DSNS(KTB,1 and 2,I)=

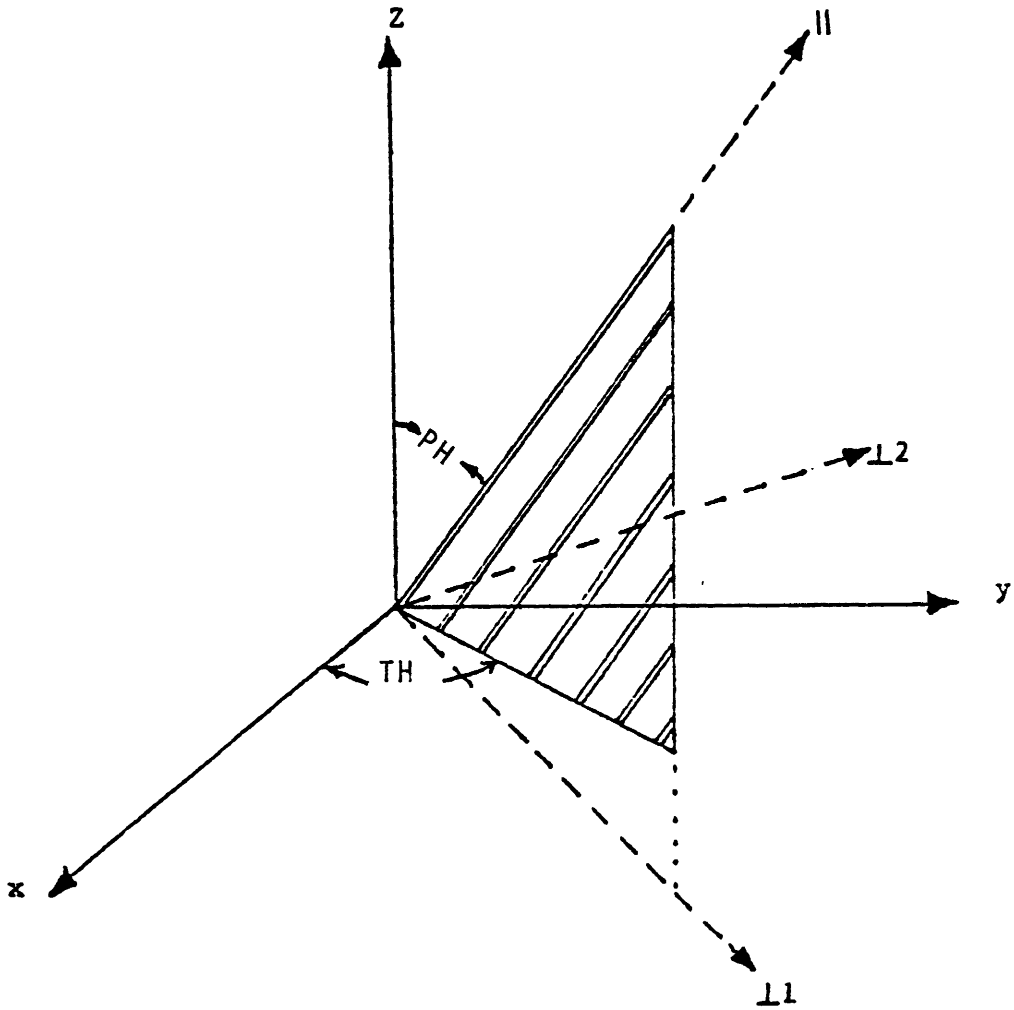
$$\frac{\text{inlet plenum volume}}{\text{total U-tube cross sectional area}}$$

The order of input is identical to that for DSEW. (Fig. F-2)

7	AHT(NKTB*NC*NZP2)	Heat Transfer Area for each tube bank in each cell. Set to zero in cells where the tube bank is not present. The order of input is the same as for DSEW.
35	TR(4*NKTB*NZP2*NC)	Initial Values of (1) primary, (2) primary-wall, (3) tube metal, and (4) secondary-wall temperatures in Deg. K. The order of input is temperature type in the order (1)→(4) given above, then tube bank, then level, and finally channel. Level one corresponds to inlet and outlet temperatures depending on the channel.

Figure F-1.

Definition of pH and tH:
tube inclination angles.



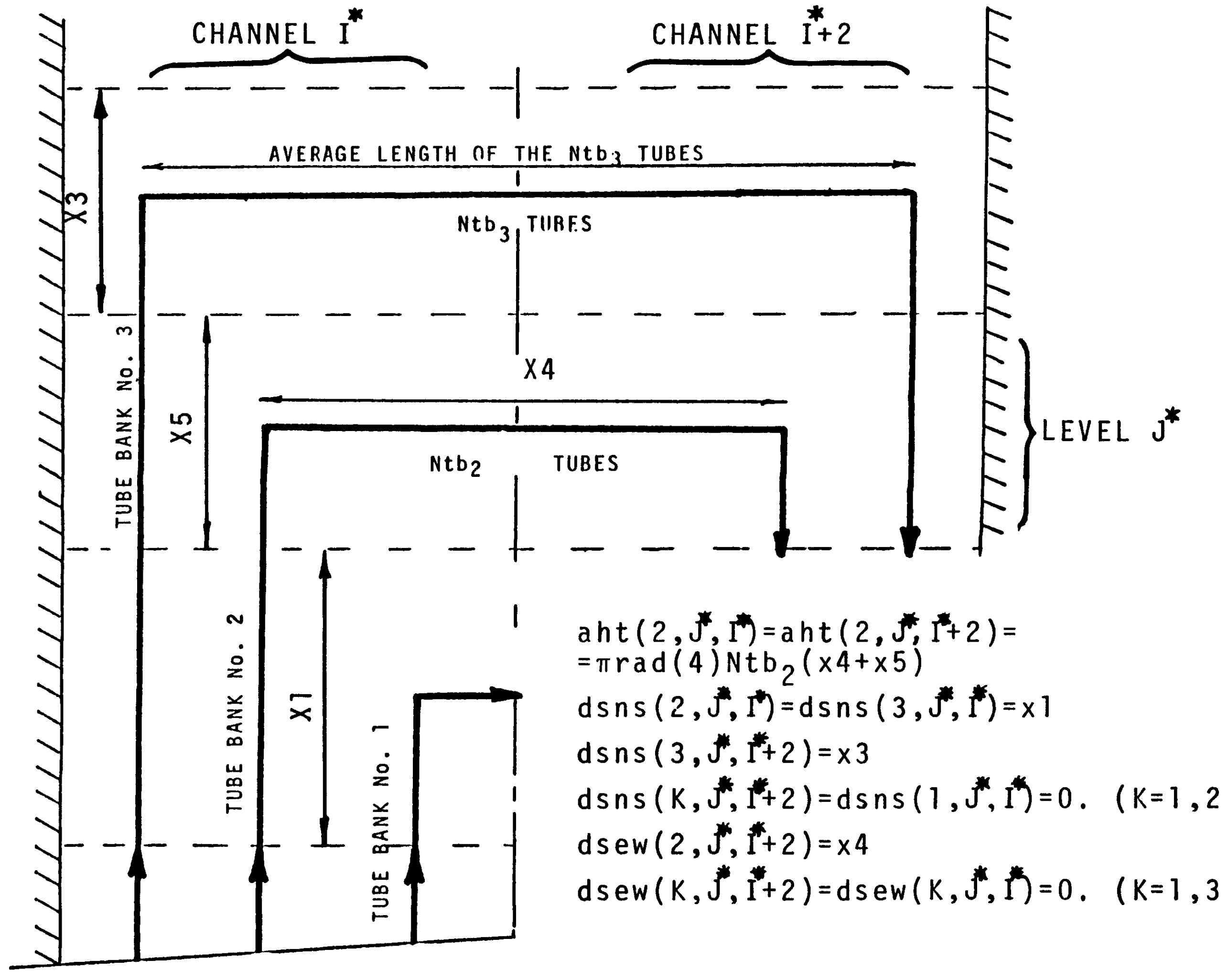


Figure F-2. Selected input parameters for primary model.

EXAMPLES OF INPUT

The next pages contain examples of input in the following order:

1. Geometric and initial conditions input for Arkansas Nuclear One.
2. Geometric and initial conditions input for Model Boiler No. 2.
3. Restart file leading to CEA drop (SG-1) initial conditions starting from the converged solution dump of the run initiated from item 1 above.
4. Restart file giving the forcing functions for CEA drop (SG-1) starting from the dump of item 3 above.
5. Restart file leading to steady-state half power result for Model Boiler No. 2 starting from the converged solution of item 2 above.

```
1
ANO-2 SG (6X13) GEOMETRIC INPUT AND INITIAL CONDITIONS
6,4,6,13,1,1,3,4345.,15513176.,15
0,1,0,1,1,1,0,1,1,0,2,0,0,0,0,0,0.0,0.0,-9.81,2.082
1,4,100,1.0e-7,1.0e-8
6.91e+8,1.8292,126.1096,166
-1 -1 1 -1 1 -1 $idn
8 9 10 $jmtb
2(8.305e-3) 8.915e-3 9.525e-3 $rad
1.11 .98  0.91 $fg
180(0.0)
21(0.)
.7544 2(0.)  0. 1.905 0.  2(0.) 2.858 15(0.0)
45(0.0) $dsew
90(0.0)
6(3.75) 3(.4064) 15(1.05225)
  0.    2(1.05225) 2(0.) .81863  15(0.)
45(0.)
3(.4064) 18(1.05225) 0. 2(.81863) 2(0.0) .81863 18(0.)
45(0.) $dsns
90(0.0)
2(379.731 2(284.799) 41.153 2(30.865)
  5(106.553 2(79.915)) 91.48 2(79.915)
 0.0 103.424 62.17 2(0.) 139.612 15(0.0)
  45(0.)) $aht
1 2 2 1 $ncr
1 0 0 1 $indent
13(0.0)
2(13(0.0) .3955 12(0.0))
13(0.0) $arx
2(13(0.0))
.3955 12(0.0)
13(0.0)
.3955 5(1.02939) 1.1561 .80762 .99377 1.3844 1.5941 2.0195 2.4462
.3955 12(0.0) $ary
2(0.0 6(.1463) .3030 .8630 1.4964 1.5752 1.0660 .6150 .6150)
0.0 5(1.6456) 1.7521 1.7754 1.8606 2.4563    2.8598 3.9008 4.9418 0.8
0.0 6(.1463) .3030 .8630 1.4964 1.5752 1.0660 .6150 .6150
0.0 5(1.6456) 1.7521 1.7754 1.8606 2.4563    2.8598 3.9008 4.9418 0.8
0.0 6(.1463) .3030 .8630 1.4964 1.5752 1.0660 .6150 .6150 $arz
2(.05946 6(.15394) .413 1.0 1.45 .8 .5 .25)
.66878 5(1.7316) 1.95577 1.36577 1.68056 2.3411    2.6958 3.4152 4.1347
.05946 6(.15394) .413 1.0 1.45 .8 .5 .25
.66878 5(1.7316) 1.95577 1.36577 1.68056 2.3411    2.6958 3.4152 4.1347
.05946 6(.15394) .413 1.0 1.45 .8 .5 .25 $vol
1.6911 2(.08651 1.6911) 1.6911 $dx
0.08651 4(1.6911) 0.08651 $dy
.01 .4064 6(1.05225) 3(.81863) 3(.60973)  .01 $dz
2(7(.1687) .3494 .9952 1.7256 1.8165 1.2293 .7092 .7092 .7092)
6(.0255) .0263 .04544 .09167 3.1436     2.737  4(2.764)
7(.1687) .3494 .9952 1.7256 1.8165 1.2293 .7092 .7092 .7092
6(.0255) .0263 .04544 .09167 3.1436     2.737 4(2.764)
7(.1687) .3494 .9952 1.7256 1.8165 1.2293 .7092 .7092 .7092 $hdz
2(15(1.0e+06))
```

```
7(.0255) .0267 .0464 .0958  4.86 4(6.76)
15(1.0e+06)
7(.0255) .0267 .0464 .0958  4.86 4(6.76)
15(1.0e+06) $hdt
24(0.0) $sij
14(02) $ifwz
2(15(0.0)) 9(0.0) 0.0 5(0.0)
15(0.0) 9(0.0) 0.0 5(0.0) 15(0.0) $ph
2(15(0.0)) 9(0.0)  (0.0)  5(0.0)  15(0.0)
9(0.0)  (0.0)  5(0.0) 15(0.0) $th
2(6279350 6277780 6282960 6277990 6273010 6268040 6263060 6258090
6258090 6252360 6246200 6240780 6236100 6231180 6228840)
6274270 6272720 6267570 6262910 6259390 6256470 6253910
6251590 6250010 6248920 6247920 6247050 6246360 6245570 6245150
6279350 6277780 6282960 6277990 6273010 6268040 6263060 6258090 6258090
6252360 6246200 6240780 6236100 6231180 6228840
6274270 6272720 6267570 6262910 6259390 6256470  6253910 6251590
6250010 6248920 6247920 6247050 6246360 6245570 6245150
6279350 6277880 6282960 6277990 6273010 6268040 6263060 6258090
6258090 6252360 6246200 6240780 6236100 6231180 6228840 $p
2(15(0.0))
2(0.0) .26 .50 .62 .70 .76 .80 .83 .85 .84 .83 3(.80)
15(0.0)
2(0.0) .26 .50 .62 .70 .76 .80 .83 .85 .84 .83 3(.80)
15(0.0) $alp
2(15(541.85))
541.85 546.7 551.7 551.6 551.6 551.55 551.5 551.5 551.45 551.44
551.42 551.41 551.4 551.38 551.38
15(541.85)
541.85 546.7 551.7 551.6 551.6 551.55 551.5 551.5 551.45 551.44
551.42 551.41 551.4 551.38 551.38
15(541.85) $tv
2(0.0 13(-1.6))
0. .833 1.008 1.376 1.769 2.184 2.664 3.229 3.491 3.824 4.332 4.008 3.868
3.730
0.0 13(-1.6)
0. .833 1.008 1.376 1.769 2.184 2.664 3.229 3.491 3.824 4.332 4.008 3.868
3.730
0.0 13(-1.6) $vvz
1 2 3 4 5 6  $icr
6(.0132) $hdh
2(15(0.0))
15(558.0)
15(0.0)
15(558.0)
1(15(0.0)) $tw
2(15(1.0))
0. 100. 92. 84. 79. 75. 73. 69. 66. 66.  5(0.0)
15(1.0)
0.0 100. 92. 84. 79. 75. 73. 69. 66. 66. 5(0.0)
1(15(1.0)) $qz
2(0.0)
0.5
(0.0)
0.5
1(0.0) $qt
```

```
1.0 0. 0.  $qr
2(0.) 4230.0 0. 4230.0 1(0.) $rn
6(1.0) $fracp
2(15(12(0.0)))
593.7 547.6 547.6 547.6 593.7 547.6 547.6 547.6 593.7 547.6 547.6 547.6
593.0 583.2 571.0 559.4 592.9 583.1 570.9 559.4 592.8 583.1 570.9 559.4
590.6 581.5 570.1 559.4 590.2 581.2 569.9 559.3 590.0 581.0 569.8 559.3
587.4 579.2 569.0 559.4 586.7 578.6 568.7 559.3 586.1 578.2 568.5 559.2
584.5 577.1 568.0 559.4 583.5 576.4 567.5 559.2 582.7 575.8 567.2 559.1
581.8 575.2 566.9 559.2 580.6 574.3 566.4 559.0 579.7 573.6 566.0 558.8
579.4 573.4 565.9 558.9 578.0 572.3 565.3 558.6 577.0 571.6 564.8 558.4
577.2 571.7 564.9 558.5 575.6 570.5 564.1 558.2 574.5 569.6 563.6 557.9
4(0.0)                  573.7 569.0 563.2 557.7 572.5 568.1 562.6 557.5
8(0.0)                                          571.0 566.8 561.8 557.1
5(12(0.0))
15(12(0.0))
566.3 547.6 547.6 547.6 561.6 547.6 547.6 547.6 558.7 547.6 547.6 547.6
566.5 563.3 559.3 555.6 561.8 559.7 557.0 554.5 558.8 557.3 555.5 553.7
567.5 564.1 560.0 556.1 562.5 560.3 557.7 555.2 559.3 557.9 556.1 554.4
568.9 565.2 560.7 556.5 563.5 561.1 558.2 555.5 560.0 558.5 556.5 554.7
570.4 566.4 561.5 557.0 564.7 562.0 558.8 555.8 560.9 559.1 556.9 554.9
572.0 567.7 562.4 557.4 565.8 563.0 559.4 556.1 561.8 559.8 557.4 555.1
573.8 569.0 563.2 557.8 567.3 564.0 560.1 556.4 562.8 560.6 557.9 555.3
575.7 570.5 564.1 558.2 568.8 565.2 560.8 556.7 564.0 561.5 558.4 555.5
4(0.0)                  570.2 566.3 561.5 557.0 565.1 562.4 559.0 555.8
8(0.0)                                          566.2 563.2 559.5 556.0
5(12(0.0))
15(12(0.0)) $tr
0 0 0 0
20. .001 .1 2. 20. .6 1
0.  0.   0. 0.  0. 0. 0
```

```
2
MODEL BOILER NO.2 DATA VERIFICATION USING THERMIT-UTSG CODE.
PRIMARY AND SECONDARY SIDES ARE COUPLED. ( AS OF NOVEMBER 20,1983 ).
6,4,6,15,1,1,3,3954.7,15513175.9,17
0,1,0,1,1,1,0,1,1,0,2,0,0,0,0,0,0.0,0.0,-9.81,2.082
1,4,100,1.0e-7,1.0e-8
1.6675e+6,3.478,1.0767,1
-1 -1 1 -1 1 -1      $idn
10 11 12             $jmtb
2(7.71e-3) 8.22e-3 8.73e-3                           $rad
1.016 0.9978 0.9825   $fg
4(17(3(0.0))) 9(3(0.0)) 0.1059 3(0.0)
0.196 3(0.0) 0.329 5(3(0.0)) 17(3(0.0))              $dsew
2(17(3(0.0))) 2(3(3.6))
3(0.117) 3(0.903) 5(3(1.02)) 3(0.567)
0.0 2(0.11) 2(0.0) 0.11 5(3(0.0))
17(3(0.0))
3(0.117) 3(0.903) 3(1.02) 3(1.02) 3(1.02) 3(1.02) 3(1.02) 3(0.567)
3(0.11) (0.0) 2(0.11) 2(0.0) 0.11 6(3(0.0))
17(3(0.0))                                           $dsns
2(17(3(0.0)))
2(2(1.580) 1.975 2(0.0513) 0.0642 2(0.3963) 0.4953 5(2(0.4476) 0.5595)
2(0.2488) 0.3110 0.0474 0.0483 0.0603 0.0 0.0671 0.1207 2(0.0) 0.1204
5(3(0.0)) 17(3(0.0)))                               $aht
1 2 2 1                                              $ncr
1 0 0 1                                               $indent
15(0.0)
2(15(0.0) 0.02536 14(0.0))
15(0.0)                                              $arx
2(15(0.0))
0.0048 14(0.0)
15(0.0)
0.0037 0.0285 5(0.0321) 0.0179 0.0035 0.0038 0.0039 0.0848 0.0229
0.0610 0.0839
0.0048 14(0.0)                                       $ary
0.0 13(3.0378e-4)
7.268e-2 9.893e-2
0. 13(2.086e-3) 4.992e-2 6.79e-2
0.0 1.079e-2 7(1.078e-2) 1.09e-2 1.19e-2 1.218e-2 1.147e-2 5.87e-3 2(5.90e
-3)
0. 13(2.086e-3) 4.992e-2 6.79e-2
0.0 1.079e-2 7(1.078e-2) 1.09e-2 1.19e-2 1.218e-2 1.147e-2 5.87e-3 2(5.90e
-3)
0.0 13(3.0378e-4)
7.268e-2 9.893e-2                                     $arz
3.554e-5 2.743e-4 5(3.099e-4) 1.722e-4 3(3.342e-5) 5.173e-4 1.397e-4
1.207e-2 1.662e-2
2(2.441e-4 1.884e-3 5(2.128e-3) 1.183e-3 3(2.295e-4) 3.553e-3 9.597e-4
8.286e-2 1.141e-1
1.263e-3 9.730e-3 5(1.099e-2) 6.11e-3 1.19e-3 1.31e-3 1.34e-3 2.90e-2
2.7e-3 7.169e-3 9.872e-3)
3.554e-5 2.743e-4 5(3.099e-4) 1.722e-4 3(3.342e-5) 5.173e-4 1.397e-4
1.207e-2 1.662e-2                                       $vol
0.0498 2(0.0061 0.0498) 0.0498                        $dx
0.0061 4(0.342) 0.0061                                $dy
0.01 0.117 0.903 5(1.02) 0.567 3(0.11) 1.703 0.46 1.22 1.68 0.01  $dz
```

```
.021737 0.02218 6(0.01135) 0.01150 5(0.01135) 1.1707 0.1610 0.1593
0.0121 0.02413 12(0.01217) 0.2262 0.3077 0.3078
0.1739 8(0.0237) 0.0239 0.03446 0.0517 0.1106 0.1400 2(0.1406) 0.14008
0.0121 0.02413 12(0.01217) 0.2262 0.3077 0.3078
0.1739 8(0.0237) 0.0239 0.03446 0.0517 0.1106 0.1400 2(0.1406) 0.14008
0.021737 0.02218 6(0.01135) 0.01150 5(0.01135) 1.1707 0.1610 0.1593 $hdz
17(4.222e-1)
17(4.596e-1)
5.08e-3 8.111e-3 6.249e-2 7.057e-2 4(7.057e-2) 3.925e-2
7.69e-3 1.10e-2 1.657e-2 5.508e-1 5.454e-1 1.4546 2.000 1.187e-2
17(4.596e-1)
5.08e-3 8.111e-3 6.249e-2 7.057e-2 4(7.057e-2) 3.925e-2
7.69e-3 1.10e-2 1.657e-2 5.508e-1 5.454e-1 1.4546 2.000 1.187e-2
17(4.222e-1)                                              $hdt
24(0.0)                                                    $sij
16(02)                                                     $fwz
2(17(0.0)) 9(0.0) 1.5708 7(0.0)
17(0.0) 9(0.0) 4.7154 7(0.0) 17(0.0)                       $ph
6(17(0.0))                                                 $th
7316860. 7316300. 7314000. 7305720. 7298170. 7285140. 7280110. 7278920.
7271430. 7270000. 7268740. 7265110. 7259230. 7258540. 7257300. 7244550.
7243920.
2(7316860. 7316300. 7314000. 7305720. 7298170. 7285140. 7280110. 7278920.
7271430. 7270000. 7268740. 7265110. 7259230. 7258540. 7257300. 7244550.
7243920.
7316400. 7315980. 7314250. 7312590. 7309120. 7294570. 7291080. 7290050.
7287520. 7283240. 7280190. 7278410. 7277580. 7271870. 7268470. 7264410.
7261540.)
7316860. 7316300. 7314000. 7305720. 7298170. 7285140. 7280110. 7278920.
7271430. 7270000. 7268740. 7265110. 7259230. 7258540. 7257300. 7244550.
7243920.                                                  $p
2(17(0.0))
2(0.0) 0.10 0.20 0.30 0.35 0.40 0.45 0.50 0.55 7(0.60)
17(0.0)
2(0.0) 0.00 0.10 0.20 0.25 0.30 0.35 0.40 0.45 7(0.60)
17(0.0)                                                    $alp
543.54 543.55 543.63 544.17 544.70 545.07 545.20 545.13 545.55 546.88 546.
72
546.10 545.17 545.17 545.17 545.17 545.17
2(543.51 543.53 543.60 544.15 544.69 545.07 545.20 545.13 545.54 546.87
546.72 546.10 545.17 545.17 545.17 545.17 545.17
545.06 545.09 554.04 561.93 561.87 561.82 561.79 561.76 561.74 561.74
561.73 561.73 561.71 561.69 561.67 561.65 561.65)
543.54 543.55 543.63 544.17 544.70 545.07 545.20 545.13 545.55 546.88 546.
72
546.10 545.17 545.17 545.17 545.17 545.17                      $tv
2(0.0 15(-0.02))
0.0 15(0.02)
0.0 15(-0.02)
0.0 15(0.02)
0.0 15(-0.02)                                               $vvz
1 2 3 4 5 6                                                $icr
6(0.0132)                                                  $hdh
2(17(0.0))
17(572.4)
17(0.0)
```

```
17(565.3)
17(0.0)                                                    $tw
2(17(1.0))
0. 100. 92. 84. 79. 75. 73. 69. 66. 66. 66. 66. 5(0.0)
17(1.0)
0. 100. 92. 84. 79. 75. 73. 69. 66. 66. 66. 66. 5(0.0)
17(1.0)                                                    $qz
2(0.0)
0.5
(0.0)
0.5
(0.0)                                                        $qt
1.0 0.0 0.0                                                $qr
2(0.0) 26.0 0.0 26.0 0.0                                   $rn
6(1.0)                                                     $fracp
2(17(12(0.0)))
581.3 577.3 577.2 577.1 581.3 577.3 577.2 577.1 581.3 577.3 577.2 577.1
581.2 576.9 571.6 566.5 581.2 576.9 571.5 566.5 581.2 576.8 571.5 566.5
580.2 576.2 571.3 566.6 580.2 576.2 571.3 566.6 580.2 576.1 571.2 566.6
578.4 575.0 570.8 566.8 578.4 575.0 570.8 566.8 578.3 574.9 570.7 566.8
576.8 573.8 570.0 566.4 576.7 573.7 569.9 566.4 576.6 573.6 569.9 566.4
575.2 572.6 569.3 566.2 575.2 572.5 569.2 566.1 575.1 572.4 569.2 566.1
573.9 571.5 568.6 565.9 573.8 571.4 568.6 565.9 573.7 571.4 568.5 565.8
572.7 570.6 568.0 565.6 572.6 570.5 568.0 565.6 572.5 570.4 567.9 565.7
571.8 569.9 567.6 565.4 571.7 569.8 567.5 565.4 571.7 569.8 567.8 565.6
571.6 569.7 567.4 565.3 571.6 569.6 567.4 565.3 571.6 569.5 567.7 565.5
4(0.0)                  571.5 569.5 567.3 565.2 571.5 569.4 567.6 565.4
8(0.0)                                          571.4 569.3 567.5 565.3
5(12(0.0))
17(12(0.0))
566.2 563.6 563.5 563.4 566.2 563.6 563.5 563.4 566.2 563.6 563.5 563.4
566.7 565.7 564.4 563.2 566.5 565.5 564.2 563.7 566.3 565.3 564.1 563.0
567.0 566.0 564.7 563.5 566.8 565.8 564.5 563.8 566.5 565.6 564.4 563.3
567.6 566.6 565.3 564.1 567.3 566.4 565.2 564.1 567.1 566.1 565.0 564.0
568.2 567.1 565.6 564.3 567.9 566.8 565.5 564.2 567.6 566.6 565.3 564.2
568.9 567.6 566.0 564.5 568.6 567.3 565.8 564.4 568.2 567.1 565.7 564.4
569.6 568.2 566.4 564.7 569.3 567.9 566.2 564.6 568.9 567.6 566.0 564.6
570.4 568.8 566.8 565.0 570.1 568.5 566.6 564.9 569.7 568.2 566.4 564.8
571.1 569.4 567.2 565.2 570.8 569.1 567.0 565.1 570.4 568.7 566.8 564.9
571.5 569.6 567.4 565.2 571.1 569.3 567.2 565.2 570.7 569.0 566.9 565.0
4(0.0)                  571.2 569.4 567.3 565.3 570.8 569.0 567.0 565.2
8(0.0)                                          570.9 569.1 567.1 565.3
5(12(0.0))
17(12(0.0))                                                      $tr
0 0 0 0
-1 0.0001 0.001 2.0 40. 0.6 1
```

```
-3
$restart vd=40.747 psdom=6421696. hris=1.8292 vt=126.1096
imixm=0 imixe=0 iflash=0 nsep=166 foul=1.0  $
$tffdata nt=3 topfac(1)=478.15,478.15,478.15  yt(1)=0.,10.,1000.
nb=2 botfac(1)=695500000.,695500000.  yb(1)=0.,1000.
ntemp=2 tinfac(1)=931.39,931.39  ytemp(1)=0.,1000.
nq=2 qfac(1)=2243.33,2243.33  yq(1)=0.,1000.
npg=0
npt=2 ptfac(1)=576.5,576.5 ypt(1)=0.,1000.  $
20. .001 .1 1. 20. .6 1
0. 0.   0. 0. 0. 0. 0
```

```
-3
$restart tauh=4.753 tauc=4.898 $
$tffdata nt=4 topfac(1)=487.15,487.15,487.15,487.15
yt(1)=0.,7.,10.,1000.
nb=19 botfac(1)=364.49,364.49,374.29,374.29,363.16,
378.49,378.49,372.19,378.49,379.33,378.49,370.79,
378.49,374.29,379.89,381.99,374.29,381.99,381.99
yb(1)=0.,1.,4.,6.,11.,14.,22.67,25.,27.,28.,30.,33.67,
36.67,38.67,42.,54.67,57.33,60.,66.
ntemp=6 tinfac(1)=931.39,931.67,922.78,916.11,913.89,
913.33 ytemp(1)=0.,7.33,20.33,36.67,49.67,66.
nq=7 qfac(1)=2243.33,2243.33,2232.22,2210.,2194.44,2188.89,
2188.89 yq(1)=0.,3.,6.67,19.33,38.,49.,66.
npt=10 ptfac(1)=576.13,576.12,576.13,576.13,574.44,573.83,
573.61,573.55,573.49,573.49
ypt(1)=0.,1.2,4.1,6.7,31.8,48.,56.6,61.2,63.5,66. $
6. .001 .1 1. 7. .6 1
24. .001 .2 3. 25. .6 1
64. .001 .6 3. 39. .6 1
0. 0. 0. 0. 0. 0. 0
```

```
-3
$restart vd=0.730 psdom=7244553.8 hris=3.36 vt=1.0767 nsep=1
epsn=1.0e-6 epsi=1.0e-7  foul=1.0 $
$tffdata nt=3 topfac(1)=460.78,460.78,460.78  yt(1)=0.,10,1000.
nb=2 botfac(1)=3335000.,3335000.  yb(1)=0.,1000.
ntemp=2 tinfac(1)=1050.73,1050.73  ytemp(1)=0.,1000.
nq=2 qfac(1)=2250.,2250.  yq(1)=0.,1000.
npg=0
npt=2 ptfac(1)=581.33,581.33  ypt(1)=0.,1000.   $
1. .0001 .02 .1 1. .6 1
0. 0. 0. 0. 0. 0. 0
```

EXAMPLES OF OUTPUT

There are two types of output resulting from a THERMIT-UTSG run:

1. long and short prints, and
2. recirculation model prints.

The long print gives the solution of the two-fluid model equations in the evaporator and the primary and tube wall temperatures as well. The long print for the MB-2 half power case follows. The short print is just the heading of the long print.

The recirculation model prints give the global parameters which are of interest for transient analyses. An example of this print for the CEA drop (SG-1) case follows after the MB-2 example.

MODEL BOILER NO.2 DATA VERIFICATION USING THERMIT-UTSG CODE.

PRIMARY AND SECONDARY SIDES ARE COUPLED (AS OF NOVEMBER 20,1983).

Array Dimensions

Number of channels	=	6
Number of rows	=	4
First level in U-bent region	=	10
Number of active axial nodes	=	15
Last level in U-bent region	=	12
Number of tube banks	=	3
Homogeneous Model Begins At iz	=	17

Thermal-Hydraulic Options In Use

Pressure boundary condition at SG top
Velocity boundary condition at SG bottom
Nigmatulin boiling model
LASL interfacial momentum exchange model
Value of iht=1 is correctly set
Explicit-qs- heat transfer calculation
iqss oK!
Biasi critical heat flux correlation
Transverse friction model - Gunter-Shaw
Transverse velocity used in transverse momentum calculations
Transverse flow is No mixing model
No mixing model
Gravitational constants (x,y,z)= 0.00000 0.00000 -9.81000
Transverse friction multiplier = 0.20820E+01

Friction Model

Axial	f =	0.184*Re**-0.200
Transverse	f =	1 920*Re**-0 145
Grid spacer	K =	3.000*Re**-0.100

Iteration Control Parameters

Dump indicator (0/1)(no/yes)	=	1
Max number of Newton iterations	=	4
Max number of inner iterations	=	100
Convergence crit. for newton iter	=	0 10000E-06
Convergence crit. for inner iter	=	0.10000E-07

Primary Model Data

Tube inner radius	=	0 77100E-02
Tube intermediate radius	=	0 82200E-02
Tube outer radius	=	0 87300E-02
Primary inlet temperature	=	581 30
primary system pressure (Pa)	=	0 15513E+08
Primary mass flux (Kg/m2/s)	=	0 39547E+04
Primary outlet temperature	=	566.01

time step no = 400 time = 7.999997 sec time step size = 0.20000E-01 sec cpu time = 1180.81 sec
number of newton iterations = 1
number of inner iterations = 2 0 reduced time steps since last print

total primary power = 1667 500 kW
total heat transfer = 1667.171 kW
flow enthalpy rise = 1662.921 kW
steam dome pressure= 7244553.8 Pa
feedwater flow rate = 0.846 Kg/s
feedwater temperature = 460.78 K

downcomer flow rate = -5.802 kg/s
riser flow rate = 5.839 kg/s
steam flow rate = 0.846 kg/s
circulation ratio = 6.85 (=1/xr)
downcomer water level = 11.237 m

primary flow parameters :
inlet temperature = 581.30 k
outlet temperature = 566.01 K
mass flux = 3953.98 Kg/m2/sec
system pressure = 15513175.9 Pa

ic	iz	P(MPa)	void	% qual	hm	hl	T vap	T liq	T sat	vvz	vlz	rov	rol	mass flux
1	1	7.31525	0.0000	0.00	1186.919	1186.919	550.54	550.54	562.01	0.000	0.000	39.40	769.16	0.0
1	2	7.31477	0.0000	0.00	1186.970	1186 970	550.55	550.55	562.00	-1.598	-1.598	39.40	769.14	-1229.1
1	3	7.31679	0.0000	0.00	1187.380	1187.380	550.63	550.63	562.02	-1.598	-1.598	39.41	769.00	-1227.5
1	4	7.31097	0.0000	0.00	1190.150	1190.150	550.17	550.17	561.97	-1.598	-1.598	39.33	768.01	-1226.0
1	5	7.30479	0.0000	0.00	1192.874	1192.874	550.70	550.70	561.91	-1.598	-1.598	39.25	767.04	-1224.9
1	6	7.29863	0.0000	0.00	1194.779	1194.779	550.07	550.07	561.85	-1.598	-1.598	39.19	766.36	-1224.5
1	7	7.29247	0.0000	0.00	1195.450	1195.450	550.20	550.20	561.80	-1.598	-1.598	39.14	766.11	-1224.7
1	8	7.28631	0.0000	0.00	1195.090	1195.090	550.13	550.13	561.74	-1.598	-1.598	39.10	766.23	-1223.5
1	9	7.28151	0 0000	0.00	1197.256	1197.256	550.55	550.55	561.69	-1.598	-1.598	39.04	765.45	-1219.6
1	10	7.27946	0.0000	0.00	1204.130	1204.130	550.88	550.88	561.67	-1.598	-1.598	38.94	763.00	-1220.0
1	11	7.27880	0.0000	0.00	1203.301	1203.301	550.72	550.72	561.67	-1.598	-1.598	38.95	763.29	-1221.9
1	12	7.27814	0.0000	0.00	1200.095	1200.095	550.10	550.10	561.66	-1.598	-1.598	38.98	764.44	-1224.6
1	13	7.27266	0.0000	0.00	1195.299	1195.299	550.17	550.17	561.61	-1.598	-1.598	39.01	766.14	-1224.6
1	14	7.26613	0.0000	0.00	1195.301	1195.301	550.17	550.17	561.55	-1.598	-1.598	38.96	766.13	-1224.6
1	15	7.26117	0.0000	0.00	1195.302	1195.302	550.17	550.17	561.50	-0.007	-0.007	38.93	766.12	-5.1
1	16	7.25027	0.0000	0.00	1195.304	1195.304	550.17	550.17	561.40	-0.005	-0.005	38.86	766.11	-3.8
1	17	7.24392	0.0000		1195.304	1195.305	550.17	550.17	561.34			38.81	766.10	
2	1	7.31499	0.0000	0.00	1186.765	1186.765	550.51	550.51	562.01	0.000	0.000	39.40	769.21	0.0
2	2	7.31451	0.0000	0.00	1186.868	1186.868	550.53	550.53	562.00	-1.588	-1.588	39.40	769.18	-1221.0
2	3	7.31793	0.0000	0.00	1187.226	1187.226	550.60	550.60	562.03	-1.588	-1.588	39.42	769.06	-1219.4
2	4	7.31198	0.0000	0.00	1190.047	1190.047	550.15	550.15	561.98	-1.588	-1.588	39.34	768.05	-1217.9
2	5	7.30567	0.0000	0.00	1192.822	1192.822	550.69	550.69	561.92	-1.588	-1.588	39.26	767.06	-1216.8
2	6	7.29937	0.0000	0.00	1194.779	1194.779	550.07	550.07	561.86	-1.588	-1.588	39.19	766.36	-1216.4
2	7	7.29307	0.0000	0.00	1195.450	1195.450	550.20	550.20	561.80	-1.588	-1.588	39.14	766.11	-1216.6
2	8	7.28677	0.0000	0.00	1195.090	1195.090	550.13	550.13	561.74	-1.588	-1.588	39.10	766.23	-1215.4
2	9	7.28187	0.0000	0.00	1197.204	1197.204	550.54	550.54	561.70	-1.588	-1.588	39.04	765.47	-1211.5
2	10	7.27978	0.0000	0.00	1204.078	1204.078	550.87	550.87	561.68	-1.588	-1.588	38.94	763.02	-1212.0
2	11	7.27911	0.0000	0.00	1203.301	1203.301	550.72	550.72	561.67	-1.588	-1.588	38.95	763.29	-1213.8
2	12	7.27843	0.0000	0.00	1200.095	1200.095	550.10	550.10	561.66	-1.588	-1.588	38.99	764.44	-1216.5
2	13	7.27283	0.0000	0.00	1195.299	1195.299	550.17	550.17	561.61	-1.588	-1.588	39.01	766.14	-1216.5
2	14	7.26616	0.0000	0 00	1195.301	1195.301	550.17	550.17	561.55	-1.588	-1.588	38.96	766.13	-1216.5
2	15	7.26117	0.0000	0.00	1195.302	1195.302	550.17	550.17	561.50	-0.066	-0.066	38.93	766.12	-50.8
2	16	7.25027	0.0000	0.00	1195.304	1195.304	550.17	550.17	561.40	-0.049	-0.049	38.86	766.11	-37.4
2	17	7.24392	0.0000		1195.304	1195.305	550.17	550.17	561.34			38.81	766.10	
3	1	7.31492	0.0000	0.00	1194.724	1194.724	550.06	550.06	562.01	0.000	0.000	39.30	766.40	0.0
3	2	7 31444	0.0000	0.00	1196.611	1196.611	551.43	551.43	562.00	0.375	0.375	39.27	765.73	287.3
3	3	7.30933	0 0000	0.00	1245.070	1245.070	556.68	556.68	561.95	0.776	0.604	38.64	748.13	451.5
3	4	7.30217	0.0168	0.13	1285.886	1283.984	561.89	561.89	561.89	0.894	0.620	38.14	733.46	447.8
3	5	7.29552	0.2568	2.58	1322.021	1283.646	561.82	561.82	561.82	0.997	0.676	38.10	733.58	378.0
3	6	7.29015	0.4156	5.11	1359.471	1283.374	561.77	561.77	561.77	1.072	0.734	38.07	733.68	331.7
3	7	7.28572	0.5206	7.65	1396 990	1283.149	561.73	561.73	561.73	1.171	0.796	38.05	733.76	303.2
3	8	7.28192	0.5939	10.05	1432.556	1282.956	561.70	561.70	561.70	1.262	0.856	38.03	733.83	283.6
3	9	7.27922	0.6263	11.26	1450.388	1282.819	561.67	561.67	561.67	1.301	0.890	38.01	733.88	275.1

3	10	7.27811	0 5741	11 47	1453 567	1282 763	561.66	561.66	561.66	1 426	0 768	38 00	733.90	271 1
3	11	7.27718	0 6311	11 86	1459 192	1282 716	561 65	561 65	561.65	1.231	0 811	38.00	733.91	249.0
3	12	7 27685	0 6397	12.14	1463 379	1282 699	561.65	561.65	561.65	1.222	0.813	38.00	733.92	244.7
3	13	7 27425	0 6410	11 82	1458 512	1282.567	561.62	561 62	561.62	1.332	0.918	37 98	733 96	274.4
3	14	7 27118	0.6679	11 69	1456.457	1282.411	561.60	561.60	561.60	2.477	1.947	37.96	734.02	537 3
3	15	7.26801	0 6947	12.37	1466 423	1282.250	561 57	561 57	561.57	2.842	2 368	37.94	734.08	605.6
3	16	7 26395	0.7249	14.95	1463.386	1282.043	561.53	561.53	561.53	3 369	2.886	37.92	734 15	731.2
3	17	7.26154	1 0000		1463 386	1282 045	561.53	561.53	561.50			37.90	734.14	
4	1	7 31521	0 0000	0.00'	1186.765	1186 765	550 51	550.51	562.01	0 000	0.000	39 40	769.21	0.0
4	2	7.31473	0 0000	0.00	1186 868	1186 868	550.53	550.53	562.00	-1 578	-1.578	39 40	769.18	-1213.7
4	3	7 31807	0 0000	0.00	1187.226	1187 226	550 60	550.60	562 03	-1.578	-1.578	39.42	769.06	-1212 1
4	4	7 31210	0 0000	0 00	1190.047	1190 047	550.15	550.15	561.98	-1 578	-1 578	39 34	768.05	-1210.6
4	5	7 30578	0.0000	0.00	1192 822	1192 822	550.69	550 69	561.92	-1.578	-1.578	39 26	767.06	-1209 5
4	6	7 29946	0.0000	0 00	1194.779	1194 779	550 07	550 07	561 86	-1.578	-1.578	39.19	766.36	-1209 1
4	7	7 29315	0 0000	0 00	1195.450	1195 450	550 20	550 20	561.80	-1 578	-1 578	39.14	766 11	-1209.3
4	8	7.28683	0 0000	0 00	1195.090	1195 090	550 13	550.13	561 74	-1 578	-1.578	39.10	766.23	-1208.1
4	9	7 28192	0 0000	0 00	1197 204	1197 204	550.54	550 54	561.70	-1 578	-1 578	39 05	765.47	-1204 2
4	10	7 27983	0 0000	0 00	1204.078	1204.078	550 87	550 87	561.68	-1 578	-1.578	38.95	763.02	-1204.7
4	11	7 27915	0 0000	0.00	1203.301	1203 301	550.72	550 72	561 67	-1 578	-1 578	38 95	763 29	-1206 5
4	12	7.27848	0.0000	0 00	1200.095	1200 095	550.10	550.10	561 66	-1 578	-1.578	38.99	764.44	-1209.2
4	13	7.27287	0 0000	0 00	1195.299	1195 299	550.17	550 17	561 61	-1 578	-1 578	39 01	766.14	-1209.2
4	14	7.26617	0 0000	0 00	1195.301	1195.301	550.17	550 17	561.55	-1 578	-1.578	38.96	766.13	-1209.2
4	15	7 26117	0 0000	0 00	1195 302	1195 302	550.17	550 17	561.50	-0 066	-0.066	38.93	766.12	-50.5
4	16	7.25027	0 0000	0.00	1195.304	1195.304	550.17	550.17	561.40	-0.048	-0.048	38.86	766.11	-37.1
4	17	7 24392	0 0000		1195 304	1195 305	550 17	550.17	561.34			38.81	766 10	
5	1	7 31514	0.0000	0.00	1194 724	1194 724	550 06	550 06	562.01	0 000	0 000	39.30	766.40	0.0
5	2	7 31467	0 0000	0 00	1189.448	1189 448	551 03	551 03	562 00	0 329	0 329	39.36	768.27	252.6
5	3	7.30936	0 0000	0 00	1211 076	1211 076	553 22	553 22	561.95	0 150	0.117	39.06	760.54	88 9
5	4	7.30216	0.0000	0.00	1267 900	1267 897	558.93	558.93	561 89	0 382	0.125	38.33	739.58	92.6
5	5	7 29551	0 2058	2.78	1325.024	1283 645	561.82	561 82	561.82	0 576	0 271	38.10	733.58	162 5
5	6	7.29015	0.3812	5.30	1362.194	1283 373	561.77	561.77	561.77	0.763	0.436	38.07	733.68	208.9
5	7	7.28572	0 4960	7 55	1395 567	1283.149	561.73	561.73	561 73	0.951	0.594	38 05	733.76	237.5
5	8	7.28192	0 5801	9 85	1429 538	1282 956	561 70	561.70	561.70	1.148	0 752	38.03	733 83	257.1
5	9	7.27921	0 6229	11 23	1449.978	1282 819	561 67	561 67	561 67	1.260	0 852	38 01	733 88	265.7
5	10	7 27811	0 5741	11.51	1454.155	1282 763	561.66	561.66	561.66	1.392	0 747	38.00	733.90	263.7
5	11	7.27720	0 6259	11 72	1457 171	1282 717	561.65	561.65	561.65	1.187	0 775	38.00	733.91	240.9
5	12	7 27687	0 6277	11 80	1458.423	1282 699	561 65	561 65	561.65	1.158	0 755	38 00	733.92	233 9
5	13	7 27425	0 6352	12 14	1463.361	1282 567	561 62	561 62	561.62	1 180	0 769	37.98	733.96	234 3
5	14	7 27118	0 6776	12 31	1465 722	1282 410	561 60	561.60	561.60	2.186	1.693	37.96	734.02	456.9
5	15	7 26801	0 6562	11 37	1451 529	1282 250	561.57	561.57	561.57	1 752	1.348	37.94	734.08	383.7
5	16	7 26395	0 6839	13 98	1452 568	1282 043	561.53	561 53	561 53	1 239	0.842	37 92	734.15	258.5
5	17	7 26154	1 0000		1452.568	1282.045	561.53	561.53	561.50			37.90	734.14	
6	1	7.31548	0 0000	0.00	1186 919	1186.919	550 64	550.64	562.01	0.000	0.000	39.40	769 16	0.0
6	2	7.31500	0 0000	0 00	1186 970	1186.970	550.54	550 54	562.01	-1 588	-1.588	39.40	769.14	-1221.5
6	3	7 31694	0 0000	0 00	1187 380	1187.380	550.61	550.61	562.02	-1 588	-1.588	39.41	769 00	-1220.0
6	4	7 31110	0 0000	0 00	1190.150	1190 150	550.17	550 17	561.97	-1 588	-1.588	39.33	768.01	-1218 4
6	5	7.30492	0 0000	0 00	1192 874	1192.874	550.61	550.61	561.91	-1.588	-1.588	39.25	767.04	-1217.4
6	6	7.29873	0.0000	0.00	1194 779	1194.779	550.07	550.07	561.85	-1 589	-1.589	39.19	766 36	-1217.0
6	7	7.29256	0 0000	0 00	1195 450	1195 450	550.20	550.20	561 80	-1.589	-1.589	39.14	766.11	-1217.2
6	8	7 28638	0 0000	0 00	1195.090	1195 090	550.15	550.15	561.74	-1 589	-1.589	39 10	766.23	-1215.9
6	9	7.28157	0.0000	0 00	1197 256	1197.256	550.55	550 55	561.69	-1.589	-1.589	39 04	765.45	-1212.1
6	10	7.27951	0 0000	0.00	1204 130	1204 130	550 88	550.88	561.67	-1.589	-1 589	38.94	763.00	-1212.5
6	11	7.27885	0 0000	0 00	1203 301	1203.301	550.72	550.72	561.67	-1.589	-1 589	38.95	763 29	-1214.4
6	12	7.27819	0 0000	0.00	1200 095	1200 095	550.10	550 10	561 66	-1 589	-1.589	38.98	764.44	-1217.1
6	13	7.27270	0 0000	0 00	1195 299	1195 299	550.17	550.17	561 61	-1.589	-1.589	39.01	766.14	-1217.1
6	14	7 26615	0 0000	0 00	1195 301	1195 301	550.17	550.17	561.55	-1.589	-1 589	38.96	766.13	-1217 1
6	15	7 26117	0.0000	0 00	1195 302	1195.302	550.17	550.17	561.50	-0 007	-0.007	38.93	766.12	-5 1
6	16	7.25027	0 0000	0 00	1195 304	1195.304	550.17	550.17	561.40	-0.005	-0.005	38.86	766.11	-3.7
6	17	7.24392	0.0000		1195 304	1195 305	550.17	550.17	561.34			38.81	766.10	

ic	vvx	vlx	vvy	vly	iz	ic	vvx	vlx	vvy	vly
1	0.000E+00	0.000E+00	0.000E+00	0.000E+00	2	2	0.000E+00	0.000E+00	0.000E+00	0.000E+00
1	0.000E+00	0.000E+00	0.000E+00	0.000E+00	3	2	0.000E+00	0.000E+00	0.000E+00	0.000E+00
1	0.000E+00	0.000E+00	0.000E+00	0.000E+00	4	2	0.000E+00	0.000E+00	0.000E+00	0.000E+00
1	0.000E+00	0.000E+00	0.000E+00	0.000E+00	5	2	0.000E+00	0.000E+00	0.000E+00	0.000E+00
1	0.000E+00	0.000E+00	0.000E+00	0.000E+00	6	2	0.000E+00	0.000E+00	0.000E+00	0.000E+00
1	0.000E+00	0.000E+00	0.000E+00	0.000E+00	7	2	0.000E+00	0.000E+00	0.000E+00	0.000E+00
1	0.000E+00	0.000E+00	0.000E+00	0.000E+00	8	2	0.000E+00	0.000E+00	0.000E+00	0.000E+00
1	0.000E+00	0.000E+00	0.000E+00	0.000E+00	9	2	0.000E+00	0.000E+00	0.000E+00	0.000E+00
1	0.000E+00	0.000E+00	0.000E+00	0.000E+00	10	2	0.000E+00	0.000E+00	0.000E+00	0.000E+00
1	0.000E+00	0.000E+00	0.000E+00	0.000E+00	11	2	0.000E+00	0.000E+00	0.000E+00	0.000E+00
1	0.000E+00	0.000E+00	0.000E+00	0.000E+00	12	2	0.000E+00	0.000E+00	0.000E+00	0.000E+00
1	0.000E+00	0.000E+00	0.000E+00	0.000E+00	13	2	0.000E+00	0.000E+00	0.000E+00	0.000E+00
1	0.000E+00	0.000E+00	0.000E+00	0.000E+00	14	2	0.000E+00	0.000E+00	0.000E+00	0.000E+00
1	0.000E+00	0.000E+00	0.000E+00	0.000E+00	15	2	0.000E+00	0.000E+00	0.000E+00	0.000E+00
1	0.000E+00	0.000E+00	0.000E+00	0.000E+00	16	2	0.000E+00	0.000E+00	0.000E+00	0.000E+00
3	1.306E-01	1.306E-01	1.012E-01	1.012E-01	2	4	0.000E+00	0.000E+00	0.000E+00	0.000E+00
3	0.000E+00	0.000E+00	0.000E+00	0.000E+00	3	4	0.000E+00	0.000E+00	0.000E+00	0.000E+00
3	0.000E+00	0.000E+00	0.000E+00	0.000E+00	4	4	0.000E+00	0.000E+00	0.000E+00	0.000E+00
3	0.000E+00	0.000E+00	0.000E+00	0.000E+00	5	4	0.000E+00	0.000E+00	0.000E+00	0.000E+00
3	0.000E+00	0.000E+00	0.000E+00	0.000E+00	6	4	0.000E+00	0.000E+00	0.000E+00	0.000E+00
3	0.000E+00	0.000E+00	0.000E+00	0.000E+00	7	4	0.000E+00	0.000E+00	0.000E+00	0.000E+00
3	0.000E+00	0.000E+00	0.000E+00	0.000E+00	8	4	0.000E+00	0.000E+00	0.000E+00	0.000E+00
3	0.000E+00	0.000E+00	0.000E+00	0.000E+00	9	4	0.000E+00	0.000E+00	0.000E+00	0.000E+00
3	0.000E+00	0.000E+00	0.000E+00	0.000E+00	10	4	0.000E+00	0.000E+00	0.000E+00	0.000E+00
3	0.000E+00	0.000E+00	0.000E+00	0.000E+00	11	4	0.000E+00	0.000E+00	0.000E+00	0.000E+00
3	0.000E+00	0.000E+00	0.000E+00	0.000E+00	12	4	0.000E+00	0.000E+00	0.000E+00	0.000E+00
3	0.000E+00	0.000E+00	0.000E+00	0.000E+00	13	4	0.000E+00	0.000E+00	0.000E+00	0.000E+00
3	0.000E+00	0.000E+00	0.000E+00	0.000E+00	14	4	0.000E+00	0.000E+00	0.000E+00	0.000E+00
3	0.000E+00	0.000E+00	0.000E+00	0.000E+00	15	4	0.000E+00	0.000E+00	0.000E+00	0.000E+00
3	0.000E+00	0.000E+00	0.000E+00	0.000E+00	16	4	0.000E+00	0.000E+00	0.000E+00	0.000E+00
5	1.298E-01	1.298E-01	-6.267E-02	-6.267E-02	2	6	0.000E+00	0.000E+00	-1.005E-01	-1.005E-01
5	0.000E+00	0.000E+00	-8.154E-02	-8.154E-02	3	6	0.000E+00	0.000E+00	0.000E+00	0.000E+00
5	0.000E+00	0.000E+00	8.962E-03	1.720E-03	4	6	0.000E+00	0.000E+00	0.000E+00	0.000E+00
5	0.000E+00	0.000E+00	6.265E-02	4.180E-02	5	6	0.000E+00	0.000E+00	0.000E+00	0.000E+00
5	0.000E+00	0.000E+00	5.051E-02	3.429E-02	6	6	0.000E+00	0.000E+00	0.000E+00	0.000E+00
5	0.000E+00	0.000E+00	3.664E-02	2.506E-02	7	6	0.000E+00	0.000E+00	0.000E+00	0.000E+00
5	0.000E+00	0.000E+00	3.057E-02	1.980E-02	8	6	0.000E+00	0.000E+00	0.000E+00	0.000E+00
5	0.000E+00	0.000E+00	3.164E-02	1.605E-02	9	6	0.000E+00	0.000E+00	0.000E+00	0.000E+00
5	0.000E+00	0.000E+00	2.933E-02	7.659E-03	10	6	0.000E+00	0.000E+00	0.000E+00	0.000E+00
5	0.000E+00	0.000E+00	-3.781E-02	-4.401E-03	11	6	0.000E+00	0.000E+00	0.000E+00	0.000E+00
5	0.000E+00	0.000E+00	-5.330E-02	-1.168E-02	12	6	0.000E+00	0.000E+00	0.000E+00	0.000E+00
5	0.000E+00	0.000E+00	-4.922E-03	-6.801E-03	13	6	0.000E+00	0.000E+00	0.000E+00	0.000E+00
5	0.000E+00	0.000E+00	6.048E-03	-1.867E-03	14	6	0.000E+00	0.000E+00	0.000E+00	0.000E+00
5	0.000E+00	0.000E+00	-4.805E-02	-2.243E-02	15	6	0.000E+00	0.000E+00	0.000E+00	0.000E+00
5	0.000E+00	0.000E+00	-4.150E-02	-2.878E-02	16	6	0.000E+00	0.000E+00	0.000E+00	0.000E+00

primary and wall parameters   time =   7.999997 sec   time step size = 0.20000E-01 sec   fouling coef. =   1.000

CHANNEL NUMBER IS i= 3 (ktb= 1)

hlfc	hlnb	hvfc	tprim	twprim	ttube	twsec	iz	sec hflux	prim hflux	hprim	%(1-E2/E1)
0.00000E+00	0.00000E+00	0.00000E+00	581.30	577.20	577.20	577.20	1	0.000000E+00	0.000000E+00	0.000000E+00	0.00
2.67767E+03	2.84183E+04	0.00000E+00	581.17	576.65	571.34	566.32	2	1.787702E+05	2.024215E+05	4.480783E+04	0.00
5.66764E+03	2.16205E+04	0.00000E+00	580.08	575.93	571.06	566.45	3	1.640543E+05	1.857579E+05	4.479511E+04	-0.00
6.57870E+03	2.15483E+04	0.00000E+00	578.22	574.73	570.64	566.78	4	1.376375E+05	1.558502E+05	4.477508E+04	0.00
6.09853E+03	2.08454E+04	0.00000E+00	576.43	573.33	569.72	566.32	5	1.211684E+05	1.372008E+05	4.418018E+04	0.00
6.76775E+03	1.82405E+04	0.00000E+00	574.87	572.15	568.99	566.01	6	1.059938E+05	1.200164E+05	4.414658E+04	0.00
7.30090E+03	1.61356E+04	0.00000E+00	573.48	571.09	568.32	565.70	7	9.308640E+04	1.054004E+05	4.411482E+04	-0.00
7.78637E+03	1.43187E+04	0.00000E+00	572.26	570.16	567.71	565.41	8	8.201015E+04	9.286019E+04	4.408511E+04	0.00
8.11288E+03	1.30906E+04	0.00000E+00	571.40	569.49	567.27	565.18	9	7.439399E+04	8.423582E+04	4.406319E+04	-0.00
8.22112E+03	1.26871E+04	0.00000E+00	571.11	569.27	567.12	565.10	10	7.189173E+04	8.140339E+04	4.405565E+04	0.00

partial primary power output =   337.762 KW   partial secondary power input =   337.760 KW

CHANNEL NUMBER IS i= 5 (ktb= 1)

hlfc	hlnb	hvfc	tprim	twprim	ttube	twsec	iz	sec hflux	prim hflux	hprim	%(1-E2/E1)
0.00000E+00	0.00000E+00	0.00000E+00	566.26	563.50	563.50	563.50	1	0.000000E+00	0.000000E+00	0.000000E+00	0.00
2.41512E+03	3.01487E+03	0.00000E+00	566.29	565.11	563.74	562.45	2	4.581399E+04	5.187300E+04	4.390849E+04	-0.00
3.05864E+03	4.47951E+03	0.00000E+00	566.63	565.41	563.99	562.66	3	4.730899E+04	5.357082E+04	4.391953E+04	0.01
1.83743E+03	1.50697E+04	0.00000E+00	567.20	566.19	565.01	563.90	4	3.942088E+04	4.463845E+04	4.394543E+04	0.01
2.64806E+03	1.60414E+04	0.00000E+00	567.79	566.67	565.37	564.15	5	4.342721E+04	4.917230E+04	4.396333E+04	-0.00
4.25423E+03	1.47012E+04	0.00000E+00	568.44	567.18	565.72	564.35	6	4.880680E+04	5.526505E+04	4.398225E+04	0.00
5.60199E+03	1.38251E+04	0.00000E+00	569.17	567.75	566.11	564.56	7	5.499080E+04	6.226863E+04	4.400299E+04	0.00
6.78732E+03	1.32650E+04	0.00000E+00	569.99	568.39	566.54	564.79	8	6.209039E+04	7.030633E+04	4.402568E+04	0.00
7.66476E+03	1.29329E+04	0.00000E+00	570.69	568.94	566.90	564.98	9	6.821671E+04	7.724253E+04	4.404450E+04	0.00
8.01664E+03	1.28324E+04	0.00000E+00	571.01	569.19	567.06	565.07	10	7.102812E+04	8.042464E+04	4.405290E+04	-0.00

partial primary power output =   152.776 KW   partial secondary power input =   152.772 KW

primary and wall parameters   time =   7.999997 sec   time step size = 0.20000E-01 sec   fouling coef. =   1.000

CHANNEL NUMBER IS i= 3 (ktb= 2)

hlfc	hlnb	hvfc	tprim	twprim	ttube	twsec	iz	sec hflux	prim hflux	hprim	%(1-E2/E1)
0.00000E+00	0.00000E+00	0.00000E+00	581.30	577.20	577.20	577.20	1	0.000000E+00	0.000000E+00	0.000000E+00	0.00
2.67767E+03	2.83428E+04	0.00000E+00	581.16	576.60	571.31	566.31	2	1.780909E+05	2.016512E+05	4.416326E+04	-0.00
5.66764E+03	2.15466E+04	0.00000E+00	580.06	575.87	571.02	566.44	3	1.633061E+05	1.849106E+05	4.415060E+04	-0.00
6.57870E+03	2.14692E+04	0.00000E+00	578.17	574.67	570.60	566.76	4	1.367514E+05	1.548457E+05	4.413072E+04	0.00
6.09853E+03	2.07466E+04	0.00000E+00	576.37	573.25	569.67	566.30	5	1.201569E+05	1.360568E+05	4.354381E+04	0.00
6.76774E+03	1.81331E+04	0.00000E+00	574.79	572.06	568.93	565.99	6	1.049222E+05	1.188041E+05	4.351038E+04	0.00
7.30088E+03	1.60221E+04	0.00000E+00	573.40	571.00	568.26	565.68	7	9.198818E+04	1.041565E+05	4.347882E+04	-0.00
7.78635E+03	1.42012E+04	0.00000E+00	572.17	570.06	567.65	565.38	8	8.090860E+04	9.161284E+04	4.344934E+04	0.00
8.11286E+03	1.29710E+04	0.00000E+00	571.31	569.40	567.21	565.15	9	7.330188E+04	8.299892E+04	4.342761E+04	-0.00
8.22109E+03	1.25142E+04	0.00000E+00	570.96	569.13	567.03	565.05	10	7.032981E+04	7.963461E+04	4.341878E+04	0.00
8.00453E+03	1.27285E+04	0.00000E+00	570.91	569.08	566.99	565.03	11	6.998336E+04	7.924202E+04	4.341727E+04	0.00

partial primary power output =   339.531 KW   partial secondary power input =   339.528 KW

CHANNEL NUMBER IS i= 5 (ktb= 2)

hlfc	hlnb	hvfc	tprim	twprim	ttube	twsec	iz	sec hflux	prim hflux	hprim	%(1-E2/E1)
0.00000E+00	0.00000E+00	0.00000E+00	566.02	563.50	563.50	563.50	1	0.000000E+00	0.000000E+00	0.000000E+00	0.00
2.41512E+03	1.97928E+03	0.00000E+00	566.06	564.89	563.55	562.29	2	4.467768E+04	5.058636E+04	4.327013E+04	-0.00
3.05864E+03	3.65105E+03	0.00000E+00	566.39	565.19	563.82	562.53	3	4.584768E+04	5.191577E+04	4.328114E+04	0.01
1.83743E+03	1.45237E+04	0.00000E+00	566.95	565.98	564.87	563.82	4	3.712940E+04	4.204410E+04	4.330691E+04	0.01
2.64805E+03	1.55232E+04	0.00000E+00	567.51	566.44	565.22	564.07	5	4.085882E+04	4.626411E+04	4.332399E+04	-0.00
4.25420E+03	1.42197E+04	0.00000E+00	568.14	566.93	565.56	564.26	6	4.600982E+04	5.209823E+04	4.334200E+04	0.00
5.60195E+03	1.33697E+04	0.00000E+00	568.84	567.48	565.93	564.47	7	5.193559E+04	5.880864E+04	4.336179E+04	0.00
6.78726E+03	1.28284E+04	0.00000E+00	569.63	568.10	566.34	564.69	8	5.874457E+04	6.651709E+04	4.338353E+04	0.00
7.66468E+03	1.25085E+04	0.00000E+00	570.31	568.62	566.69	564.87	9	6.462634E+04	7.317664E+04	4.340161E+04	0.00
8.01656E+03	1.24123E+04	0.00000E+00	570.62	568.86	566.85	564.96	10	6.732715E+04	7.623424E+04	4.340969E+04	-0.00
7.79823E+03	1.27892E+04	0.00000E+00	570.72	568.94	566.90	564.97	11	6.837107E+04	7.741565E+04	4.341233E+04	-0.00

partial primary power output =   149.426 KW   partial secondary power input =   149.422 KW primary and wall parameters time = 7.999997 sec time step size = 0.20000E-01 sec fouling coef. = 1.000

CHANNEL NUMBER IS i= 3 (ktb= 3)

hlfc	hlnb	hvfc	tprim	twprim	ttube	twsec	iz	sec hflux	prim hflux	hprim	%(1-E2/E1)
0.00000E+00	0.00000E+00	0.00000E+00	581.30	577.20	577.20	577.20	1	0.000000E+00	0.000000E+00	0.000000E+00	0.00
2.67767E+03	2.82776E+04	0.00000E+00	581.16	576.55	571.28	566.30	2	1.775055E+05	2.009881E+05	4.361958E+04	-0.00
5.66764E+03	2.14827E+04	0.00000E+00	580.05	575.82	570.99	566.43	3	1.626619E+05	1.841815E+05	4.360698E+04	0.00
6.57870E+03	2.14009E+04	0.00000E+00	578.14	574.61	570.56	566.75	4	1.359895E+05	1.539836E+05	4.358722E+04	0.00
6.09852E+03	2.06617E+04	0.00000E+00	576.32	573.18	569.63	566.28	5	1.192913E+05	1.350743E+05	4.300706E+04	0.00
6.76773E+03	1.80411E+04	0.00000E+00	574.73	571.99	568.89	565.97	6	1.040074E+05	1.177668E+05	4.297378E+04	-0.00
7.30086E+03	1.59246E+04	0.00000E+00	573.33	570.93	568.21	565.65	7	9.105021E+04	1.030952E+05	4.294239E+04	-0.00
7.78633E+03	1.41005E+04	0.00000E+00	572.09	569.98	567.60	565.35	8	7.997017E+04	9.054964E+04	4.291311E+04	-0.00
8.11284E+03	1.28685E+04	0.00000E+00	571.23	569.32	567.15	565.12	9	7.237131E+04	8.194526E+04	4.289154E+04	-0.00
8.22106E+03	1.24110E+04	0.00000E+00	570.88	569.05	566.98	565.02	10	6.940450E+04	7.858774E+04	4.288276E+04	0.00
8.00450E+03	1.25702E+04	0.00000E+00	570.77	568.96	566.91	564.99	11	6.858849E+04	7.766261E+04	4.287992E+04	0.00
7.75498E+03	1.28220E+04	0.00000E+00	570.72	568.92	566.88	564.96	12	6.821067E+04	7.723604E+04	4.287849E+04	0.00

partial primary power output = 431.855 KW partial secondary power input = 431.854 KW

CHANNEL NUMBER IS i= 5 (ktb= 3)

hlfc	hlnb	hvfc	tprim	twprim	ttube	twsec	iz	sec hflux	prim hflux	hprim	%(1-E2/E1)
0.00000E+00	0.00000E+00	0.00000E+00	565 79	563.50	563.50	563.50	1	0.000000E+00	0.000000E+00	0.000000E+00	0.00
2.41512E+03	8.21930E+02	0.00000E+00	565.83	564.67	563.36	562.12	2	4.378726E+04	4.957897E+04	4.273052E+04	-0.00
3.05864E+03	2.76868E+03	0.00000E+00	566.16	564.98	563.64	562.39	3	4.453089E+04	5.042597E+04	4.274160E+04	0.01
1.83743E+03	1.39703E+04	0.00000E+00	566.70	565.78	564.73	563.75	4	3.488789E+04	3.950670E+04	4.276731E+04	0.01
2.64804E+03	1.49962E+04	0 00000E+00	567.24	566.22	565.07	564.00	5	3.832600E+04	4.339625E+04	4.278358E+04	-0.00
4.25417E+03	1.37274E+04	0.00000E+00	567 83	566.69	565.39	564.18	6	4.323418E+04	4.895469E+04	4.280066E+04	0.00
5.60191E+03	1.29012E+04	0 00000E+00	568.50	567.21	565.75	564.37	7	4.888080E+04	5.534891E+04	4.281949E+04	0.00
6.78719E+03	1.23763E+04	0.00000E+00	569.26	567 80	566.14	564.59	8	5.537298E+04	6.270032E+04	4.284024E+04	0.00
7.66460E+03	1.20669E+04	0.00000E+00	569.91	568.30	566.48	564.76	9	6.098696E+04	6.905592E+04	4.285754E+04	0.00
8.01648E+03	1 19742E+04	0.00000E+00	570.21	568.53	566.63	564.84	10	6.356615E+04	7.197543E+04	4.286529E+04	-0.00
7.79814E+03	1.23418E+04	0.00000E+00	570 31	568.60	566.67	564.86	11	6.455332E+04	7.309429E+04	4.286782E+04	0.00
7.45097E+03	1.28743E+04	0.00000E+00	570.41	568.68	566.72	564.87	12	6.555892E+04	7.423099E+04	4.287038E+04	-0.00

partial primary power output = 186.840 KW partial secondary power input = 186.835 KW

Estimate of Forward Time Values

N	m2=kg	h2a=J/Kg	t2a=K	ro2a=Kg/m3	qw=MW	M1=Kg	t1=K	h1=J/kg	ro1=Kg/m3	qs=MW	tsat=K
0	134 046	1281056 73	561.34488	734 50	0.000	13.108	561.34	2771685 09	37.81	0.000	561.345
11	134 032	1281056 73	561 34488	734 50	0.000	13.106	561.34	2771685.09	37.81	0.000	561.345

New Boundary Conditions

ro2(kg/m3)	t2(k)	h2(j/Kg)	vd(m3)	xl2(m)	pd(Pa)	pr(Pa)	dm1dt=Kg/s	dm2dt=Kg/s	h2err(J/Kg)	h1err(J/Kg)
842.94	550 06	0 1210926E+07	0.73000	11.2367	7.23851E+06	7.27482E+06	-2.9210E-01	-7.0303E-01	0.00	0.00

dpris= 2 016E+04 dpsep= 3 027E+04 Pdome= 7 244554E+06 Pa t2b= 550.06 K h2b= 941452.65 J/Kg ro2b= 846 69 hg= 2771685.09J/Kg

Initial Conditions

wd	wr	xr	roavg	ws	wfd	cverr	icase	hfd	hf	nfeed	nzlvl	nmin
11 61	11 61	0.1460	611.63	1.69	1.69	0.00028	4	0 8003231E+06	0.1281057E+07	12	15	13

English Units

Pdome= 1050.74 psia wfd= 13.43 Klbm/hr ws= 13.43 Klbm/hr level= percent time= 0.800E+01 sec dt= 0.02000 sec
wfla= 0 00Klbm/hr(0.00 Kg/s) wcond= 0.00 Klbm/hr(0.00 Kg/s) xmfd= 0.00000

APPENDIX G

CODE LISTING

```
c                    THERMIT-UTSG version of December 20,1983
c
c          Text reference of this version is "Thermohydraulic Analysis
c          of U-Tube Steam Generators" by Hugo C. da Silva Jr.,
c          Ph.D Thesis, MIT, 1984.
c ***********************************************************************
c
c                DESCRIPTION OF DATA BASE FOR CODE
c
c
c common /ic/     integer constants
c
c nstep   Time step number
c nitmax  Maximum number of Newton iterations
c nitno   Counter for Newton iterations
c iitmax  Maximum number of inner iterations
c iitot   Total inner iterations for time step
c iic     Inner iteration count for one Newton iteration
c kred    Counter for reduced time steps
c nm      Total number of basic unknowns
c ntc     Number of title cards
c nc      Number of cells in x-y plane
c nrods   Number of fuel rods or fuel rod sections to be modeled
c          (ie: for subchannel analysis each rod is
c           divided into four sections,for UTSG analysis nrods=nc)
c nz      Number of axial cells
c nr      Number of rows of cells in x-y plane
c ncp     nc+1
c nzp     nz+1
c nzp2    nz+2
c iflash  Phase change indicator (0/1/2)(Nigmatulin/suppressed/subcooled
)
c itb     Top boundary condition indicator (0/1)(pressure/velocity)
c         (for UTSG applications set to zero)
c ibb     Bottom boundary condition indicator (0/1)(pressure/velocity)
c         (for UTSG applications set to one)
c icpu    Initial cpu time
c iwft    Wall friction indicator for transverse direction
c          (0/1)(no transverse friction/Gunter-Shaw correlation)
c ivec    Indicator for transverse velocity to be used in
c         friction calculations
c          (0/1)  (Vx used/ magnitude of V used)
c         The next 5 variables have no meaning for UTSG's and
c         should be set to the values given in the reference text
c nodes   Number of temperature nodes in fuel pin
c ndm1    Number of cells in fuel pin (nodes-1)
c ncf     number of cells in fuel
c ncc     Number of cells in clad
c ng      Cell in which gap is located
c iht     Heat transfer indicator:(0/1)  (no heat transfer/normal
c         calculation)
c iss     Indicator for Heat Transfer calculations (1/2)
c         (chf test/no chf test)
c iqss    Indicator for heat flux boundary condition (0 in UTSG version)
c ichf    Indicator for chf correlation
```

```
 c          (biasi/w-3/cise)  (1/2/3)
 c itam    Transverse area indicator (0/1) (no transverse flow/normal)
 c ires    Restart indicator
 c idump   Dump indicator
 c ntitle  Number of title words (1 word = 4 characters)
 c itwmax  Channel location of maximum wall temperature
 c jtwmax  Axial location of maximum wall temperarure
 c itrmax  Channel location of maximum rod temperature
 c jtrmax  Axial location of maximum rod temperature
 c imchfr  Channel location of minimum chf ratio
 c jmchfr  Axial location of minimum chf ratio
 c ichf    CHF indicator
 c ifintr  Interfacial friction model indicator (0/1) (MIT/LASL)
 c ierr    Error code
 c lerr    Logical error flag
 c imixm   Indicator for momentum mixing (0/1) (no mixing/mixing included)
 c imixe   Indicator for energy mixing (0/1) (no mixing/mixing included)
 c iafm    Axial friction model indicator (0/1) (default/user supplied)
 c itfm    Transverse friction model indicator (0/1) (default/user supplie
d)
 c igfm    Grid friction model (0/1) (default/user supplied)
 c
 c
 c common /rc/     real constants
 c
 c delt    Time step size
 c rdelt   1/delt
 c errn    Newton iteration error
 c epsn    Newton iteration convergence criterion
 c erri    Inner iteration error
 c epsi    Inner iteration convergence criterion
 c dtmin   Minimum time step size
 c dtmax   Maximum time step size
 c tend    End of time zone
 c dtsp    Short print time interval
 c dtlp    Long print time interval
 c rtnsp   Real time at next short print
 c rtnlp   Real time at next long print
 c gravx    Gravitational constant (normally 0.0)
 c gravy    Gravitational constant (normally 0.0)
 c gravz    Gravitational constant (normally -9.81)
 c rtime   Real time
 c hdt     Hydraulic diameter in transverse direction
 c velx    Velocity multiplier for transverse friction correlation
 c q       Total power level (w)
 c q0      Initial total power (w)
 c twmax   Maximum wall temperature
 c trmax   Maximum rod temperature
 c amchfr  Minimum critical heat flux ratio
 c
 c
 c
 c common /force/    Transient forcing function data
 c                   (consult text for detailed explanation
 c                   of this input)
```

```
c
c botfac(30)    Feedwater flow rate values(Kg/s) (transient) or
c               Power Level for whole UTSG (Watt) (steady-state)
c yb(30)        Time vector for botfac
c topfac(30)    Feedwater Temperatures (K)
c yt(30)        Time vector for topfac
c tinfac(30)    Steam Dome Pressures (psi)
c ytemp(30)     Time vector for tinfac
c qfac(30)      Primary Pressure values(Pa)
c yq(30)        Time vector for qfac
c ptfac(30)     Inlet Primary Temperatures(k)
c ypt(30)       Time vector for ptfac
c pgfac(30)     Average Primary Mass flux(kg/m2/sec)
c ypg(30)       Time vector for pgfac
c nb            Number of entries in botfac table   <=30
c nt            Number of entries in topfac table   <=30
c ntemp         Number of entries in tinfac table   <=30
c nq            Number of entries in qfac table   <=30
c npt           Number of entries in ptfac table <=30
c npg           Number of entries in pgfac table <=30
c
c
c common /fricmd/      constants in friction model
c
c fcon(1,1-4)    Axial model
c fcon(2,1-4)    Grid model
c fcon(3,1-4)    Grid model with funnel effect
c fcon(4,1-4)    Transverse model
c
c common /point/     pointers for all arrays
c
c lncr    ncr(nr)               Number of channels per row
c lindnt  indent(nr)            Indentation of each row
c licc    icc(4,nc)             Indices for four adjacent subchannels
c liwfz   iwfz(nzp)             Indicator for axial wall friction
c lihtr   ihtr(nz,nrods)        Heat transfer regime indicator
c licr    icr(nrods)            Indices of channels adjacent to rods
c lirc    irc(4,nc)             Indices of rods adjacent to channels
c
c lp      p(nzp2,ncp)           Old pressure
c lalp    alp(nzp2,ncp)         Old vapor volume fraction
c lrov    rov(nzp2,ncp)         Old vapor density
c lrol    rol(nzp2,ncp)         Old liquid density
c lev     ev(nzp2,ncp)          Old vapor specific internal energy
c lel     el(nzp2,ncp)          Old liquid specific internal energy
c ltv     tv(nzp2,ncp)          Old vapor temperature
c ltl     tl(nzp2,ncp)          Old liquid temperature
c ltr     tr(4*nktb,nzp2,nc)    Old primary + wall temperatures
c
c lpn     pn(nzp2,ncp)          New Pressure
c lalpn   alpn(nzp2,ncp)        New vapor volume fraction
c lrovn   rovn(nzp2,ncp)        New vapor density
c lroln   roln(nzp2,ncp)        New liquid density
c levn    evn(nzp2,ncp)         New vapor specific internal energy
```

```
c leln   eln(nzp2,ncp)       New liquid specific internal energy
c ltvn   tvn(nzp2,ncp)       New vapor temperature
c ltln   tln(nzp2,ncp)       New liquid temperature
c ltrn   trn(4*nktb,nzp2,nrods) New primary + wall temperatures
c
c lvvx   vvx(nzp2,ncp)       Old vapor velocity in x direction
c lvlx   vlx(nzp2,ncp)       Old liquid velocity in x direction
c lvvy   vvy(nzp2,ncp)       Old vapor velocity in y direction
c lvly   vly(nzp2,ncp)       Old liquid velocity in y direction
c lvvz   vvz(nzp,ncp)        Old vapor velocity in z direction
c lvlz   vlz(nzp,ncp)        Old liquid velocity in z direction
c lhv    hv(nzp2,ncp)        Old vapor specific enthalpy
c lhl    hl(nzp2,ncp)        Old liquid specific enthalpy
c lhvs   hvs(nzp2,ncp)       Old vapor saturation enthalpy
c lhls   hls(nzp2,ncp)       Old liquid saturation enthalpy
c ltsat  tsat(nzp2,ncp)      Saturation temperature
c lvisv  visv(nzp2,ncp)      Viscosity of vapor
c lvisl  visl(nzp2,ncp)      Viscosity of liquid
c
c ldx    dx(ncp)             Mesh spacing in x direction
c ldy    dy(ncp)             Mesh spacing in y direction
c ldz    dz(nzp2)            Mesh spacing in z direction
c larx   arx(nz,ncp)         Mesh cell areas in x direction
c lary   ary(nz,ncp)         Mesh cell areas in y direction
c larz   arz(nzp,nc)         Mesh cell areas in z direction
c lvol   vol(nz,nc)          Mesh cell volumes
c lcpvx  cpvx(nz,nc)         Pressure coefficient in x dir vap mom eq
c lcpvy  cpvy(nz,nc)         Pressure coefficient in y dir vap mom eq
c lcpvz  cpvz(nzp,nc)        Pressure coefficient in z dir vap mom eq
c lcplx  cplx(nz,nc)         Pressure coefficient in x dir liq mom eq
c lcply  cply(nz,nc)         Pressure coefficient in y dir liq mom eq
c lcplz  cplz(nzp,nc)        Pressure coefficient in z dir liq mom eq
c lfvx   fvx(nz,nc)          Explicit terms in x direction vap mom eq
c lfvy   fvy(nz,nc)          Explicit terms in y direction vap mom eq
c lfvz   fvz(nzp,nc)         Explicit terms in z direction vap mom eq
c lflx   flx(nz,nc)          explicit terms in x direction liq mom eq
c lfly   fly(nz,nc)          Explicit terms in y direction liq mom eq
c lflz   flz(nzp,nc)         Explicit terms in z direction liq mom eq
c lajm1  ajm1(3,nz,nc)       Tridiagonal part of Jacobian matrix
c lajm2  ajm2(4,nz,nc)       Remainder of Jacobian matrix
c lcpa   cpa(6,nz,nc)        Pressure coefficients in eq for alpha
c lcptv  cptv(6,nz,nc)       Pressure coefficients in eq for vapor temp
c lcptl  cptl(6,nz,nc)       Pressure coefficients in eq for liquid temp
c lrhs   rhs(nz,nc,4)        Right hand side of all equations
c ldp    dp(nzp2,nc)         Pressure change
c lqpp   qpp(nzp2,nrods)       Total linear heat flux
c lqv    qv(nzp2,nrods)        Heat Flux to vapor for transition boiling
c lql    ql(nzp2,nrods)        Heat flux to liquid for transition boilin
g
c lhvfc  hvfc(nzp2,nrods)      Heat transfer coefficient to vapor
c lhlnb  hlnb(nzp2,nrods)      Heat transfer coefficient to liquid
c                             (nucleate boiling)
c lhlfc  hlfc(nz,nrods)      Heat transfer coefficient to liquid
c                             (forced convection)
c ldtrn dtrn(nktb,nzp2,nc)  secondary heat flux for each tube bank
```

```
 c                                during fluid dynamics iterations
 c ldtw    dtw(nzp2,nrods)          Change in tw per fluid temp change
 c ltw     tw(nzp2,nrods)           Wall surface temperature
 c lchfr   chfr(nzp2,nrods)         Critical heat flux ratio
 c lfrac   fracp(nrods)           Fraction of total heated perimeter
 c                                facing its adjacent channel
 c lhdz    hdz(nzp2,ncp)          Hydraulic diameter in parallel direction
 c lhdt    hdt(nzp2,ncp)          Hydraulic diameter in transverse direction
 c lhdh    hdh(nc)                Equivalent heated diameter of each channel
 c lqz     qz(nzp2,nrods)           Axial power shape function
 c lqt     qt(nrods)              Transverse power shape function
 c lqr     qr(ndm1)               Fuel pin radial power shape
 c lqppp   qppp(ndm1)             Fuel pin power density
 c lrn     rn(nc)                 Number of fuel rods in each channel
 c lcnd    cnd(ndm1)              Fuel pin conductivities
 c lrcp    rcp(ndm1)              Fuel pin density times specific heat
 c lrrdr   rrdr(ndm1)             r/(delta r) at fuel pin cell centers
 c lvp     vp(nodes)              Volume of half cell outside a fuel pin node
 c lvm     vm(nodes)              Volume of half cell inside a fuel pin node
 c lrad    rad(nodes)             Radii of fuel pin nodes
 c
 c lbotbc  botbc(nc)              Initial bottom boundary conditions
 c ltopbc  topbc(nc)              Initial top boundary conditions
 c ltmpbc  tmpbc(nc)              Initial inlet temperatures
 c lsij    sij(4,nc)              Channel gap widths(m)
 c lph     ph(nzp2,nc)            angle between z-axis and rod axis
 c lth     th(nzp2,nc)            angle between x-axis and rod axis projectio
n
 c
 c lend                           First free core location
 c                                  (some space beyond this pointer is used
 c                                   as scratch pads.)
 c
 c ************************************************************************
       common/sg3/qqx(30),qqy(30),k1111,d1111
        common/sg1/ xxzz1,xxzz2,nxzz3,nxzz4,xxzz5,xxzz6
        common/sg/ vd,xl2,vfeed,xfeed,nonce,ksh,nfeed,nzlvl,nmin,ti,
      1 pdi,t2i,ro2i,h2i,v2i,wd,xr,wr,roavg,xlr,pd,pr,ro2,t2,h2,v2,
      2 dpdz,dro2,dpris,dpsep,wsf,wsb,wfdf,wfdb,t2a,h2a,ro2a,
      3 t2b,h2b,ro2b,psdom,ireci,xl2i,hris,vt,nsep
        common/sgpt1/pprim,pmflx,tpin,tpout,xpp,ypp,xm2,xm3,ymm,xss,
      1 c(15,12),xxxx2
       common a(1)
       common /point/ lncr,lindnt,licc,liwfz,lihtr,licr,lirc,lidwn,ljmtb,
      1  lp,lalp,lrov,lrol,lev,lel,ltv,ltl,ltr, lpn,lalpn,lrovn,
      2   lroln,levn,leln,ltvn,ltln,ltrn, lvvx,lvlx,lvvy,lvly,lvvz,lvlz,
      3   lhv,lhl,lhvs,lhls,ltsat,lvisv,lvisl, ldx,ldy,ldz,larx,lary,
      4   larz,lvol, lcpvx,lcpvy,lcpvz,lcplx,lcply,lcplz, lfvx,lfvy,
      5   lfvz,lflx,lfly,lflz, lajm1,lajm2,lcpa,lcptv,lcptl,lrhs,
      6ldp,lqpp,lqv,lql,lhvfc,lhlnb,lhlfc, ldtrn,ldtw,ltw,lchfr,lfrac,
      7lhdz,lhdt,lhdh,lqz,lqt,lqr,lqppp,lrn,lcnd,lrcp,lrrdr,lvp,lvm,lrad
      8,lbotbc,ltopbc,ltmpbc,lsij,lph,lth,lfg,ldsew,ldsns,laht,lend
       common /ic/ nstep,nitmax,nitno,iitmax,iitot,iic,kred,nm,ntc,
      1   nc,nrods,nz,nr,ncp,nzp,nzp2, iflash,itb,ibb,icpu,iwft,ivec,
      2   nodes,ndm1,ncf,ncc,ng,iht, iss,iqss,itam,ires,idump,ntitle,
```

```
     3   itwmax,jtwmax,itrmax,jtrmax,imchfr,jmchfr,ichf,ifintr,ierr,lerr
     4    ,imixm,imixe,iafm,itfm,igfm,nktb,jhem
      logical lerr
      common /rc/ delt,rdelt,errn,epsn,erri,epsi,dtmin,dtmax,tend,
     1   dtsp,dtlp,rtnsp,rtnlp,gravx,gravy,gravz,rtime,velx,
     2   q,q0,twmax,trmax,amchfr


      common /units/ ntty,ninp,nout,ntzc,nres,ndump
      common /force/ botfac(30),yb(30),topfac(30),yt(30),
     1          tinfac(30),ytemp(30),qfac(30),yq(30),nb,nt,ntemp,nq,
     2  ptfac(30),npt,ypt(30),pgfac(30),npg,ypg(30)
      common /fricmd/ fcon(4,4)
c
c  the following statements initialize commons for multics probe
c
      a(1)=a(1)
      lp=lp
      delt=delt
      botfac(1)=botfac(1)
      fcon(1,1)=fcon(1,1)
c
c
c  set I/O units
c
      ntty=0
      ninp=5
      nout=6
      nres=7
      ndump=8
      ntzc=ninp
      ires=0
      ierr=0
      lerr=.false.
c   set flags for sg restart
       ksh=0
       nonce=0
       ireci=1
c
c  read input data
c
  100 call input
c  print primary variables
       call pout(a(ltr),a(lfg),a(ldsew),a(ldsns),a(laht),
     1 a(ljmtb),4*nktb,nktb,nzp2,nc,a(lidwn))
      if (lerr) go to 200
c
c  initialize arrays
c
      if(ires.eq.0) call init
c
c  perform transient calculation
c
      call trans
```

```
      if (lerr) go to 200
c
c  take restart dump
c
      if(idump.eq.1)call dump
c
c  return for another case
c
      go to 100
c
c  error detected
c
  200 call error
      stop
      end
      block data
      common /fricmd/ fcon(4,4)
      data  fcon  / 64.,    1502.11,    .184, -.2,
     1              0.,     1.,         3., -.1,
     2              0.,     1.,         3., -.1,
     3              180.,   202.5,     1.92, -.145  /
      end
      subroutine error
c
c  prints uncoded error message
c
      common /ic/ nstep,nitmax,nitno,iitmax,iitot,iic,kred,nm,ntc,
     1   nc,nrods,nz,nr,ncp,nzp,nzp2, iflash,itb,ibb,icpu,iwft,ivec,
     2   nodes,ndm1,ncf,ncc,ng,iht, iss,iqss,itam,ires,idump,ntitle,
     3   itwmax,jtwmax,itrmax,jtrmax,imchfr,jmchfr,ichf,ifintr,ierr,lerr
     4     ,imixm,imixe,iafm,itfm,igfm,nktb,jhem
      logical lerr
      common /units/ ntty,ninp,nout,ntzc,nres,ndump
c
      go to (10,20,30,40,50,60,70,80,90,100), ierr
c
   10 write(ntty,1001)
      write(nout,1001)
      return
   20 write(ntty,1002)
      write(nout,1002)
      return
   30 write(ntty,1003)
      write(nout,1003)
      return
   40 write(ntty,1004)
      write(nout,1004)
      return
   50 write(ntty,1005)
      write(nout,1005)
      return
   60 write(ntty,1006)
      write(nout,1006)
      return
```

```
   70 write(ntty,1007)
      write(nout,1007)
      return
   80 write(ntty,1008)
      write(nout,1008)
      return
   90 write(ntty,1009)
      write(nout,1009)
  100 write(ntty,1010)
      write(nout,1010)
      return
 1001  format(41h input error in integer or real parameter)
 1002  format(21h input error in array)
 1003  format(41h pressure problem not diagonally dominant)
 1004  format(41h pressure out of range of state functions)
 1005  format(51h liquid temperature out of range of state functions)
 1006  format(50h vapor temperature out of range of state functions)
 1007  format(23h negative void fraction)
 1008  format(23h void fraction over one)
 1009  format(37h Newton iterations failed to converge)
 1010  format(43h convective time step limit less than dtmin)
      end
      subroutine start
c
c  prepares to restart a previous calculation by reading in
c   commons from an unformatted data file
c
      common a(1)
      common /point/ lncr,lindnt,licc,liwfz,lihtr,licr,lirc,lidwn,ljmtb,
     1  lp,lalp,lrov,lrol,lev,lel,ltv,ltl,ltr, lpn,lalpn,lrovn,
     2   lroln,levn,leln,ltvn,ltln,ltrn, lvvx,lvlx,lvvy,lvly,lvvz,lvlz,
     3   lhv,lhl,lhvs,lhls,ltsat,lvisv,lvisl, ldx,ldy,ldz,larx,lary,
     4   larz,lvol, lcpvx,lcpvy,lcpvz,lcplx,lcply,lcplz, lfvx,lfvy,
     5   lfvz,lflx,lfly,lflz, lajm1,lajm2,lcpa,lcptv,lcptl,lrhs,
     6ldp,lqpp,lqv,lql,lhvfc,lhlnb,lhlfc, ldtrn,ldtw,ltw,lchfr,lfrac,
     7lhdz,lhdt,lhdh,lqz,lqt,lqr,lqppp,lrn,lcnd,lrcp,lrrdr,lvp,lvm,lrad
     8,lbotbc,ltopbc,ltmpbc,lsij,lph,lth,lfg,ldsew,ldsns,laht,lend
      common /ic/ nstep,nitmax,nitno,iitmax,iitot,iic,kred,nm,ntc,
     1   nc,nrods,nz,nr,ncp,nzp,nzp2, iflash,itb,ibb,icpu,iwft,ivec,
     2   nodes,ndm1,ncf,ncc,ng,iht, iss,iqss,itam,ires,idump,ntitle,
     3   itwmax,jtwmax,itrmax,jtrmax,imchfr,jmchfr,ichf,ifintr,ierr,lerr
     4    ,imixm,imixe,iafm,itfm,igfm,nktb,jhem
      logical lerr
      common /rc/ delt,rdelt,errn,epsn,erri,epsi,dtmin,dtmax,tend,
     1   dtsp,dtlp,rtnsp,rtnlp,gravx,gravy,gravz,rtime,velx,
     2   q,q0,twmax,trmax,amchfr


      common /units/ ntty,ninp,nout,ntzc,nres,ndump
      common/force/ botfac(30),yb(30),topfac(30),yt(30),
     1      tinfac(30),ytemp(30),qfac(30),yq(30),nb,nt,ntemp,nq,
     2  ptfac(30),npt,ypt(30),pgfac(30),npg,ypg(30)
      common /fricmd/ fcon(4,4)
      common/sg/vd,x12,vfeed,xfeed,nonce,ksh,nfeed,nzlvl,nmin,ti,
```

```
     1pdi,t2i,ro2i,h2i,v2i,wd,xr,wr,roavg,xlr,pd,pr,ro2,t2,h2,v2,
     2dpdz,dro2,dpris,dpsep,wsf,wsb,wfdf,wfdb,t2a,h2a,ro2a,
     3t2b,h2b,ro2b,psdom,ireci,xl2i,hris,vt,nsep
       common/sgpt1/pprim,pmflx,tpin,tpout,zz1,zz2,zz3,zz4,zz5,zz6,
     1 zz7(15,12),zz8
      common/sg1/pi,pri,nmini,nzli,t2ai,dpdzi,tfeed,wfeed
      dimension d1(1),d2(1),d3(1),d4(1),d5(1),d6(1),d7(1),d8(1),d9(1)
      equivalence (d1(1),lncr),(d2(1),nstep),(d3(1),delt),
     1            (d5(1),botfac(1)),(d6(1),fcon(1,1)),
     2  (d7(1),vd),(d8(1),pprim),(d9(1),pi)
c
c  restore commons
c
      rewind(nres)
      read(nres) (d1(i),i=1,102)
      read(nres) (d2(i),i=1,51)
      read(nres) (d3(i),i=1,23)
      read(nres) (d5(i),i=1,366)
      read(nres) (d6(i),i=1,16)
      read(nres) (d7(i),i=1,46)
      read(nres) (d8(i),i=1,191)
       read(nres)  (d9(i),i=1,8)
      read(nres) (a(i),i=1,lend)
      write(ntty,1001) nres
 1001  format(23h commons read from file,i3)
      return
      end

      subroutine dump
c
c  at the end of a calculation, this subroutine dumps all commons
c   onto an unformatted data file for later use with restart
c
      common a(1)
      common /point/ lncr,lindnt,licc,liwfz,lihtr,licr,lirc,lidwn,ljmtb,
     1  lp,lalp,lrov,lrol,lev,lel,ltv,ltl,ltr, lpn,lalpn,lrovn,
     2   lroln,levn,leln,ltvn,ltln,ltrn, lvvx,lvlx,lvvy,lvly,lvvz,lvlz,
     3   lhv,lhl,lhvs,lhls,ltsat,lvisv,lvisl, ldx,ldy,ldz,larx,lary,
     4   larz,lvol, lcpvx,lcpvy,lcpvz,lcplx,lcply,lcplz, lfvx,lfvy,
     5   lfvz,lflx,lfly,lflz, lajm1,lajm2,lcpa,lcptv,lcptl,lrhs,
     6ldp,lqpp,lqv,lql,lhvfc,lhlnb,lhlfc, ldtrn,ldtw,ltw,lchfr,lfrac,
     7lhdz,lhdt,lhdh,lqz,lqt,lqr,lqppp,lrn,lcnd,lrcp,lrrdr,lvp,lvm,lrad
     8,lbotbc,ltopbc,ltmpbc,lsij,lph,lth,lfg,ldsew,ldsns,laht,lend
      common /ic/ nstep,nitmax,nitno,iitmax,iitot,iic,kred,nm,ntc,
     1   nc,nrods,nz,nr,ncp,nzp,nzp2, iflash,itb,ibb,icpu,iwft,ivec,
     2   nodes,ndm1,ncf,ncc,ng,iht, iss,iqss,itam,ires,idump,ntitle,
     3   itwmax,jtwmax,itrmax,jtrmax,imchfr,jmchfr,ichf,ifintr,ierr,lerr
     4    ,imixm,imixe,iafm,itfm,igfm,nktb,jhem
      logical lerr
      common /rc/ delt,rdelt,errn,epsn,erri,epsi,dtmin,dtmax,tend,
     1   dtsp,dtlp,rtnsp,rtnlp,gravx,gravy,gravz,rtime,velx,
     2   q,q0,twmax,trmax,amchfr
```

```
      common /units/ ntty,ninp,nout,ntzc,nres,ndump
      common/force/ botfac(30),yb(30),topfac(30),yt(30),
     1  tinfac(30),ytemp(30),qfac(30),yq(30),nb,nt,ntemp,nq,
     2  ptfac(30),npt,ypt(30),pgfac(30),npg,ypg(30)
      common /fricmd/ fcon(4,4)
       common/sg/vd,xl2,vfeed,xfeed,nonce,ksh,nfeed,nzlvl,nmin,ti,
     1 pdi,t2i,ro2i,h2i,v2i,wd,xr,wr,roavg,xlr,pd,pr,ro2,t2,h2,v2,
     2 dpdz,dro2,dpris,dpsep,wsf,wsb,wfdf,wfdb,t2a,h2a,ro2a,
     3 t2b,h2b,ro2b,psdom,ireci,xl2i,hris,vt,nsep
       common/sgpt1/pprim,pmflx,tpin,tpout,zz1,zz2,zz3,zz4,zz5,zz6,
     1 zz7(15,12),zz8
       common/sg1/pi,pri,nmini,nzli,t2ai,dpdzi,tfeed,wfeed
      dimension d1(1),d2(1),d3(1),d4(1),d5(1),d6(1),d7(1),d8(1),d9(1)
      equivalence (d1(1),lncr),(d2(1),nstep),(d3(1),delt)
     1             ,(d5(1),botfac(1)),(d6(1),fcon(1,1)),
     1   (d7(1),vd),(d8(1),pprim),(d9(1),pi)
c
c  dump commons
c
      rewind(ndump)
      write(ndump) (d1(i),i=1,102)
      write(ndump) (d2(i),i=1,51)
      write(ndump) (d3(i),i=1,23)
      write(ndump) (d5(i),i=1,366)
      write(ndump)  (d6(i),i=1,16)
       write(ndump)  (d7(i),i=1,46)
       write(ndump)  (d8(i),i=1,191)
       write(ndump)  (d9(i),i=1,8)
      write(ndump) (a(i),i=1,lend)
      write(ntty,1001) ndump
 1001  format(23h commons dumped on file,i3)
      return
      end
      subroutine nips(ih1,ih2,inx,x,nk,ierr,iflag)
c
c  a built-in free format input interpreter for array data
c
      common /units/ ntty,ninp,nout,ntzc,nres,ndump
      dimension x(nk),inx(nk),nmult(10),ind(10)
      integer ined(18),a(80),xx
      data ined/4h    ,4h0   ,4h1   ,4h2   ,4h3   ,4h4   ,
     1  4h5   ,4h6   ,4h7   ,4h8   ,4h9   ,4h.   ,4h-   ,4h+   ,
     2     4he   ,4h(   ,4h)   ,4h$   /
c
c  print echo check
c
      write(nout,1238) ih1,ih2
 1238 format(1h ,2a4)
      read(ninp,1234) a
 1234 format(80a1)
      write(nout,1235) a
 1235 format(1h ,'card image>',80a1,'<')
      npar  = 1
      ixa=1
```

```
      nn = 0
      kounta=1
      if (iflag.eq.1) go to 500
      n=5
      go to 58
    7 if (kounta-80) 9,9,8
    8 kounta = 1
      read(ninp,1234) a
      write(nout,1235) a
    9 xx=a(kounta)
      kounta=kounta+1
      do 100 ipt = 1, 19
      if(xx-ined(ipt)) 100,200,100
  100 continue
      go to 301
  200 k=ipt
c
cc    k is the character number ( range - 1 to 18 )
cc    n is the operation number
cc          operation branch for comments
c
 10   continue
      if (n)11,11,13
11    continue
      if (k-17)7,12,7
   12 n=5
      go to 7
   13 continue
      go to (14,15,16,17,18,19,20,21,22,23,24,25,27,30,33,60,80,250)
     1,k
cc
cc
cc
cc          character branch
cc
   14 continue
      go to (50,48,45,301,7),n
   15 xt=0.0
      go to 40
   16 xt=1.0
      go to 40
   17 xt=2.0
      go to 40
   18 xt=3.0
      go to 40
   19 xt=4.0
      go to 40
   20 xt=5.0
      go to 40
   21 xt=6.0
      go to 40
   22 xt=7.0
      go to 40
   23 xt=8.0
```

```
      go to 40
   24 xt=9.0
      go to 40
cc          operation branch for .
   25 go to (26,301,301,301,26),n
   26 n=2
      go to 7
cc          operation branch for -
   27 go to (301,28,28,301,29),n
   28 isgnb=-1
   31 n=3
      go to 7
   29 isgna=-1
      n=1
      go to 7
cc          operation branch for +
   30 go to (301,31,7,7,32),n
   32 n=1
      go to 7
cc          operation branch for e
   33 if (n-2)301,31,301
cc
cc          operation branch for (
cc
   60 if (npar.gt.10) go to 70
      if (n.eq.5) go to 65
      if (n.ne.1) go to 62
      if (isgna.lt.0) go to 62
      nmult(npar) = xl-1
      xl = 0.
      xr = 10.
   61 ind(npar) = ixa
      npar = npar+1
      n = 5
      go to 7
   62 write(nout,64)
   64 format(1h ,'multiplier incorrect')
      go to 301
   65 nmult(npar) = 0
      go to 61
   70 write(nout,75)
   75 format(1h ,'greater than ten levels of parentheses')
      go to 301
cc
cc          operation branch for )
cc
   80 if (npar.le.1) go to 86
      npar = npar-1
      if (n.eq.5) go to 82
      nn = -1
      go  to 14
   82 nel = ixa-ind(npar)
      if (nel.eq.0) go to 85
      if (nmult(npar).eq.0) go to 85
```

```
      jtemp=nmult(npar)
      do 89 jj = 1,jtemp
      do 84 i = 1,nel
      if (ixa.gt.nk) go to 400
      x(ixa) = x(ind(npar)+i-1)
   84 ixa = ixa+1
   89 continue
   85 nn = 0
      go to 58
   86 write(nout,88)
   88 format(1h ,'Unexpected right parenthesis found')
      go to 301
cc
cc          operation branch for integers
   40 go to (41,42,43,43,44,41),n
   41 xl=xl*10.0+xt
      go to 7
c  the real divide - xt/xr on cdc 6600  is inaccurate in place 15.
c  the scale factor below is to eliminate the problem caused when xl
c  is later stored into an integer in rdvl etc.  truncation problem.
c  not sure that this scale factor is big enough for ibm
   42 xl=xl+(xt/xr)*1.
      xr=xr*10.0
      go to 7
   43 ixe=ixe*10+k-2
      go to 7
   44 n=1
      xl=xt
      go to 7
cc          termination for floating value with exponent
   45 if (isgnb)46,47,47
   46 ixe=-ixe
   47 xl=xl*10.0**ixe
cc          termination for floating value
   50 continue
   48 if (isgna)49,55,55
   49 xl=-xl
      go to 55
cc          load the value
   55 if (ixa.gt.nk) go to 400
      x(ixa) = xl
      ixa=ixa+1
cc          initialize for next value
      if (nn) 82,58,255
   58 xl = 0.0
      xr=10.0
      ixe=0
      isgna=1
      isgnb=1
      n=5
      go to 7
cc
cc          operation branch for termination
cc
```

```
  250 if (n.eq.5) go to 255
      nn = 1
      go to 14
  255 ixa1 = ixa-1
      if (ixa1.ne.nk) go to 400
c 260 write(nout,1239) x
 1239 format(1h ,10e12.4)
      return
  500 n = 0
      go to 592
  507 if (kounta-80) 509,509,508
  508 kounta = 1
      read(ninp,1234) a
      write(nout,1235) a
  509 xx = a(kounta)
      kounta = kounta + 1
      do 520 ipt = 1,18
      if (xx-ined(ipt)) 520,540,520
  520 continue
      go to 301
  540 go to (545,550,550,550,550,550,550,550,550,
     1     550,550,301,555,560,301,570,580,600), ipt
cc
  545 if (n) 590,507,590
cc
  550 ii = ipt-2
      it = it*10+ii
      if (n.ne.0) go to 507
      n = 1
      go to 507
cc
cc     operation branch for -
cc
  555 if (n) 301,556,301
  556 n = -1
      go to 507
cc
cc          operation branch for +
cc
  560 if (n) 301,562,301
  562 n = 1
      go to 507
cc
cc          operation branch for (
cc
  570 if (npar.gt.10) go to 70
      if (n) 301,574,571
  571 nmult(npar) = it-1
      ii = 0
      it = 0
  572 ind(npar) = ixa
      npar = npar+1
      n = 0
      go to 507
```

```
  574 nmult(npar) = 0
      go to 572
cc
cc           operation branch for )
cc
  580 if (npar.le.1) go to 86
      npar = npar-1
      if (n.eq.0) go to 583
      if (ixa.gt.nk) go to 400
      inx(ixa) = it*n
      ixa = ixa+1
  583 nel = ixa-ind(npar)
      if (nel.eq.0) go to 589
      if (nmult(npar).eq.0) go to 589
      jtemp=nmult(npar)
      do 588 ixx = 1,jtemp
      do 584 ixy = 1,nel
      if (ixa.gt.nk) go to 400
      inx(ixa) = inx(ind(npar)+ixy-1)
  584 ixa = ixa+1
  588 continue
  589 go to 592
cc
  590 if (ixa.gt.nk) go to 400
      inx(ixa) = it*n
      ixa = ixa+1
  592 n = 0
      it = 0
      ii = 0
      go to 507
cc
cc           operation branch for termination
cc
  600 if (n.eq.0) go to 604
  602 inx(ixa) = it*n
      ixa = ixa+1
  604 ixa1 = ixa-1
      if (ixa1.ne.nk) go to 400
c     write(nout,1267) inx
 1267 format(1h ,20(1x,i5))
      return
cc
cc           error return
cc
  301 continue
      kounta = kounta - 1
      write(nout,1266) kounta, ih1,ih2
 1266 format(1h ,'  error at column',i4,' trying to read ',2a4)
      ierr = -1
      return
  400 ierr = -1
      write(nout,444) ih1,ih2
  444 format(1h ,'nips error - wrong number of values',
     1           ' while trying to read ',2a4)
```

```
      return
      end
      subroutine input
c
c  reads one full block of input data
c   also sets pointers for variably dimensioned arrays
c
      real*8 bctype(2),fptype(4,4),qpptyp(2),chftyp(3),trnflo(2,2),
     1       mixmod(3,2)
      common a(1)
      common/sgpt1/pprim,pmflx,tpin,tpout,xmout,tauh,tauc,toutn,z1,z2,
     1c(15,12),foul
      common /point/ lncr,lindnt,licc,liwfz,lihtr,licr,lirc,lidwn,ljmtb,
     1  lp,lalp,lrov,lrol,lev,lel,ltv,ltl,ltr, lpn,lalpn,lrovn,
     2   lroln,levn,leln,ltvn,ltln,ltrn, lvvx,lvlx,lvvy,lvly,lvvz,lvlz,
     3   lhv,lhl,lhvs,lhls,ltsat,lvisv,lvisl, ldx,ldy,ldz,larx,lary,
     4   larz,lvol, lcpvx,lcpvy,lcpvz,lcplx,lcply,lcplz, lfvx,lfvy,
     5   lfvz,lflx,lfly,lflz, lajm1,lajm2,lcpa,lcptv,lcptl,lrhs,
     6ldp,lqpp,lqv,lql,lhvfc,lhlnb,lhlfc, ldtrn,ldtw,ltw,lchfr,lfrac,
     7lhdz,lhdt,lhdh,lqz,lqt,lqr,lqppp,lrn,lcnd,lrcp,lrrdr,lvp,lvm,lrad
     8,lbotbc,ltopbc,ltmpbc,lsij,lph,lth,lfg,ldsew,ldsns,laht,lend
      common /ic/ nstep,nitmax,nitno,iitmax,iitot,iic,kred,nm,ntc,
     1   nc,nrods,nz,nr,ncp,nzp,nzp2, iflash,itb,ibb,icpu,iwft,ivec,
     2   nodes,ndm1,ncf,ncc,ng,iht, iss,iqss,itam,ires,idump,ntitle,
     3   itwmax,jtwmax,itrmax,jtrmax,imchfr,jmchfr,ichf,ifintr,ierr,lerr
     4    ,imixm,imixe,iafm,itfm,igfm,nktb,jhem
      common /fricmd/ fcon(4,4)
      logical lerr
      common /rc/ delt,rdelt,errn,epsn,erri,epsi,dtmin,dtmax,tend,
     1   dtsp,dtlp,rtnsp,rtnlp,gravx,gravy,gravz,rtime,velx,
     2   q,q0,twmax,trmax,amchfr
       common/sg/vd,xl2,vfeed,xfeed,nonce,ksh,nfeed,nzlvl,nmin,ti,
     1  pdi,t2i,ro2i,h2i,v2i,wd,xr,wr,roavg,xlr,pd,pr,ro2,t2,h2,v2,
     2  dpdz,dro2,dpris,dpsep,wsf,wsb,wfdf,wfdb,t2a,h2a,ro2a,
     3 t2b,h2b,ro2b,psdom,ireci,xl2i,hris,vt,nsep
      common /units/ ntty,ninp,nout,ntzc,nres,ndump
       common/force/botfac(30),yb(30),topfac(30),yt(30),tinfac(30),
     1             ytemp(30),qfac(30),yq(30),nb,nt,ntemp,nq,
     2  ptfac(30),npt,ypt(30),pgfac(30),npg,ypg(30)
      dimension ia(1)
      dimension gtype(3,3),fitype(2),httype(3,3),chfchk(5,3),
     1          tftype(3,2),tvtype(3,2)
      equivalence (a(1),ia(1))
      namelist/restart/nitmax,iitmax,iflash,itb,ibb,iwft,iht,iss,epsn,
     1 epsi,gravz,q0,itam,ichf,idump,iqss,ivec,imixm,imixe,tauh,tauc,
     2 vd,psdom,nsep,vt,hris,ifintr,foul
      namelist/tffdata/botfac,topfac,tinfac,qfac,yb,yt,ytemp,yq,nb,nt,
     1     ntemp,nq,npt,ptfac,ypt,npg,pgfac,ypg
c
c  data for thermal-hydraulic options
c
      data bctype/8hPressure,8hVelocity/
      data gtype/4hNigm,4hatul,4hin  ,4hSupp,4hress,4hed  ,4hSubc,
     1          4hoole,4hd   /
```

```
      data fitype/4hMIT ,4hLASL/
      data fptype/8hATTENTIO,8hN Please,8h set iht,8h = 1    ,
     1            8hValue of,8h iht=1 i,8hs correc,8htly set ,
     2            8hATTENTIO,8hN Please,8h set iht,8h = 1    ,
     3            8hATTENTIO,8hN Please,8h set iht,8h = 1    /
      data httype/4hSet ,4hiss=,4hlor2,
     1            4hExpl,4hicit,4h-qs-,
     2            4hExpl,4hicit,4h-qs-/
      data chfchk/4h    ,4h    ,4h    ,4h    ,4h    ,4h    ,4h    ,
     1         4h    ,4h    ,4h    ,4hwith,4hout ,4hCHF ,4hchec,4hk   /
      data qpptyp/8hiqss oK!,8hiqss=0!!/
      data chftyp/8hBiasi   ,8hW-3     ,8hCISE-4  /
      data tftype/4hNone,4h    ,4h    ,4hGunt,4her-S,4hhaw /
      data tvtype/4hTran,4hsver,4hse  ,4hMagn,4hitud,4he of/
      data trnflo/8hnot calc,8hulated  ,8hcalculat,8hed      /
      data mixmod/8hNo mixin,8hg model ,8h        ,
     1            8hSet imix,8hm to zer,8ho       /
c
c  test for restart
c
      if(ires.ne.0)go to 100
c
c  read title
c
      read(ninp,1001)ntc
c
c  if ntc=0 stop
c
      if(ntc.eq.0)stop
c
c  if ntc<0 restart
c
      ires=iabs(ntc)
      if(ntc.lt.0)go to 100
      ires=0
      ntitle=20*ntc
      read(ninp,1002) (a(i),i=1,ntitle)
      write(nout,1011)
      write(nout,1003) (a(i),i=1,ntitle)
c
c  read array dimensions
c
      read(ninp,1004)  nc,nr,nrods,nz,ncf,ncc,nktb,pmflx,pprim,jhem
c
c   read thermal-hydraulic data
c
      read(ninp,1004)  itb,ibb,iflash,ifintr,iht,iss,iqss,ichf,
     1              iwft,ivec,itam,imixm,imixe,iafm,itfm,igfm,
     2              gravx,gravy,gravz,velx
       if(iafm.eq.1)  read(ninp,1004)  (fcon(j,1),j=1,4)
       if(itfm.eq.1)  read(ninp,1004)  (fcon(j,4),j=1,4)
       if(igfm.ne.1)  go to 7
      read(ninp,1004)  (fcon(j,2),j=3,4)
      fcon(3,3)=fcon(3,2)
```

```
      fcon(4,3)=fcon(4,2)
 7    continue
c
c   read iteration control data
c
      read(ninp,1004) idump,nitmax,iitmax,epsn,epsi
      if (max0(itb,ibb,iwft,ivec,imixm,imixe,iqss,idump,ifintr,
     1     iafm,itfm,igfm) .gt.1) go to 901
      if (max0(iflash,itam,iss).gt.2) go to 901
      if (ichf.gt.3) go to 901
      if (min0(nc,nrods,nz,nr,iflash,itb,ibb,iwft,ivec,ncf,ncc,iht,
     1    iafm,itfm,igfm,
     t        iss,iqss,ichf,itam,imixe,imixm,ifintr).lt.0) go to 901
      if(iht.eq.0) go to 10
c
c   read nominal power and geometrical quantities
c
      read(ninp,1004)q0,hris,vt,nsep
   10 continue
c
c  set up storage requirements
c
c  integer arrays
c
      lncr=ntitle+1
      lindnt=lncr+nr
      licc=lindnt+nr
      liwfz=licc+4*nc
      lihtr=liwfz+nz+1
      licr = lihtr+ nz*nrods
      lirc = licr + nrods
      lidwn = lirc + 4*nc
      ljmtb = lidwn + nc
      iend = ljmtb + nktb
c  real arrays
c
      nzp=nz+1
      nzp2=nzp+1
      ncp=nc+1
      l1=nzp2*ncp
      l2=nzp2*nc
      l3=nzp*nc
      l4=nz*nc
      l5=nz*ncp
      l6 = nrods*nzp2
      nodes=4*nktb
      l7=nodes*nzp2*nc
      l8=nktb*12
      if (iht.eq.0) nodes=0
      ndm1=(ncf+ncc+2)-1
      if (iht.eq.0) ndm1=0
      nm=8*l1+(ncf+ncc+2)*nz*nrods
      if(iht.eq.0) nm=8*l1
c
```

```
lp=iend
lalp=lp+l1
lrov=lalp+l1
lrol=lrov+l1
lev=lrol+l1
lel=lev+l1
ltv=lel+l1
ltl=ltv+l1
ltr=ltl+l1
lpn=ltr+l7
lalpn=lpn+l1
lrovn=lalpn+l1
lroln=lrovn+l1
levn=lroln+l1
leln=levn+l1
ltvn=leln+l1
ltln=ltvn+l1
ltrn=ltln+l1
lvvx=ltrn+l7
lvlx=lvvx+l1
lvvy=lvlx+l1
lvly=lvvy+l1
lvvz=lvly+l1
lvlz=lvvz+nzp*ncp
lhv=lvlz+nzp*ncp
lhl=lhv+l1
lhvs = lhl +l1
lhls=lhvs+l1
ltsat=lhls+l1
lvisv=ltsat+l1
lvisl=lvisv+l1
ldx=lvisl+l1
ldy=ldx+ncp
ldz=ldy+ncp
larx=ldz+nzp2
lary=larx+l5
larz=lary+l5
lvol=larz+l3
lcpvx=lvol+l4
lcpvy=lcpvx+l4
lcpvz=lcpvy+l4
lcplx=lcpvz+l3
lcply=lcplx+l4
lcplz=lcply+l4
lfvx=lcplz+l3
lfvy=lfvx+l4
lfvz=lfvy+l4
lflx=lfvz+l3
lfly=lflx+l4
lflz=lfly+l4
lajm1=lflz+l3
lajm2=lajm1+3*l4
lcpa=lajm2+4*l4
lcptv=lcpa+6*l4
```

```
      lcptl=lcptv+6*14
      lrhs=lcptl+6*14
      ldp=lrhs+4*14
      lqpp=ldp+12
      lqv=lqpp+16
      lql=lqv+16
      lhvfc=lql+16
      lhlnb=lhvfc+16
      lhlfc=lhlnb+16
      ldtrn=lhlfc+16
      ldtw=ldtrn+nktb*12
      ltw=ldtw+16
      lchfr=ltw+16
      lfrac =lchfr+16
      lhdz=lfrac+nrods
      lhdt=lhdz+11
      lhdh=lhdt+11
      lqz=lhdh+nc
      lqt=lqz+16
      lqr=lqt+nrods
      lqppp=lqr+ndm1
      lrn=lqppp+ndm1
      lcnd=lrn+nrods
      lrcp=lcnd+ndm1
      lrrdr=lrcp+ndm1
      lvp=lrrdr+ndm1
      lvm=lvp+nodes
      lrad=lvm+nodes
      lbotbc=lrad+nodes
      ltopbc=lbotbc+nc
      ltmpbc=ltopbc+nc
      lsij= ltmpbc+nc
      lph = lsij +4*nc
      lth = lph + 11
      lfg=lth+11
      ldsew=lfg+nktb
      ldsns=ldsew+12*nktb
      laht=ldsns+12*nktb
      lend=laht+nktb*12
c--space in array  a  is used beyond lend, as scratch space.
c  see calls to inner and rtemp.
c
c  clear all arrays
c
      i1=lncr
      i2=iend-1
      do 15 i=i1,i2
   15 ia(i)=0
      call clear (0.0,a(lp),lend-lp)
        do 17 i=1,30
        botfac(i) =1.0
        topfac(i) =1.0
        tinfac(i) =1.0
        qfac(i)   =1.0
```

```
      yb(i) =0.
      yt(i) =0.
      ytemp(i) =0.
      yq(i) =0.
 17   continue
c
c  read arrays
c
c   read geometrical data
c
      write(nout,3429)
3429  format(/,/,/,/,1x,'Input Parameters Check List',/)
c
c  primary side geometrical data
      call nips(4hidow,4hn    ,ia(lidwn),rdum,nc,ierr,1)
      call nips(4hjmtb,4h     ,ia(ljmtb),rdum,nktb,ierr,1)
      call nips(4hrad ,4h     ,idum,a(lrad),4,ierr,0)
      call nips(4hfg  ,4h     ,idum,a(lfg),nktb,ierr,0)
      call nips(4hdsew,4h     ,idum,a(ldsew),18,ierr,0)
      call nips(4hdsns,4h     ,idum,a(ldsns),18,ierr,0)
      call nips(4haht ,4h     ,idum,a(laht),18,ierr,0)
c  secondary side geometrical data
      call nips(4h ncr,4h     ,ia(lncr),rdum,nr,ierr,1)
      call nips(4h ind,4hent  ,ia(lindnt),rdum,nr,ierr,1)
      call nips(4harx ,4h     ,idum,a(larx),14,ierr,0)
      call nips(4hary ,4h     ,idum,a(lary),14,ierr,0)
      call nips(4harz ,4h     ,idum,a(larz),13,ierr,0)
      call nips(4hvol ,4h     ,idum,a(lvol),14,ierr,0)
      call nips(4hdx  ,4h     ,idum,a(ldx),nc,ierr,0)
      call nips(4hdy  ,4h     ,idum,a(ldy),nc,ierr,0)
      call nips(4hdz  ,4h     ,idum,a(ldz),nzp2,ierr,0)
      call nips(4hhdz ,4h     ,idum,a(lhdz),12,ierr,0)
      call nips(4hhdt ,4h     ,idum,a(lhdt),12,ierr,0)
      call nips(4hsij ,4h     ,idum,a(lsij),4*nc,ierr,0)
c
c   read axial friction model
c
      call nips(4hind ,4hfric,ia(liwfz),rdum,nzp,ierr,1)
      call nips(4h phi,4h     ,idum,a(lph),12,ierr,0)
      call nips(4hthed,4ha    ,idum,a(lth),12,ierr,0)
c
c   read  initial conditions
c
      call nips(4hp   ,4h     ,idum,a(lp),12,ierr,0)
      call nips(4halp ,4h     ,idum,a(lalp),12,ierr,0)
      call nips(4htemp,4h     ,idum,a(ltv),12,ierr,0)
      call nips(4hvz  ,4h     ,idum,a(lvvz),13,ierr,0)
      if(iht.eq.0) go to 20
      call nips(4hind ,4hrods,ia(licr),rdum,nrods,ierr,1)
      call nips(4hhdh ,4h     ,idum,a(lhdh),nc,ierr,0)
      call nips(4htw  ,4h     ,idum,a(ltw),16,ierr,0)
      call nips(4h    ,4h  qz,idum,a(lqz),16,ierr,0)
      call nips(4h    ,4h  qt,idum,a(lqt),nrods,ierr,0)
      call nips(4h    ,4h  qr,idum,a(lqr),ndm1,ierr,0)
```

```
      call nips(4h    ,4h  rn,idum,a(lrn),nrods,ierr,0)
      call nips(4hfrac,4h Ph ,idum,a(lfrac),nrods,ierr,0)
c  primary and wall initial temperatures
      call nips(4h tr ,4h    ,idum,a(ltr),17,ierr,0)
   20 continue
c   write problem data
c
      write(nout,1005) nc,nr,ia(ljmtb),nz,ia(ljmtb+nktb-1),nktb,jhem
      write(nout,1006) bctype(itb+1),bctype(ibb+1),(gtype(i,iflash+1)
     1       ,i=1,3),fitype(ifintr+1),(fptype(i,iht+1),i=1,4),(httype(i,
     2       iss+1),i=1,3),(chfchk(i,iss+1),i=1,5),qpptyp(iqss+1),
     3       chftyp(ichf),(tftype(i,iwft+1),i=1,3),(tvtype(i,ivec+1),
     4       i=1,3),(trnflo(i,itam+1),i=1,2),(mixmod(i,imixm+1),i=1,3),
     5       gravx,gravy,gravz,velx
      write(nout,1066) fcon(3,1),fcon(4,1),fcon(3,4),fcon(4,4),
     1                 fcon(3,2),fcon(4,2)
      write(nout,1007) idump,nitmax,iitmax,epsn,epsi
      if(iht.eq.0) go to 11
      write(nout,1008) (a(lrad+i),i=1,3),tpin,pprim,pmflx,tpout
 11   continue
c
c  read in transient forcing functions
c
      read(ninp,1004) nb,nt,ntemp,nq
      if(min0(nb,nt,ntemp,nq).lt.0) go to 902
       if(nb.gt.0) read(ninp,1004) (botfac(i),yb(i),i=1,nb)
       if(nt.gt.0) read(ninp,1004)  (topfac(i),yt(i),i=1,nt)
       if(ntemp.gt.0) read(ninp,1004)  (tinfac(i),ytemp(i),i=1,ntemp)
       if(nq.gt.0) read(ninp,1004)  (qfac(i),yq(i),i=1,nq)
c
c
c  test for nips-detected error
c
      if(ierr.ne.0) go to 902
      return
c
c  restart option
c
  100 continue
      itemp=ires
      if(ires.ge.2)call start
      if(ntzc.eq.ntty)write(ntty,1009)
      read(ntzc,restart)
      ires=itemp
      icpu=0
      if(ntzc.eq.ntty) write(ntty,1024)
      read(ntzc,tffdata)
      if(ires.ne.3) go to 108
      delt= 0.
      rtime= 0.
      nstep= 0
c
c  set boundary conditions
c
```

```
      call initfd(a(lp),a(lalp),a(lrov),a(lrol),a(lev),a(lel),
     1     a(ltv),a(ltl),a(lvvx),a(lvlx),a(lvvy),a(lvly),a(lvvz),
     2     a(lvlz),a(lhv),a(lhl),a(lpn),a(ltsat),a(lvisv),
     3     a(lvisl),a(lbotbc),a(ltopbc),a(ltmpbc),a(lhls),a(lhvs),nzp,
     4     nzp2,2)
 108  continue
c
c  if q0<0, then q0 is set to current q
c
      if(q0.lt.0.0) q0=q
c
c  print new problem data
c
      write(nout,1010)
      write(nout,1003) (a(i),i=1,ntitle)
      write(nout,1005)nc,nr,ia(ljmtb),nz,ia(ljmtb+nktb-1),nktb,jhem
      write(nout,1006) bctype(itb+1),bctype(ibb+1),(gtype(i,iflash+1)
     1      ,i=1,3),fitype(ifintr+1),(fptype(i,iht+1),i=1,4),(httype(i,
     2      iss+1),i=1,3),(chfchk(i,iss+1),i=1,5),qpptyp(iqss+1),
     3      chftyp(ichf),(tftype(i,iwft+1),i=1,3),(tvtype(i,ivec+1),
     4      i=1,3),(trnflo(i,itam+1),i=1,2),(mixmod(i,imixm+1),i=1,3),
     5      gravx,gravy,gravz,velx
      write(nout,1066) fcon(3,1),fcon(4,1),fcon(3,4),fcon(4,4),
     1                 fcon(3,2),fcon(4,2)
      write(nout,1007) idump,nitmax,iitmax,epsn,epsi
      if(iht.eq.0) go to 110
      write(nout,1008) (a(lrad+i),i=1,3),tpin,pprim,pmflx,tpout
  110 continue
      if(ires.ge.2)
     a call edit(a(laht),a(lpn),a(lalpn),a(lrovn),a(lroln),a(levn),
     1 a(leln),a(lidwn),
     1a(ltvn),a(ltln),a(lvvx),a(lvlx),a(lvvy),a(lvly),a(lvvz),a(lvlz),
     2a(ltsat),a(lhv),a(lhl),a(ltrn),a(lihtr),a(lqpp),a(ldz),a(larz),
     3a(lrn),a(lfrac),a(lchfr),nzp2,nzp,nodes,nz,nktb)
      return
c
c  input data error
c
  901 ierr = 1
      go to 999
  902 ierr = 2
  999 lerr = .true.
      return
c
 1001 format(v)
 1002 format(20a4)
 1003 format((25x,20a4/)/)
 1004 format(v)
 1005 format(1h0,/,13x,16hArray Dimensions,//,
     1       35h Number of channels                =,i4,/,
     2       35h Number of rows                    =,i4,/,
     3       35h First level in U-bent region      =,i4,/,
     4       35h Number of active axial nodes      =,i4,/,
     5       35h Last level in U-bent region       =,i4,/,
```

```
     6        35h Number of tube banks              =,i4,/,
     7        35h Homogeneous Model Begins At iz    =,i4)
 1006 format(1h0,//,10x,33hThermal-Hydraulic Options In Use ,//,
     1    1x,a8,32h boundary condition at SG top    ,/,
     2    1x,a8,33h boundary condition at SG bottom ,/,
     3    1x,3a4,13hboiling model,/,
     4    1x,a4,36h interfacial momentum exchange model,/,
     5    1x,4a8,/,
     6    1x,3a4,27h heat transfer calculation ,5a4,/,
     7    1x,a8,/,
     8    1x,a8,31h critical heat flux correlation,/,
     9    29h Transverse friction model - ,3a4,/,
     a    1x,3a4,50h velocity used in transverse momentum calculations,/,
     b    20h Transverse flow is ,2a8,/,
     b    1x,3a8,/,
     c    34h Gravitational constants (x,y,z)= ,f10.5,f10.5,f10.5,/,
     d    34h Transverse friction multiplier = ,e12.5)

 1066 format(1h0,//,10x,14hFriction Model,//,
     1        18h Axial        f = ,f6.3,5h*Re**,f6.3,/,
     2        18h Transverse   f = ,f6.3,5h*Re**,f6.3,/,
     3        18h Grid spacer  K = ,f6.3,5h*Re**,f6.3)
 1007 format(1h0,//,10x,28hIteration Control Parameters,//,
     1        35h Dump indicator (0/1)(no/yes)      =,i4,/,
     1        35h Max number of Newton iterations   =,i4,/,
     2        35h Max number of inner iterations    =,i4,/
     a        35h Convergence crit. for newton iter=,e12.5,/
     1        35h Convergence crit. for inner iter =,e12.5)
 1008 format(1h0,//,24h     Primary Model Data //,
     1        35h Tube inner radius                 =,e12.5,/,
     2        35h Tube intermediate radius          =,e12.5,/,
     3        35h Tube outer radius                 =,e12.5,/,
     4        35h Primary inlet temperature         =,f7.2,/,
     5        35h primary system pressure (Pa)      =,e12.5,/,
     6        35h Primary mass flux (Kg/m2/s)       =,e12.5,/,
     7        35h Primary outlet temperature        =,f7.2)
 1014 format(1h0,//,30h    Constants for MATPRO Model,//,
     5        35h Fraction of theor. density (fuel)=,f8.5,/
     6        35h Fraction PuO2                     =,f8.5,/
     8        35h Fuel contact pressure             =,e12.5,/
     9        35h Coefficient of fuel pressure      =,e12.5,/
     a        35h Exponent of fuel pressure         =,f8.5,/
     b        35h Gap roughness                     =,e12.5,/
     c        35h Gap gas pressure                  =,e12.5,/
     d        35h Helium fraction                   =,f8.5,/
     e        35h Argon fraction                    =,f8.5,/
     f        35h Krypton fraction                  =,f8.5,/
     g        35h Xenon fraction                    =,f8.5,/
     h        35h Burnup                            =,e12.5)
 1009 format(26h please enter restart data)
 1010 format(1h1)
 1011 format(1h1,52x,31h THERMIT STEAM GENERATOR 6/6/83,//)
c
 1024  format(' please enter transient forcing function data ')
```

```
      end
      subroutine init
c
c  calls individual initialization subroutines and initial state
c   printout
c
      common a(1)
      common /point/ lncr,lindnt,licc,liwfz,lihtr,licr,lirc,lidwn,ljmtb,
     1  lp,lalp,lrov,lrol,lev,lel,ltv,ltl,ltr, lpn,lalpn,lrovn,
     2   lroln,levn,leln,ltvn,ltln,ltrn, lvvx,lvlx,lvvy,lvly,lvvz,lvlz,
     3   lhv,lhl,lhvs,lhls,ltsat,lvisv,lvisl, ldx,ldy,ldz,larx,lary,
     4   larz,lvol, lcpvx,lcpvy,lcpvz,lcplx,lcply,lcplz, lfvx,lfvy,
     5   lfvz,lflx,lfly,lflz, lajm1,lajm2,lcpa,lcptv,lcptl,lrhs,
     6ldp,lqpp,lqv,lql,lhvfc,lhlnb,lhlfc, ldtrn,ldtw,ltw,lchfr,lfrac,
     7lhdz,lhdt,lhdh,lqz,lqt,lqr,lqppp,lrn,lcnd,lrcp,lrrdr,lvp,lvm,lrad
     8,lbotbc,ltopbc,ltmpbc,lsij,lph,lth,lfg,ldsew,ldsns,laht,lend
      common /ic/ nstep,nitmax,nitno,iitmax,iitot,iic,kred,nm,ntc,
     1   nc,nrods,nz,nr,ncp,nzp,nzp2, iflash,itb,ibb,icpu,iwft,ivec,
     2   nodes,ndm1,ncf,ncc,ng,iht, iss,iqss,itam,ires,idump,ntitle,
     3   itwmax,jtwmax,itrmax,jtrmax,imchfr,jmchfr,ichf,ifintr,ierr,lerr
     4    ,imixm,imixe,iafm,itfm,igfm,nktb,jhem
      logical lerr
      common /rc/ delt,rdelt,errn,epsn,erri,epsi,dtmin,dtmax,tend,
     1   dtsp,dtlp,rtnsp,rtnlp,gravx,gravy,gravz,rtime,velx,
     2   q,q0,twmax,trmax,amchfr
      dimension ia(1)
       equivalence (a(1),ia(1))
c
c  print map of channels
c
      if(nc.gt.1) call mapper(a(lncr),a(lindnt),nr,nc)
c
c  determine channel coupling array
c
      call seticc(a(licc),a(lncr),a(lindnt),nr,nc)
c
c  determine rod and channel coupling
c
      do 6 i=1,nrods
      ir=ia(licr+(i-1))
      do 5 j=1,4
      if(ia(lirc+4*(ir-1)+(j-1)).ne.0) go to 5
      ia(lirc+4*(ir-1)+(j-1)) = i
      go to 6
 5     continue
 6     continue
c
c  initialize fluid dynamics arrays
c
      call initfd(a(lp),a(lalp),a(lrov),a(lrol),a(lev),a(lel),
     1 a(ltv),a(ltl),a(lvvx),a(lvlx),a(lvvy),a(lvly),a(lvvz),
     2 a(lvlz),a(lhv),a(lhl),a(lpn),a(ltsat),a(lvisv),a(lvisl),
     3 a(lbotbc),a(ltopbc),a(ltmpbc),a(lhls),a(lhvs),nzp,nzp2,1)
c
```

```
c  initialize heat source distribution arrays
c
      if(iht.eq.0) go to 10
      call initrc(a(lrad),a(lrrdr),a(lvm),a(lvp),a(lcnd),a(lrcp),
     1 a(lqz),a(lqt),a(lqr),a(lrn),a(lfrac),a(ldz),a(ltw),a(ltr),
     2  a(ltrn),a(lql),a(laht),nktb,4*nktb,nzp2)
   10 continue
      rtime=0.0
      rtnsp=0.
      rtnlp=0.
      nstep=0
      nitno=0
      iitot=0
      delt=0.0
      kred=0
      icpu=0
      q=q0
      twmax =0.0
      trmax =0.0
      amchfr=0.0
      itwmax=0
      itrmax=0
      jtwmax=0
      jtrmax=0
      imchfr=0
      jmchfr=0
c
c  print initial conditions
c
      call edit(a(laht),a(lpn),a(lalpn),a(lrovn),a(lroln),a(levn),
     1 a(leln),a(lidwn),
     1 a(ltvn),a(ltln),a(lvvx),a(lvlx),a(lvvy),a(lvly),a(lvvz),a(lvlz),
     2 a(ltsat),a(lhv),a(lhl),a(ltrn),a(lihtr),a(lqpp),a(ldz),a(larz),
     3 a(lrn),a(lfrac),a(lchfr),nzp2,nzp,nodes,nz,nktb)
      return
      end
      subroutine clear(x,a,n)
      dimension a(1)
      do 10 i=1,n
        a(i)=x
   10   continue
      return
      end
      subroutine mapper(ncr,indent,nr,nc)
c
c       function: prints a map of the channel overlay
c
      common /units/ ntty,ninp,nout,ntzc,nres,ndump
      dimension ncr(1),indent(1),idigit(10),iform(6)
      data idigit/4h1   ,4h2   ,4h3   ,4h4   ,4h5   ,
     1            4h6   ,4h7   ,4h8   ,4h9   ,4h0   /
      data  iform/4h(1h ,4h,   ,2*4h    ,4hx,20,4hi4/)/
c
c
```

```
   10 format(1h1/56x,20h< channel  overlay >//)
   20 format(1h )
      write(nout,10)
c
      indsml = indent(1)
      isize = indent(1)+ncr(1)
      do 2 i=1,nr
        if(indent(i).lt.indsml) indsml = indent(i)
        if(indent(i)+ncr(i).gt.isize) isize = indent(i)+ncr(i)
    2   continue
      isize = isize-indsml
      ispace = (130-4*isize)/2
c
      icf = nc-ncr(nr)+1
      icl = nc
c
      do 4 i=1,nr
        ir = nr+1-i
        if(i.eq.1) go to 3
        icf = icf-ncr(ir)
        icl = icl-ncr(ir+1)
    3   continue
c
c  construct format(1h ,nnx,20i4/), where nn=10*n1+n2
c
        indmod = indent(ir)-indsml
        nn = 4*indmod+ispace
        n1 = nn/10
        iform(3) = idigit(10)
        if(n1.ne.0) iform(3) = idigit(n1)
        n2 = nn-10*n1
        iform(4) = idigit(10)
        if(n2.ne.0) iform(4) = idigit(n2)
c
        write(nout,iform) (j,j=icf,icl)
        write(nout,20)
    4   continue
c
      return
      end
      subroutine seticc(icc,ncr,indent,nr,nc)
      dimension icc(4,1),ncr(1),indent(1)
c
c  sets an array which, for each channel, determines its (up to four)
c    neighbors
c
c     arguments:
c           icc = Indicator for Channel Coupling
c           ncr = Number of Channels per Row
c        indent = INDENTation of each row
c            nr = Number of Rows
c            nc = Number of Channels
c
c  the convention for neighbor numbering via  icc  is:
```

```
c
c             4
c           2 x 3
c             1
c
      do 10 ic=1,nc
        do 10 kc=1,4
   10     icc(kc,ic) = nc+1
c
      irow = 1
      irp = 1
      do 100 ic=1,nc
        if(irp.ne.1)          icc(2,ic) = ic-1
        if(irp.ne.ncr(irow))  icc(3,ic) = ic+1
        if(irow.eq.1) go to 20
        ix = indent(irow)+irp-indent(irow-1)
        if(ix.ge.1.and.ix.le.ncr(irow-1)) icc(1,ic) =
     1    ic-ncr(irow-1)-indent(irow-1)+indent(irow)
   20   continue
        if(irow.eq.nr) go to 30
        ix = indent(irow)+irp-indent(irow+1)
        if(ix.ge.1.and.ix.le.ncr(irow+1)) icc(4,ic) =
     1    ic+ncr( irow )+indent(irow)-indent(irow+1)
   30   continue
        irp = irp+1
        if(irp.le.ncr(irow)) go to 100
        irow = irow+1
        irp = 1
  100   continue
      return
      end
      subroutine initfd(p,alp,rov,rol,ev,el,tv,tl,vvx,vlx,vvy,
     1 vly,vvz,vlz,hv,hl,pn,tsat,visv,visl,botbc,topbc,tmpbc,
     2 hls,hvs,m2,m1,ipart)
c
c  initialize fluid dynamics arrays
c
      common /ic/ nstep,nitmax,nitno,iitmax,iitot,iic,kred,nm,ntc,
     1   nc,nrods,nz,nr,ncp,nzp,nzp2, iflash,itb,ibb,icpu,iwft,ivec,
     2   nodes,ndm1,ncf,ncc,ng,iht, iss,iqss,itam,ires,idump,ntitle,
     3   itwmax,jtwmax,itrmax,jtrmax,imchfr,jmchfr,ichf,ifintr,ierr,lerr
     4    ,imixm,imixe,iafm,itfm,igfm,nktb,jhem
      logical lerr
      dimension p(m1,1),alp(m1,1),rov(m1,1),rol(m1,1),
     1 ev(m1,1),el(m1,1),tv(m1,1),tl(m1,1),vvx(m1,1),
     2 vlx(m1,1),vvy(m1,1),vly(m1,1),vvz(m2,1),vlz(m2,1),
     3 hv(m1,1),hl(m1,1),pn(m1,1),tsat(m1,1),visv(m1,1),visl(m1,1)
     4 ,botbc(1),topbc(1),tmpbc(1),hls(m1,1),hvs(m1,1)
      if(ipart.eq.2) go to 215
c
c  set old variables
c
      do 100 j=1,nzp2
        do 50 i=1,nc
```

```
          tl(j,i)=tv(j,i)
          call state(p(j,i),tv(j,i),tl(j,i),rov(j,i),rol(j,i),ev(j,i),
     1   el(j,i),tsat(j,i),hvs(j,i),hls(j,i),d1,d2,d3,d4,d5,d6,d7,d8,
     2   d9,0,ierr)
          hv(j,i)=ev(j,i)+p(j,i)/rov(j,i)
          hl(j,i)=el(j,i)+p(j,i)/rol(j,i)
          visv(j,i)=visvp(tv(j,i))
          visl(j,i)=vislq(tl(j,i))
   50     continue
        p(j,ncp)=0.0
        alp(j,ncp)=0.0
        tv(j,ncp)=0.0
        tl(j,ncp)=0.0
        rov(j,ncp)=0.0
        rol(j,ncp)=0.0
        ev(j,ncp)=0.0
        el(j,ncp)=0.0
        tsat(j,ncp)=0.0
        hv(j,ncp)=0.0
        hl(j,ncp)=0.0
        visv(j,ncp)=0.0
        visl(j,ncp)=0.0
  100   continue
c
c  set new variables
c
      call move(p,pn,8*nzp2*ncp)
c
c  set velocities
c
      do 200 i=1,ncp
        do 200 j=1,nzp
  200     vlz(j,i)=vvz(j,i)
      do 210 i=1,ncp
        do 210 j=1,nzp2
          vvx(j,i)=0.0
          vlx(j,i)=0.0
          vvy(j,i)=0.0
          vly(j,i)=0.0
  210     continue
  215    continue
c
c   save initial boundary conditions
c
       do 260 i=1,nc
       if(ibb.eq.0) go to 230
       botbc(i) = vlz(1,i)
       go to 240
 230    botbc(i) = pn(1,i)
 240    tmpbc(i) = tl(1,i)
       if(itb.eq.0) go to 250
       topbc(i) = vlz(nzp,i)
       go to 260
 250   topbc(i) = pn(nzp2,i)
```

```
 260   continue
      return
      end
      subroutine initrc(rad,rrdr,vm,vp,cnd,rcp,qz,qt,qr,rn,fracp,
     1 dz,tw,tr,trn,ql,aht,m0,m1,m2)
c
c  initialize rod conduction arrays
c   and make initializing call to gap conductance subroutine
c
      common /ic/ nstep,nitmax,nitno,iitmax,iitot,iic,kred,nm,ntc,
     1   nc,nrods,nz,nr,ncp,nzp,nzp2, iflash,itb,ibb,icpu,iwft,ivec,
     2   nodes,ndm1,ncf,ncc,ng,iht, iss,iqss,itam,ires,idump,ntitle,
     3   itwmax,jtwmax,itrmax,jtrmax,imchfr,jmchfr,ichf,ifintr,ierr,lerr
     4    ,imixm,imixe,iafm,itfm,igfm,nktb,jhem
      logical lerr
      common /rc/ delt,rdelt,errn,epsn,erri,epsi,dtmin,dtmax,tend,
     1   dtsp,dtlp,rtnsp,rtnlp,gravx,gravy,gravz,rtime,velx,
     2   q,q0,twmax,trmax,amchfr


      dimension rad(1),rrdr(1),vm(1),vp(1),cnd(1),rcp(1),qz(m2,1),qt(1),
     1 qr(1),rn(1),fracp(1),dz(1),tw(m2,1),tr(m1,m2,1),trn(m1,m2,1)
     2 , ql(m2,1),aht(m0,nzp2,1)
      data pi/3.1415926/
c
c
c  heat source distribution arrays
c
       do 225 i=1,nrods
      sum=0.0
      do 210 j=1,nzp2
  210   sum=sum+qz(j,i)*dz(j)
      do 220 j=1,nzp2
  220   qz(j,i)=qz(j,i)/sum
  225  continue
      sum=0.0
      do 230 i=1,nrods
  230   sum=sum+qt(i)*rn(i)*fracp(i)
      do 240 i=1,nrods
  240   qt(i)=qt(i)/sum
       if(iqss.ne.0) go to 410
       do 400 i=1,nrods
       do 400 j=1,nzp2
       ql(j,i)=qz(j,i)*qt(i)*q0*dz(j)*rn(i)
        area=0.
        do 340 ktb=1,nktb
340     area=area+aht(ktb,j,i)
        if(area.ne.0.)ql(j,i)=ql(j,i)/area
 400   continue
 410   continue
      return
      end
      subroutine trans
c
```

```
c  governs transient calculation
c
       common/sgpt1/pprim,pmflx,tpin,tpout,xpp,ypp,xm2,xm3,ymm,xss,
     1 c(15,12),xyz1
      common/sg3/qqx(30),qqy(30),k1111,d1111
       common/sg1/xxzz1,xxzz2,xxzz3,nxzz4,xxzz5,xxzz6,xxzz7,xxzz8
       common/sg/ vd,xl2,vfeed,xfeed,nonce,ksh,nfeed,nzlvl,nmin,ti,
     1 pdi,t2i,ro2i,h2i,v2i,wd,xr,wr,roavg,xlr,pd,pr,ro2,t2,h2,v2,
     2 dpdz,dro2,dpris,dpsep,wsf,wsb,wfdf,wfdb,t2a,h2a,ro2a,
     3 t2b,h2b,ro2b,psdom,ireci,xl2i,hris,vt,nsep
      common a(1)
      common /point/ lncr,lindnt,licc,liwfz,lihtr,licr,lirc,lidwn,ljmtb,
     1  lp,lalp,lrov,lrol,lev,lel,ltv,ltl,ltr, lpn,lalpn,lrovn,
     2   lroln,levn,leln,ltvn,ltln,ltrn, lvvx,lvlx,lvvy,lvly,lvvz,lvlz,
     3   lhv,lhl,lhvs,lhls,ltsat,lvisv,lvisl, ldx,ldy,ldz,larx,lary,
     4   larz,lvol, lcpvx,lcpvy,lcpvz,lcplx,lcply,lcplz, lfvx,lfvy,
     5   lfvz,lflx,lfly,lflz, lajm1,lajm2,lcpa,lcptv,lcptl,lrhs,
     6ldp,lqpp,lqv,lql,lhvfc,lhlnb,lhlfc, ldtrn,ldtw,ltw,lchfr,lfrac,
     7lhdz,lhdt,lhdh,lqz,lqt,lqr,lqppp,lrn,lcnd,lrcp,lrrdr,lvp,lvm,lrad
     8,lbotbc,ltopbc,ltmpbc,lsij,lph,lth,lfg,ldsew,ldsns,laht,lend
      common /ic/ nstep,nitmax,nitno,iitmax,iitot,iic,kred,nm,ntc,
     1   nc,nrods,nz,nr,ncp,nzp,nzp2, iflash,itb,ibb,icpu,iwft,ivec,
     2   nodes,ndm1,ncf,ncc,ng,iht, iss,iqss,itam,ires,idump,ntitle,
     3   itwmax,jtwmax,itrmax,jtrmax,imchfr,jmchfr,ichf,ifintr,ierr,lerr
     4    ,imixm,imixe,iafm,itfm,igfm,nktb,jhem
      logical lerr
      common /rc/ delt,rdelt,errn,epsn,erri,epsi,dtmin,dtmax,tend,
     1   dtsp,dtlp,rtnsp,rtnlp,gravx,gravy,gravz,rtime,velx,
     2   q,q0,twmax,trmax,amchfr
      common /units/ ntty,ninp,nout,ntzc,nres,ndump
c
c  begin one time zone, obtain initial cpu time
c
      call timing(icpu)
c
c   obtain time step control parameters
c
   10 continue
      if(ntzc.eq.ntty)write(ntty,1001)
      read(ntzc,1000)tend,dtmin,dtmax,dtsp,dtlp,clm,iredmx
      rtnsp=rtime+dtsp
      rtnlp=rtime+dtlp
      if(tend.gt.0.0)go to 20
      ires=0
      if(tend.eq.0.0) return
      if(ntzc.eq.ninp) go to 15
      ires=1
      return
 15   ntzc=ntty
      go to 10
c
c  begin one time step, determine time step size
c
   20 continue
```

```
      call timstp(a(lvvz),a(ldz),clm,delt,dtmin,dtmax,rtime,tend,
     1 ini,ird,nzp)
      ired = 0
   21 rdelt=1./delt
      if(ini.eq.1)go to 10
c
      call sgbcp  (a(lidwn),a(lpn),a(lrovn),a(lroln),a(levn),a(leln),
     1  a(lalpn),a(ltvn),a(ltln),a(ltsat),a(lvvz),a(lvlz),a(larz),
     2          a(lvol),a(ldz),delt,nzp,nzp2,nz,nc,rtime,rtnlp,qb)
        lerr=.false.
c  determine power level from the primary
c
       iqss=1
       if(iht.eq.0.or.iqss.eq.0) go to 25
       q=0.
c   obtain heat transfer coefficients and explicit
c   heat flux to secondary side for each tube bank
c
      do 337 ktb=1,nktb
      call hconv(a(lihtr),a(lqv),a(lql),a(lhvfc),a(lhlnb),a(lhlfc),
     1 a(ltw),a(ltl),a(ltv),a(lp),a(lalp),a(lrov),a(lrol),
     2 a(lhv),a(lhl),a(lhvs),a(lhls),a(lqpp),a(ldz),a(lfrac),a(ltsat),
     3a(lvvz),a(lvlz),a(ltr),a(licr),a(lhdh),a(lchfr),a(ldtrn),
     4ktb,nz,nzp2,nzp,nodes,4*nktb,nktb)
c
c
c  solve for primary temperatures
c
c  estimate a fouling parameter
       if(xyz1.eq.0.) xyz1=1.
c
       call ptemp(a(lrcp),a(lrad),a(ltr),a(ltrn),a(ldtrn),
     1 a(ltw),a(ldtw),a(lhvfc),a(lhlnb),a(lhlfc),a(ltvn),
     2 a(ltln),a(ltsat),a(ldz),delt,a(lfg),a(ldsew),a(ldsns),
     3 a(laht),a(ljmtb),nktb,nz,nzp2,4*nktb,ktb,nc,
     4 a(lql),a(lqv),rtime,xyz1)
       rtime=rtime+delt
       call pout1(a(ltr),4*nktb,nktb,nzp2,nc,a(lidwn),a(ldtrn),
     1 a(lhlnb),a(lhlfc),a(lhvfc),ktb,a(ljmtb),a(ldtw),a(laht),a(lrad),
     2 a(ltrn),q,xyz1)
       rtime=rtime-delt
337     continue
c
c  adjust pmflx to yield prescribed power and inlet temperature
       if(qb.ne.0.) pmflx=qb*pmflx/(2.*q)
c
c  determine total heat flux to secondary and zero
c  all variables in core version heat transfer approach
c
       call htprm(a(lql),a(lidwn),nzp2,nktb,a(ldtrn),a(laht),
     1 a(lqv),a(lhlnb),a(lhlfc),a(lhvfc),a(ltw),a(ldtw))
c
25      continue
       iqss=0
```

```
c
c  perform fluid dynamics calculation
c
        call newton
       if (lerr) go to 100
       rtime=rtime+delt
       nstep=nstep+1
c
c  compute total linear heat flux for printout
        if(iht.ne.0) call heat(a(lql),a(lqv),a(lqpp),a(lhlnb),
     1 a(lhlfc),a(lhvfc),a(ltw),a(ltln),a(ltvn),a(ltsat),a(lidwn),
     2 nzp2,nz,nc)
c
c  print results
c
   30 call edit(a(laht),a(lpn),a(lalpn),a(lrovn),a(lroln),a(levn),
     1a(leln),a(lidwn),a(ltvn),a(ltln),a(lvvx),a(lvlx),a(lvvy),
     1a(lvly),
     2 a(lvvz),a(lvlz),a(ltsat),a(lhv),a(lhl),a(ltrn),a(lihtr),a(lqpp),
     3 a(ldz),a(larz),a(lrn),a(lfrac),a(lchfr),nzp2,nzp,nodes,nz,nktb)
c
c
       if(lerr.or.tend.eq.0.0) return
c
c  stop if time step is less than dtmin
c
       if(delt.ge.dtmin) go to 45
       lerr=.true.
       ierr=10
       return
c
c  advance time step
c
 45     call move(a(lpn),a(lp),nm)
c
c  end of one time step, return for another time step
c
       ierr = 0
       go to 20
c
c  error detected during Newton iterations,
c    back up and reduce time step
c
  100 if ((ired.eq.iredmx).or.(dtmin.ge.dtmax)) go to 200
       ts = 0.1
       if ((ird.eq.1).and.(ired.eq.0)) ts = 0.5
       delt = ts*delt
       if ((ird.eq.0).and.(ired.eq.0)) kred = kred+1
       ired = ired + 1
       if (delt.lt.dtmin) go to 200
       call move(a(lp),a(lpn),nm)
       go to 21
c
c  too much reduction, error stops calculation
```

```
c
  200 rtnlp = rtime
      go to 30
c
 1000 format(v)
 1001 format (32h please enter new time zone card )
      end
c
      subroutine heat(ql,qv,qpp,hlnb,hlfc,hvfc,tw,tln,tvn,tsat,
     1 idown,nzp2,nz,nc)
c
      dimension ql(nzp2,1),qv(nzp2,1),qpp(nzp2,1),hlnb(nzp2,1),
     1 hvfc(nzp2,1),hlfc(nzp2,1),tw(nzp2,1),tln(nzp2,1),tvn(nzp2,1),
     2 tsat(nzp2,1),idown(1)
c
      do 1 i=1,nc
      if(idown(i).eq.-1) go to 1
      do 2 j=1,nzp2
2     qpp(j,i)=hvfc(j,i)*(tw(j,i)-tvn(j,i))+qv(j,i)+ql(j,i)+
     + hlnb(j,i)*(tw(j,i)-tsat(j,i))+hlfc(j,i)*(tw(j,i)-tln(j,i))
1     continue
      return
      end
      subroutine timstp (vvz,dz,clm,delt,dtmin,dtmax,rtime,tend,ini,
     1 ird,m1)
c
c  determines time step size
c    if dtmin>=dtmax,  delt=dtmin
c    otherwise,        delt=minimum( dtmax, dtconv*clm, 2*dtprev )
c  the convective limit value  dtconv  is determined only
c    from the axial vapor velocities.
c
      dimension vvz(m1,1),dz(1)
      common /ic/ nstep,nitmax,nitno,iitmax,iitot,iic,kred,nm,ntc,
     1   nc,nrods,nz,nr,ncp,nzp,nzp2, iflash,itb,ibb,icpu,iwft,ivec,
     2   nodes,ndm1,ncf,ncc,ng,iht, iss,iqss,itam,ires,idump,ntitle,
     3   itwmax,jtwmax,itrmax,jtrmax,imchfr,jmchfr,ichf,ifintr,ierr,lerr
     4    ,imixm,imixe,iafm,itfm,igfm,nktb,jhem
      logical lerr
      data eps/1.e-7/
c
      ird=0
      ini=1
      if (rtime .gt. tend-eps) return
      ini=0
      dtprev = delt
      if (dtprev.eq.0.) dtprev = dtmax
      delt = dtmin
      if (dtmin.ge.dtmax) return
c
      ts = dtmax
      do 20 j=1,nz
        ts1 = 0.
        jj=j+1
```

```
        do 10 i=1,nc
   10     ts1 = amax1 (ts1, abs(vvz(j,i)) )
        if(ts1.gt.0.) ts = amin1 (ts, clm*dz(jj)/ts1,clm*dz(j)/ts1)
   20   continue
      delt = amin1( ts, 2.*dtprev )
      if (delt.lt.0.9*ts) ird = 1
      kred = kred + ird
      return
      end
      subroutine table(fx,x,f,y,n)
c  Performs tabular look-up
c  fx  =  quantity to be found
c   x  =  independent variable
c   f  =  array containing ordinate values
c   y  =  array containing abcissa values
c   n  =  number of values in f or y arrays
c
      dimension f(1),y(1)
      do 120 i=1,n
      if(x-y(i)) 130,110,120
  110  if (i.eq.n) go to 140
  120  continue
      go to 170
  130  if(i.eq.1) go to 160
  140  slope=(x-y(i-1))/(y(i)-y(i-1))
  150  fx=f(i-1)+ slope*(f(i)-f(i-1))
       return
  160  fx=1.0
       return
  170  fx=f(n)
      return
      end
      subroutine newton
c
c  performs Newton iterations for fluid dynamics equations
c
      common a(1)
      common /point/ lncr,lindnt,licc,liwfz,lihtr,licr,lirc,lidwn,ljmtb,
     1  lp,lalp,lrov,lrol,lev,lel,ltv,ltl,ltr, lpn,lalpn,lrovn,
     2   lroln,levn,leln,ltvn,ltln,ltrn, lvvx,lvlx,lvvy,lvly,lvvz,lvlz,
     3   lhv,lhl,lhvs,lhls,ltsat,lvisv,lvisl, ldx,ldy,ldz,larx,lary,
     4   larz,lvol, lcpvx,lcpvy,lcpvz,lcplx,lcply,lcplz, lfvx,lfvy,
     5   lfvz,lflx,lfly,lflz, lajm1,lajm2,lcpa,lcptv,lcptl,lrhs,
     6ldp,lqpp,lqv,lql,lhvfc,lhlnb,lhlfc, ldtrn,ldtw,ltw,lchfr,lfrac,
     7lhdz,lhdt,lhdh,lqz,lqt,lqr,lqppp,lrn,lcnd,lrcp,lrrdr,lvp,lvm,lrad
     8,lbotbc,ltopbc,ltmpbc,lsij,lph,lth,lfg,ldsew,ldsns,laht,lend
      common /ic/ nstep,nitmax,nitno,iitmax,iitot,iic,kred,nm,ntc,
     1   nc,nrods,nz,nr,ncp,nzp,nzp2, iflash,itb,ibb,icpu,iwft,ivec,
     2   nodes,ndml,ncf,ncc,ng,iht, iss,iqss,itam,ires,idump,ntitle,
     3   itwmax,jtwmax,itrmax,jtrmax,imchfr,jmchfr,ichf,ifintr,ierr,lerr
     4     ,imixm,imixe,iafm,itfm,igfm,nktb,jhem
      logical lerr
      common /rc/ delt,rdelt,errn,epsn,erri,epsi,dtmin,dtmax,tend,
     1   dtsp,dtlp,rtnsp,rtnlp,gravx,gravy,gravz,rtime,velx,
```

```
     2   q,q0,twmax,trmax,amchfr
c
c  perform explicit calculations
c
      call explct(a(licc),a(lirc),a(ldx),a(ldy),a(ldz),a(lvvx),a(lvvy),
     1 a(lvvz),a(lvlx),a(lvly),a(lvlz),a(lp),a(lalp),a(lrov),a(lrol),
     2 a(lcpvx),a(lcplx),a(lcpvy),a(lcply),a(lcpvz),a(lcplz),a(lfvx),
     3 a(lflx),a(lfvy),a(lfly),a(lfvz),a(lflz),a(larx),a(lary),
     4 a(larz),a(liwfz),a(lvisv),a(lvisl),a(lihtr),a(lqv),a(lql),
     5 a(lhdz),a(lhdt),a(lfrac),a(lsij),a(lph),a(lth),nc,nz,nzp,nzp2)
c
c  begin newton iteration
c
      iitot=0
      lim = iabs(nitmax)
      do 100 it=1,lim
        nitno=it
c
c  set up jacobian matrix
c
        call jacob(a(licc),a(licr),a(lirc),a(ldz),a(lvol),a(lrov),
     1  a(lrol),a(lalp),a(lp),a(lev),a(lel),a(lrovn),
     2  a(lroln),a(lalpn),a(lpn),a(levn),a(leln),a(ltvn),a(ltln),
     3  a(ltw),a(larx),a(lary),a(larz),a(lcpvx),a(lcplx),a(lcpvy),
     4  a(lcply),a(lcpvz),a(lcplz),a(lfvx),a(lflx),a(lfvy),a(lfly),
     5  a(lfvz),a(lflz),a(lrn),a(lqv),a(lhvfc),a(lhlnb),
     6  a(lhlfc),a(lajm1),a(lajm2),a(lcpa),a(lcptv),a(lcptl),a(lrhs),
     7  a(lvvx),a(lvvy),a(lvvz),a(lvlx),a(lvly),a(lvlz),a(ldtw),
     8  a(lihtr),a(lvisl), a(ltv), a(lhdz),a(laht),
     9  nzp2,nzp,nz,nc,nktb)
        if (lerr) return
c
c  solve for new pressures
c
        if ((itam.eq.0).or.(itam.eq.2)) go to 5
        call innbgs
        go to 10
  5     call clear (0., a(ldp), nzp2*nc)
        call inner (a(ldp),a(lajm1),a(lajm2),a(lrhs),a(lend),a(licc),
     1  nz,nc,nzp2,ncp,epsi,iitmax,iic,erri,a(lp) )
 10     continue
        iitot=iitot+iic
c
c  obtain remaining variables and compute error
c
      call update(a(lidwn),a(licc),a(lirc),a(lpn),a(lalpn),a(ltvn),
     1a(ltln),a(lrovn),a(lroln),a(levn),a(leln),a(ldp),a(lrhs),a(lcpa),
     2a(lcptv),a(lcptl),a(ltsat),a(ltw),a(ldtw),a(lhlfc),a(lhlnb),
     3a(lhvfc),a(lhvs),a(lhls),nz,nc,nzp,nzp2)
        if (lerr) return
        if(nitmax.eq.1) go to 200
        call newerr(a(lpn),a(ldp),errn,nzp2*nc)
 20     continue
c
```

```
c  test error
c
         if(errn.le.epsn) go to 200
  100    continue
      if (nitmax.gt.0) go to 200
c
c  Newton iterations failed to converge as required
c
      ierr = 9
      lerr = .true.
      return
c
c  compute velocities
c
  200 call vset(a(licc),a(lpn),a(lvvx),a(lvlx),a(lvvy),a(lvly),a(lvvz),
     1 a(lvlz),a(lcpvx),a(lcplx),a(lcpvy),a(lcply),a(lcpvz),
     2 a(lcplz),a(lfvx),a(lflx),a(lfvy),a(lfly),a(lfvz),a(lflz),
     3 nzp2,nzp,nz)
c
c  determine quantities in top and bottom fictitious cells
c
      call qset(a(lpn),a(lalpn),a(lrovn),a(lroln),a(levn),a(leln),
     1          a(ltvn),a(ltln),a(lvvz),a(lvlz),a(ltsat),
     2  a(lidwn),a(ldz),a(lvisl),a(lvisv),a(liwfz),a(lhdz),gravz,
     3          nzp2,nzp,nc,itb,ibb)
c
c  compute properties
c
      call pset(a(lpn),a(lrovn),a(lroln),a(levn),a(leln),a(ltvn),
     1 a(ltln),a(lhv),a(lhl),a(lvisv),a(lvisl),nzp2)
      return
      end
      subroutine newerr(pn,dp,errn,n)
c
c  find largest relative pressure change for the preceding
c   Newton iteration
c
      dimension pn(1),dp(1)
c
      errn=0.0
      do 10 i=1,n
         if(pn(i).ne.0.0) errn=amax1(errn,abs(dp(i)/pn(i)) )
   10    continue
      return
      end
      subroutine edit(aht,pn,alpn,rovn,roln,evn,eln,idown,tvn,tln,vvx,
     1 vlx,vvy,vly,vvz,vlz,tsat,hv,hl,trn,ihtr,qpp,dz,arz,rn,
     2 fracp,chfr,m1,m2,m3,m4,m5)
c
       common/sgpt1/pprim,pmflx,tpin,tpout,zeb1,zeb2,zeb3,zeb4,zeb5,
     1 zeb6,zeb7(15,12),zeb8
c  prints out state of flow at the end of specified time steps
c
      common/sg3/qqx(30),qqy(30),k1111,d1111
```

```
       common/sg1/xxzz1,xxzz2,nxzz3,nxzz4,xxzz5,xxzz6,tfeed,wfeed
      common/sg/vd,xl2,vfeed,xfeed,nonce,ksh,nfeed,nzlvl,nmin,ti,
     1 pdi,t2i,ro2i,h2i,vdi,wd,xr,wr,roavg,xlr,pd,pr,ro2,t2,h2,v2,
     2 dpdz,dro2,dpris,dpsep,wsf,wsb,wfdf,wfdb,t2a,h2a,ro2a,
     3 t2b,h2b,ro2b,psdom,ireci,xl2i,hris,vt,nsep
      common /ic/ nstep,nitmax,nitno,iitmax,iitot,iic,kred,nm,ntc,
     1   nc,nrods,nz,nr,ncp,nzp,nzp2, iflash,itb,ibb,icpu,iwft,ivec,
     2   nodes,ndml,ncf,ncc,ng,iht, iss,iqss,itam,ires,idump,ntitle,
     3   itwmax,jtwmax,itrmax,jtrmax,imchfr,jmchfr,ichf,ifintr,ierr,lerr
     4    ,imixm,imixe,iafm,itfm,igfm,nktb,jhem
      logical lerr
      common /rc/ delt,rdelt,errn,epsn,erri,epsi,dtmin,dtmax,tend,
     1   dtsp,dtlp,rtnsp,rtnlp,gravx,gravy,gravz,rtime,velx,
     2   q,q0,twmax,trmax,amchfr
      common /units/ ntty,ninp,nout,ntzc,nres,ndump
      dimension pn(m1,1),alpn(m1,1),rovn(m1,1),roln(m1,1),evn(m1,1),
     1 eln(m1,1),tvn(m1,1),tln(m1,1),vvx(m1,1),vlx(m1,1),vvy(m1,1),
     2 vly(m1,1),vvz(m2,1),vlz(m2,1),tsat(m1,1),hv(m1,1),hl(m1,1),
     3 trn(m3,m1,1),ihtr(m4,1),qpp(m1,1),dz(1),arz(m2,1),rn(1),fracp(1)
     4 ,chfr(m1,1),aht(m5,m1,1),idown(1)
      logical long,short
      idon(v) = int(.51 + sign(.5,v) )
c
c  obtain current cpu time
c
      call timing(ncpu)
      cpu=.01*(ncpu-icpu)
c
c  monitor print
c
      short=.true.
      long=.true.
      rtm=rtime+0.5*delt
      if(nstep.gt.0.and.icpu.gt.0) go to 10
        write(ntty,1011)
        cpu=0.0
        kcpu = icpu + 6000
        go to 19
c
   10 continue
c  short print
      if(rtime.ge.rtnsp)go to 11
      if(rtnsp.lt.rtm)go to 11
      short=.false.
c  long print
   11 continue
      if(rtime.ge.rtnlp)go to 12
      if(rtnlp.lt.rtm)go to 12
      long=.false.
   12 continue
      if (ncpu.lt.kcpu) go to 15
        write(ntty,1018) rtime,delt,cpu,kred
        kcpu = kcpu + 6000
   15 if (.not.(long.or.short)) return
```

```
c
c  heading for both short and long print
c
   19 hsum = 0.
      qsum = 0.
      wsumi = 0.
      wsumo = 0.
       wstm=0.
       wdown=0.
       wrise=0.
      do 20 i=1,nc
        ji =     2 - idon(vvz(  1,i))
        jo = nzp2 - idon(vvz(nzp,i))
      wi = alpn(ji,i)*rovn(ji,i)*vvz(  1,i)*arz(  1,i)
      wo = alpn(jo,i)*rovn(jo,i)*vvz(nzp,i)*arz(nzp,i)
       w1=wo
      wsumi = wsumi+wi
      wsumo = wsumo+wo
      hsum=hsum+hv(jo,i)*wo-hv(ji,i)*wi
        ji =     2 - idon(vlz(  1,i))
        jo = nzp2 - idon(vlz(nzp,i))
      wi = (1.-alpn(ji,i))*roln(ji,i)*vlz(  1,i)*arz(  1,i)
      wo = (1.-alpn(jo,i))*roln(jo,i)*vlz(nzp,i)*arz(nzp,i)
       w2=wo
      wsumi = wsumi+wi
      wsumo = wsumo+wo
      hsum=hsum+hl(jo,i)*wo-hl(ji,i)*wi
       w3=w1+w2
       if(idown(i).eq.-1) wdown=wdown+w3
       if(idown(i).ne.1) go to 20
       wstm=wstm+w1
       wrise=wrise+w3
20       continue
       if(wstm.ne.0.) circ=wrise/wstm
      do 30 i=1,nrods
        do 30 j=1,nzp2
       ahtot=0.
       do 123 k=1,nktb
123        ahtot=ahtot+aht(k,j,i)
30         qsum= qsum + qpp(j,i)*ahtot
c
      kcpu = ncpu + 6000
      qkw = 0.001*q
      hsum = 0.001*hsum
      qsum = 0.001*qsum
      werr = wsumi - wsumo
      qerr = qsum - hsum
   40 continue
      ireci=1
      write(ntty,1000) nstep,rtime,delt,nitno,iitot,cpu,werr,qerr
       wfd2=wfeed/2.
      if(long) write(nout,999)
      write (nout,1001) nstep,rtime,delt,cpu,nitno,iitot,kred,
     1 qkw,wdown,qsum,wrise,tpin,hsum,wstm,tpout,
```

```
     2 psdom,circ,pmflx,wfd2,xl2,pprim,tfeed
      kred = 0
      if (long) go to 200
c
c  short print
c
      rtnsp=rtnsp+dtsp
      return
c
c  long print
c
  200 continue
      rtnlp=rtnlp+dtlp
      if(short)rtnsp=rtnsp+dtsp
      write (nout,1006)
      do 220 i=1,nc
        do 210 j=1,nzp
          jv = j + 1 - idon(vvz(j,i))
          jl = j + 1 - idon(vlz(j,i))
          qual = alpn(jv,i)*rovn(jv,i)*vvz(j,i)
          g = qual + (1.-alpn(jl,i))*roln(jl,i)*vlz(j,i)
          if (g.ne.0.)  qual = 100.*qual/g

          hm = (1.-qual/100.)*hl(j,i) + qual/100.*hv(j,i)
          write (nout,1007) i,j,pn(j,i),alpn(j,i),qual,hm,hl(j,i),
     1      tvn(j,i),tln(j,i),tsat(j,i),vvz(j,i),vlz(j,i),rovn(j,i),
     2      roln(j,i),g
  210     continue
        write (nout,1017) i,nzp2,pn(nzp2,i),alpn(nzp2,i),hm,
     1    hl(nzp2,i),tvn(nzp2,i),tln(nzp2,i),tsat(nzp2,i),
     2    rovn(nzp2,i),roln(nzp2,i)
  220   continue
c
      if (nc.eq.1.or.itam.eq.0) go to 250
      write (nout,1008)
      do 240 i=1,nc,2
        ip = i+1
        if (ip.gt.nc) go to 232
        write (nout,1009) (i,vvx(j,i),vlx(j,i),vvy(j,i),vly(j,i),j,
     1    ip,vvx(j,ip),vlx(j,ip),vvy(j,ip),vly(j,ip), j=2,nzp)
        go to 233
  232     write (nout,1015) (i,vvx(j,i),vlx(j,i),vvy(j,i),vly(j,i), j,
     1      j=2,nzp)
  233   write (nout,1014)
  240   continue
c
  250 if (iht.eq.0.or.iqss.eq.0) go to 280
      write (nout,1012)
      do 270 i=1,nrods
        do 260 j=1,nzp2
          ki=4
          qp = qpp(j,i)*6.2831853*rad(ki)
       jj=j+1
          write (nout,1013) i,jj,ihtr(j,i),qp,chfr(jj,i),
```

```
     1                           (trn(k,jj,i), k=1,nodes)
  260      continue
        write (nout,1014)
  270   continue
c
  280 if (.not.lerr) return
      write (nout,1016)
      return
c
c
 1000 format(i5,e12.5,e10.3,2i4,f8.2,1x,e12.5,2x,e12.5)
999   format(1h1)
 1001 format(15h time step no =,i5,10x,6htime =,f11.6,4h sec,6x,
     1 16htime step size =,e12.5,4h sec,10x,10hcpu time =,f8.2,4h sec,/
     2 30h number of newton iterations =, i4,/
     3 30h number of inner iterations = , i4,20x,i4,
     a   36h reduced time steps since last print//
     4 22h total primary power =,f12.3,3h kW,7x,
     5 21hdowncomer flow rate =,f12.3,5h kg/s,9x,
     6 27h primary flow parameters : ,/
     7 22h total heat transfer =,f12.3,3h kW,10x,
     8 18h riser flow rate =,f12.3,5h kg/s,9x,
     9 22h inlet temperature  = ,f9.2,2h k,/
     1 22h flow enthalpy rise = ,f12.3,3h kW,10x,
     1 19h steam flow rate = ,f11.3,5h kg/s,9x,
     2 22h outlet temperature = ,f9.2,2h K,/
     2 22h steam dome pressure= ,f12.1,3h Pa,9x,
     2 20hcirculation ratio = ,f7.2,8h (=1/xr),10x,
     3 13h mass flux = ,f10.2,10h Kg/m2/sec,/
     3 22h feedwater flow rate = ,f10.3,5h Kg/s,5x,
     3 24hdowncomer water level = ,f8.3,2h m,15x,
     3 19h system pressure = ,f12.1,3h Pa,/
     4 25h feedwater temperature = ,f9.2,2h K,/)
 1002 format(9h pressure,/)
 1003 format(i4,10 f10.5,/(4x,10 f10.5) )
 1053 format(i4,-6p 10f10.6,/ (4x,10f10.6) )
 1004 format(6h alpha,/)

 1006 format (52h  ic  iz  P(MPa)   void   % qual     hm       hl   ,
     1  24h  T vap   T liq   T sat ,5x,3hvvz,5x,3hvlz,
     2  6x,3hrov,6x,3hrol,5x,9hmass flux,//)
 1007 format (2i4, -6p f9.5, 0p f8.4,f7.2, -3p f11.3,  f9.3,
     1   0p 3f8.2, 2x,2f8.3,2f9.2,f10.1)
 1017 format (2i4, -6p f9.5, 0p f8.4, 7x , -3p f11.3,f9.3,
     1         0p 3f8.2,18x,2f9.2/)
 1008 format (//4h  ic, 4x,3hvvx,8x,3hvlx,8x,3hvvy,8x,3hvly,4x,
     1  8h  iz  ic, 4x,3hvvx,8x,3hvlx,8x,3hvvy,8x,3hvly,//)
 1009 format (i4, 1p 4e11.3, 2i4, 1p 4e11.3)
c
 1011 format(41h step  real time   delta t  nit iit   cpu,
     1      35h   delta mass kg/s  delta energy kw)
c
 1012 format (//13h  ir  iz ihtr, 2x,14hheat flow(W/m), 2x,
     1   12h chfr       ,
```

```
     1  16hrod temperatures,8h (deg K),//)
 1013 format (2i4,i5, f10.0,7x,f7.4,5x, 15f7.1, (30x,15f7.1) )
c
 1014 format (/)
 1015 format(i4,1p4e11.3,i4)
 1016 format (//38h State of flow when error was detected/)
 1018 format (5x,e12.5,e10.3,8x,f8.2, i8,8h reduced)
c
      end
      subroutine move(pn,p,nm)
      dimension pn(1),p(1)
c
      do 10 i=1,nm
   10   p(i)=pn(i)
      return
      end
      subroutine vset(icc,pn,vvx,vlx,vvy,vly,vvz,vlz,cpvx,cplx,cpvy,
     1  cply,cpvz,cplz,fvx,flx,fvy,fly,fvz,flz,m1,m2,m3)
c
c  computes velocities after Newton iterations terminate
c
      common /ic/ nstep,nitmax,nitno,iitmax,iitot,iic,kred,nm,ntc,
     1   nc,nrods,nz,nr,ncp,nzp,nzp2, iflash,itb,ibb,icpu,iwft,ivec,
     2   nodes,ndm1,ncf,ncc,ng,iht, iss,iqss,itam,ires,idump,ntitle,
     3   itwmax,jtwmax,itrmax,jtrmax,imchfr,jmchfr,ichf,ifintr,ierr,lerr
     4     ,imixm,imixe,iafm,itfm,igfm,nktb,jhem
      logical lerr
      dimension icc(4,1),pn(m1,1),vvx(m1,1),vlx(m1,1),vvy(m1,1),
     1  vly(m1,1),vvz(m2,1),vlz(m2,1),cpvx(m3,1),cplx(m3,1),cpvy(m3,1),
     2  cply(m3,1),cpvz(m2,1),cplz(m2,1),fvx(m3,1),flx(m3,1),
     3  fvy(m3,1),fly(m3,1),fvz(m2,1),flz(m2,1)
c
      do 100 i=1,nc
        ixm=icc(2,i)
        iym=icc(1,i)
        do 50 j=1,nz
          jj=j+1
          vvx(jj,i)=cpvx(j,i)*(pn(jj,i)-pn(jj,ixm))+fvx(j,i)
          vlx(jj,i)=cplx(j,i)*(pn(jj,i)-pn(jj,ixm))+flx(j,i)
          vvy(jj,i)=cpvy(j,i)*(pn(jj,i)-pn(jj,iym))+fvy(j,i)
          vly(jj,i)=cply(j,i)*(pn(jj,i)-pn(jj,iym))+fly(j,i)
          vvz(j,i)=cpvz(j,i)*(pn(jj,i)-pn(j,i))+fvz(j,i)
          vlz(j,i)=cplz(j,i)*(pn(jj,i)-pn(j,i))+flz(j,i)
   50     continue
        vvz(nzp,i)=cpvz(nzp,i)*(pn(nzp2,i)-pn(nzp,i))+fvz(nzp,i)
        vlz(nzp,i)=cplz(nzp,i)*(pn(nzp2,i)-pn(nzp,i))+flz(nzp,i)
  100   continue
      return
      end
      subroutine pset(pn,rovn,roln,evn,eln,tvn,tln,hv,hl,visv,visl,m1)
c
c  computes fluid properties (enthalpy, viscosity) at end of
c   Newton iterations
c
```

```
      common /ic/ nstep,nitmax,nitno,iitmax,iitot,iic,kred,nm,ntc,
     1   nc,nrods,nz,nr,ncp,nzp,nzp2, iflash,itb,ibb,icpu,iwft,ivec,
     2   nodes,ndm1,ncf,ncc,ng,iht, iss,iqss,itam,ires,idump,ntitle,
     3   itwmax,jtwmax,itrmax,jtrmax,imchfr,jmchfr,ichf,ifintr,ierr,lerr
     4    ,imixm,imixe,iafm,itfm,igfm,nktb,jhem
      logical lerr
      common /rc/ delt,rdelt,errn,epsn,erri,epsi,dtmin,dtmax,tend,
     1   dtsp,dtlp,rtnsp,rtnlp,gravx,gravy,gravz,rtime,velx,
     2   q,q0,twmax,trmax,amchfr
      dimension pn(m1,1),rovn(m1,1),roln(m1,1),evn(m1,1),eln(m1,1),
     1  tvn(m1,1),tln(m1,1),hv(m1,1),hl(m1,1),visv(m1,1),visl(m1,1)
c
      do 100 i=1,nc
        do 100 j=1,nzp2
          hv(j,i)=evn(j,i)+pn(j,i)/rovn(j,i)
          hl(j,i)=eln(j,i)+pn(j,i)/roln(j,i)
          visv(j,i)=visvp(tvn(j,i))
          visl(j,i)=vislq(tln(j,i))
  100     continue
      return
      end
      subroutine qset(pn,alpn,rovn,roln,evn,eln,tvn,tln,vvz,vlz,tsat,
     1         idown,dz,visl,visv,iwfz,hdz,gravz,nzp2,nzp,nc,itb,ibb)
c
c  computes quantities for top and bottom layers of fictitious cells
c
      dimension pn(nzp2,1),alpn(nzp2,1),rovn(nzp2,1),roln(nzp2,1),
     1          evn(nzp2,1),eln(nzp2,1),tvn(nzp2,1),tln(nzp2,1),
     2          tsat(nzp2,1),vvz(nzp,1),vlz(nzp,1),visl(nzp2,1),
     3          visv(nzp2,1),dz(2),hdz(nzp2,1),iwfz(1),idown(1)
c
      idon(i1,i2,v)  = i1+(i2-i1)*int(.51+sign(.5,v))
c                    = i1 if v<0,  = i2 if v>0
      do 100 i=1,nc
        jdn = idon(   2,   1,vvz(  1,i))
        tvn(   1,i) = tvn(jdn,i)
        alpn(   1,i) = alpn(jdn,i)
        jdn = idon(   2,   1,vlz(  1,i))
        tln(   1,i) = tln(jdn,i)
        jdn = idon(nzp2, nzp,vvz(nzp,i))
        tvn(nzp2,i) = tvn(jdn,i)
        alpn(nzp2,i) = alpn(jdn,i)
        if(idown(i).ne.-1)alpn(nzp2,i)=1.
        jdn = idon(nzp2, nzp,vlz(nzp,i))
        tln(nzp2,i) = tln(jdn,i)
c
c   compute pressure in fictitous cell if ibb=1
c
        if(ibb.ne.1) go to 30
        dza=.5* (dz(1) + dz(2))
        rdz = 0.5 /dza
        rdz2 = 2. * rdz
        hdza=rdz*(hdz(1,i)*dz(1) + hdz(2,i)*dz(2))
        rola = rdz*(roln(1,i)*dz(1) + roln(2,i)*dz(2))
```

```
         rova = rdz*(rovn(1,i)*dz(1) + rovn(2,i)*dz(2))
         alpa = rdz*(alpn(1,i)*dz(1) + alpn(2,i)*dz(2))
         alpal = 1.-alpa
         visva = rdz*(visv(1,i)*dz(1) + visv(2,i)*dz(2))
         visla = rdz*(visl(1,i)*dz(1) + visl(2,i)*dz(2))
         pa = rdz*(pn(1,i)*dz(1) + pn(2,i)*dz(2))
         cfv=0.0
         cfl=1.0
         call fwall(fwv,fwl,hdza,1.,cfv,cfl,pa,alpa,rova,rola,
     1             vvz(1,i),vlz(1,i),visva,visla,rdz2,iwfz(1),i,1)
         conv=0.0
         if(vlz(1,i).lt.0.) conv=(vlz(2,i)-vlz(1,i))*vlz(1,i)/dz(2)
         pn(1,i) = pn(2,i) + dza*(rola*(conv-gravz)+fwl*vlz(1,i)/alpal)
   30    continue
c
         call state(pn(    1,i),tvn(    1,i),tln(    1,i),rovn(    1,i),
     1             roln(    1,i),evn(    1,i),eln(    1,i),tsat(    1,i),
     2             d0,d00,d1,d2,d3,d4,d5,d6,d7,d8,d9,2,ierr)
         call state(pn(nzp2,i),tvn(nzp2,i),tln(nzp2,i),rovn(nzp2,i),
     1             roln(nzp2,i),evn(nzp2,i),eln(nzp2,i),tsat(nzp2,i),
     2             d0,d00,d1,d2,d3,d4,d5,d6,d7,d8,d9,2,ierr)
  100    continue
       return
       end
       subroutine explct(icc,irc,dx,dy,dz,vvx,vvy,vvz,
     1  vlx,vly,vlz,p,alp,rov,rol,cpvx,cplx,cpvy,cply,
     2  cpvz,cplz,fvx,flx,fvy,fly,fvz,flz,arx,ary,arz,iwfz,visv,visl,
     3  ihtr,qv,ql,hdz,hdt,fracp,sij,ph,th,m4,m3,m2,m1)
c
c  computes the coefficients of P and of velocities in linear
c   momentum equations
c
c  since all terms are assumed to be linear in implicit variables, this
c   calculation is only done once per time step
c
       common /ic/ nstep,nitmax,nitno,iitmax,iitot,iic,kred,nm,ntc,
     1   nc,nrods,nz,nr,ncp,nzp,nzp2, iflash,itb,ibb,icpu,iwft,ivec,
     2   nodes,ndml,ncf,ncc,ng,iht, iss,iqss,itam,ires,idump,ntitle,
     3   itwmax,jtwmax,itrmax,jtrmax,imchfr,jmchfr,ichf,ifintr,ierr,lerr
     4    ,imixm,imixe,iafm,itfm,igfm,nktb,jhem
       logical lerr
       common /rc/ delt,rdelt,errn,epsn,erri,epsi,dtmin,dtmax,tend,
     1   dtsp,dtlp,rtnsp,rtnlp,gravx,gravy,gravz,rtime,velx,
     2   q,q0,twmax,trmax,amchfr
       dimension icc(4,1),irc(4,1),dx(1),dy(1),dz(1),vvx(m1,1),
     1  vvy(m1,1),vvz(m2,1),p(m1,1),alp(m1,1),rov(m1,1),rol(m1,1),
     2  cpvx(m3,1),cplx(m3,1),cpvy(m3,1),cply(m3,1),
     3  cpvz(m2,1),cplz(m2,1),fvx(m3,1),flx(m3,1),
     4  fvy(m3,1),fly(m3,1),fvz(m2,1),flz(m2,1),
     5  vlx(m1,1),vly(m1,1),vlz(m2,1),arx(m3,1),ary(m3,1),arz(m2,1),
     6  iwfz(1),visv(m1,1),visl(m1,1),ihtr(m3,1),qv(m1,1),ql(m1,1)
     7   ,fracph(1),ph(m1,1),th(m1,1),hdz(m1,1),hdt(m1,1)
        dimension sij(4,1)
       data  itrns / 5 /
```

```
c--value of ihtr for transition boiling regime. it is assumed that
c  for ihtr<5, liquid coats rods, for ihtr>5, vapor contacts rods
      idon(i1,i2,v)=i1+(i2-i1)*int(.51+sign(.5,v))
c                   =i1 if v<0,   =i2 if v>0
c
      jstart=1
      jstop=nzp
      if(ibb.eq.1)jstart=2
      if(itb.eq.1)jstop=nz
c
c  momentum equations
c
      do 1000 i=1,nc
        iym=icc(1,i)
        ixm=icc(2,i)
        ixp=icc(3,i)
        iyp=icc(4,i)
        if(ixm.eq.ncp) go to 200
        if(itam.eq.0)  go to 200
        ixmyp = icc(4,ixm)
c
c  x direction
c
        do 100 j=1,nz
          if(arx(j,i).eq.0.0) go to 100
          jj=j+1
          rdx=1./(dx(i)+dx(ixm))
          rdx2=2.*rdx
          hdza=rdx*(hdz(jj,i)*dx(i)  + hdz(jj,ixm)*dx(ixm))
          hdta=rdx*(hdt(jj,i)*dx(i)  + hdt(jj,ixm)*dx(ixm))
          sinph=rdx*(sin(ph(jj,i))*dx(i)  + sin(ph(jj,ixm))*dx(ixm))
          sinth=rdx*(sin(th(jj,i))*dx(i)  + sin(th(jj,ixm))*dx(ixm))
          cosph=rdx*(cos(ph(jj,i))*dx(i)  + cos(ph(jj,ixm))*dx(ixm))
          costh=rdx*(cos(th(jj,i))*dx(i)  + cos(th(jj,ixm))*dx(ixm))
          pa=rdx*(p(jj,i)*dx(i)+p(jj,ixm)*dx(ixm))
          alpa=rdx*(alp(jj,i)*dx(i)+alp(jj,ixm)*dx(ixm))
          rova=rdx*(rov(jj,i)*dx(i)+rov(jj,ixm)*dx(ixm))
          rola=rdx*(rol(jj,i)*dx(i)+rol(jj,ixm)*dx(ixm))
          arv=alpa*rova
          arl=(1.-alpa)*rola
          visva=rdx*(visv(jj,i)*dx(i)+visv(jj,ixm)*dx(ixm))
          visla=rdx*(visl(jj,i)*dx(i)+visl(jj,ixm)*dx(ixm))
c--set liquid and vapor wall contact fractions for fwall
          cfv=0.
          cfl=1.
          if (iht.eq.0) go to 90
          idn = idon(i,ixm,vix(jj,i))
          itest = 0
          do 20 n=1,4
          nn = irc(n,idn)
          if(nn.eq.0) go to 30
          if(itest.lt.ihtr(j,nn)) itest = ihtr(j,nn)
   20     continue
   30     continue
```

```
      if(itest-itrns) 40,50,80
40    go to 90
50    qlsave = 0.0
      qvsave = 0.0
      do 60 n=1,4
      nn = irc(n,idn)
      if(nn.eq.0) go to 70
      qlsave = qlsave + ql(j+1,nn)*fracp(nn)
      qvsave = qvsave + qv(j+1,nn)*fracp(nn)
60    continue
70    cfl = qlsave/(qlsave+qvsave)
      cfv = 1.0- cfl
      go to 90
80    cfv=1.
      cfl=0.
90    vzl= .25*(vlz(jj,i) + vlz(j,i) +vlz(jj,ixm) + vlz(j,ixm))
      vzv = .25*(vvz(jj,i) + vvz(j,i) +vvz(jj,ixm) + vvz(j,ixm))
      vyv = .25*(vvy(jj,i)+vvy(jj,ixm)+vvy(jj,iyp) +vvy(jj,ixmyp))
      vyl = .25*(vly(jj,i)+vly(jj,ixm)+vly(jj,iyp)+vly(jj,ixmyp))
      vxv=vvx(jj,i)
      vxl=vlx(jj,i)
c   parallel
      vl=vxl*sinph*costh + vyl*sinph*sinth + vzl*cosph
      vv=vxv*sinph*costh + vyv*sinph*sinth + vzv*cosph
       rdz2=2./dz(jj)
       iwf=iwfz(j)
       call fwall(fwv1,fwl1,hdza,1.,cfv,cfl,pa,alpa,rova,rola,
     1          vv,vl,visva,visla,rdz2,iwf,i,jj)
       fwl=fwl1*costh*costh*sinph*sinph
       fwv=fwv1*costh*costh*sinph*sinph
       fwlex=fwl1*costh*sinph*(sinph*sinth*vyl + cosph*vzl)
       fwvex=fwv1*costh*sinph*(sinph*sinth*vyv + cosph*vzv)
c   perpendular 1
       vl=vxl*cosph*costh + vyl*cosph*sinth - vzl*sinph
       vv=vxv*cosph*costh + vyv*cosph*sinth - vzv*sinph
       iwf=iwft + 30
       if (iwfz(j).eq.0) iwf=30
       call fwall(fwv1,fwl1,hdta,velx,cfv,cfl,pa,alpa,rova,rola,
     1           vv,vl,visva,visla,rdx2,iwf,i,jj)
       fwl=fwl + fwl1*cosph*cosph*costh*costh
       fwv=fwv + fwv1*cosph*cosph*costh*costh
       fwlex=fwlex + fwl1*cosph*costh*(cosph*sinth*vyl - sinph*vzl)
       fwvex=fwvex + fwv1*cosph*costh*(cosph*sinth*vyv - sinph*vzv)
c   perpendular 2
       vl= -vxl*sinth + vyl*costh
       vv= -vxv*sinth + vyv*costh
       rdy2=2./dy(i)
       call fwall(fwv1,fwl1,hdta,velx,cfv,cfl,pa,alpa,rova,rola,
     1          vv,vl,visva,visla,rdy2,iwf,i,jj)
       fwl=fwl + fwl1*sinth*sinth
       fwv=fwv + fwv1*sinth*sinth
       fwlex=fwlex - fwl1*sinth*costh*vyl
       fwvex=fwvex - fwv1*sinth*costh*vyv
       vr=abs(vxv-vxl)
```

```
      fiv=1.0e+10
      fil=1.0e+10
      if(j.ge.jhem) go to 11
         call finter(fiv,fil,alpa,rova,rola,vr,visva,visla,hdta,ifintr)
11    continue
         call cvecx(tcvx,dx,dy,dz,icc,iwfz,vvx,vvy,vvz,jj,i,
     1     ixm,ixp,iym,iyp,nzp,nzp2)
         call cvecx(tclx,dx,dy,dz,icc,iwfz,vlx,vly,vlz,jj,i,
     1     ixm,ixp,iym,iyp,nzp,nzp2)
c     if(((i.eq.5).or.(i.eq.6)).and.(j.eq.2))tcvx=0.0
c      if(((i.eq.5).or.(i.eq.6)).and.(j.eq.2))tclx=0.0
c     if(((i.eq.9).or.(i.eq.10)).and.(j.eq.2))tcvx=0.0
c     if(((i.eq.9).or.(i.eq.10)).and.(j.eq.2))tclx=0.0
         fexvx=(rdelt*vvx(jj,i)-tcvx+gravx)*arv - fwvex
         fexlx=(rdelt*vlx(jj,i)-tclx+gravx)*arl - fwlex
         aa=rdelt*arv+fiv+fwv
         bb=rdelt*arl+fil+fwl
         rd=1./(aa*bb-fil*fiv)
         rvdx=rdx2*alpa
         rldx=rdx2*(1.-alpa)
         cpvx(j,i)=-rd*(bb*rvdx+fiv*rldx)
         cplx(j,i)=-rd*(aa*rldx+fil*rvdx)
         fvx(j,i)=rd*(bb*fexvx+fiv*fexlx)
         flx(j,i)=rd*(aa*fexlx+fil*fexvx)
  100    continue
  200  continue
       if(iym.eq.ncp) go to 400
       if(itam.eq.0)  go to 400
       iymxp=icc(3,iym)
c
c  y direction
c
       do 300 j=1,nz
         if(ary(j,i).eq.0.0) go to 300
         jj=j+1
         rdy=1./(dy(i)+dy(iym))
         rdy2=2.*rdy
         hdza=rdy*(hdz(jj,i)*dy(i)  + hdz(jj,iym)*dy(iym))
        hdta=rdy*(hdt(jj,i)*dy(i)  + hdt(jj,iym)*dy(iym))
         sinph=rdy*(sin(ph(jj,i))*dy(i)  + sin(ph(jj,iym))*dy(iym))
         sinth=rdy*(sin(th(jj,i))*dy(i)  + sin(th(jj,iym))*dy(iym))
         cosph=rdy*(cos(ph(jj,i))*dy(i)  + cos(ph(jj,ixm))*dy(iym))
         costh=rdy*(cos(th(jj,i))*dy(i)+ cos(th(jj,iym))*dy(iym))
         pa=rdy*(p(jj,i)*dy(i)+p(jj,iym)*dy(iym))
         alpa=rdy*(alp(jj,i)*dy(i)+alp(jj,iym)*dy(iym))
         rova=rdy*(rov(jj,i)*dy(i)+rov(jj,iym)*dy(iym))
         rola=rdy*(rol(jj,i)*dy(i)+rol(jj,iym)*dy(iym))
         arv=alpa*rova
         arl=(1.-alpa)*rola
         visva=rdy*(visv(jj,i)*dy(i)+visv(jj,iym)*dy(iym))
         visla=rdy*(visl(jj,i)*dy(i)+visl(jj,iym)*dy(iym))
c--set liquid and vapor wall contact fractions for fwall
         cfv=0.
         cfl=1.
```

```
         if (iht.eq.0) go to 290
         idn = idon(i,iym,vly(jj,i))
         itest = 0
         do 220 n=1,4
         nn = irc(n,idn)
         if(nn.eq.0) go to 230
         if(itest.lt.ihtr(j,nn)) itest = ihtr(j,nn)
 220     continue
 230     continue
         if(itest -itrns) 240,250,280
 240     go to 290
 250     qlsave = 0.0
         qvsave = 0.0
         do 260 n=1,4
         nn = irc(n,idn)
         if(nn.eq.0) go to 270
         qlsave = qlsave + ql(j+1,nn)*fracp(nn)
         qvsave = qvsave + qv(j+1,nn)*fracp(nn)
 260     continue
 270     cfl = qlsave/(qlsave+qvsave)
         cfv = 1.0 - cfl
         go to 290
 280     cfv=1.
         cfl=0.
 290     vzv= .25*(vvz(jj,i) + vvz(jj,iym) + vvz(j,i) + vvz(j,iym))
         vzl= .25*(vlz(jj,i) + vlz(jj,iym) + vlz(j,i) + vlz(j,iym))
         vxv= .25*(vvx(jj,iym)+vvx(jj,i)+vvx(jj,iymxp)+vvx(jj,ixp))
         vxl= .25*(vlx(jj,iym)+vlx(jj,i)+vlx(jj,iymxp)+vlx(jj,ixp))
         vyv=vvy(jj,i)
         vyl=vly(jj,i)
         iwf=iwfz(j)
         rdz2=2./dz(jj)
c   parallel
        vl=vxl*sinph*costh + vyl*sinph*sinth + vzl*cosph
        vv=vxv*sinph*costh + vyv*sinph*sinth + vzv*cosph
         call fwall(fwvl,fwll,hdza,1.,cfv,cfl,pa,alpa,rova,rola,
     1              vv,vl,visva,visla,rdz2,iwf,i,jj)
       if(sinph.lt.1.0e-8) sinph=0.
       if(sinth.lt.1.0e-08) sinth=0.
         fwl=fwll*sinph*sinph*sinth*sinth
         fwv=fwvl*sinph*sinph*sinth*sinth
         fwlex=fwll*sinph*sinth*(sinph*costh*vxl + cosph*vzl)
         fwvex=fwvl*sinph*sinth*(sinph*costh*vxv + cosph*vzv)
c   perpendular 1
         vl=vxl*cosph*costh + vyl*cosph*sinth - vzl*sinph
         vv=vxv*cosph*costh + vyv*cosph*sinth - vzv*sinph
         rdx2=2./dx(i)
         iwf=iwft+30
         if(iwfz(j).eq.0)  iwf = 30
         call fwall(fwvl,fwll,hdta,velx,cfv,cfl,pa,alpa,rova,rola,
     1              vv,vl,visva,visla,rdx2,iwf,i,jj)
         fwl=fwl + fwll*sinth*sinth*cosph*cosph
         fwv=fwv + fwvl*sinth*sinth*cosph*cosph
         fwlex=fwlex + fwll*sinth*cosph*(cosph*costh*vxl - sinph*vzl)
```

```
          fwvex=fwvex + fwv1*sinth*cosph*(cosph*costh*vxv - sinph*vzv)
c    perpendular 2
          vl= -vxl*sinth + vyl*costh
          vv= -vxv*sinth + vyv*costh
          call fwall(fwv1,fwl1,hdta,velx,cfv,cfl,pa,alpa,rova,rola,
     1               vv,vl,visva,visla,rdy2,iwf,i,jj)
          fwl=fwl + fwl1*costh*costh
          fwv=fwv + fwv1*costh*costh
          fwlex=fwlex - fwl1*costh*sinth*vxl
          fwvex=fwvex - fwv1*costh*sinth*vxv
          vr = abs(vyv-vyl)
        fiv=1.0e+10
        fil=fiv
        if(j.ge.jhem) go to 12
          call finter(fiv,fil,alpa,rova,rola,vr,visva,visla,hdta,ifintr)
12      continue
          call cvecy(tcvy,dx,dy,dz,icc,iwfz,vvx,vvy,vvz,jj,i,
     1      ixm,ixp,iym,iyp,nzp,nzp2)
          call cvecy(tcly,dx,dy,dz,icc,iwfz,vlx,vly,vlz,jj,i,
     1      ixm,ixp,iym,iyp,nzp,nzp2)
c       if(((i.eq.1).or.(i.eq.2)).and.(j.eq.2))tcvy=0.0
c       if(((i.eq.1).or.(i.eq.2)).and.(j.eq.2))tcly=0.0
c       if(((i.eq.7).or.(i.eq.8)).and.(j.eq.2))tcvy=0.0
c       if(((i.eq.7).or.(i.eq.8)).and.(j.eq.2))tcly=0.0
          fexvy=(rdelt*vvy(jj,i)-tcvy+gravy)*arv - fwvex
          fexly=(rdelt*vly(jj,i)-tcly+gravy)*arl - fwlex
          aa=rdelt*arv+fiv+fwv
          bb=rdelt*arl+fil+fwl
          rd=1./(aa*bb-fil*fiv)
          rvdy=rdy2*alpa
          rldy=rdy2*(1.-alpa)
          cpvy(j,i)=-rd*(bb*rvdy+fiv*rldy)
          cply(j,i)=-rd*(aa*rldy+fil*rvdy)
          fvy(j,i)=rd*(bb*fexvy+fiv*fexly)
          fly(j,i)=rd*(aa*fexly+fil*fexvy)
  300     continue
  400   continue
c
c   z direction
c
        do 500 j=jstart,jstop
          if(arz(j,i).eq.0.0) go to 500
          jp=j+1
          rdz=1./(dz(j)+dz(jp))
          rdz2=2.*rdz
          hdza=rdz*(hdz(j,i)*dz(j) + hdz(jp,i)*dz(jp))
          hdta=rdz*(hdt(j,i)*dz(j) + hdt(jp,i)*dz(jp))
          sinph=rdz*(sin(ph(j,i))*dz(j) + sin(ph(jp,i))*dz(jp))
          sinth=rdz*(sin(th(j,i))*dz(j) + sin(th(jp,i))*dz(jp))
          cosph=rdz*(cos(ph(j,i))*dz(j) + cos(ph(jp,i))*dz(jp))
          costh=rdz*(cos(th(j,i))*dz(j) + cos(th(jp,i))*dz(jp))
          pa=rdz*(p(j,i)*dz(j)+p(jp,i)*dz(jp))
          alpa=rdz*(alp(j,i)*dz(j)+alp(jp,i)*dz(jp))
          rova=rdz*(rov(j,i)*dz(j)+rov(jp,i)*dz(jp))
```

```
      rola=rdz*(rol(j,i)*dz(j)+rol(jp,i)*dz(jp))
      arv=alpa*rova
      arl=(1.-alpa)*rola
      visva=rdz*(visv(j,i)*dz(j)+visv(jp,i)*dz(jp))
      visla=rdz*(visl(j,i)*dz(j)+visl(jp,i)*dz(jp))
c--set liquid and vapor wall contact fractions for fwall
      cfv=0.
      cfl=1.
      if (iht.eq.0) go to 490
      jdn = idon(jp,j,vlz(j,i)) - 1
      if ((jdn.eq.0).or.(jdn.eq.nzp)) go to 490
      itest=0
      do 420 n=1,4
      nn = irc(n,i)
      if(nn.eq.0) go to 430
      if(itest.lt.ihtr(jdn,nn)) itest = ihtr(jdn,nn)
 420  continue
 430  continue
      if(itest -itrns) 440,450,480
 440  go to 490
 450  qlsave = 0.0
      qvsave = 0.0
      do 460 n=1,4
      nn = irc(n,i)
      if(nn.eq.0) go to 470
      qlsave = qlsave + ql(jdn+1,nn)*fracp(nn)
      qvsave = qvsave + qv(jdn+1,nn)*fracp(nn)
 460  continue
 470  cfl = qlsave/(qlsave+qvsave)
      go to 490
 480  cfv=1.
      cfl=0.
 490  iwf = iwfz(j)
      vxv = .25*(vvx(j,i) + vvx(jp,i) + vvx(j,ixp) + vvx(jp,ixp))
      vxl = .25*(vlx(j,i) + vlx(jp,i) +vlx(j,ixp) + vlx(jp,ixp))
      vyv =.25*(vvy(j,i) + vvy(jp,i) + vvy(j,iyp) + vvy(jp,iyp))
      vyl = .25*(vly(j,i)+ vly(jp,i) + vly(j,iyp) +vly(jp,iyp))
      vzv = vvz(j,i)
      vzl= vlz(j,i)
c   parallel
      vv=vxv*sinph*costh + vyv*sinph*sinth + vzv*cosph
      vl=vxl*sinph*costh + vyl*sinph*sinth + vzl*cosph
      call fwall(fwvl,fwll,hdza,1.,cfv,cfl,pa,alpa,rova,rola,
     1           vv,vl,visva,visla,rdz2,iwf,i,j)
      fwl=fwll*cosph*cosph
      fwv=fwvl*cosph*cosph
      fwlex=fwll*cosph*sinph*(costh*vxl + sinth*vyl)
      fwvex=fwvl*cosph*sinph*(costh*vxv + sinth*vyv)
c   perpendular 1
      vl=vxl*cosph*costh + vyl*cosph*sinth - vzl*sinph
      vv=vxv*cosph*costh + vyv*cosph*sinth - vzv*sinph
      rdx2=2./dx(i)
      iwf = iwft + 30
      if(iwfz(j) .eq. 0)  iwf = 30
```

```
          call  fwall(fwv1,fwl1,hdta,velx,cfv,cfl,pa,alpa,rova,rola,
     1          vv,vl,visva,visla,rdx2,iwf,i,j)
         fwl=fwl + fwl1*sinph*sinph
         fwv=fwv + fwv1*sinph*sinph
         fwlex=fwlex - fwl1*sinph*cosph*(costh*vxl + sinth*vyl)
         fwvex=fwvex - fwv1*sinph*cosph*(costh*vxv + sinth*vyv)
          vr = abs(vzv-vzl)
       fiv=1.0e+10
       fil=fiv
       if(j.ge.jhem) go to 14
       call finter(fiv,fil,alpa,rova,rola,vr,visva,visla,hdza,ifintr)
14     continue
          call cvecz(tcvz,dx,dy,dz,icc,vvx,vvy,vvz,j,i,
     1      ixm,ixp,iym,iyp,nzp,nzp2)
          call cvecz(tclz,dx,dy,dz,icc,vlx,vly,vlz,j,i,
     1      ixm,ixp,iym,iyp,nzp,nzp2)
c      if(j.eq.2)tcvz = 0.0
c      if(j.eq.2)tclz = 0.0
c
c   calculate turbulent mixing pressure gradient
c
          dpdztv=0.0
          dpdztl=0.0
          if(itam.eq.0.or.imixm.eq.0) go to 496
          do 495 ij=1,4
          ic=icc(ij,i)
  495     continue
          rarz=1./arz(j,i)
          dpdztv=dpdztv*rarz
          dpdztl=dpdztl*rarz
  496     continue
          fexvz=(rdelt*vvz(j,i)-tcvz+gravz)*arv-dpdztv - fwvex
          fexlz=(rdelt*vlz(j,i)-tclz+gravz)*arl-dpdztl - fwlex
          aa=rdelt*arv+fiv+fwv
          bb=rdelt*arl+fil+fwl
          rd=1./(aa*bb-fil*fiv)
          rvdz=rdz2*alpa
          rldz=rdz2*(1.-alpa)
          cpvz(j,i)=-rd*(bb*rvdz+fiv*rldz)
          cplz(j,i)=-rd*(aa*rldz+fil*rvdz)
          fvz(j,i)=rd*(bb*fexvz+fiv*fexlz)
          flz(j,i)=rd*(aa*fexlz+fil*fexvz)
  500     continue
 1000   continue
c
c  set top and bottom velocity boundary conditions
c
      if(ibb.eq.0) go to 1200
      do 1100 i=1,nc
        fvz(1,i)=vvz(1,i)
        flz(1,i)=vlz(1,i)
 1100   continue
 1200 continue
      if(itb.eq.0) go to 1400
```

```
      do 1300 i=1,nc
        fvz(nzp,i)=vvz(nzp,i)
        flz(nzp,i)=vlz(nzp,i)
 1300   continue
 1400 continue
      return
      end
      subroutine cvecx(tcx,dx,dy,dz,icc,iwfz,vx,vy,vz,izp,ic,
     1    ixm,ixp,iym,iyp,nzp,nzp2)
c
c  calculates (explicit) convective terms in x direction momentum
c   equations       (one call for each phase)
c
c  includes axial funnel effect where iwfz >= 20
c
      dimension dx(1),dy(1),dz(1),icc(4,1),iwfz(1),
     1          vx(nzp2,1),vy(nzp2,1),vz(nzp,1)
c
      ixmyp = icc(4,ixm)
c
      vy1 = 0.5*(vy(izp  ,ixm  )+vy(izp  ,ic ))
      vy4 = 0.5*(vy(izp  ,ixmyp)+vy(izp  ,iyp))
      vz5 = 0.5*(vz(izp-1,ixm  )+vz(izp-1,ic ))
      vz6 = 0.5*(vz(izp  ,ixm  )+vz(izp  ,ic ))
c
      vxa = vx(izp,ic)
      vya = 0.5*(vy1+vy4)
      vza = 0.5*(vz5+vz6)
c
      dvxxm = (vx(izp  ,ic )-vx(izp  ,ixm))/ dx(ixm)
      dvxxp = (vx(izp  ,ixp)-vx(izp  ,ic ))/ dx(ic )
      dvxym = (vx(izp  ,ic )-vx(izp  ,iym))/(dy(ic )+dy(iym  ))*2.0
      dvxyp = (vx(izp  ,iyp)-vx(izp  ,ic ))/(dy(iyp)+dy(ic   ))*2.0
      vxinz =                vx(izp-1,ic )
      if (iwfz(izp-1).ge.20) vxinz=0.
      dvxzm = (vx(izp  ,ic )-vxinz        )/(dz(izp)+dz(izp-1))*2.0
      vxinz =  vx(izp+1,ic )
      if (iwfz(izp  ).ge.20) vxinz=0.
      dvxzp = (vxinz        -vx(izp  ,ic ))/(dz(izp)+dz(izp+1))*2.0
c
      tcxxd = 0.5*((vxa+abs(vxa))*dvxxm+(vxa-abs(vxa))*dvxxp)
      tcxyd = 0.5*((vya+abs(vya))*dvxym+(vya-abs(vya))*dvxyp)
      tcxzd = 0.5*((vza+abs(vza))*dvxzm+(vza-abs(vza))*dvxzp)
c
      tcx = tcxxd+tcxyd+tcxzd
      return
      end
      subroutine cvecy(tcy,dx,dy,dz,icc,iwfz,vx,vy,vz,izp,ic,
     1    ixm,ixp,iym,iyp,nzp,nzp2)
c
c  calculates (explicit) convective terms in y direction momentum
c   equations       (one call for each phase)
c
c  includes axial funnel effect where iwfz >= 20
```

```
c
      dimension dx(1),dy(1),dz(1),icc(4,1),iwfz(1),
     1          vx(nzp2,1),vy(nzp2,1),vz(nzp,1)
c
      iymxp = icc(3,iym)
c
      vx2 = 0.5*(vx(izp  ,iym  )+vx(izp  ,ic ))
      vx3 = 0.5*(vx(izp  ,iymxp)+vx(izp  ,ixp))
      vz5 = 0.5*(vz(izp-1,iym  )+vz(izp-1,ic ))
      vz6 = 0.5*(vz(izp  ,iym  )+vz(izp  ,ic ))
c
      vxa = 0.5*(vx2+vx3)
      vya = vy(izp,ic)
      vza = 0.5*(vz5+vz6)
c
      dvyxm = (vy(izp  ,ic )-vy(izp  ,ixm))/(dx(ic )+dx(ixm  ))*2.0
      dvyxp = (vy(izp  ,ixp)-vy(izp  ,ic ))/(dx(ixp)+dx(ic   ))*2.0
      dvyym = (vy(izp  ,ic )-vy(izp  ,iym))/ dy(iym)
      dvyyp = (vy(izp  ,iyp)-vy(izp  ,ic ))/ dy(ic )
      vyinz =              vy(izp-1,ic )
      if (iwfz(izp-1).ge.20) vyinz=0.
      dvyzm = (vy(izp  ,ic )-vyinz        )/(dz(izp)+dz(izp-1))*2.0
      vyinz =  vy(izp+1,ic )
      if (iwfz(izp  ).ge.20) vyinz=0.
      dvyzp = (vyinz        -vy(izp  ,ic ))/(dz(izp)+dz(izp+1))*2.0
c
      tcyxd = 0.5*((vxa+abs(vxa))*dvyxm+(vxa-abs(vxa))*dvyxp)
      tcyyd = 0.5*((vya+abs(vya))*dvyym+(vya-abs(vya))*dvyyp)
      tcyzd = 0.5*((vza+abs(vza))*dvyzm+(vza-abs(vza))*dvyzp)
c
      tcy = tcyxd+tcyyd+tcyzd
      return
      end
      subroutine cvecz(tcz,dx,dy,dz,icc,vx,vy,vz,iz,ic,
     1    ixm,ixp,iym,iyp,nzp,nzp2)
c
c  calculates (explicit) convective terms in z direction momentum
c   equations       (one call for each phase)
c
      dimension dx(1),dy(1),dz(1),icc(4,1),
     1          vx(nzp2,1),vy(nzp2,1),vz(nzp,1)
c
      vx2 = 0.5*(vx(iz,ic )+vx(iz+1,ic ))
      vx3 = 0.5*(vx(iz,ixp)+vx(iz+1,ixp))
      vy1 = 0.5*(vy(iz,ic )+vy(iz+1,ic ))
      vy4 = 0.5*(vy(iz,iyp)+vy(iz+1,iyp))
c
      vxa = 0.5*(vx2+vx3)
      vya = 0.5*(vy1+vy4)
      vza = vz(iz,ic)
c
      dvzxm = (vz(iz,ic )-vz(iz,ixm))/(dx(ic )+dx(ixm))*2.0
      dvzxp = (vz(iz,ixp)-vz(iz,ic ))/(dx(ixp)+dx(ic ))*2.0
      dvzym = (vz(iz,ic )-vz(iz,iym))/(dy(ic )+dy(iym))*2.0
```

```
      dvzyp =  (vz(iz,iyp)-vz(iz,ic ))/(dy(iyp)+dy(ic ))*2.0
c     dvzzm =  vz(iz,ic)/dz(iz)
      dvzzm = 0.0
      if(iz.ne.  1) dvzzm =  (vz(iz,ic)-vz(iz-1,ic))/dz(iz  )
c     dvzzp = -vz(iz,ic)/dz(iz)
      dvzzp = 0.0
      if(iz.ne.nzp) dvzzp =  (vz(iz+1,ic)-vz(iz,ic))/dz(iz+1)
c
      tczxd = 0.5*((vxa+abs(vxa))*dvzxm+(vxa-abs(vxa))*dvzxp)
      tczyd = 0.5*((vya+abs(vya))*dvzym+(vya-abs(vya))*dvzyp)
      tczzd = 0.5*((vza+abs(vza))*dvzzm+(vza-abs(vza))*dvzzp)
c
      tcz = tczxd+tczyd+tczzd
      return
      end
      subroutine jacob(icc,icr,irc,dz,vol,rov,rol,alp,p,ev,el,
     1  rovn,roln,alpn,pn,evn,eln,tvn,tln,tw,arx,ary,arz,
     2  cpvx,cplx,cpvy,cply,cpvz,cplz,fvx,flx,fvy,fly,
     3  fvz,flz,rn,qv,hvfc,hlnb,hlfc,
     4  ajm1,ajm2,cpa,cptv,cptl,rhs,vvx,vvy,vvz,vlx,vly,vlz,dtw,
     5  ihtr,visl, tv,hdz,aht,
     6  m1,m2,m3,m4 ,m5)
c
c  obtain Jacobian matrix by linearizing mass and energy equations,
c   substituting in the momentum equations to eliminate velocities,
c   reduce to pressure problem
c
      double precision ca,c,cp,f,t,rd
      common /ic/ nstep,nitmax,nitno,iitmax,iitot,iic,kred,nm,ntc,
     1   nc,nrods,nz,nr,ncp,nzp,nzp2, iflash,itb,ibb,icpu,iwft,ivec,
     2   nodes,ndm1,ncf,ncc,ng,iht, iss,iqss,itam,ires,idump,ntitle,
     3   itwmax,jtwmax,itrmax,jtrmax,imchfr,jmchfr,ichf,ifintr,ierr,lerr
     4    ,imixm,imixe,iafm,itfm,igfm,nktb,jhem
      logical lerr
      common /rc/ delt,rdelt,errn,epsn,erri,epsi,dtmin,dtmax,tend,
     1   dtsp,dtlp,rtnsp,rtnlp,gravx,gravy,gravz,rtime,velx,
     2   q,q0,twmax,trmax,amchfr
      dimension icc(4,1),icr(1),irc(4,1),rov(m1,1),rol(m1,1),
     1  alp(m1,1),p(m1,1),ev(m1,1),el(m1,1),
     2  rovn(m1,1),roln(m1,1),alpn(m1,1),
     3  pn(m1,1),evn(m1,1),eln(m1,1),tvn(m1,1),tln(m1,1),
     4  tw(m1,1),dtw(m1,1),arx(m3,1),ary(m3,1),arz(m2,1)
      dimension qv(m1,1),hvfc(m1,1),hlnb(m1,1),hlfc(m1,1)
      dimension dz(1),vol(m3,1),rn(1),
     1  vvx(m1,1),vvy(m1,1),vvz(m2,1),vlx(m1,1),vly(m1,1),vlz(m2,1)
      dimension cpvx(m3,1),cplx(m3,1),cpvy(m3,1),cply(m3,1),
     1  cpvz(m2,1),cplz(m2,1),fvx(m3,1),flx(m3,1),
     2  fvy(m3,1),fly(m3,1),fvz(m2,1),flz(m2,1)
      dimension ajm1(3,m3,1),ajm2(4,m3,1),cpa(6,m3,1),
     1  cptv(6,m3,1),cptl(6,m3,1),rhs(m3,m4,4)
      dimension ihtr(m3,1),visl(m1,1),hdz(m1,1),tv(m1,1)
c
      dimension arb(6),vvb(6),vlb(6),cc(4,6),cp(4,11),f(4),
     1  tc(4),c(4,4),ija(6),ca(4,11)
```

```
      dimension aht(m5,m1,1)
      equivalence (ca(1,1),c(1,1)),(ca(1,5),cp(1,1)),(ca(1,11),f(1))
      data ija/0,0,0,0,-1,1/
      data pi2/6.2831853/
       data  h10,h11,h12,h13,h14,h15/5.7474718e5,2.0920624e-1,
     1    -2.8051070e-8,2.3809828e-15,-1.0042660e-22,1.6586960e-30/
       data  hv0,hv1,hv2,hv3,hv4/2.7396234e6,3.758844e-2,
     1       -7.1639909e-9,4.2002319e-16,-9.8507521e-24 /
c
      gam=0.0
      dgdp=0.0
      dgda=0.0
      dgdtv=0.0
      dgdtl=0.0
c
c
      do 1000 i=1,nc
c
        iym=icc(1,i)
        ixm=icc(2,i)
        ixp=icc(3,i)
        iyp=icc(4,i)
c
        do 1000 j=1,nz
          jj=j+1
c
c  determine heat transfer area per unit volume
c
          rvol=1./vol(j,i)
          area=0.
          do 340 ktb=1,nktb
340       area=area+aht(ktb,jj,i)
          areah=area*rvol
c
c  set donor cell quantities for divergence terms
c
          arb(1)=ary(j,i)
          arb(2)=arx(j,i)
          arb(3)=arx(j,ixp)
          arb(4)=ary(j,iyp)
          arb(5)=arz(j,i)
          arb(6)=arz(jj,i)
          vvb(1)=vvy(jj,i)
          vlb(1)=vly(jj,i)
          vvb(2)=vvx(jj,i)
          vlb(2)=vlx(jj,i)
          vvb(3)=vvx(jj,ixp)
          vlb(3)=vlx(jj,ixp)
          vvb(4)=vvy(jj,iyp)
          vlb(4)=vly(jj,iyp)
          vvb(5)=vvz(j,i)
          vlb(5)=vlz(j,i)
          vvb(6)=vvz(jj,i)
          vlb(6)=vlz(jj,i)
```

```
c
          do 150 k=1,6
            ja=jj+ija(k)
            ia=i
            if(k.le.4)ia=icc(k,i)
            rva=rvol*arb(k)
            wtv=0.5+sign(0.5,vvb(k))
            wtl=0.5+sign(0.5,vlb(k))
c             for sides 1,2,5, positive velocity means incoming
            if(k.eq.3.or.k.eq.4.or.k.eq.6) go to 110
c             for sides 3,4,6, positive velocity means outgoing
            wtv=1.-wtv
            wtl=1.-wtl
  110       cc(1,k)=rva*(wtv*alp(jj,i)*rov(jj,i)+(1.-wtv)*alp(ja,ia)
     1        *rov(ja,ia))
            cc(2,k)=rva*(wtl*(1.-alp(jj,i))*rol(jj,i)+(1.-wtl)*
     1        (1.-alp(ja,ia))*rol(ja,ia))
            cc(3,k)=rva*(wtv*alp(jj,i)*(rov(jj,i)*ev(jj,i)+p(jj,i))
     1        +(1.-wtv)*alp(ja,ia)*(rov(ja,ia)*ev(ja,ia)+p(jj,i)))
            cc(4,k)=rva*(wtl*(1.-alp(jj,i))*(rol(jj,i)*el(jj,i)+p(jj,i))
     1        +(1.-wtl)*(1.-alp(ja,ia))*(rol(ja,ia)*el(ja,ia)+p(jj,i)))
  150       continue
c
c  eliminate surrounding liquid velocities for pressures
c
          do 200 l=2,4,2
            cp(l,1)  =  cc(l,1)*cply(j,i)
            cp(l,2)  =  cc(l,2)*cplx(j,i)
            cp(l,3)  =  cc(l,3)*cplx(j,ixp)
            cp(l,4)  =  cc(l,4)*cply(j,iyp)
            cp(l,5)  =  cc(l,5)*cplz(j,i)
            cp(l,6)  =  cc(l,6)*cplz(j+1,i)
           f(l) =
     +        cc(l,1)*(fly(j,i)    +cply(j,i)   *(pn(jj,i)-pn(jj,iym)))  -
     -        cc(l,4)*(fly(j,iyp)  +cply(j,iyp) *(pn(jj,iyp)-pn(jj,i)))  +
     +        cc(l,2)*(flx(j,i)    +cplx(j,i)   *(pn(jj,i)-pn(jj,ixm)))  -
     -        cc(l,3)*(flx(j,ixp)  +cplx(j,ixp) *(pn(jj,ixp)-pn(jj,i)))  +
     +        cc(l,5)*(flz(j,i)    +cplz(j,i)   *(pn(jj,i)-pn(jj-1,i)))  -
     -        cc(l,6)*(flz(j+1,i)  +cplz(j+1,i) *(pn(jj+1,i)-pn(jj,i)))
  200       continue
c
c  eliminate surrounding vapor velocities for pressures
c
          do 201 l=1,3,2
            cp(l,1)  =  cc(l,1)*cpvy(j,i)
            cp(l,2)  =  cc(l,2)*cpvx(j,i)
            cp(l,3)  =  cc(l,3)*cpvx(j,ixp)
            cp(l,4)  =  cc(l,4)*cpvy(j,iyp)
            cp(l,5)  =  cc(l,5)*cpvz(j,i)
            cp(l,6)  =  cc(l,6)*cpvz(j+1,i)
            f(l) =
     +        cc(l,1)*(fvy(j,i)    +cpvy(j,i)   *(pn(jj,i)-pn(jj,iym)))  -
     -        cc(l,4)*(fvy(j,iyp)  +cpvy(j,iyp) *(pn(jj,iyp)-pn(jj,i)))  +
     +        cc(l,2)*(fvx(j,i)    +cpvx(j,i)   *(pn(jj,i)-pn(jj,ixm)))  -
```

```
     -        cc(1,3)*(fvx(j,ixp)  +cpvx(j,ixp)*(pn(jj,ixp)-pn(jj,i))) +
     +        cc(1,5)*(fvz(j,i)    +cpvz(j,i)   *(pn(jj,i)-pn(jj-1,i))) -
     -        cc(1,6)*(fvz(j+1,i)  +cpvz(j+1,i)*(pn(jj+1,i)-pn(jj,i)))
  201       continue
c
          tc(1)=cp(1,1)+cp(1,2)+cp(1,3)+cp(1,4)+cp(1,5)+cp(1,6)
          tc(2)=cp(2,1)+cp(2,2)+cp(2,3)+cp(2,4)+cp(2,5)+cp(2,6)
          tc(3)=cp(3,1)+cp(3,2)+cp(3,3)+cp(3,4)+cp(3,5)+cp(3,6)
          tc(4)=cp(4,1)+cp(4,2)+cp(4,3)+cp(4,4)+cp(4,5)+cp(4,6)
c
c  obtain thermodynamic derivatives
c
          call state(pn(jj,i),tvn(jj,i),tln(jj,i),d1,d2,d3,d4,tsatn,
     1 hvsn,hlsn,dtsdp,deldp,devdp,deldt,devdt,drldp,drvdp,drldt,drvdt,
     2      1,ierr )
      hvs=hv0 + p(jj,i)*(hv1 + p(jj,i)*(hv2 + p(jj,i)*(hv3 + p(jj,i)
     1     *hv4)))
      hls = hl0 + p(jj,i)*(hl1 + p(jj,i)*(hl2 + p(jj,i)*(hl3 +
     1      p(jj,i)*(hl4 + p(jj,i)*hl5))))
      dhlsdp = hl1 + pn(jj,i)*(2.*hl2 + pn(jj,i)*(3.*hl3 + pn(jj,i)
     1         *(4.*hl4 + pn(jj,i)*5.*hl5)))
      dhvsdp = hv1 + pn(jj,i)*(2.*hv2 + pn(jj,i)*(3.*hv3 + pn(jj,i)
     1         *4.*hv4))
c
c  heat transfer terms in energy equation
      qwl=0.
      qwv=0.
      dqwdtv=0.
      dqwdtl=0.
      dqwdp=0.
      atotal = 0.
      itest = 0
      do 205 jr=1,4
      ir=irc(jr,i)
      if(ir.eq.0) go to 206
      qwv= qwv+ areah*(qv(jj,ir)+hvfc(jj,ir)*(tw(jj,ir)-tvn(jj,i)))
      qwl=qwl+areah*(qv(jj,ir+nrods)+hlfc(jj,ir)*(tw(jj,ir)-tln(jj,i))
     1                + hlnb(jj,ir)*(tw(jj,ir)-tsatn))
c    Derivatives of wall heat flux are negatives of actual derivatives
      dqwdtv= dqwdtv+ areah*hvfc(jj,ir)*(1.-hvfc(jj,ir)*dtw(jj,ir))
      dqwdtl= dqwdtl+ areah*hlfc(jj,ir)*(1.-(hlfc(jj,ir)+hlnb(jj,ir))*
     1                      dtw(jj,ir))
      dqwdp= dqwdp+areah*hlnb(jj,ir)*dtsdp*(1.-(hlfc(jj,ir)+
     1 hlnb(jj,ir))*dtw(jj,ir))
      atotal = atotal + areah
      if(itest.lt.ihtr(j,ir)) itest = ihtr(j,ir)
 205   continue
 206    continue
c
c   turbulent mixing mass and energy terms
c
      vv=0.5*(vvz(j,i)+vvz(jj,i))
      vl=0.5*(vlz(j,i)+vlz(jj,i))
      qvtm=0.
```

```
      qltm=0.
      wvtm=0.
      wltm=0.
      dwvdp=0.
      dwvda=0.
      dwvdtv=0.
      dwldp=0.
      dwlda=0.
      dwldtl=0.
      dqvdp=0.
      dqvda=0.
      dqvdtv=0.
      dqldp=0.
      dqlda=0.
      dqldtl=0.
      if(itam.eq.0.or.imixe.eq.0) go to 208
      rarz=1./arz(j,i)
      arv=alp(jj,i)*rov(jj,i)
      arl=(1.-alp(jj,i))*rol(jj,i)
      do 207 ij=1,4
      ic=icc(ij,i)
  207 continue
  208 continue
      if (iflash.eq.0.and.itest.lt.5)
     +      call gamma (gam, dgdp,dgda,dgdtv,dgdtl, alp(jj,i),
     1        alpn(jj,i),tvn(jj,i),tln(jj,i),rov(jj,i),rol(jj,i),
     2        tsatn,dtsdp )
      if (iflash.eq.2.and.itest.lt.5)
     +      call gamsub (gam, dgdp,dgda,dgdtv,dgdtl, p(jj,i),
     1       alp(jj,i),tv(jj,i),tln(jj,i),tsatn,pn(jj,i),alpn(jj,i),
     2       tvn(jj,i),tln(jj,i),tsatn,dtsdp,deldt, rov(jj,i),
     3       rol(jj,i),vv,vl,hdz(jj,i),atotal,visl(jj,i),hvs,hls,
     4       qwl,qwv,dqwdtl,dqwdp,
     5       itest )
c
c       Saha post-CHF vapor generation rate corr.
c
       if (iflash.ne.1.and.itest.ge.5)
     +   call gamsup (gam,dgdp,dgda,dgdtv,dgdtl,p(jj,i),
     1   rov(jj,i),tv(jj,i),tln(jj,i),vv,alpn(jj,i),tvn(jj,i),
     2   tsatn,hvs,hls,dtsdp,hdz(jj,i))
        if((iflash.eq.0.and.itest.lt.5).or.iflash.eq.1)
     +   call qinter(qi,dqdp,dqda,dqdtv,dqdtl,alpn(jj,i),pn(jj,i),
     1   tvn(jj,i),tln(jj,i),tsatn,dtsdp,alp(jj,i),p(jj,i),
     2   rov(jj,i),rol(jj,i) )
       if (iflash.eq.2.and.itest.lt.5)
     +   call qisub(qi,dqdp,dqda,dqdtv,dqdtl,hvsn,gam,dgdp,dgda,dgdtv,
     1   dgdtl,dhvsdp,tvn(jj,i),tsatn,dtsdp)
c
c       calculate post-CHF interfacial heat transfer rate
c
       if (iflash.ne.1.and.itest.ge.5)
     +   call qisup(qi,dqdp,dqda,dqdtv,dqdtl,gam,dgdp,dgda,dgdtv,
     1   dgdtl,hlsn,dhlsdp,tsatn,tln(jj,i),dtsdp)
```

```
c
c      vapor mass equation
c
          c(1,1) = alpn(jj,i)*drvdp*rdelt - dgdp +dwvdp
          c(1,2) = rovn(jj,i)*rdelt - dgda +dwvda
          c(1,3) = alpn(jj,i)*drvdt*rdelt - dgdtv +dwvdtv
          c(1,4) = -dgdtl
          f(1) = f(1) - rdelt*( alpn(jj,i)*rovn(jj,i) -
     -               alp(jj,i)*rov(jj,i)  )   + gam - wvtm
          c(1,1) = c(1,1) - tc(1)
c
c   liquid mass equation
c
          alpnl = 1.- alpn(jj,i)
          c(2,1) = alpnl*drldp*rdelt + dgdp + dwldp
          c(2,2) = -roln(jj,i)*rdelt + dgda +dwlda
          c(2,3) = dgdtv
          c(2,4) = alpnl*drldt*rdelt + dgdtl +dwldtl
          f(2) = f(2) - rdelt*( alpnl*roln(jj,i) -
     -                  (1.-alp(jj,i))*rol(jj,i)  )  - gam - wltm
          c(2,1) = c(2,1) - tc(2)
c
c   vapor energy equation
c
          c(3,1) = alpn(jj,i) *(rovn(jj,i)*devdp + evn(jj,i)*drvdp) *
     *                            rdelt - dqdp +dqvdp
          c(3,2) = (rovn(jj,i)*evn(jj,i) + p(jj,i)) *rdelt -dqda+dqvda
          c(3,3) = alpn(jj,i)* (rovn(jj,i)*devdt + evn(jj,i)*drvdt) *
     *               rdelt - dqdtv +
     +      dqwdtv + dqvdtv
          c(3,4) = -dqdtl
          f(3) = f(3) - rdelt*( alpn(jj,i)*rovn(jj,i)*evn(jj,i) -
     -                        alp(jj,i)*rov(jj,i)*ev(jj,i) +
     +            p(jj,i) * (alpn(jj,i) - alp(jj,i))  )  + qi +
     +         qwv - qvtm
          c(3,1) = c(3,1) - tc(3)
c
c   liquid energy equation
c
          c(4,1) = alpnl* (roln(jj,i)*deldp + eln(jj,i)*drldp) *
     *                     rdelt + dqdp +
     +      dqwdp + dqldp
          c(4,2) = (-roln(jj,i)*eln(jj,i) - p(jj,i))* rdelt +dqda+dqlda
          c(4,3) = dqdtv
          c(4,4) = alpnl* (roln(jj,i)*deldt + eln(jj,i)*drldt)  *
     *                     rdelt + dqdtl +
     +      dqwdtl + dqldtl
c--note: above assumes hvfc=0 if hlfc+hlnb>0 and vice versa
          f(4) = f(4) - rdelt*( alpnl*roln(jj,i)*eln(jj,i) -
     -                      (1.-alp(jj,i))*rol(jj,i)*el(jj,i) +
     +            p(jj,i) * (alp(jj,i) - alpn(jj,i))  )  - qi +
     +        qwl - qltm
c
          c(4,1) = c(4,1) - tc(4)
```

```
c
c  invert 4x4 to eliminate al, tv, tl, for pressure alone
c
          rd=1./(c(1,1)*c(2,2)-c(1,2)*c(2,1))
          do 210 k=3,11
            t=rd*(c(2,2)*ca(1,k)-c(1,2)*ca(2,k))
            ca(2,k)=rd*(-c(2,1)*ca(1,k)+c(1,1)*ca(2,k))
  210       ca(1,k)=t
          do 220 k=3,11
            ca(4,k)=ca(4,k)-c(4,1)*ca(1,k)-c(4,2)*ca(2,k)
  220       ca(3,k)=ca(3,k)-c(3,1)*ca(1,k)-c(3,2)*ca(2,k)
          rd=1./(c(3,3)*c(4,4)-c(4,3)*c(3,4))
          do 230 k=5,11
            t=rd*(c(4,4)*ca(3,k)-c(3,4)*ca(4,k))
            ca(4,k)=rd*(-c(4,3)*ca(3,k)+c(3,3)*ca(4,k))
  230       ca(3,k)=t
          do 240 k=5,11
            ca(1,k)=ca(1,k)-c(1,4)*ca(4,k)-c(1,3)*ca(3,k)
  240       ca(2,k)=ca(2,k)-c(2,4)*ca(4,k)-c(2,3)*ca(3,k)
c
c  set up jacobian matrix, store pressure coefficients
c
          ajm1(1,j,i)=1.0
          ajm1(2,j,i)=cp(1,5)
          ajm1(3,j,i)=cp(1,6)
          ajm2(1,j,i)=cp(1,1)
          ajm2(2,j,i)=cp(1,2)
          ajm2(3,j,i)=cp(1,3)
          ajm2(4,j,i)=cp(1,4)
          do 270 k=1,4
  270       rhs(j,i,k)=f(k)
          ddom = 0.
          do 300 k=1,6
            ddom = ddom + cp(1,k)
            cpa(k,j,i)=cp(2,k)
            cptv(k,j,i)=cp(3,k)
  300       cptl(k,j,i)=cp(4,k)
          if (ddom.lt.-1.) go to 903
c
 1000     continue
      return
c
c  error detected, pressure problem is not diagonally dominant
c
  903 ierr = 3
      lerr = .true.
      return
      end
      subroutine inner (dp,ajm1,ajm2,rhs,t,icc,nz,nc,nzp2,ncp,
     1                  epsi,initmx,initct,erri,pref)
      dimension dp(nzp2,ncp),ajm1(3,nz,nc),ajm2(4,nz,nc),rhs(nz,nc),
     1          t(nz),icc(4,nc)
c
c  solve the pressure problem, of the form
```

```
c          ([ajm1] + [ajm2]) * [dp] = [rhs]
c   by block Gauss-Seidel iteration
c
c     arguments:
c            dp = new pressure increment (for this Newton iteration)
c          ajm1 = tridiagonal portion of pressure coefficient matrix
c          ajm2 = remainder of pressure coefficient matrix
c           rhs = right hand side of equation
c             t = temporary storage (virtual rhs)
c           icc = indicator for channel coupling
c            nz = number of axial levels
c            nc = number of channels
c          nzp2 = number of axial levels including boundaries
c           ncp = number of channels plus one
c          epsi = convergence criterion
c        initmx = maximum number of inner iterations allowed
c        initct = actual number of inner iterations (count)
c          erri = inner iteration error
c
c     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
c
c  perform "lu" decomposition
c
      do 10 ic=1,nc
        ajm1(1, 1,ic) = 1.0/ajm1(1, 1,ic)
        ajm1(3, 1,ic) = ajm1(3, 1,ic)*ajm1(1, 1,ic)
c
        do 10 iz=2,nz
          ajm1(1,iz,ic) = 1.0/
     1       (ajm1(1,iz,ic) - ajm1(2,iz,ic)*ajm1(3,iz-1,ic))
          ajm1(3,iz,ic) = ajm1(3,iz,ic)*ajm1(1,iz,ic)
   10     continue
c
      initct = 0
c
c  enter iterative process
c
   20 continue
      erri = 0.0
      initct = initct+1
c
c  form virtual rhs and perform forward elimination
c
      do 40 ic=1,nc
        iym = icc(1,ic)
        ixm = icc(2,ic)
        ixp = icc(3,ic)
        iyp = icc(4,ic)
c
        t( 1) = -ajm2(1, 1,ic)*dp(  2,iym)-ajm2(2, 1,ic)*dp(  2,ixm)
     1          -ajm2(3, 1,ic)*dp(  2,ixp)-ajm2(4, 1,ic)*dp(  2,iyp)
     2          +rhs( 1,ic)               -ajm1(2, 1,ic)*dp(  1, ic)
        t( 1) = t( 1)*ajm1(1, 1,ic)
c
```

```
      do 30 iz=2,nz
        izp = iz+1
        t(iz) = -ajm2(1,iz,ic)*dp(izp,iym)-ajm2(2,iz,ic)*dp(izp,ixm)
     1          -ajm2(3,iz,ic)*dp(izp,ixp)-ajm2(4,iz,ic)*dp(izp,iyp)
     2          +rhs(iz,ic)
        t(iz) = (t(iz)-ajm1(2,iz,ic)*t(iz-1))*ajm1(1,iz,ic)
   30   continue
c
c  perform backward substitution and compute maximum error
c
      do 40 iz=1,nz
        jz = nz+1-iz
        jzp  = jz+1
        dpold = dp(jzp,ic)
        dp(jzp,ic) = t(jz)-ajm1(3,jz,ic)*dp(jzp+1,ic)
        erri = amax1(erri, abs(dpold-dp(jzp,ic)) )
   40   continue
c
     erri = erri / pref
     if(erri.le.epsi.or.initct.ge.initmx) return
     go to 20
     end
     subroutine update(idown,icc,irc,pn,alpn,tvn,tln,rovn,roln,
    1  evn,eln,dp,rhs,cpa,cptv,cptl,tsat,tw,dtw,hlfc,hlnb,hvfc,
    2  hvs,hls,m1,m2,m3,m4)
c
c  determine values of alpha, tl, tv  from new pressures
c   evaluate new functions of state
c
      common/sg/z1,z2,z3,z4,z5,z6,z7,z8,nmin,z10,
    1 z11,z12,z13,z14,z15,z16,z17,z18,z19,z20,z21,z22,z23,z24,z25,
    2 z26,z27,z28,z29,z30,z31,z32,z33,z34,z35,z36,z37,
    3 z38,z39,z40,z41,z42,z43,z44,z45,z46
     common /ic/ nstep,nitmax,nitno,iitmax,iitot,iic,kred,nm,ntc,
    1   nc,nrods,nz,nr,ncp,nzp,nzp2, iflash,itb,ibb,icpu,iwft,ivec,
    2   nodes,ndm1,ncf,ncc,ng,iht, iss,iqss,itam,ires,idump,ntitle,
    3    itwmax,jtwmax,itrmax,jtrmax,imchfr,jmchfr,ichf,ifintr,ierr,lerr
    4     ,imixm,imixe,iafm,itfm,igfm,nktb,jhem
     logical lerr
     dimension icc(4,1),irc(4,1),pn(m4,1),alpn(m4,1),tvn(m4,1),
    1  tln(m4,1),rovn(m4,1),roln(m4,1),evn(m4,1),idown(m2),
    2  eln(m4,1),dp(m4,1),rhs(m1,m2,4),cpa(6,m1,1),
    3  cptv(6,m1,1),cptl(6,m1,1),tsat(m4,1),tw(m4,1),
    4  dtw(m4,1),hlfc(m4,1),hlnb(m4,1),hvs(m4,1),hls(m4,1),hvfc(m4,1)
c
     do 200 i=1,nc
         itype=idown(i)
       iym=icc(1,i)
       ixm=icc(2,i)
       ixp=icc(3,i)
       iyp=icc(4,i)
       do 200 j=1,nz
         jj=j+1
         pn(jj,i) = pn(jj,i) + dp(jj,i)
```

```
      alpn(jj,i) = alpn(jj,i)+rhs(j,i,2)-cpa(1,j,i)*dp(jj,iym)-
     1   cpa(2,j,i)*dp(jj,ixm)-cpa(3,j,i)*dp(jj,ixp)-
     2   cpa(4,j,i)*dp(jj,iyp)-cpa(5,j,i)*dp(j,i)-
     3   cpa(6,j,i)*dp(jj+1,i)
      dtvn = rhs(j,i,3)-cptv(1,j,i)*dp(jj,iym)-
     1   cptv(2,j,i)*dp(jj,ixm)-cptv(3,j,i)*dp(jj,ixp)-
     2   cptv(4,j,i)*dp(jj,iyp)-cptv(5,j,i)*dp(j,i)-
     3   cptv(6,j,i)*dp(jj+1,i)
      if(jj.ge.nmin.and.itype.eq.-1)dtvn=0.
      tvn(jj,i) = tvn(jj,i) + dtvn
      dtln = rhs(j,i,4)-cptl(1,j,i)*dp(jj,iym)-
     1   cptl(2,j,i)*dp(jj,ixm)-cptl(3,j,i)*dp(jj,ixp)-
     2   cptl(4,j,i)*dp(jj,iyp)-cptl(5,j,i)*dp(j,i)-
     3   cptl(6,j,i)*dp(jj+1,i)
       if(jj.ge.nmin.and.itype.eq.-1) dtln=0.
      tln(jj,i) = tln(jj,i) + dtln
      tsold = tsat(jj,i)
      call state(pn(jj,i),tvn(jj,i),tln(jj,i),rovn(jj,i),
     1 roln(jj,i),evn(jj,i),eln(jj,i),tsat(jj,i),hvs(jj,i),hls(jj,i),
     2   d1,d2,d3,d4,d5,drldp,drvdp,d8,d9,2,ierr)
      if (ierr.ne.0) go to 904
      dtsat = tsat(jj,i) - tsold
      do 100 n=1,4
      ir=irc(n,i)
      if(ir.eq.0) go to 105
      chtw= dtw(jj,ir)*(hvfc(jj,ir)*dtvn+hlfc(jj,ir)*dtln
     1 + hlnb(jj,ir)*dtsat)
 100  tw(jj,ir)=tw(jj,ir)+chtw
 105  continue
      ts = 0.1*rovn(jj,i)/roln(jj,i)
      if (alpn(jj,i).lt.-ts*drldp/drvdp) go to 907
      if (alpn(jj,i).lt.1.e-10) alpn(jj,i)=0.
      if (alpn(jj,i).gt.1.+ts) go to 908
      if (alpn(jj,i).gt.1.) alpn(jj,i)=1.
 200  continue
      return
c
c  range of state functions or void fraction exceeded
c
 904  ierr = ierr + 3
      go to 999
 907  ierr = 7
      go to 999
 908  ierr = 8
 999  lerr = .true.
      return
      end
      subroutine film(h,alp,rov,rol,vva,vla,hd,rhd,tl,tv,tw,tsat,hfg,
     1 cpv,cpl,p,visv,visl,betav,sig,ihtr,x)
      data gcon,pi2/9.8066,6.2831853/
c
c  note: In Bromley's and McAdams' correlations vapor properties
c        are evaluated at bulk vapor temperature and NOT
c        at vapor film temperature.
```

```
c        In Groeneveld's correlation the vapor Prandtl number
c        is evaluated at bulk vapor temperature and NOT
c        at wall temperature.
c
c
c   high flow film boiling
c   Groeneveld 5.7 or modified Dittus-Boelter (for low pressure)
c
      cndv = condv(p,tv)
      rev = hd*rov*(vla+alp*(vva-vla))/visv
      prv = visv*cpv/cndv
      if(p.lt.1.33e6) go to 10
      y = 1.-.1*pow((1.-x)*((rol/rov)-1.),.4)
      hgdb = .052*cndv*rhd *pow(rev,.688)*pow(prv,1.26)*pow(y,-1.06)
      ihtr = 6
      go to 20
   10 hgdb = .023*cndv*rhd *pow(rev,.800)*pow(prv,0.40)
      ihtr = 7
   20 continue
      h = hgdb
c
c   test for low or high flow
c
      ajg =      alp *rov*vva/sqrt(gcon*hd*rov*(rol-rov))
      ajf = (1.-alp)*rol*vla/sqrt(gcon*hd*rol*(rol-rov))
      aj = sqrt(ajg)+sqrt(ajf)
      if(aj.ge.2.0) return
c
c   low flow film boiling
c   Bromley plus max of McAdams and forced convection(as for high flow)
c
      clam = pi2*sqrt(sig/(rol-rov))
      hfgp = hfg+0.5*cpv*(tw-tsat)
      t1 = gcon*(rol-rov)*rov*(cndv**3)*hfgp/(clam*visv*(tw-tsat))
      hmb = .62*pow(t1,.25)
c
      t1 = rov*rov*gcon*betav*cpv*abs(tw-tv)/(visv*cndv)
      hma = .13*cndv*pow(t1,.333333)
c
      h = (1.-alp)*hmb + alp*amax1(hgdb,hma)
      ihtr = 8
c
      return
      end
      subroutine chf(qchf,alp,rov,rol,g,p,x,hd,hfg,sig,hlin,
     1      xequil,gbrit,ichf,hls,ftong)
c
c  determines critical heat flux
c
      data gcon/9.8066/
      data hconvt/4.302e-4/
      data ee /2.7182818/
      if(ichf.gt.1) go to 200
c
```

```
      pbar=1.0e-5*p
      ghi=135.0
      glo=27.0
      if(pbar.ge.83.0.and.x.ge.0.5)ghi=270.0
      if(g.lt.glo)go to 20
c
c  Biasi correlation for high flow
c
      en=-0.4
      if(hd.lt.0.01)en=-0.6
      gt=amax1(g,ghi)
      q10=0.0
      if(gt.lt.300.0)go to 10
      f=.7249 + .099*pbar*pow(ee,-.032*pbar)
      g6=pow(gt,-.166667)
      q10=2.764e7*pow(100.*hd,en)*g6*(1.468*f*g6-x)
   10 continue
      h=-1.159 + .149*pbar*pow(ee,-.019*pbar) + 8.99*pbar/
     1 (10.+pbar*pbar)
      q11=15.048e7*h*pow(100.*hd,en)*pow(gt,-.6)*(1.0-x)
      qb=amax1(q10,q11)
      qchf=qb
c
c
      if(g.ge.ghi)go to 100
   20 continue
c
c  chf-void correlation for low flow
c
      t1=sig*gcon*gcon*(rol-rov)*rov*rov
      qvc=.1178*(1.-alp)*hfg*pow(t1,.25)
      qchf=qvc
c
c
      if(g.le.glo)go to 100
c
c  linear interpolation between Biasi and chf-void
c
      wt=(g-glo)/(ghi-glo)
      qchf=wt*qb+(1.-wt)*qvc
c
  100 continue
      return
  200 continue
      hf = hls * hconvt
      pbrit = p* 1.4503684e-4
      dh = hd * 39.370079
      hin = hlin * hconvt
c
c     w-3 for uniform heat flux
c
      chfw3 = ((2.022 - .0004302*pbrit) + (.1722 - .0000984*pbrit)*
     1         exp((18.177-0.004129*pbrit)*xequil)) * ((0.1484 -
     2         1.596* xequil + 0.1729*xequil* abs(xequil))* gbrit +
```

```
     3        1.037 ) * (1.157-0.869*xequil) * (0.2664+0.8357 *
     4        exp( -3.151 *dh)) * (0.8258 + 0.000794*(hf-hin))
c    w-3 chf with non-uniform axial heat flux correction
c
      qchf = chfw3 * 3.1546031e6 / ftong
      qchf= amax1(qchf,2.84e5)
       return
       end
      subroutine cpr(qchf,qchen,xequil,g,p,d,bl,hls,hlin,hfg)
c
c
c     calculates the critical quality and approximates the critical
c     power.  The critical power ratio is used to estimate qchf
c
      data pcr/2.2106e7/
c
c     CISE-4 Correlation
c     determines critical quality
c     xcrit = critical quality
c     bl    = boiling length
c
      gstar=3375.*(1-p/pcr)**3
      a=1./(1.+1.481e-4*(1-p/pcr)**-3*g)
      if(g.gt.gstar) a=(1.-p/pcr)/(g*.001)**.33333333
      b=0.199*(pcr/p-1)**0.4*g*d**1.4
      xcrit=a*bl/(b+bl)
      cpr1=1.+(xcrit-xequil)/(xequil+(hls-hlin)/hfg)
      if(bl.le.0.) cpr1=99.
      qchf=qchen*cpr1
      return
      end
      function pow(a,b)
c
c  this function is called whenever a low accuracy exponentiation
c   would be adequate
c
      pow=a**b
      return
      end
      function  condl (p, tl)
c
c  thermal conductivity of liquid water
c    W/m deg K    function of    Pascal,  deg K
c
c  error of approximation < 5 percent for  273 < tl < 573 deg K
c  value at 150 bar, 300 deg C = .55
c
      ts = tl - 415.
      condl = .686 - 5.87e-6*ts*ts + 7.3e-10*p
      return
      end
      function  condv (p, tv)
c
c  thermal conductivity of dry steam
```

```
c    W/m deg K    function of    Pascal,  deg K
c
c  error of approximation < 10 percent for  373 < tv < 623 and
c    p in superheated region
c  for low p, condv depends more on tv, for p > 50 bar condv depends
c    more on p.
c  value at saturation for 70 bar = .061
c
      condv = -.0123 + p*(7.8e-9 + p*2.44e-16) +
     +            1.25e-11*tv*(80.e5 - p)
      return
      end
      function vislq (tl)
c
c  viscosity of saturated liquid water
c    kg/m sec    function of    deg K
c
c  error of approximation = 6 percent for  273 < tl < 623 deg K
c    may also be used for non-saturated conditions at same tl
c    this fit has a singularity at tl = 251 deg K
c  value at 250 deg c = .107e-3
c
      vislq = 25.3 / (-8.58e4 + tl*(91.+ tl))
      return
      end
      function  visvp (tv)
c
c  viscosity of saturated steam
c    kg/m sec    function of    deg K
c
c  error of approximation = 3 percent for  373 < tv < 623 de K
c    may also be used for non-saturated conditions at same tv
c    this fit has a singularity at tv = 822 deg K
c  value at 250 deg c = .174e-4
c
      if(tv.gt.623) go to 50
      visvp = 11.4 / (1.37e6 - tv*(844.+ tv))
      return
 50    visvp=4.07e-8*tv -3.7e-7
      return
      end
      function surten (tl)
c
c  surface tension of liquid water
c    kg(f)/m    function of    deg K
c  ( 1 kg(f)= 9.80665 kg m/sec**2 )
c  also equal to surface tension / gravitational acceleration constant
c  in units of kg/m
c
c  error of approximation = 2 percent for 373 < tl < 623 deg K
c  value at 250 deg c = .0026
c
      surten = (80.72 - tl*.126) / (5140.+ tl)
      if(surten.lt.0.0) surten=0.0
```

```
      return
      end
      subroutine fwall (fwv,fwl,hdiam,velx,cfv,cfl,
     1  p,alp,rov,rol,vv,vl,visv,visl,rd,iwf,i,j)
c
c  computes wall friction for liquid and vapor momentum equations
c
c  return values  fwv,fwl  should appear in the momentum equations
c    for one of the three directions, multiplied by the appropriate
c    velocity
c
c  input:
c    hdiam    hydraulic diameter
c    velx     velocity multiplier, converts transverse velocity to
c               maximum value (in gap)
c    cfv,cfl  wall contact fractions for vapor and liquid;
c               normally  cfv+cfl=1  and  cfv=0 below chf
c    rd       inverse of dz; used for form loss
c    iwf      selects formula to be used:
c      tens digit = 0: axial friction only
c                 = 1 or 2: form loss + axial friction
c                 = 3: transverse friction
c      iwf = 01: 2 phase Martinelli with Xtt parameter
c          = 02: 2 phase Martinelli-Nelson with mass flux effect (Jones)
c          = 03: 2 phase Levy
c          = 04: Rough wall correlation
c          = 10: form loss without axial friction
c          = 1n: form loss + 0n axial friction
c          = 2n: same as 1n (convection funnel effect)
c          = 31: Gunter and Shaw lateral with Martinelli 2 phase
c
c  N.B.: In two-phase flow, a friction multiplier is applied assuming
c    both phases to be turbulent if one is. No two-phase multiplier is
c    used if both phases are laminar.
c
      common /fricmd/ fcon(4,4)
c
      logical tt
      data  cax, ct /20., 8./
c
c-----------------------------------------------------
c
      all = 1.- alp
      fwv = 0.
      fwl = 0.
      if (hdiam.eq.0.)  return
      rhd = 1./hdiam
      gv = abs(alp*rov*vv*velx)
      gl = abs(all*rol*vl*velx)
      g = gv + gl
      rev = gv/(visv*rhd)
      rel = gl/(visl*rhd)
      it = int (float(iwf)*.1)
      iu = iwf - it*10
```

```
      fwv = 0.
      fwl = 0.
c
    5 a0 = fcon(1,it+1)
      rex = fcon(2,it+1)
      a = fcon(3,it+1)
      b = fcon(4,it+1)
      tt = (rel.gt.rex) .or. (rev.gt.rex)
      reyl = rel
      if (tt) reyl = amax1(rel,rex)
      reyv = rev
      if (tt) reyv = amax1(rev,rex)
c
      if (it.eq.0) go to 10
      go to (100,100,200),it
   10 if (iu.eq.0) return
c
c  axial friction
c
      if (tt) go to 20
        fwv = fwv + .5*a0*cfv*visv*rhd*rhd
        fwl = fwl + .5*a0*cfl*visl*rhd*rhd
        return
   20 continue
      go to (30,40,50,60),iu
   30 fwl = fwl + .5*rhd*a*cfl*( (reyl**b)*all*gl +
     +               cax *sqrt((rol/rov)*(reyv*reyl)**b) *all*gv  )
      fwv = fwv + .5*rhd*a*(reyv**b)*alp*gv*cfv
      return
c
   40 xq = gv / g
      g0 = g/950. - 1.
      if (g0) 41,42,42
   41   fgp = 1.43 + g0*(.07 - 7.35e-8*p)
        go to 43
   42   fgp = 1.43 + (950./g -1.)*(.17 - 6.e-8*p)
   43 fwl = fwl + .5*rhd*a*(reyl**b)*all*g*cfl*
     *                (1.+ fgp*1.2*(rol/rov -1.)*(xq**.824) )
      fwv = fwv + .5*rhd*a*(reyv**b)*gv*cfv
      return
c
   50 reyl = abs(rol*vl)/(visl*rhd)
      reyv = abs(rov*vv)/(visv*rhd)
      fwl = fwl + .5*rhd*a*(reyl**b)*gl*cfl
      fwv = fwv + .5*rhd*a*(reyv**b)*gv*cfv
      return
  60  rrel=18.7/g*rhd*visl
      t=6.5
      do 65 k=1,10
      if(t.le.0.) t=10.
      df= t- 1.74 + .868589*log(.02+rrel*t)
      dfdt= 1+ .868589/(.02 + rrel*t)*rrel
      dt= -df/dfdt
      t= t + dt
```

```
      if(abs(dt).le. 0.01) go to 70
   65 continue
   70 continue
      f= 1/t**2
      fwl=fwl + .5*rhd*f*g**.2*(rol*abs(vl))**.8*cfl
      fwv=fwv + .5*rhd*f*g**.2*(rov*abs(vv))**.8*cfv
      return
c
c
c  axial form loss
c
  100 rey = amax1 (rex, g/(visl*rhd) )
      ts = .5*rd*a*(rey**b)
      fwv = ts*alp*gv
      fwl = ts*(all*g + alp*all*rol*abs(vv) )
      it = 0
      if (iu.le.0) return
      go to 5
c
c  transverse friction
c
  200 continue
      if (iu.eq.0) return
      if (tt) go to 220
        fwv = .5*a0*alp*cfv*visv*rhd*rhd*velx
        fwl = .5*a0*all*cfl*visl*rhd*rhd*velx
        return
  220 fwv = .5*rhd*a*(reyv**b)*alp*cfv*gv*velx
      fwl = .5*rhd*a*cfl*( (reyl**b)*all*gl +
     +             ct *sqrt((rol/rov)*(reyv*reyl)**b) *all*gv )*velx
      return
      end
      subroutine finter(fiv,fil,alp,rov,rol,vr,visv,visl,hd,ifintr)



      data an/4.18888e7/
      ts = amax1(alp,0.1)
      rrb = (1.01-ts)/(ts*hd)
      fil = rrb*(rrb*visl + 0.5*rov*vr)*1.0
      fiv = fil
      if(alp.le.0.) go to 50
      if(alp.eq.1.)go to 50
      if(ifintr.eq.0)go to 50
c      alternative interfacial model
       rho = alp*rov +(1.-alp)*rol
      visc = alp*visv/rov + (1.-alp)*visl/rol
      alpha = alp
      if(alp.gt.0.5)alpha=1.-alp
      a = an**.33333*alpha**.666667
      rb = (alpha/an)**.33333
      fil = .375*rho*(.5*vr+12*visc/rb)*a*1.0
   50 fiv = fil
       return
```

```
      end
      subroutine gamma(gam,dgdp,dgda,dgdtv,dgdtl,
     1                 alp,alpn,tvn,tln,rov,rol,tsatn,dtsdp)
      data an,srrg/1.33333e+14,21.4942/
c
c  function: calculates gamma and its derivatives
c
c   note: an=4/3*nr. of bubbles(or droplets) per cubic metre
c         srrg=sqrt of the gas constant for water vapor
c
      talp=amax1(alp,.0001)
      talp=amin1(talp,.9999)
      wt = .5+sign(.5,talp-.5)
      talpx = talp*(1.-wt)+(1.-talp)*wt
      area = an**.333333 * talpx**.666667
      ce = area*rol*    talp *srrg
      cc = area*rov*(1.-talp)*srrg
c
      alame = .05+sign(.05,tln-tsatn)
      alamc = .05+sign(.05,tsatn-tvn)
      clame = ce*alame
      clamc = cc*alamc
      rrtsat = 1./sqrt(tsatn)
      dtl = (tln-tsatn)*rrtsat
      dtv = (tvn-tsatn)*rrtsat
c
      gam = clame*(1.-alpn)*dtl+clamc*alpn*dtv
      dgdp = -.5*(clame*(1.-alpn)*(tsatn+tln)+clamc*alpn*(tsatn+tvn))
     1      *rrtsat/tsatn*dtsdp
      dgda = -clame*dtl+clamc*dtv
      dgdtv = clamc*    alpn *rrtsat
      dgdtl = clame*(1.-alpn)*rrtsat
      return
      end
      subroutine qinter (qi, dqdp,dqda,dqdtv,dqdtl, alpn,pn,tvn,
     1   tln,tsatn,dtsdp, alp,p,rov,rol )
c
c  computes interfacial energy exchange
c
c  the current model sets this term to a large constant times
c   (tvn-tln), with the result that tvn should always be
c   nearly equal to tln.
c
      data hinter/1.e+11/
c
      qi = hinter*(tln-tvn)
      dqdp = 0.
      dqda = 0.
      dqdtv = -hinter
      dqdtl = hinter
c
      return
      end
      subroutine qisub(qi,dqdp,dqda,dqdtv,dqdtl,hvs,gam,dgdp,dgda,
```

```
     +          dgdtv,dgdtl,dhvsdp,tvn,tsatn,dtsdp)
c
c      interfacial energy exchange term for use with
c      subcooled boiling model in gamsub
c
c      the form here has the effect of forcing tv=tsat
c      below film boiling
c
       data hinter /1.e11/
c
       qi = hinter*(tsatn-tvn) + gam*hvs
       dqdp = hinter*dtsdp + hvs*dgdp + gam*dhvsdp
       dqda = hvs*dgda
       dqdtv = -hinter + hvs*dgdtv
       dqdtl = hvs*dgdtl
       return
       end
       subroutine qisup(qi,dqdp,dqda,dqdtv,dqdtl,gam,dgdp,dgda,dgdtv,
     +                 dgdtl,hls,dhlsdp,tsat,tl,dtsdp)
c
c      calculates interfacial heat transfer rate in the
c      post-CHF flow regime.
c                             Kao, 5-22-80
c
       data hinter/1.0e11/
c
       qi = gam*hls-hinter*(tsat-tl)
       dqdp = hls*dgdp+gam*dhlsdp-hinter*dtsdp
       dqda = hls*dgda
       dqdtv = hls*dgdtv
       dqdtl = hls*dgdtl+hinter
       return
       end
       subroutine gamsub (gam,dgdp,dgda,dgdtv,dgdtl,p,alp,tv,tl,tsat,
     1  pn,alpn,tvn,tln,tsatn,dtsdp,deldt,rov,rol,vv,vl,dh,area,visl,
     2  hv,hl,qleff,qveff,dqwdtl,dqwdp,ihtr)
c
c   phase change rate gamma, including a model for subcooled boiling
c
       data alpbf,cbf /0.1, 9./
       data itrns /5/
       data third /0.3333333/
c
       cpl = deldt
       cndl = condl(p,tl)
       tst = 9.0395*pow(p,.223) + 255.2
       rhvl = 1./(hv-hl)
c
c   subcooled or nucleate boiling phase change
c
       hh = 0.
       if (tln.ge.tsatn) go to 10
         g = alp*rov*abs(vv) + (1.-alp)*rol*abs(vl)
         re = g*dh/visl
```

```
        prl = visl*cpl/cndl
        xnuah = 2.44*pow(re,.5)*pow(prl*(hv-hl)/hl,third)
        hh = cndl*xnuah/dh
   10 qbl = qleff - hh *(tsatn-tln)*area
      w = (.5 + sign(.5,qbl)) * rhvl
c--w is zero when tln < bubble detachment temperature
c
   20 gam = qbl*w
      dgdp  = -w*(dqwdp + hh*dtsdp*area)
      dgda  = 0.
      dgdtv = 0.
      dgdtl = -w*(dqwdtl - hh*area)
c
c  condensation or vaporization at liquid-vapor interface
c
      surfb = 0.
      rh = 0.5*dh
        rb = rh
        ve = pow( (1.-alp)*abs(vl), third)
        rb0 = 0.45 * pow( surten(tst)/(rol-rov), .5) / (1.+ 1.34*ve)
        if (alp.ge.1.) go to 110
        rb = rb0
        if (alp.gt.alpbf) rb = rb0*pow(cbf*alp/(1.-alp), third)
        rb = amin1(rb, rh)
  110   surfb = 3./ rb
      dl = rb0 * 0.015
      if (tvn.gt.tln)  go to 150
c  vaporization
        cnd = cndl /dl
        go to 160
c  condensation
  150   cndv = condv(p,tv)
      dv = rb0 * 0.010
        cnd = cndv*cndl /(cndl*dv + cndv*dl)
c
  160 cnd = cnd*rhvl
      s = surfb*alpn
      scnd = s*cnd
      dtlv = tln-tvn
      gam = gam + scnd*dtlv
      dgdp = dgdp
      dgda = dgda + surfb*cnd*dtlv
      dgdtv = dgdtv - scnd
      dgdtl = dgdtl + scnd
c
      return
      end
      subroutine gamsup(gam,dgdp,dgda,dgdtv,dgdtl,p,rov,tv,tl,vv,
     1                  alpn,tvn,tsatn,hg,hf,dtsdp,d)
c
c    post-chf gamma model of saha
c
      data gcon,pcrit/9.80665,2.2105e7/
c
```

```
      sig=surten(tl)*gcon
      conv=condv(p,tv)
      rd=1./d
      hfg=hg-hf
      w=6300.*(1.-p/pcrit)**2*(rov*vv*vv*d/sig)**0.5*rd*rd*conv/hfg
      gam=w*(tvn-tsatn)*(1.-alpn)
      if(gam.lt.0.) gam=0.
      dgdp=-w*(1.-alpn)*dtsdp
      dgda=-w*(tvn-tsatn)
      dgdtv=w*(1.-alpn)
      dgdtl=0.
      return
      end
      function theta(rhol,rhov,g,re,d,sij,x)
      data a1,a2,grav/0.4,0.6,9.81/
      theta=1.0
      if(x.le.0..or.x.ge.1.0) return
      xm=(a1*sqrt(rhol*(rhol-rhov)*grav*d)/g+a2)/(sqrt(rhol/rhov)+a2)
      betal=0.04*(sij/d)**1.5
      thetam= 5.
      theta=1.+(thetam-1.)*x/xm
      if(x.le.xm) return
      x0=xm*0.57*re**0.0417
      theta=1.+(thetam-1.)*(xm-x0)/(x-x0)
      return
      end
      subroutine innbgs
c
      common a(1)
      common /point/ lncr,lindnt,licc,liwfz,lihtr,licr,lirc,lidwn,ljmtb,
     1 lp,lalp,lrov,lrol,lev,lel,ltv,ltl,ltr, lpn,lalpn,lrovn,
     2  lroln,levn,leln,ltvn,ltln,ltrn, lvvx,lvlx,lvvy,lvly,lvvz,lvlz,
     3  lhv,lhl,lhvs,lhls,ltsat,lvisv,lvisl, ldx,ldy,ldz,larx,lary,
     4  larz,lvol, lcpvx,lcpvy,lcpvz,lcplx,lcply,lcplz, lfvx,lfvy,
     5  lfvz,lflx,lfly,lflz, lajm1,lajm2,lcpa,lcptv,lcptl,lrhs,
     6ldp,lqpp,lqv,lql,lhvfc,lhlnb,lhlfc, ldtrn,ldtw,ltw,lchfr,lfrac,
     7lhdz,lhdt,lhdh,lqz,lqt,lqr,lqppp,lrn,lcnd,lrcp,lrrdr,lvp,lvm,lrad
     8,lbotbc,ltopbc,ltmpbc,lsij,lph,lth,lfg,ldsew,ldsns,laht,lend
      common /ic/ nstep,nitmax,nitno,iitmax,iitot,iic,kred,nm,ntc,
     1  nc,nrods,nz,nr,ncp,nzp,nzp2, iflash,itb,ibb,icpu,iwft,ivec,
     2  nodes,ndm1,ncf,ncc,ng,iht, iss,iqss,itam,ires,idump,ntitle,
     3  itwmax,jtwmax,itrmax,jtrmax,imchfr,jmchfr,ichf,ifintr,ierr,lerr
     4   ,imixm,imixe,iafm,itfm,igfm,nktb,jhem
      logical lerr,loutput
      common /rc/ delt,rdelt,errn,epsn,erri,epsi,dtmin,dtmax,tend,
     1  dtsp,dtlp,rtnsp,rtnlp,gravx,gravy,gravz,rtime,velx,
     2  q,q0,twmax,trmax,amchfr
c
c perform explicit calculations
c
      dimension ia(1),ltable(3)
      equivalence (a(1),ia(1))
c
      omega = 1.0
```

```
      nx = ia(lncr)
      ny = nr
c
c  provide scratch storage for auxiliary iteration matrices,
c   virtual rhs and working area.
c
      laa = lend
      nband = 2*nx+1
      lenaa = nc*nband*nz
      lbb = laa+lenaa
      lenbb = nc*nz
      lxl = lbb+lenbb
      lenxlc = nc*(nx+1)
      lenxl = lenxlc*nz
c
c..set initial dp = 0.0
c
c     call clear(0.0,a(ldp),nzp2*ncp)
c
c..perform plane SOR iterations
c
       call innxyb( a(ldp),a(lajm1),a(lajm2),a(lrhs),a(laa),a(lbb),
     1       a(lxl),ia(licc),
     2       nz,nc,nzp2,ncp,nx,nband,lenxlc,
     3       epsi,iitmax,iic,erri,omega,a(lp) )
c
      return
      end
      subroutine innxyb( dp,ajm1,ajm2,rhs,aa,b,xl,icc,
     1            nz,nc,nzp2,ncp,nx,nband,lenxlc,
     2            epsi,iitmax,iic,erri,omega,pref )
      dimension dp(nzp2,ncp),ajm1(3,nz,nc),ajm2(4,nz,nc),rhs(nz,nc),
     1           aa(nc,nband,nz),b(nc,nz),xl(lenxlc,nz),icc(4,nc)
c  function: solves the system [ajm1+ajm2]*[dp] = [rhs]
c   by successive xy-block relaxation.
c
      j1 = 1
      j2 = nx
      j0 = nx+1
      j3 = j0+1
      j4 = j0+nx
      js1 = j1+1
      jf1 = j2-1
      js2 = j3+1
      jf2 = j4-1
c
      ijob = 1
c
c..form matrix[aa] and perform LU factorization
c
      do 40 iz=1,nz
      do 30 ic=1,nc
      aa(ic,j1,iz) = ajm2(1,iz,ic)
       do 10 j=js1,jf1
```

```
   10   aa(ic,j,iz) = 0.0
        aa(ic,j2,iz) = ajm2(2,iz,ic)
        aa(ic,j0,iz) = ajm1(1,iz,ic)
        aa(ic,j3,iz) = ajm2(3,iz,ic)
         do 20 j=js2,jf2
   20    aa(ic,j,iz) = 0.0
        aa(ic,j4,iz) = ajm2(4,iz,ic)
   30 continue
c
        call leqt1b(aa(1,1,iz),nc,nx,nx,nc,b(1,iz),1,nc,ijob,xl(1,iz),ier)
c
   40 continue
c
        ijob = 2
        omega1 = 1.0-omega
c
c..enter iterative cycle
c
        do 100 it=1,iitmax
        iic = it
        erri = 0.0
c
c...form virtual RHS and solve the X-Y block
c
        do 60 iz=1,nz
        jz = iz+1
        jzm = jz-1
        jzp = jz+1
c
        do 50 ic=1,nc
        b(ic,iz) = rhs(iz,ic) - ajm1(2,iz,ic)*dp(jzm,ic)
       1                      - ajm1(3,iz,ic)*dp(jzp,ic)
   50 continue
c
        call leqt1b(aa(1,1,iz),nc,nx,nx,nc,b(1,iz),1,nc,ijob,xl(1,iz),ier)
c
        do 60 ic=1,nc
        dpold = dp(jz,ic)
        dp(jz,ic) = omega*b(ic,iz) + omega1*dpold
        erri = amax1(erri,abs(dp(jz,ic)-dpold) )
        erri = erri/pref
   60 continue
c
        if(erri.le.epsi) return
c
  100 continue
        return
        end
        subroutine hconv(ihtr,qv,ql,hvfc,hlnb,hlfc,tw,tl,tv,p,alp,
       1 rov,rol,hv,hl,hvs,hls,qpp,dz,fracp,tsat,vvz,vlz,tr,icr,hdh,
       2 chfr,dtrn,ktb,m1,m2,m3,m4,m5,m6)
c
c  explicit portion of heat transfer (beginning of time step)
c
```

```
      dimension ihtr(m1,1),qv(m2,1),ql(m2,1),hvfc(m2,1),hlnb(m2,1),
     1 hlfc(m2,1),tw(m2,1),tl(m2,1),tv(m2,1),p(m2,1),alp(m2,1),
     2 rov(m2,1),rol(m2,1),vvz(m3,1),vlz(m3,1),tr(m5,m2,1),icr(1),
     3 hdh(1),qpp(m2,1),hl(m2,1),hv(m2,1),hls(m2,1),hvs(m2,1),
     4 chfr(m2,1),dz(1),fracp(1),tsat(m2,1),dtrn(m6,m2,1)
      common /ic/ nstep,nitmax,nitno,iitmax,iitot,iic,kred,nm,ntc,
     1   nc,nrods,nz,nr,ncp,nzp,nzp2, iflash,itb,ibb,icpu,iwft,ivec,
     2   nodes,ndm1,ncf,ncc,ng,iht, iss,iqss,itam,ires,idump,ntitle,
     3   itwmax,jtwmax,itrmax,jtrmax,imchfr,jmchfr,ichf,ifintr,ierr,lerr
     4    ,imixm,imixe,iafm,itfm,igfm,nktb,jhem
      logical lerr
      common /rc/ delt,rdelt,errn,epsn,erri,epsi,dtmin,dtmax,tend,
     1   dtsp,dtlp,rtnsp,rtnlp,gravx,gravy,gravz,rtime,velx,
     2   q,q0,twmax,trmax,amchfr
       data gconv/7.3732572e-4/
c
      amchfr = 99.0
c
      do 200 i=1,nrods
      jboil = 0
        do 200 j=1,nz
        bl = 0.
          jj=j+1
          tw(jj,i) = tr(4*(ktb-1)+4,jj,i)
           ii=icr(i)
c
c  compute average velocities in z direction
c   note: transverse velocities are ignored at present
c
          vv=0.5*(vvz(j,ii)+vvz(jj,ii))
          vl=0.5*(vlz(j,ii)+vlz(jj,ii))
c
          if(ichf.lt.2) go to 190
          if(tw(jj,i).lt.tsat(jj,ii)) go to 190
           rolalp= (1. - alp(jj,ii))*abs(vl)*rol(jj,ii)
          rovalp = alp(jj,ii) * abs(vv)*rov(jj,ii)
c      compute equilibrium quality

          xequil = ((rolalp*hl(jj,ii) + rovalp*hv(jj,ii))/(rolalp
     1    +rovalp) - hls(jj,ii))/(hvs(jj,ii) -hls(jj,ii))
       if(ichf.ne.2) go to 90
c
c      compute F factor for w-3
c
       z=0.
       sum=0.
       gbrit = (abs(vl)*rolalp+abs(vv)*rovalp)* gconv
       c = 0.15*pow((1.-xequil),4.31)*pow(gbrit, -0.478)
       do 50 jx=1,j
       jxp1 = jx +1
       zp1 = z + dz(jxp1) * 39.37079
       sum = sum + qpp(jx+1,i)*(exp(c*zp1)-exp(c*z))
       z = zp1
 50    continue
```

```
      ftong = sum/qpp(jj,i)/(exp(c*z)-1.)
      go to 190
 90   continue
c
c     compute boiling length and power input over boiling length
c
      if(jboil.gt.1) go to 160
      if(xequil.le.0.) go to 190
      jboil=jj
 160  continue
      do 170 jx=jboil,jj
      bl = bl + dz(jx)
 170  continue
 190  continue
c  obtain heat transfer coefficients and/or heat flux
c
         call htcor(ihtr(j,i),qv(jj,i),ql(jj,i),hvfc(jj,i),hlnb(jj,i),
     1    hlfc(jj,i),tw(jj,i),tl(jj,ii),tv(jj,ii),p(jj,ii),alp(jj,ii),
     2    rov(jj,ii),rol(jj,ii),vv,vl,hdh(ii),iss,chfr(jj,i),hl(1,ii),
     3    hls(jj,ii),hvs(jj,ii),xequil,gbrit,ftong,bl,ichf,
     4    hv(jj,ii),hl(jj,ii),tsat1)
c
c  find mininum critical heat flux ratio
c
         if(amchfr.le.chfr(jj,i)) go to 201
         amchfr = chfr(jj,i)
         imchfr = i
         jjmchfr = jj
201   continue
c   SG version ; explicit ht to secondary
c   Determine Heat Input to Liq. for tube bank ktb
c   Store Heat Flux from each tube bank in dtrn(ktb,j,i)
      dtrn(ktb,jj,i)=ql(jj,i)+qv(jj,i)+hlfc(jj,i)*(tw(jj,i)-
     1tl(jj,ii))+hlnb(jj,i)*(tw(jj,i)-tsat(jj,i))+hvfc(jj,i)*(
     2tw(jj,i)-tv(jj,ii))
 200     continue
c
      if(amchfr.ne.1.0.and.amchfr.ne.99.0) return
      imchfr = 0
      jmchfr = 0
      return
      end
      subroutine htcor(ihtr,qv,ql,hvfc,hlnb,hlfc,tw,tl,tv,p,alp,rov,rol,
     1              vv,vl,hd,iss,chfr,hlin,hls,hvs,
     2                  xequil,gbrit,ftong,bl,ichf,hv,hl,tsat1)
c
c  this routine computes heat transfer coefficients and/or heat
c  fluxes
c
c  the total heat flux is assumed to be of the form:
c  q=qv+ql+hvfc(tw-tv)+hlnb(tw-tsat)+hlfc(tw-tl)
c
c  normally qv and ql will be zero and one or more of the heat
c  transfer coefficients hvfc, hlnb, and hlfc will be non-zero.
```

```
c  In transition boiling, however, the heat transfer coefficients are
c  zero and q=qv+ql.
c
c  Nomenclature:
c
c  qv     Heat flux to vapor (w/m**2)
c  ql     Heat flux to liquid (w/m**2)
c  hvfc   Convection heat transfer coefficient to vapor (w/m**2 K)
c  hlnb   Nucleate boiling heat transfer coefficient (w/m**2 K)
c  hlfc   Convection heat transfer coefficient to liquid (w/m**2 K)
c  hvfilm Film boiling heat transfer coefficient to vapor (w/m**2 K)
c  tw     Wall temperature (k)
c  tl     Liquid temperature (k)
c  tv     Vapor temperature (k)
c  p      Pressure (p)
c  alp    Vapor volume fraction
c  rov    Vapor density (kg/m**3)
c  rol    Liquid density (kg/m**3)
c  vv     Vapor velocity (m/s)
c  vl     Liquid velocity (m/s)
c  hd     Hydraulic diameter (m)
c  chfr   Critical heat flux ratio
c
c
c  note: the following quantities are available and,
c  if desired, could be added to the argument list of
c  htcor and the corresponding call statement:
c
c  tchf   Temperature at critical heat flux
c  tmsfb  Minimum stable film boiling temperature
c  qchf   Critical heat flux
c  qmsfb  Heat flux at tmsfb
c
c
      data gcon/9.8066/
      data pi/3.1416/
      hvfc=0.0
      hlfc=0.0
      hvfilm=0.0
      hlnb=0.0
      chfr=1.0
      qchf = 0.
      qv=0.0
      ql=0.0
      ihtr=0
      if (tw.eq.0.0) return
      vva=abs(vv)
      vla=abs(vl)
      rhd=1./hd
c
c  obtain fluid properties
c  (running time could be shortened by replacing the
c  following call to state and the subsequent computation of
c  hfg, betav, betal, cpv, and cpl by appropriate fits to
```

```
c   these quantities)
       tsat1 = 9.0395*pow(p,.223) + 255.2
       call state(p,tsat1,tsat1,rovs,rols,ev,el,tsat,d0,d00,dtsdp,
      1 deldp,devdp,deldt,devdt,drldp,drvdp,drldt,drvdt,2,ierr)
       tvf = (tw+tv)*.5
       spvv = 1./rovs
       spvl = 1./rols
       hfg = hvs - hls
       betav = -drvdt*spvv
       betal = -drldt*spvl
       cpv = devdt -p*drvdt*spvv*spvv
       cpl = deldt -p*drldt*spvl*spvl
       visv = visvp(tvf)
       visl = vislq(tl)
       cndv = condv(p,tvf)
       cndl = condl(p,tl)
       sig = surten(tl)
c
c   compute quality assuming vl=vv if flow is countercurrent or if
c   abs(vv) .lt. abs(vl)
c
       gv = alp*rov*vva
       gl = (1.-alp)*rol*vla
       g = gv + gl
       if((vv-vl)*vl.le.0.0)go to 5
       x = gv / g
       go to 10
     5 continue
       x=alp*rov/(alp*rov+(1.-alp)*rol)
    10 continue
       hm=x*hv+(1.-x)*hl
       xequil=(hm-hls)/hfg
       x=xequil
       if (x.lt.0.) x=0.
       if(x.gt.1.) x=1.0
c
c           ... determine heat transfer regime ...
c
c   test quality
c
       if(x.ge.0.99)go to 300
c
c   test for cold wall
c
       if(tw.le.tsat)go to 200
       if(iss.ne.0) go to 30
c
c
c   compute minimum stable film boiling temperature
c
       if (p.gt.68.96e5) go to 20
       thn = 581.5 + .01876*sqrt( amax1(p-1.0345e5,0.) )
       go to 25
    20 thn = 630.37 + .00432*sqrt(p-68.96e5)
```

```
   25 continue
      psi=0.0
      if (p.lt.4.827e5) psi = 127.3 - 26.37e-5*p
c     call mpc(tw,rcp,cond)
c     rrkcpw = 1./(rcp*cond)
c  inverse of rocp of zircaloy times conductivity of oxide
      rrkcpw = 3.1e-7 - 1.3e-10*tw
      rkcpl=rol*cndl*cpl
      tl = rrkcpw*rkcpl
crrkcpw becomes negative for tw>2384.6 deg K
      if(tl.lt.0.0) tl=0.0
c
      tmsfb = thn + (thn-tl)*pow(tl,.5) - psi
c
c  test whether twall exceeds tmsfb
c
      if(tw.lt.tmsfb)go to 30
c
c  original BEEST model would now
c  compute film boiling heat transfer coefficient
c
c     call film(hvfc,alp,rov,rol,vva,vla,hd,rhd,tl,tv,tw,tsat,hfg,
c    1 cpv,cpl,p,visv,visl,betav,sig,ihtr,x)
      go to 300
c
c
c  calculate critical heat flux
c
   30 continue
c
      if(ichf.lt.3) call chf(qchf,alp,rov,rol,g,p,x,hd,hfg,sig,
     1          hlin,xequil,gbrit,ichf,hls,ftong)
c
c  determine heat transfer coefficients using Chen correlation
c
      rvisl = 1./visl
      xtti=pow(x/(1.-x),.9) *pow(rol/rov,.5) *pow(visv*rvisl,.1)
      f=1.0
      gx = g
      if(tl.lt.tsat-0.5) go to 32
      if(xtti.gt.0.1)f=2.35*pow(xtti+.213,.736)
      gx = gl
   32 prl = visl*cpl/cndl
      rel = gx*hd*rvisl
      hlf = .023*f*cndl*rhd* pow(rel,.8) *pow(prl,.4)
      retp = rel *pow(f,1.25)*1.e-4
      s=.1
      if(retp.lt.70.0.and.retp.ge.32.5) s=1./(1.+.42*pow(retp,.78))
      if(retp.lt.32.5) s=1./(1.+.12*pow(retp,1.14))
      hs = .00122*s* pow(cndl*cpl/(sig*gcon),.5) *pow(prl,-.29) *
     *        pow(rol,.25) *pow(cpl*rol/(hfg*rov),.24)
      pwall = (.11062558*(tw-255.2))**4.4843049
      hln = hs*pow(tw-tsat,.24)*pow(pwall-p,.75)
c
```

```
c  compute heat flux as predicted by Chen's correlation and
c  compare against the critical heat flux
c  or check critical power ratio
c
      qchen = hlf*(tw-tl) + hln*(tw-tsat)
      if(ichf.eq.3) call cpr(qchf,qchen,xequil,g,p,hd,bl,hls,hlin,hfg)
      if(iss.eq.2) go to 400
      if(qchen.le.qchf) go to 400
      if(iss.eq.1) go to 300
c
c
c  solve the equation
c       hlfc*(tchf-tl) +hlnb*(tchf-tsat)**1.24*(pwall-p)**.75  = qchf
c  for tchf using Newton's iteration
c
      tchf=amax1(tl,tsat+.1)
      do 35 k=1,10
        tcs=amax1(tchf-tsat,0.0)
        pwall=(.11062558*(tchf-255.2))**4.4843049
        dq = qchf - hlf*(tchf-tl) - hs*pow(tcs,1.24)*pow(pwall-p,.75)
        dqdt = hlf + hs*pow(tcs,.24)*pow(pwall-p,.75) *
     *     (1.24 + 3.3632287*tcs*pwall/((tchf-255.2)*(pwall-p)) )
        dtchf = dq/dqdt
        tchf = tchf + dtchf
        if(abs(dtchf).le.0.1)go to 40
        if(tchf.le.tsat) go to 36
   35   continue
   36   if(tchf.lt.tsat) tchf=tsat
   40 continue
      go to 500
c
c          ... individual correlations follow ...
c
c
c  convection to single phase liquid
c  max of Sieder-Tate and McAdams correlations
c
c  Note: McAdams should evaluate properties at a liquid film temp
c
  200 continue
      tl=rol*rol*gcon*betal*cpl*abs(tw-tl)/(visl*cndl)
      hma=.13*cndl*pow(tl,.333333)
      rel=rol*vla*hd/visl
      prl=visl*cpl/cndl
      visw=vislq(tw)
      hst=.023*cndl*rhd *pow(rel,.8) *pow(prl,.33) *pow(visl/visw,.14)
      hlfc=amax1(hma,hst)
      chfr=100.0
      ihtr=1
      if(hma.gt.hst) ihtr=2
      go to 1000
c
c  convection to single phase vapor
c  max of Sieder-Tate and McAdams correlations
```

```
c
c  Note:  McAdams should evaluate properties at a vapor film temp
c
  300 continue
      t1=rov*rov*gcon*betav*cpv*abs(tw-tv)/(visv*cndv)
      hma=.13*cndv*pow(t1,.333333)
      rev=rov*vva*hd/visv
      prv=visv*cpv/cndv
      visw=visvp(tw)
      hst=.023*cndv*rhd *pow(rev,.8) *pow(prv,.33) *pow(visv/visw,.14)
      hvfc=amax1(hma,hst)
      ihtr=9
      if(hma.gt.hst) ihtr=10
      go to 1000
c
c  subcooled or saturated nucleate boiling
c  Chen correlation
c
  400 continue
      hlfc = hlf
      hlnb = hln
      chfr=qchf/qchen
      ihtr=4
      if(tl.lt.tsat-0.5) ihtr=3
      go to 1000
c
c
c  transition boiling
c
  500 continue
      if(tv.gt.tmsfb) go to 300
      call film(hvtb,alp,rov,rol,vva,vla,hd,rhd,tl,tv,tmsfb,tsat,hfg,
     1 cpv,cpl,p,visv,visl,betav,sig,ihtr,x)
      rdtmc = 1./(tmsfb-tchf)
      eps =  (tmsfb-tw)*rdtmc
      eps2 = eps*eps
      qmsfb=hvtb*(tmsfb-tv)
      qv= qmsfb
      ql= qchf
      hlnb= eps2
      dqldtw = -2.*eps*qchf*rdtmc
      dqvdtw = 2.*eps*qmsfb*rdtmc
      hlfc = dqldtw
      hvfc = dqvdtw
      ihtr=5
c
 1000 continue
      return
      end
      subroutine htprm(ql,idown,nzp2,nktb,dtrn,aht,qv,hlnb,
     1 hlfc,hvfc,tw,dtw)
      dimension ql(nzp2,1),idown(1),dtrn(nktb,nzp2,1),
     1 aht(nktb,nzp2,1),qv(nzp2,1),hlnb(nzp2,1),hlfc(nzp2,1),
     2 hvfc(nzp2,1),tw(nzp2,1),dtw(nzp2,1)
```

```
c
      do 339 i=3,5,2
      do 339 j=1,nzp2
      ql(j,i)=0.
      area=0.
      do 340 ktb=1,nktb
      ql(j,i)=ql(j,i)+dtrn(ktb,j,i)*aht(ktb,j,i)
      area=aht(ktb,j,i)+area
340   continue
      if(area.ne.0.)ql(j,i)=ql(j,i)/area
      qv(j,i)=0.
      hlnb(j,i)=0.
      hlfc(j,i)=0.
      hvfc(j,i)=0.
      tw(j,i)=0.
      dtw(j,i)=0.
339   continue

      return
      end
      subroutine pout1(tr,m1,nktb,nzp2,nc,idown,dtrn,hlnb,hlfc,
     1hvfc,k,jmtb,dtw,aht,rad,trn,qorig,xxzz1)
      common/units/ ntty,ninp,nout,ntzc,nres,ndump
      common/rc/ delt,d1,d2,d3,d4,d5,d6,d7,d8,d9,d10,rtnsp,rtnlp,
     1 d11,d12,d13,rtime,d14,d15,d16,d17,d18,d19
      dimension tr(m1,nzp2,nc),idown(nc),dtrn(nktb,nzp2,1),
     1hlnb(nzp2,1),hlfc(nzp2,1),hvfc(nzp2,1),jmtb(1),dtw(nzp2,1),
     2aht(nktb,nzp2,1),rad(nc),trn(m1,nzp2,nc)
      logical long,short
      long=.true.
      short=.true.
      rtm=rtime+0.5*delt
c   short print
      if(rtime.ge.rtnsp) go to 21
      if(rtnsp.lt.rtm) go to 21
      short=.false.
c   long print
21    continue
      if(rtime.ge.rtnlp) go to 22
      if(rtnlp.lt.rtm) go to 22
      long=.false.
22    continue
      if(long) write(nout,1001)rtime,delt,xxzz1
1001  format(29h1primary and wall parameters ,5x,6htime =,f11.6,4h sec
     1  ,6x,16htime step size =,e12.5,4h sec,6x,'fouling coef. = ',
     2 f7.3,//)
      do 12 i=1,nc
      p1=0.
      p2=0.
      if(idown(i).eq.-1) go to 13
      if(.not.(long)) go to 23
      write(nout,106)i,k
      write(nout,101)
23    continue
```

```
      do 11 j=1,jmtb(k)
      phflx=dtw(j,i)*(trn(4*(k-1)+1,j,i)-trn(4*(k-1)+2,j,i))
      p1=phflx*rad(2)/rad(4)*aht(k,j,i)+p1
      p2=dtrn(k,j,i)*aht(k,j,i)+p2
     qorig=qorig+phflx*rad(2)*aht(k,j,i)/rad(4)
      se=aht(k,j,i)*(phflx*rad(2)/rad(4)-dtrn(k,j,i))
      if(phflx.ne.0.)se=se/(phflx*aht(k,j,i)*rad(2)/rad(4))*100.
      if(long) write(nout,102) hlfc(j,i),hlnb(j,i),hvfc(j,i),
     1 (tr(4*(k-1)+ii,j,i),ii=1,4),j,dtrn(k,j,i),phflx,dtw(j,i),se
11    continue
      p1=p1/1000.
      p2=p2/1000.
      if(long) write(nout,103)p1,p2
13    continue
12    continue
106   format(1x,'CHANNEL NUMBER IS i=',i3,'    (ktb=',i3,') ')
101   format(t6,'hlfc',t17,'hlnb',t26,'     hvfc     ',t39,'tprim',
     1 t45,'twprim',t53,'ttube',t60,'twsec',t67,'iz',
     2 t75,'sec hflux',t89,'prim hflux',t105,'hprim',t117,'%(1-E2/E1)')
102    format(3(1pe12.5),4(0pf7.2),i4,3(1pe15.6),0pf9.2)
103   format(1x,t2,'partial primary power output = ',f11.3,1x,'KW',
     1 t72,'partial secondary power input = ',f11.3,1x,'KW',/)
      return
      end
      subroutine pout(tr,fg,dsew,dsns,aht,jmtb,m1,nktb,nzp2,
     1nc,idown)
      common/units/ ntty,ninp,nout,ntzc,nres,ndump
      dimension tr(m1,nzp2,nc),fg(nktb),dsew(nktb,nzp2,nc),
     1  dsns(nktb,nzp2,nc),aht(nktb,nzp2,nc),jmtb(nktb),idown(nc)
      do 14 k=1,nktb
      write(nout,300) k,tr(4*(k-1)+1,1,3),tr(4*(k-1)+1,1,5),
     1 fg(k),jmtb(k),dsew(k,jmtb(k),5)
      do 12 i=1,nc
      if(idown(i).eq.-1) go to 13
      write(nout,106) i
      write(nout,101)
      do 11 j=1,jmtb(k)
      write(nout,102) dsew(k,j,i),dsns(k,j,i),aht(k,j,i),
     1 (tr(4*(k-1)+ii,j,i),ii=1,4),j
11    continue
13    continue
12    continue
14    continue
300   format(/,/,/,/,t2,'INITIAL CONDITIONS AND GEOMETRICAL PARAMETERS
     1 FOR TUBE BANK NO.',i4,/,/,
     2 27h  Inlet Temperature        =,f7.2,/,
     3 27h  Outlet Temperature       =,f7.2,/,
     4 27h  Flow Split Parameter     =,f7.2,/,
     5 27h  Tube Bank bends at j     =,i4,/,
     6 27h  Horizontal Bank Length  =,f7.2)
106   format(/,/,1x,'CHANNEL NUMBER IS i=',i3,/)
101   format(t6,'dsew',t17,'dsns',t26,'sec ht area',t39,'tprim',
     1 t45,'twprim',t53,'ttube',t60,'twsec',t67,'level')
102   format(3(1pe12.5),4(0pf7.2),i4)
```

```
      subroutine state(p,tv,tl,rov,rol,ev,el,tsat,hvs,hls,dtsdp,
     1   deldp,devdp,deldt,devdt,drldp,drvdp,drldt,drvdt,iop,ierr)
c
c
c     subroutine state calculates the state dynamic properties of
c     water. the present version uses fits due to bill rivard of
c     group t-3 of the lasl theoretical division.
c     taken from TRAC and recoded to improve efficiency.
c     the compressed liquid properties have been reprogrammed
c     by john kelly of mit(5,79).
c     SI units are used
c
c     input variables
c        1. p    pressure
c        2. tl   temperature of the liquid
c        3. tv   temperature of the vapor
c        4. iop option selector - not in present version
c
c     output variables
c        1. ev      internal energy of the vapor
c        2. el      internal energy of the liquid
c        3. tsat    saturation temperature
c        4. hvs     vapor saturation enthalpy
c        5. hls     liquid saturation enthalpy
c        6. rol     density of the liquid
c        5. rov     density of the vapor
c        6. dtsdp   derivative of tsat wrt pressure
c        7. deldp   derivative of tl wrt pressure
c        8. devdp   derivative of tv wrt pressure
c        9. deldt   derivative of el wrt tl
c       10. devdt   derivative of ev wrt tv
c       11. drldp   derivative of rol wrt pressure
c       12. drvdp   derivative of rov wrt pressure
c       13. drldt   derivative of rol wrt tl
c       14. drvdt   derivative of rov wrt tv
c       15. ierr    error flag (input variable out of range)
c
c     constants used in fits
c
c     for tsat, cps
      data tsc1,tsc2, tsexp  /9.0395, 255.2, 0.223/
      data cps1,cps2, cpsexp  /9.5875e2, .00132334, -0.8566/
c        cps2 = -cpsexp * tcrinv
c     for es, gams  if p < 20 bars
      data g11,g12,g13  /2.6194106e6, -4.995e10, 3.403e5/
      data g14,g15,g16  /1.0665544, 1.02e-8, -2.548e-15/
c             g11,g14 are adjusted so that es resp. gams jumps less than
c                1 part in 1.e-8 across p = 20 bars.
      data g17 /-5.096e-15/
c        g17 = 2.* g16
c     for es, gams  if p > 20 bars
      data g21,g22,g23  /2.5896e6, 6.350e-3, -1.0582e-9/
      data g24,g25,g26  /1.0764, 3.625e-10, -9.063e-17/
      data g27,g28  /-2.1164e-9, -18.126e-17/
```

```
c        g27 = 2.* g23,   g28 = 2.* g26
c
c    for hls and hvs
       data  hl0,hl1,hl2,hl3,hl4,hl5/5.7474718e5,2.0920624e-1,
     1    -2.8051070e-8,2.3809828e-15,-1.0042660e-22,1.6586960e-30/
       data  hv0,hv1,hv2,hv3,hv4/2.7396234e6,3.758844e-2,
     1        -7.1639909e-9,4.2002319e-16,-9.8507521e-24 /
c
      data p20b /2.0e6/
      data tcrit /647.3/
      data tcrinv /.00154488/
      data cc,cci,ccm  /1.3, .76923, 0.3/
c
c      for rol if t < 576.5 deg K
c
      data rl0,rl1,rl2,rl3 /1735.3320,-4.6406842,1.0431090e-2,
     1                          -9.4367085e-6/
      data rl22,rl33 /2.086218e-2,-2.8310126e-5 /
c       rl22 = 2.*rl2       rl33 =3.*rl3
c
c       for rol   if t > 576.5 deg K
c

      data rh0,rh1,rh2,rh3,rh4 /-1.1755984e6,8.1437361e3,-2.1136559e1,
     1          2.4381598e-2,-1.0549747e-5 /
      data rh22,rh33,rh44 /-4.2273118e1,7.3144794e-2,-4.2198988e-5/
c       rh22 = 2.*rh2    rh33 = 3.*rh3     rh44 = 4.*rh4
c
      data rp0,rp1,rp2 /-14.643890,1.1283357e-3,1.2670366e-2/
c
      data sp0,sp1,sp2,sp3 / -42.0218,.2116,-4.4587e-4,3.251e-7/
      data sp22,sp33 /-8.9174e-4,9.753e-7/
c
c    for el  if tl < 576.5 deg K
      data sl0,sl1,sl2,sl3,sl4 /-460.26818e3,-2.8634045e3,27.450693,
     1          -4.8108323e-2,3.2059316e-5/
      data sl22,sl33,sl44 /54.901386,-.14432497,1.2823726e-4/
c      sl22 = 2.* sl2,   sl33 = 3.* sl3
c    for el  if tl > 576.5 deg K
      data sh0,sh1,sh2,sh3,sh4 /1.2426455e9,-8.6082251e6,2.2364564e4,
     1          -2.5815959e1,1.1178766e-2/
      data sh22,sh33,sh44 /4.4729128e4,-77.447877,4.4715064e-2/
c      sh22 = 2.* sh2,   sh33 = 3.* sh3
c
c    for vapor
      data a11,a12,a13 /1.2959e-3,  593.59, 1.6847e-3/
c
      data  half,zero,one,two  /0.5, 0., 1., 2./
c
c  ----------------------------------------
c
c     check that p, tl, tv, are within range of fits
c
       tlsave = tl
```

```
      if(tl.gt.647.) tl = 647.
     if (p.ge.1.0e+3.and.p.le.190.0e+5) go to 5
       ierr = 1
       return
   5 if (tl.ge.280.0.and.tl.le.647.0) go to 10
       ierr = 2
       return
  10 if(tv.ge.280.0) go to 20
       ierr = 3
       return
  20 ierr = 0
c
c    calculate saturation properties
c
c       1. tsat    saturation temperature
c       2. dtsdp    derivative of tsat wrt pressure
c       3. es      saturation internal energy
c       4. dpes    derivative of es wrt pressure
c       5. gams    gamma sub s
c       6. dpgams  derivative of gams wrt pressure
c       7. cps     c sub ps
c       8. dpcps   derivative of cps wrt pressure
c       9. gamsm   gams-one
c      10. hvs     vapor saturation enthalpy
c      11. hls     liquid saturation enthalpy
c
      tsat = tsc1* p**tsexp
      hvs = hv0 + p*(hv1 + p*(hv2 + p*(hv3 + p*hv4)))
      hls = hl0 + p*(hl1 + p*(hl2 + p*(hl3 + p*(hl4 + p*hl5))))
      pinv = one/ p
      dtsdp = tsat*tsexp*pinv
      tsat = tsat + tsc2
c
      tl = one - tsat*tcrinv
      cps = cps1* tl**cpsexp
      dpcps = cps2*cps/tl *dtsdp
c
      if (p.gt.p20b) go to 150
        t2 = one/ (g13+p)
        t1 = t2*g12
        es = g11 + t1
        dpes = -t1*t2
        gams = g14 + p*(g15 + p*g16)
        dpgams = g15+g17*p
        go to 200
 150 continue
        es = g21+(g23*p+g22)*p
        dpes = g22+g27*p
        gams = g24+(g26*p+g25)*p
        dpgams = g25 + g28*p
 200 gamsm = gams - one
c
c
c     calculate liquid properties
```

```
c
c
c        1. internal energy and its derivatives
c
      dp=p - 150.e5
      deldp = -exp.(sp0+tl*(sp1+tl*(sp2+tl*sp3)))
      del = deldp * dp
      if (tl.ge.576.5) go to 210
        el = sl0 + tl*(sl1 + tl*(sl2 + tl*(sl3+tl*sl4))) + del
        deldt = sl1 + tl*(sl22 + tl*(sl33+tl*sl44))
     1            + del * (sp1 + tl*(sp22 + tl*sp33))
        go to 220
  210 continue
        el =(sh0 + tl*(sh1 + tl*(sh2 + tl*(sh3+ tl*sh4)))) + del
        deldt = (sh1 + tl*(sh22 + tl*(sh33 + tl*sh44)))
     1          + del * (sp1 + tl*(sp22 +tl*sp33))
  220 continue
c
c        2. density and its derivatives
c
      drldp = exp(rp0 + rp1*exp(rp2 * tl))
      drl = drldp * dp
          if (tl.ge.576.5) go to 230
      rol = rl0 + tl*(rl1 + tl*(rl2+ tl*rl3)) + drl
      drldt = rl1 + tl*(rl22 + tl*rl33) + drl*rp1*rp2*exp(rp2*tl)
      go to 240
 230  continue
      rol = rh0 + tl*(rh1 + tl*(rh2 + tl*(rh3 + tl*rh4))) + drl
      drldt = rh1 + tl*(rh22 + tl*(rh33 + tl*rh44)) +
     1        drl*rp1*rp2*exp(rp2*tl)
240   continue
       tl = tlsave
c
c     calculate vapor properties
c
      dt = tv-tsat
      if (dt.le.zero) go to 250
c
c     superheated vapor
c
c        1. beta a working parameter
c        2. capk a working parameter
c        3. dbetap  derivative of beta wrt pressure
c        4. dcapkp  derivative of capk wrt pressure
c        5. devdt
c        6. devdp
c        7. rov
c        8. drvde
c        9. drvdp
c
      t1 = one/(a11*cps-one)
      t1sq = t1*t1
      beta = tsat*tsat*(one - t1sq)
      t2 = tsat*t1
```

```
      de = a12*(dt+sqrt(tv*tv-beta)-t2)
      ev = es + de
      capk = a13*de+tsat+t2
      dbetap = two*(beta*dtsdp+t2*t2*t2*a11*dpcps)/tsat
      dcapkp = -a13*dpes + (one + t1)*dtsdp
     1  -tsat*a11*t1sq*dpcps
      t3 = one-beta/(capk*capk)
      devdt = one/(half*t3*a13)
      devdp = -half*(t3*dcapkp+dbetap/capk)*devdt
      t4 = one/(gamsm*es+ccm*de)
      rov = p*t4
      drvde = -rov*ccm*t4
      drvdt = drvde*devdt
      drvdp = rov*(pinv-(es*dpgams+(gamsm-ccm)*dpes)*t4)
     1   + drvde*devdp
      go to 300
  250 continue
c
c     subcooled vapor
c
      devdt = cps * cci
      de = dt * devdt
      ev = es + de
      t1 = one/ cps
      devdp = -(dtsdp -cc*t1*(dpes +de*dpcps*t1) )*devdt
      t1 = one/ gamsm
      t2 = one/ ev
      rov = p *t1*t2
      drvde = -rov *t2
      drvdt = drvde * devdt
      drvdp = rov *(pinv - dpgams*t1) + drvde*devdp
c
  300 continue
      return
      end
```

```
 subroutine deriv(p,tsat,ro1,ro2,ev,el,dtsdp,deldp,devdp,deldt,
1devdt,dr1dp,drvdp,drvdt,dr1dt,dh2dt,dh2dp,ro2dh,ro2dp,dh1dt,dh1dp
2,ro1dh,ro1dp,hf,hg,drgdp,drfdp,dhgdp,dhfdp,rog,rof)
 dh2dt=deldt-p*dr1dt/ro2**2.
 dh2dp=deldp+1./ro2-p*dr1dp/ro2**2.
 ro2dh=dr1dt/dh2dt
 ro2dp=dr1dp-dr1dt*dh2dp/dh2dt
 dh1dt=devdt-p*drvdt/ro1**2.
 dh1dp=devdp+1./ro1-p*drvdp/ro1**2.
 ro1dh=drvdt/dh1dt
 ro1dp=drvdp-drvdt*dh1dp/dh1dt
 call state(p,tsat,tsat,rog,rof,egs,els,d00,do1,d02,dtsdp,
1deldp,devdp,deldt,devdt,dr1dp,drvdp,dr1dt,drvdt,2,ierr)
 hf=p/rof+els
 hg=p/rog+egs
 drgdp=drvdp+drvdt*dtsdp
 drfdp=dr1dp+dr1dt*dtsdp
 dh1dt=deldt-p*dr1dt/rof**2.
 dhvdt=devdt-p*drvdt/rog**2.
 dh1dp=deldp+1./rof-p*dr1dp/rof**2.
 dhvdp=devdp+1./rog-p*drvdp/rog**2.
 dhgdp=dhvdp+dhvdt*dtsdp
 dhfdp=dh1dp+dh1dt*dtsdp
 return
 end
```

```
      subroutine ptemp(rcp,rad,tr,trn,dtrn,tw,dtw,hvfc,hlnb,hlfc,
     1 tvn,tln,tsat,dz,delt,fg,dsew,dsns,aht,jmtb,nktb,nz,nzp2,
     2 ml,ktb,nc,ql,qv,rtime,foul)
c
      common/sgpt1/pprim,pmflx,tpin,tpout,xmout,tauh,tauc,toutn,z1,z2,
     1 c(15,12)
      common /force/botfac(30),yb(30),topfac(30),yt(30),tinfac(30),
     1   ytemp(30),qfac(30),yq(30),nb,nt,ntemp,nq,
     2   ptfac(30),npt,ypt(30),pgfac(30),npg,ypg(30)
c
      dimension rcp(nc),rad(nc),tr(ml,nzp2,nc),dtrn(nktb,nzp2,1),
     1 trn(ml,nzp2,nc),tw(nzp2,1),dtw(nzp2,1),hvfc(nzp2,1),
     2 hlnb(nzp2,1),hlfc(nzp2,1),tvn(nzp2,1),tln(nzp2,1),tsat(nzp2,1)
     3,dz(1),fg(1),dsew(nktb,nzp2,1),dsns(nktb,nzp2,1),
     4 aht(nktb,nzp2,nc),jmtb(1),ql(nzp2,1),qv(nzp2,1),f(4,2)
      data rl0,rl1,rl2,rl3/1735.332,-4.6406842,1.043109e-02,
     ,-9.4367085e-06/
      data rl22,rl33/2.086218e-2,-2.83101265e-5/
      data sl0,sl1,sl2,sl3,sl4/-460.26818e3,-2.8634045e3,
     ,27.450693,-4.8108323e-2,3.2059316e-5/
      data sl22,sl33,sl44/54.901386,-0.14432497,1.2823726e-4/
      data sp0,sp1,sp2,sp3/-42.0218,0.2116,-4.4587e-4,3.251e-7/
      data sp22,sp33/-8.917e-4,9.753e-7/
      data rh0,rh1,rh2,rh3,rh4/-1.1755984e6,8.1437361e3,-2.1136559e1,
     ,2.4381598e-2,-1.0549747e-5/
      data rh22,rh33,rh44/-4.2273118e1,7.3144794e-2,-4.2198988e-5/
      data rp0,rp1,rp2/-14.64389,1.1283357e-3,1.2670366e-2/
       data sh1,sh22,sh33,sh44/-8.6082251e6,4.4729128e4,
     1 -77.447877,4.4715064e-2/
      data xlamb,xlaml/0.8,0.33333/
      xss=(3.*rad(4)+rad(3))*(rad(4)-rad(3))/(8.*delt)
      xpp=(rad(3)-rad(2))*(3.*rad(2)+rad(3))/(8.*delt)
      ypp=0.5*(rad(3)+rad(2))/(rad(3)-rad(2))
      xm2=(rad(4)-rad(3))*(3.*rad(3)+rad(4))/(8.*delt)
      xm3=(rad(3)-rad(2))*(3.*rad(3)+rad(2))/(8.*delt)
      ymm=(rad(4)+rad(3))/(rad(4)-rad(3))/2.
c   jlim is the topmost cell in the heat adition region
      jlim=jmtb(ktb)
      jmax=jmtb(1)+nktb
c    sweep does not include downcomer cells: i=1,2,4,6
c     for an arrangement in which hot side and cold side
c    are not cells 3 and 5 respectively, the values of
c    k,jns,and iew below must be corrected
      do 100 i=3,5,2
c   sweep does not include cells in the riser region:j>lim
c  k=1 means hot side : k=2 means cold side
      k=(i-1)/2
      do 100 j3=1,jlim
      j=(k-1)*(jlim+1-j3)+(2-k)*j3
      delz=dz(j)
      if(j.ge.jlim)delz=dsew(ktb,j,i)
      if(j.eq.1)delz=3.75
c  Read in primary boundary conditions
      if(nq.eq.0) go to 300
```

```
      call table(pprim,rtime+delt,qfac,yq,nq)
      pprim=pprim*6894.7448
300   continue
      tpin=tr(4*(ktb-1)+1,1,3)
      if(npt.eq.0) go to 301
      call table(phold,rtime,ptfac,ypt,npt)
      call table(phnew,rtime+delt,ptfac,ypt,npt)
      tpin=phnew+(phnew-phold)*tauh/delt
301   continue
       if(npg.ne.0) call table(pmflx,rtime+delt,pgfac,ypg,npg)
c  check j and i below
       jns=(k-1)+(k-2)
       iew=(k-1)*2
c  section missing here
       t=tr(4*(ktb-1)+1,j,i)
       drldp=exp(rp0+rp1*exp(rp2*t))
       drl=drldp*(pprim-150.e5)
       deldp=-exp(sp0+t*(sp1+t*(sp2+t*sp3)))
       del=deldp*(pprim-150.e5)
       if(t-576.5)220,230,230
220    rop=rl0+t*(rl1+t*(rl2+t*rl3))+drl
       drldt=rl1+t*(rl22+t*rl33)+drl*rp1*rp2*exp(rp2*t)
       deldt=sl1+t*(sl22+t*(sl33+t*sl44))+del*(sp1+t*(sp22
     ++t*sp33))
       go to 240
230    rop=rh0+t*(rh1+t*(rh2+t*(rh3+t*rh4)))+drl
       drldt=rh1+t*(rh22+t*(rh33+t*rh44))+drl*rp1*rp2*exp(rp2*t)
       deldt=(sh1+t*(sh22+t*(sh33+t*sh44)))+del*(sp1+
     +t*(sp22+t*sp33))
240    cp=deldt-pprim*drldt/(rop**2.)
       tpm=(tr(4*(ktb-1)+2,j,i)+tr(4*(ktb-1)+3,j,i))/2.-273.15
       rcp(2)=0.2409353*tpm*tpm+1216.0654*tpm+3757094.7
       cond2=18.8+0.016*(tpm-300.)
       if(foul.ne.0.) cond2=cond2/foul
       tms=(tr(4*(ktb-1)+3,j,i)+tr(4*(ktb-1)+4,j,i))/2.-273.15
       rcp(3)=0.2409353*tms*tms+1216.0654*tms+3757094.7
       cond3=18.8+0.016*(tms-300.)
       if(foul.ne.0.) cond3=cond3/foul
       cond=condl(pprim,t)
       pr=vislq(t)*cp/cond
       re=2.*rad(2)*fg(ktb)*pmflx/vislq(t)
       twll=tr(4*(ktb-1)+2,j,i)
       xmew=(vislq(t)/vislq(twll))**0.14
       dtw(j,i)=0.027*(re**xlamb)*(pr**xlaml)*cond/(2.*rad(2))*xmew
c Primary ht coef. must be set to zero at level 1
       dtw(1,i)=0.
       xp=rcp(2)*xpp
       yp=cond2*ypp
       xm=rcp(2)*xm2+rcp(3)*xm3
       ym=cond3*ymm
       xs=rcp(3)*xss
       det=(xs+ym)*(xp*yp+(xp+yp)*xm)+ym*xs*(xp+yp)
       ap=((xs+ym)*(xm+yp)+xs*ym)*rad(2)/det
       bp=-yp*ym*rad(4)/det
```

```
      cpn=ap*xp*tr(4*(ktb-1)+2,j,i)/rad(2)-bp*xs*tr(4*(ktb-1)+4,j,i)/
     1rad(4)+yp*xm*(xs+ym)*tr(4*(ktb-1)+3,j,i)/det
      as=yp*ym*rad(2)/det
      bs=-((xp+yp)*(xm+ym)+xp*yp)*rad(4)/det
      csn=-bs*xs*tr(4*(ktb-1)+4,j,i)/rad(4)-bp*xp*tr(4*(ktb-1)+2,j,i)/
     1rad(4)+ym*xm*(xp+yp)*tr(4*(ktb-1)+3,j,i)/det
      at=rop*cp/delt
      aew=0.
      ans=0.
      if(dsew(ktb,j,i).ne.0.) aew=fg(ktb)*cp*pmflx/dsew(ktb,j,i)
      delta=(dsns(ktb,j,i)+delz)/2.
      if(dsns(ktb,j,i).ne.0.)ans=fg(ktb)*cp*pmflx/(delta)
      awl=2./rad(2)
      dhtr=delz+dsew(ktb,j,i)+dsns(ktb,j,i)
      vs1=awl/(1.+ap*dtw(j,i))
      v=at+aew+ans+vs1*dtw(j,i)*delz/dhtr
      vs2=vs1*dtw(j,i)/v
      vew=aew/v
     vns=ans/v
      vs=vs2*bp*delz/dhtr
      vt=at/v-vs2*ap*dtw(j,i)*delz/dhtr

81    continue

      hs=hvfc(j,i)+hlnb(j,i)+hlfc(j,i)
      bws0=-bs/(1.-bs*hs)
      aws0=as/(1.-bs*hs)
        cws0=-bws0*(qv(j,i)+ql(j,i)+csn/bs)
      tws0=bws0*(hlfc(j,i)*tln(j,i)+hlnb(j,i)*tsat(j,i)+
     1hvfc(j,i)*tvn(j,i))+aws0*dtw(j,i)*(tr(4*(ktb-1)+1,j,i)
     1-tr(4*(ktb-1)+2,j,i))+cws0
      bwp0=1./(1.+ap*dtw(j,i))
     twp0=(ap*dtw(j,i)*tr(4*(ktb-1)+1,j,i)+bp*dtrn(ktb,j,i)+cpn)*bwp0
      vp0=awl*dtw(j,i)*twp0*delz/(dhtr*v)
      x1=aws0*dtw(j,i)*bwp0*bp
      jota=j+jns
      tew=trn(4*(ktb-1)+1,j,i-iew)
      tns=tpin
      if(jota.gt.0)  tns=trn(4*(ktb-1)+1,jota,i)
     if(jota.gt.0)  vp0=vp0+dsew(ktb,j,i)*c(j,i-iew)/(dhtr*v)
     ++dsns(ktb,j,i)*
     1c(jota,i)/(dhtr*v)
c  computing tp* (advanced prim temp. with q"sec taken explicitly)
      trn(4*(ktb-1)+1,j,i)=vp0+vt*tr(4*(ktb-1)+1,j,i)+vns*tns+vew*tew
     trn(4*(ktb-1)+1,1,3)=tpin
      c(j,i)=awl*dtw(j,i)*((twp0+ap*dtw(j,i)*bwp0*(
     1trn(4*(ktb-1)+1,j,i)
     1-tr(4*(ktb-1)+1,j,
     ,i)))-trn(4*(ktb-1)+1,j,i))
c computing twp* (* has analogous meaning)
      trn(4*(ktb-1)+2,j,i)=twp0+ap*dtw(j,i)*bwp0*
     1(trn(4*(ktb-1)+1,j,i)-tr(4*(ktb-1)+1,j,i))
c determining the first guess for advanced sec. side wall temp.
      trn(4*(ktb-1)+4,j,i)=tws0+aws0*dtw(j,i)*
```

```
     1(trn(4*(ktb-1)+1,j,i)-trn(4*(ktb-1)+2,j,i)-
     2tr(4*(ktb-1)+1,j,i)+tr(4*(ktb-1)+2,j,i))
      trn(4*(ktb-1)+3,j,i)=(xm*tr(4*(ktb-1)+3,j,i)+
     1 ym*trn(4*(ktb-1)+4,j,i)+yp*trn(4*(ktb-1)+2,j,i))/(xm+ym+yp)
c   update old primary and wall temperatures
      tr(4*(ktb-1)+1,j,i)=trn(4*(ktb-1)+1,j,i)
      tr(4*(ktb-1)+2,j,i)=trn(4*(ktb-1)+2,j,i)
      tr(4*(ktb-1)+3,j,i)=trn(4*(ktb-1)+3,j,i)
      tr(4*(ktb-1)+4,j,i)=trn(4*(ktb-1)+4,j,i)
100   continue
c  Outlet temperatures
      if(ktb.gt.1) go to 10
      toutn=tpout
      tpout=0.
      xmout=0.
10    continue
      tpout=tpout+cp*fg(ktb)*trn(4*(ktb-1)+1,1,5)*aht(ktb,1,5)
       xmout=xmout+fg(ktb)*aht(ktb,1,5)*cp
      if(ktb.ne.nktb) return
      xinew=tpout/xmout
c  RDT lag
      taucl=tauc
      if(npt.eq.0) taucl=0.
      tpout=(xinew*delt+toutn*taucl)/(taucl+delt)
      return
      end
```

```
      subroutine sgbcp (idown,pn,rovn,roln,evn,eln,alpn,tvn,tln,tsat,
     1            vvz,vlz,arz,vol,dz,delt,m1,m2,m3,m4,rtime,rtnlp,qb)
c    links recirculation model to thermit
c
c    this routine is for SG analysis. For a chosen cell lay out cells
c    representing downcomer  must be marked with idown(i)=-1, after no5
      common/sg3/qsl(30),qsw(30),konce,dthtr
      common/sg1/pi,pri,nmini,nzli,t2ai,dpdzi,tfeed,wfeed
       common/sg/ vd,xl2,vfeed,xfeed,nonce,ksh,nfeed,nzlvl,nmin,ti,
     1 pdi,t2i,ro2i,h2i,vdi,wd,xr,wr,roavg,rlevl,pd,pr,ro2,t2,h2,v2,
     2 dpdz,dro2,dpris,dpsep,t1,t1i,ro1,h1,t2a,h2a,ro2a,
     3 t2b,h2b,ro2b,p,ireci,xl2i,hris,vt,nsep
      common/force/botfac(30),yb(30),topfac(30),yt(30),tinfac(30),
     1              ytemp(30),qfac(30),yq(30),nb,nt,ntemp,nq,
     2   ptfac(30),npt,ypt(30),pgfac(30),npg,ypg(30)
      common/ic/ nstep,nitmax,nitno,iitmax,iitot,iic,kred,nm,ntc,
     1   nc,nrods,nz,nr,ncp,nzp,nzp2,iflash,itb,ibb,icpu,iwft,ivec,
     2   nodes,ndml,ncf,ncc,ng,iht, iss,iqss,itam,ires,idump,ntitle,
     3   itwmax,jtwmax,itrmax,jtrmax,imchfr,jmchfr,ichf,ifintr,ierr,lerr
     4   ,imixm,imixe,iafm,itfm,igfm,nktb,jhem
      logical lerr
      dimension pn(m2,m4),rovn(m2,m4),roln(m2,m4),evn(m2,m4),
     1          eln(m2,m4),alpn(m2,m4),tvn(m2,m4),tln(m2,m4),
     2          tsat(m2,m4),vvz(m1,m4),vlz(m1,m4),arz(m1,m4),
     3          vol(m3,m4),dz(m2),idown(m4)
      data kih2/2/
      data kih3/3/
      idon(v)=int(.51+sign(.5,v))
      if(nt.lt.2) return
c    initialize sumation dumies
      zlvl=0.
      rlevl=0.
      wd=0.
      wr=0.
      xnum=0.
      xden=0.
      ronum=0.
      roden=0.
       idome=0
c    Initialization of Steam Dome Model
c    once only calculations
      if(nonce.eq.1) go to 1
      konce=0
      call wlvl(vd,xl2,vfeed,xfeed)
      call state(p,tvn(nzp2,1),tln(nzp2,1),d0,d1,d2,d3,ts,d16,h2a,
     1 d4,d5,d6,d7,d8,d9,d10,d11,d12,2,ierr)
      t1=ts
      t2a=ts
1     nonce=1
c    transfer ref. lev.,w.lev. index,fw. lev. index and p. grad
      do 40 k=3,nzp
      rlevl=rlevl+dz(k)
      if(rlevl.le.xl2) nzlvl=k
      if(rlevl.le.xfeed) nfeed=k
```

```
40    continue
      rlevl=rlevl+dz(nzp2)
       v2a=10.
       do 33 j=1,nz
       do 34 i=1,nc
       if(idown(i).ge.0) go to 35
       v=v+vol(j,i)*2.
35     continue
34     continue
       if(v.ge.(vd-v2a)) go to 36
33     continue
36     jmin=j
          nmin=min(nfeed,nzlvl,nzp2,jmin)
      rlevl=rlevl-dz(nzp2)/2.
      do 41 kl=3,nzlvl
41    zlvl=dz(kl)+zlvl
      zlvl=zlvl-dz(nzlvl)/2.
c ******************************************************
       dpdz1=(pn(nzlvl,1)-pn(nzp2,1))/(rlevl-zlvl)
       dpdz2=(pn(nzlvl,2)-pn(nzp2,2))/(rlevl-zlvl)
       dpdz4=(pn(nzlvl,4)-pn(nzp2,4))/(rlevl-zlvl)
       dpdz6=(pn(nzlvl,6)-pn(nzp2,6))/(rlevl-zlvl)
       dpdz=(dpdz1+dpdz2+dpdz4+dpdz6)/4.
c ******************************************************
c  calculate pressure correction term
       pc1=-p+pn(nzp2,1)+dpdz1*(rlevl-xl2)
       pc2=-p+pn(nzp2,2)+dpdz2*(rlevl-xl2)
       pc4=-p+pn(nzp2,4)+dpdz4*(rlevl-xl2)
       pc6=-p+pn(nzp2,6)+dpdz6*(rlevl-xl2)
c   recuperate previous values in case of tmstp reduction
       if(.not.(lerr))go to 45
      p=pi
      vd=vdi
      nmin=nmini
      dpdz=dpdzi
      nzlvl=nzli
      xl2=xl2i
      t2a=t2ai
      tl=tli
      do 55 i=1,nc
      itype=idown(i)
      if(itype) 65,75,75
65    continue
      pn(nzp2,i)=pdi
      do 46 k=nmin,nzp2
      tln(k,i)=t2i
      tvn(k,i)=t2i
      call state(pn(k,i),tvn(k,i),tln(k,i),rovn(k,i),
     1 roln(k,i),evn(k,i),eln(k,i),tsat(k,i),
     2 d0,d00,d1,d2,d3,d4,d5,d6,d7,d8,d9,2,ierr)
46    continue
      go to 55
75    pn(nzp2,i)=pri
      call state(pn(nzp2,i),tvn(nzp2,i),tln(nzp2,i),
```

```
     1 rovn(nzp2,i),roln(nzp2,i),evn(nzp2,i),eln(nzp2,i),
     2 tsat(nzp2,i),d0,d00,d1,d2,d3,d4,d5,d6,d7,d8,d9,
     3 2,ierr)
55      continue
45      continue
c   determine internal BC's
        do 10 i=1,nc
        jv=nzp+1-idon(vvz(nzp,i))
        jl=nzp+1-idon(vlz(nzp,i))
        qual=alpn(jv,i)*rovn(jv,i)*vvz(nzp,i)
        g=qual+(1.-alpn(jl,i))*roln(jl,i)*vlz(nzp,i)
        itype=idown(i)
        if(itype) 20,20,30
20      wd=wd+roln(nzlvl,i)*vlz(nzlvl-1,i)*arz(nzlvl-1,i)
        go to 10
30      xnum=xnum+qual
        xden=xden+g
        wr=wr+g*arz(nzp,i)
        ronum=ronum+(alpn(jv,i)*rovn(jv,i)+(1.-alpn(jl,i))
       1*roln(jl,i))*vol(jl-1,i)
        roden=roden+vol(jl-1,i)
10      continue
        wd=-2.*wd
c *****************************************************
        xr=xnum/xden
        if(xr.lt.0.) xr=1.
        xr=amin1(1.,xr)
        wr=2.*wr
        roavg=ronum/roden
         sep=float(nsep)/4.
c   for different riser pressures on hot(ch3) + cold(ch5) sides
         w3=((1.-alpn(nzp,3))*roln(nzp,3)*vlz(nzp,3)+
       +alpn(nzp,3)*rovn(nzp,3)*vvz(nzp,3))*arz(nzp,3)/sep
         ro3=(1.-alpn(nzp,3))*roln(nzp,3)+alpn(nzp,3)*rovn(nzp,3)
         q3=wr/166./roavg
         ro3=roavg
         w5=((1.-alpn(nzp,5))*roln(nzp,5)*vlz(nzp,5)+
       +alpn(nzp,5)*rovn(nzp,5)*vvz(nzp,5))*arz(nzp,5)/sep
         ro5=(1.-alpn(nzp,5))*roln(nzp,5)+alpn(nzp,5)*rovn(nzp,5)
         ro5=roavg
         q5=wr/166/roavg
         dpsp3=ro3*(4494.508*q3**2.+30.5)/20.
         dpsp5=ro5*(4494.508*q5**2.+30.5)/20.
         dpsp3=dpsp3*sign(1.,w3)
         dpsp5=dpsp5*sign(1.,w5)
c *****************************************************
c   determine new BC's
        pd=pn(nzp2,1)
        t2=tln(nmin,1)
        ro2=roln(nmin,1)
        h2=pn(nmin,1)/roln(nmin,1)+eln(nmin,1)
        ro2bi=ro2
        if(xl2.le.xfeed) go to 47
        t2=t2a
```

```
       call state(p,t1,t2a,d0,ro2a,d1,e2a,d3,d4,d5,d6,d7,
     1 d8,d9,a10,d11,d12,d13,d14,2,ierr)
      h2a=p/ro2a+e2a
      h2=h2a
      ro2=ro2a
47     continue
       do 48 i=1,nc
      tln(1,i)=tln(kih2,i)
      tvn(1,i)=tvn(kih2,i)
48    continue
      call recirc (delt,rtime,idome,ro2bi,v2a,qb)
      do 50 i=1,nc
      itype=idown(i)
      if(itype) 60,70,70
60    continue
c *****************************************************
       data grav/9.81/
       data fact/1./
       pn(nzp2,1)=p+(dpdz1+dro2*grav)*(xl2-rlevl)+pc1*fact
       pn(nzp2,2)=p+(dpdz2+dro2*grav)*(xl2-rlevl)+pc2*fact
       pn(nzp2,4)=p+(dpdz4+dro2*grav)*(xl2-rlevl)+pc4*fact
       pn(nzp2,6)=p+(dpdz6+dro2*grav)*(xl2-rlevl)+pc6*fact
c *****************************************************
      do 43 k=nmin,nzp2
      tln(k,i)=t2
      tvn(k,i)=t2
      call state (pn(k,i),tvn(k,i),tln(k,i),rovn(k,i),
     1            roln(k,i),evn(k,i),eln(k,i),tsat(k,i),
     2             d0,d00,d1,d2,d3,d4,d5,d6,d7,d8,d9,2,ierr)
43    continue
      go to 50
c *******************************************************
70     continue
       pn(nzp2,3)=p+dpsp3
       pn(nzp2,5)=p+dpsp5
       a1=float(idon(vlz(nzp,i)))
c *******************************************************
      tln(nzp2,i)=a1*tln(nzp2,i)+(1.-a1)*tvn(nzp2,i)
      call state (pn(nzp2,i),tvn(nzp2,i),tln(nzp2,i),rovn(nzp2,i),
     1            roln(nzp2,i),evn(nzp2,i),eln(nzp2,i),tsat(nzp2,i),
     3             d0,d00,d1,d2,d3,d4,d5,d6,d7,d8,d9,2,ierr)
50    continue
      ti=rtime
      ksh=1
      return
      end
      subroutine recirc (dt,rtime,idome,ro2bi,v2a,qb)
      common/sg3/qsl(30),qsw(30),konce,dthtr
      common/sg1/pi,pri,nmini,nzli,t2ai,dpdzi,tfeed,wfeed
       common/sg/ vd,xl2,vfeed,xfeed,nonce,ksh,nfeed,nzlvl,nmin,ti,
     1 pdi,t2i,ro2i,h2i,vdi,wd0,xr,wr,roavg,xlr,pd,pr,ro2,t2,h2,v2,
     2 dpdz,dro2,dpris,dpsep,t1,t1i,ro1,h1,t2a,h2a,ro2a,
     3 t2b,h2b,ro2b,p,ireci,xl2i,hris,vt,nsep
c
```

```
c  all variable groups preceded by 'c *' must be input by user
c* define time dependent external bc functions
      common/force/botfac(30),yb(30),topfac(30),yt(30),tinfac(30),
     1             ytemp(30),qfac(30),yq(30),nb,nt,ntemp,nq,
     2   ptfac(30),npt,ypt(30),pgfac(30),npg,ypg(30)
      data nzzz/9/.
c     store initial values for possible tmstp reduction
      pdi=pd
      t2i=t2
      h2i=h2
      vdi=vd
      pdoti=(p-pi)/dt
      pi=p
      pri=pr
      nmini=nmin
      nzli=nzlvl
      t2ai=t2a
      dpdzi=dpdz
      t1i=t1
      xl2i=xl2
      ro2i=ro2
c  determine feedwater h from its p and t.
      pfeed=p+ro2*g*(xl2-xfeed)
      call table(tfeed,rtime,topfac,yt,nt)
      if(xl2.le.xfeed) pfeed=p
      call state(pfeed,t1,tfeed,d01,rfeed,d03,efeed,tfsat,
     1hgsat,hfsat,d04,d05,d06,d07,d08,d09,d10,d11,
     2d12,2,ierr)
      hfd=efeed+(pfeed/rfeed)
c   choose liq. control volume
      vlr=vd-vfeed
       xmfd=0.
       v2=amax1(vlr,v2a)
       if(vlr.lt.v2a) xmfd=1.
c
      data g/9.81/
      t=rtime
c  internal BS's are transfered via recirc's arguments
c  same for time step size dt=delt
c  parameters for the numerical solution
      data epsi,nmax,icase/.0001,10,-1/
c   determine whether print will be required
       long=ireci
c
c  zero the iteration couter
      n=0
c
c   external BC's at present time
      call table(p1,t,tinfac,ytemp,ntemp)
c
c  external boundary conditions at advanced time
      delt=dt
      t=t+dt
      call table(p2,t,tinfac,ytemp,ntemp)
```

```
       call table(wfd0,t,botfac,yb,nb)
       qb=0.
       if(ntemp.le.2.and.idome.eq.0) qb=wfd0
      prfdt=(p2-p1)/dt*6894.7448
      if(idome.eq.1) prfdt=pdoti
      ws=p2
c
c   Property Determination
c    p,t system derivatives begin with d
c    p,h system derivatives begin with variable name
c
      call state(p,t1,t2,ro1,d02,ev,el,tsat,hg,hf,dtsdp,
     1deldp,devdp,deldt,devdt,dr1dp,drvdp,dr1dt,drvdt,
     22,ierr)
      h1=p/ro1+ev
      h1i=h1
      ro1i=ro1
      xm1i=ro1*(vt-vd)
      xm2i=ro2*v2
      xm1=xm1i
      xm2=xm2i
      call deriv(p,tsat,ro1,ro2,ev,el,dtsdp,deldp,devdp,deldt,
     1devdt,dr1dp,drvdp,drvdt,dr1dt,dh2dt,dh2dp,ro2dh,ro2dp,dh1dt,
     2dh1dp,ro1dh,ro1dp,hf,hg,drgdp,drfdp,dhgdp,dhfdp,rog,rof)
      if(long.eq.0) go to 104
      write(nzzz,100)
100   format(/,'*****************************************************
     1***************************************************************
     2*********',
     2          t37,'Estimate of Forward Time Values',/,t3,'N',t9,
     !' m2=kg',t20,'h2a=J/Kg',t35,'t2a=K',t47,'ro2a=Kg/m3',t59,
     2' qw=MW',t70,'M1=Kg',t83,'t1=K',t92,'h1=J/kg',
     3 t104,'ro1=Kg/m3',t115,'qs=MW',t126,'tsat=K')
      write(nzzz,200)n,xm2,h2,t2,ro2,qw,xm1,t1,h1,ro1,qs,tsat
104   continue
c  determination of case number
      if(t1.le.tsat.and.t2.lt.tsat) icase=2
      if(t1.gt.tsat.and.t2.ge.tsat) icase=3
      if(t1.le.tsat.and.t2.ge.tsat) icase=4
      if(t1.gt.tsat.and.t2.lt.tsat) icase=1
      if(prfdt.le.0.) konce=0
      if(prfdt.ge.0.) icase=1
      if(ntemp.le.2.and.xmfd.eq.0.) icase=4
      if(tfeed.lt.tsat.and.x12i.le.xfeed) icase=2
      go to (501,502,503,504) icase
c
c SOLUTION PROCEDURE BEGINS
c
c CASE NO. 1 - RISING PRESSURE OR
c              FALLING PRESSURE,SUPERH. VAPOR,SUBC. LIQ.
501   continue
      qs=0.
      qw=0.
c      if(prfdt.gt.0.) call sink(rtime,qs,qw,p)
```

```
      do 511 n=1,nmax
      pden1=xm1*ro1dh/ro1**3.
      pden2=xm1*ro1dp/ro1**2.
      pden3=xm2*ro2dh/ro2**3.
      pden=pden1+pden2+pden3
      pnum1=(wr*xr*(hg-h1)+qs)*ro1dh/ro1**2.
      pnum2=(wr*(1.-xr)*(hf-h2)+qw)*ro2dh/ro2**2.
      if(ntemp.le.2.and.idome.eq.0) wfd0=qb/(h1-hfd)
      wfd=wfd0*xmfd
      wd=wd0-(1.-xmfd)*wfd0
      pnum3=wfd*(hfd-h2)*ro2dh/ro2**2.
      pnum=pnum1+pnum2+pnum3
      dm2dt=(1.-xr)*wr+wfd-wd
      ws=idome*ws+(1-idome)*(wr*xr-ro1*(prfdt*pden+pnum-dm2dt/ro2))
       if(ntemp.le.2.and.idome.eq.0) ws=wfd0
      dm1dt=wr*xr-ws
      prfdt=(1-idome)*prfdt+idome*(dm1dt/ro1+dm2dt/ro2-pnum)/pden
      xm2=xm2i+dm2dt*dt
      xm1=xm1i+dm1dt*dt
      p=pi+prfdt*dt
      h1dot=(wr*xr*(hg-h1)+qs+prfdt*xm1/ro1)/xm1
      h2dot=(wr*(1.-xr)*(hf-h2)+wfd*(hfd-h2)+qw+prfdt*xm2/ro2)/xm2
      t1dot=(h1dot-dh1dp*prfdt)/dh1dt
      t2dot=(h2dot-dh2dp*prfdt)/dh2dt
      t1=t1i+t1dot*dt
      t2=t2i+t2dot*dt
      call state(p,t1,t2,ro1,ro2,ev,el,tsat,hg,hf,dtsdp,deldp,
     1devdp,deldt,devdt,drldp,drvdp,drldt,drvdt,2,ierr)
      h10=h1
      h20=h2
      h1=p/ro1+ev
      h2=p/ro2+el
      cverr=abs((xm2/ro2+xm1/ro1)/(xm1i/ro1i+xm2i/ro2i)-1.)
      h2err=abs(h2-h20)
      h1err=abs(h1-h10)
      convc=amax1(h2err,h1err)
      if(convc.le.1.0.and.cverr.le.epsi) go to 600
      call deriv(p,tsat,ro1,ro2,ev,el,dtsdp,deldp,devdp,
     1deldt,devdt,drldp,drvdp,drvdt,drldt,dh2dt,dh2dp,
     2ro2dh,ro2dp,dh1dt,dh1dp,ro1dh,ro1dp,hf,hg,drgdp,
     3drfdp,dhgdp,dhfdp,rog,rof)
511   continue
      go to 600
c
c  CASE NO 2 - FALLING PRESSURE,SAT. VAPOR, SUBC. LIQUID
502   CONTINUE
      do 512 n=1,nmax
      t1=tsat
      pden1=xm1*(dhgdp-1./rog)/(hg-hf)*(1./rog-1./ro2+
     1ro2dh*hf/ro2**2.-ro2dh*h2/ro2**2.)
      pden2=xm1*drgdp/rog**2.
      pden3=xm2*ro2dh/ro2**3.
      pden=pden1+pden2+pden3
      pum2=(ro2dh*hf/ro2**2.-ro2dh*h2/ro2**2.)*wr*(1.-xr)
```

```
      if(ntemp.le.2.and.idome.eq.0) wfd0=qb/(hg-hfd)
      wfd=xmfd*wfd0
      wd=wd0-(1.-xmfd)*wfd0
      pnum1=(wr*(1.-xr)+wfd-wd)/ro2
      pnum3=(ro2dh*hfd/ro2**2.-ro2dh*h2/ro2**2.)*wfd
      pnum=pnum1-pnum2-pnum3
      ws=idome*ws+(1-idome)*(rog*(pnum-pden*prfdt)+wr*xr)
      if(ntemp.le.2.and.idome.eq.0) ws=wfd0
      prfdt=(1-idome)*prfdt+idome*((wr*xr-ws)/rog+pnum)/pden
      wcond=prfdt*xm1/(hg-hf)*(dhgdp-1./rog)
      dm2dt=(1.-xr)*wr+wfd+wcond-wd
      dm1dt=wr*xr-ws-wcond
      xm1=xm1i+dm1dt*dt
      xm2=xm2i+dm2dt*dt
      h2dot=((1.-xr)*wr*(hf-h2)+wfd*(hfd-h2)+xm2*prfdt/ro2+
     1wcond*(hf-h2))/xm2
      t2dot=(h2dot-dh2dp*prfdt)/dh2dt
      t2=t2i+t2dot*dt
      p=pi+prfdt*dt
      call state(p,t1,t2,ro1,ro2,ev,el,tsat,hg,hf,dtsdp,deldp,
     1devdp,deldt,devdt,drldp,drvdp,drldt,drvdt,2,ierr)
      h10=h1
      h20=h2
      h1=p/ro1+ev
      h2=p/ro2+el
      h1err=abs(h1-h10)
      h2err=abs(h2-h20)
      convc=amax1(h1err,h2err)
      cverr=abs((xm2/ro2+xm1/ro1)/(xm1i/ro1i+xm2i/ro2i)-1.)
      if(convc.le.1.0.and.cverr.le.epsi) go to 600
      call deriv(p,tsat,ro1,ro2,ev,el,dtsdp,deldp,devdp,deldt,
     1devdt,drldp,drvdp,drvdt,drldt,dh2dt,dh2dp,ro2dh,ro2dp,
     2dh1dt,dh1dp,ro1dh,ro1dp,hf,hg,drgdp,drfdp,dhgdp,dhfdp,rog,rof)
512   continue
      go to 600
c
c  CASE NO. 3 - FALLING PRESSURE,SUPERH.VAPOR,SAT.LIQUID
503   continue
      do 513 n=1,nmax
      t2=tsat
      qw=0.
      qs=0.
       pden1=xm2/(hg-hf)*(dhfdp-
     11./rof)*(1./ro1-1./rof-ro1dh*hg/(ro1**2.)+ro1dh*h1/ro1**2.)
      pden2=xm2*drfdp/rof**2.
      pden3=xm1*(ro1dp/ro1**2+ro1dh/ro1**3.)
      pden=pden1+pden2+pden3
      if(ntemp.le.2.and.idome.eq.0) wfd0=qb/(hg-hfd)
      wfd=xmfd*wfd0
      wd=wd0-(1.-xmfd)*wfd0
      pnum1=(wr*xr+wfd*(hfd-hf)/(hg-hf)+qw/(hg-hf))/ro1
      pnum2=(wr*(1.-xr)+wfd*(hg-hfd)/(hg-hf)-wd-qw/(hg-hf))/rof
      pnum3=ro1dh*((wr*xr+wfd*(hfd-hf)/(hg-hf)+qw/(hg-hf))*(hg
     1-h1)-qs)/(ro1**2.)
```

```
      pnum=pnum1+pnum2-pnum3
      ws=idome*ws+(1-idome)*(ro1*(pnum-pden*prfdt))
      if(ntemp.le.2.and.idome.eq.0) ws=wfd0
      prfdt=(1-idome)*prfdt+idome*(pnum-ws/ro1)/pden
      wfla=prfdt*xm2/(hg-hf)*(1./rof-dhfdp)+wfd*(hfd-hf)/(hg-
     1hf)+qw/(hg-hf)

      dm1dt=wr*xr+wfla-ws
      dm2dt=(1.-xr)*wr+wfd-wd-wfla
      xm1=xm1i+dm1dt*dt
      xm2=xm2i+dm2dt*dt
      h1dot=(wr*xr+wfla)*(hg-h1)/xm1+prfdt/ro1+qs/xm1
      t1dot=(h1dot-dh1dp*prfdt)/dh1dt
      t1=t1i+t1dot*dt
      p=pi+prfdt*dt
      call state(p,t1,t2,ro1,ro2,ev,el,tsat,hg,hf,dtsdp,deldp,
     1devdp,deldt,devdt,drldp,drvdp,drldt,drvdt,2,ierr)
      h10=h1
      h20=h2
      h1=p/ro1+ev
      h2=p/ro2+el
      h1err=abs(h1-h10)
      h2err=abs(h2-h20)
      convc=amax1(h1err,h2err)
      cverr=abs((xm1/ro1+xm2/ro2)/(xm1i/ro1i+xm2i/ro2i)-1.)
      if(convc.le.1.0.and.cverr.le.epsi) go to 600
      call deriv(p,tsat,ro1,ro2,ev,el,dtsdp,deldp,devdp,
     1deldt,devdt,drldp,drvdp,drvdt,drldt,dh2dt,dh2dp,ro2dh,
     2ro2dp,dh1dt,dh1dp,ro1dh,ro1dp,hf,hg,drgdp,drfdp,
     3dhgdp,dhfdp,rog,rof)
513   continue
      go to 600
c
c  CASE NO.4 - FALLING PRESSURE,SAT. VAPOR,SAT. LIQUID
504     continue
      do 514 n=1,nmax
      t1=tsat
      t2=tsat
      qs=0.
      qw=0.
      pden1=xm1*(1./rog-1./rof)/(hg-hf)*(dhgdp-1./rog)
      pden2=xm1*drgdp/rog**2.
      pden3=xm2*(1./rog-1./rof)/(hg-hf)*(dhfdp-1./rof)
      pden4=xm2*drfdp/rof**2.
      pden=pden1+pden2+pden3+pden4
      pnum1=(wr*xr+(qs+qw)/(hg-hf))/rog
      pnum2=((1.-xr)*wr-(qs+qw)/(hg-hf))/rof
      if(ntemp.le.2.and.idome.eq.0) wfd0=qb/(hg-hfd)
      wfd=xmfd*wfd0
      wd=wd0-(1.-xmfd)*wfd0
      pnum=pnum1+pnum2+(wfd-wd)/rof
      ws=idome*ws+(1-idome)*(pnum-pden*prfdt)*rog
      if(ntemp.le.2.and.idome.eq.0) ws=wfd0
      prfdt=(1-idome)*prfdt+idome*(pnum-ws/rog)/pden
```

```
      wfla=prfdt*xm2/(hg-hf)*(1./rof-dhfdp)+qw/(hg-hf)
      wcond=prfdt*xm1/(hg-hf)*(dhgdp-1./rog)-qs/(hg-hf)
      dm1dt=wr*xr-ws+wfla-wcond
      dm2dt=wr*(1.-xr)+wfd-wd+wcond-wfla
      xm1=xm1i+dm1dt*dt
      xm2=xm2i+dm2dt*dt
      p=pi+prfdt*dt
      call state(p,t1,t2,ro1,ro2,ev,el,tsat,hg,hf,dtsdp,
     1deldp,devdp,deldt,devdt,drldp,drvdp,drldt,drvdt,2,ierr)
      h10=h1
      h20=h2
      h1=p/ro1+ev
      h2=p/ro2+el
      h1err=abs(h1-h10)
      h2err=abs(h2-h20)
      convc=amax1(h1err,h2err)
      cverr=abs((xm1/ro1+xm2/ro2)/(xm1i/ro1i+xm2i/ro2i)-1.)
      if(convc.le.1.0.and.cverr.le.epsi) go to 600
      call deriv(p,tsat,ro1,ro2,ev,el,dtsdp,deldp,devdp,
     1deldt,devdt,drldp,drvdp,drvdt,drldt,dh2dt,dh2dp,
     2ro2dh,ro2dp,dh1dt,dh1dp,ro1dh,ro1dp,hf,hg,drgdp,
     3drfdp,dhgdp,dhfdp,rog,rof)
514   continue
      go to 600
c  UPDATE PROPERTIES
600   continue
      dro2=ro2-ro2i
      dv=xm2/ro2-xm2i/ro2i
       if(ntemp.le.2) dv=0.
      vd=vd+dv
      t2a=t2
      h2a=h2
      ro2a=ro2

c
c  update water level
      call wlvl (vd,xl2,vfeed,xfeed)
c
c update boundary conditions
      q=(wr/roavg)/nsep
      dpris=roavg*g*hris
      dpsep=roavg*(4494.508*q**2.+30.499)
      pr=p+dpsep
      pd=p+(dpdz+dro2*g)*(xl2-xlr)
c   determine new temp. ,dens and pd if xl2>xfeed.Also correct wd & wfd
      if(xl2i.le.xfeed) go to 45
      wd=wd0
      wfd=wfd0
      h2b=h2a-(1.-xmfd)*wfd*(h2a-hfd)/wd
      t2b=t2a+(h2b-h2a)/dh2dt
      tcp=t2a
      do 11 i=1,15
      tcp=(tcp+t2b)/2.
      call state(p,t1,tcp,d01,rcp,d03,ecp,d05,d06,d07,d08,
```

```
      1d09,d10,deldt,d12,d13,d14,drldt,d16,2,ierr)
       cp2=deldt-p*drldt/rcp**2.
       hcp=ecp+p/rcp
       t2b=tcp+(h2b-hcp)/cp2
11     continue
       h2=h2b
       t2=t2b
       ro2=ro2b
       call state(pd,t2b,t2b,d0,ro2b,d6,d7,d8,d9,d10,d11,d1,d13,d14,
      1 d15,d16,d17,d18,d19,2,ierr)
       do 1 i=1,3
       pd=p+(dpdz+dro2*g)*(x12-xlr)
       call state(pd,t2b,t2b,d0,ro2b,d6,d7,d8,d9,d10,d11,d12,d13,d14,
      1 d15,d16,d17,d18,d19,2,ierr)
1      dro2=ro2b-ro2bi
45     continue
        wfeed=wfd
       if (long.eq.0) return
       q2=qw/1.e6
       q1=qs/1.e6
       write(nzzz,200)n,xm2,h2a,t2a,ro2a,q2,xm1,t1,h1,ro1,q1,tsat
200    format(t2,i2,t7,f9.3,t19,f10.2,t33,f9.5,t47,f7.2,t59,f6.3,
      1t68,f9.3,t82,0pf7.2,t90,f11.2,t104,f7.2,t111,f9.3,t124,f8.3)
         write(nzzz,300)
       write(nzzz,400) ro2,t2,h2,vd,x12,pd,pr,dm1dt,dm2dt,h2err,
      1h1err,dpris,dpsep,p,t2b,h2b,ro2b,hg
       write(nzzz,401) wd,wr,xr,roavg,ws,wfd,cverr,icase,hfd,hf
      1,nfeed,nzlvl,nmin
        ireci=0
c     English Units Output
       perct=(x12-7.0422)*100./(11.2332-7.0422)
       w1=wfd*3.6*2.20462
       w2=ws*3.6*2.20462
       p2=p/6894.7448
       w3=wfla*3.6*2.20462
       w4=wcond*3.6*2.20462
       write(nzzz,402) p2,w1,w2,perct,t,dt,w3,wfla,w4,wcond,xmfd
       return
 400   format(t3,f7.2,t11,f7.2,t19,e14.7,t34,f9.5,t43,f8.4,t52,1pe12.5,
      )t66,1pe12.5,t79,1pe11.4,t91,1pe11.4,t102,0pf11.2,t120
      1,0pf11.2,/,/,t2,'dpris=',t8,1pe10.3,t20,'dpsep=',t26,
      21pe10.3,t37,'Pdome=',t43,e13.6,t57,'Pa',t61,'t2b=',
      3t65,0pf7.2,t73,'K',t76,'h2b=',t80,f11.2,t92,'J/Kg',
      4t98,'ro2b=',t103,f7.2,t110,'Kg/m3',t113,'hg=',t116,f11.2,t127,
      5'J/Kg')
401    format(1x,/,/,t37,'Initial Conditions',/,t4,'wd',t13,'wr',
      1t22,'xr',t30,'roavg',t40,'ws',t50,'wfd',t60,'cverr',
      2t70,'icase',t82,'hfd',t98,'hf',t110,'nfeed',t117,'nzlvl',t124,
      3'nmin',/,t1,f8.2,
      3t10,f8.2,t19,f7.4,t28,f8.2,t38,f8.2,t48,f8.2,
      4t58,f8.5,t68,i6,t78,e14.7,t94,e14.7,t111,i3,t118,i3,t125,i3
      5,/,/)
 300   format(1x,/,/,t37,'New Boundary Conditions',/,t2,'ro2(kg/m3)',
      1t13,'t2(k)'
```

```
     1,t23,'h2(j/Kg)',t36,'vd(m3)',t45,'xl2(m)',t56,'pd(Pa)',
     2t70,'pr(Pa)',t80,'dm1dt=Kg/s',t92,'dm2dt=Kg/s',
     3t104,'h2err(J/Kg)',t121,'h1err(J/Kg)')
402   format(t37,'English Units',/,t2,'Pdome=',t8,f8.2,
     1t17,'psia',t24,'wfd=',t28,f8.2,t37,'Klbm/hr',
     2t47,'ws=',t50,f8.2,t59,'Klbm/hr',t69,'level=',
     3t75,f6.2,t82,'percent',t92,'time=',t98,1pe10.3,t109,'sec',
     4t115,'dt=',t119,0pf8.5,t129,'sec',/,t3,'wfla=',t8,f8.2,t16,
     5'Klbm/hr(',t24,f7.2,t32,'Kg/s)',t45,'wcond=',t52,f8.2,t61,
     6'Klbm/hr(',t69,f7.2,t77,'Kg/s)',t91,'xmfd=',t96,f8.5,/)
      end
      subroutine wlvl(v2,xl2,vfeed,xfeed)
c  determine working parameters for calculations
      data xlone,v2one,xltwo,a2,b2,c2,sum2,v2two,xlthr,
     1     v2thr,sum3,xlfor,v2for,d,e5,a5,b5,c5,sum4,xlast,apipe,e6,
     2     a6,b6,c6,a7,b7,c7,sum7,v2sev,f7,sum8,v2eig,
     3     f9,sum9,v2nin,f10,v2ten,v2six,v2fiv,k,xleig,xlnin,xlten/
     4     0.,0.,0.,0.,0.,0.,0.,0.,0.,
     5     0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,
     6     0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,
     7     0.,0.,0.,0.,0.,0.,0.,0,0.,0.,0./
      if(k.eq.1) go to 55
      call geomet(xlone,v2one,xltwo,a2,b2,c2,sum2,v2two,xlthr,
     1          v2thr,sum3,xlfor,v2for,d,e5,a5,b5,c5,sum4,xlast,apipe,
     2          e6,a6,b6,c6,a7,b7,c7,sum7,v2sev,f7,sum8,v2eig,
     3          f9,sum9,v2nin,f10,v2ten,v2six,v2fiv,k,xleig,xlnin,xlten)
55    k=1
      vfeed=v2thr
      xfeed=sum3
c  determine level region
      if(v2.gt.v2nin.and.v2.lt.v2ten) go to 10
      if(v2.gt.v2eig.and.v2.lt.v2nin) go to 9
      if(v2.gt.v2sev.and.v2.lt.v2eig) go to 8
      if(v2.gt.v2six.and.v2.lt.v2sev) go to 7
      if(v2.gt.v2fiv.and.v2.lt.v2six) go to  6
      if(v2.gt.v2for.and.v2.lt.v2fiv) go to 5
      if(v2.gt.v2thr.and.v2.lt.v2for)go to 4
      if(v2.gt.v2two.and.v2.lt.v2thr) go to 3
      if(v2.gt.v2one.and.v2.lt.v2two) go to 2
      if(v2.gt.0.000.and.v2.lt.v2one) go to 1
      if(v2.lt.0.) write(6,100)
      if(v2.gt.v2ten) write(6,200)
      return
c  determine the water level
1     xl2=v2*xlone/v2one
      return
2     d2=v2one-v2
      guess=xltwo/2.
      call raph(a2,b2,c2,d2,guess,x2)
      xl2=x2+xlone
      return
3     xl2=sum2+(v2-v2two)*xlthr/(v2thr-v2two)
      return
4     xl2=sum3+(v2-v2thr)*xlfor/(v2for-v2thr)
```

```
      return
5     d5=d+v2for-v2
      guess=(xlast+e5)/2.
       call raph(a5,b5,c5,d5,guess,x2)
      xl2=sum4+xlast-x2
      return
6     d6=d+v2for+apipe*e5-v2
      guess=(e5+e6)/2.
      call raph(a6,b6,c6,d6,guess,x2)
      xl2=sum4+xlast-x2
      return
7     d7=v2sev-v2
      guess=e6/2.
      call raph(a7,b7,c7,d7,guess,x2)
      xl2=sum4+xlast-x2
      return
8     xl2=sum7+((v2-v2sev)/(v2eig-v2sev))*xleig
      return
9     xl2=sum8+((v2-v2eig)/(v2nin-v2eig))*xlnin
      return
10    xl2=sum9+((v2-v2nin)/(v2ten-v2nin))*xlten
      return
100   format(1x,'Downcomer has dried out')
200   format(1x,'Water has covered steam dryers')
      end
      subroutine geomet(xlone,v2one,xltwo,a2,b2,c2,sum2,v2two,xlthr,
     1v2thr,sum3,xlfor,v2for,d,e5,a5,b5,c5,sum4,xlast,apipe,e6,
     2a6,b6,c6,a7,b7,c7,sum7,v2sev,f7,sum8,v2eig,
     3f9,sum9,v2nin,f10,v2ten,v2six,v2fiv,k,xleig,xlnin,xlten)
c  subroutine transforms sg geometrical info into working
c  parameters for water level determination as function of volume
c  geometrical info must be supplied in data cards directly  into
c  this routine
      data rone,rtwo,rthr,rfor,rfiv,rsix,rsev/
     1     1.9082,2.0034,2.9242,2.7686,0.,0.,0.4572/
      data    pi,dpipe,npipe,nsep,rsep/
     1  3.14159,0.1524,    6, 166,0.057/
      xlone=6.3135
      xltwo=1.8034
      xlthr=0.
      xlfor=0.
      xlfiv=1.0593
      xlsix=1.0414
      xlsev=0.
      xleig=0.
      xlnin=0.381
      xlten=0.99695
      xlast=xlfiv+xlsix+xlsev
      v2one=pi*(rtwo**2.-rone**2.)*xlone
      a2=pi/3.*((rthr-rtwo)/xltwo)**2.
      b2=pi*rtwo*(rthr-rtwo)/xltwo
      c2=pi*(rtwo**2.-rone**2.)
      v2two=a2*xltwo**3.+b2*xltwo**2.+c2*xltwo+v2one
      v2thr=pi*(rthr**2.-rone**2.)*xlthr+v2two
```

```
      v2for=pi*(rthr**2.-rsix**2.+rfiv**2.-rone**2.)*xlfor+v2thr
      a5=pi/3.*((rfor-rone)/xlast)**2.
      b5=pi*rfor*(rone-rfor)/xlast
      c5=(rfor**2.-rthr**2.)*pi
      d=pi*xlast*((rthr**2.-rfor**2.)+(rfor-(rfor-rone)/3.)*(rfor-rone))
      e5=xlast-xlfiv
      v2fiv=a5*e5**3.+b5*e5**2.+c5*e5+d+v2for
      a6=a5
      b6=b5
      apipe=npipe*pi*(dpipe**2./4.)*sqrt(((rone-rsev)/xlsix)**2.+1.)
      c6=c5-apipe
      e6=e5-xlsix
      v2six=a6*e6**3.+b6*e6**2.+c6*e6+d+v2for+apipe*e5
      a7=a5
      b7=b5
      f7=pi*rsev**2.
      c7=c6-f7
      v2sev=d+v2for+apipe*(xlsix+e6)+f7*e6
      v2eig=f7*xleig+v2sev
       f9=pi*(rthr**2.-rfor**2.+rsev**2.)
      v2nin=f9*xlnin+v2eig
      f10=pi*(rthr**2.-nsep*rsep**2.)
      v2ten=f10*xlten+v2nin
      sum1=xlone
      sum2=sum1+xltwo
         sum3=sum2+xlthr
      sum4=sum3+xlfor
      sum5=sum4+xlfiv
      sum6=sum5+xlsix
      sum7=sum6+xlsev
      sum8=sum7+xleig
      sum9=sum8+xlnin
      return
      end
      subroutine raph (a,b,c,d,guess,x2)
      nmax=5000
      x1=guess
1     fx1=a*x1**3.+b*x1**2.+c*x1+d
      fpx1=3.*a*x1**2.+2.*b*x1+c
      x2=x1-fx1/fpx1
      x=abs(fx1)
      n=n+1
      if(x.lt.0.0001.or.n.gt.nmax) return
      x1=x2
      go to 1
      end
```

```
      subroutine sink(rtime,qs,qw,p0)
c  determines heat removalrate from steam dome during presurre rise
      common/force/botfac(30),yb(30),topfac(30),yt(30),tinfac(30),
     1             ytemp(30),qfac(30),yq(30),nb,nt,ntemp,nq
      common/sg3/qsl(30),qsw(30),k,dt
      dimension sum(30),g(30),h(30),sewal(30),seliq(30),p(30),v(30)
      dimension difus(20),areas(20),denss(20),cps(20),thick(20),
     1qswp(20,30)
      a(x)=(((2*x+1.)*pi/(2*thick(m)))**2.)*difus(m)
      data a1,b,c,pi/9.0395,0.223,255.2,3.14159/
      if(k.eq.1) go to 11
      t=rtime
      rewind(5)
      read(5,102) max,jmax,nmax,delt
102   format(/,/,/,/,/,/,/,/,/,/,/,/,/,/,/,/,/,/,/,i3,i3,i4,f6.3)
      dt=delt*1.
      read(5,103) diful,areal,densl,cpl
103   format(f12.9,f6.2,f6.1,f7.1)
      do 1 j=1,jmax
      t=t+delt
      call table(z,t,tinfac,ytemp,ntemp)
  1   p(j)=z*6894.7448
      t0=a1*(p0**b)+c
      do 3 m=1,max
      read(5,101) difus(m),areas(m),denss(m),cps(m),thick(m)
101   format(e11.2,f6.2,f7.1,f6.1,e11.2)
      alpha=2.*difus(m)*areas(m)*denss(m)*cps(m)/thick(m)
      a0=a(0)
      a0str=a0*delt
      sum1=0.
      if(abs(a0str).gt.72.) go to 519
      sum1=(a0str-1.+exp(-a0str))/(a0**2.)
519   continue
      do 10 n=1,nmax
      an=a(n)
      anstr=an*delt
      aexp=0.
      if(anstr.gt.78.) go to 10
      aexp=exp(-anstr)
10    sum1=(anstr-1.+aexp)/(an**2.)+sum1
      g0=sum1*alpha/delt
      h0=4*areal*densl*cpl*sqrt(diful*delt/pi)/3.
      do 20 j=1,jmax
      sum(j)=0.
      if(abs(j*a0str).gt.72) go to 517
      sum(j)=exp(-j*a0str)*(exp(a0str)+exp(-a0str)-2.)/(a0**2.)
517   continue
      do 30 n=1,nmax
      an=a(n)
      anstr=an*delt
      e1=0.
      e2=0.
      e3=0.
      e11=j*anstr
```

```
      e22=(j+1)*anstr
      e33=(j-1)*anstr
      if(e11.ge.78.) go to 31
      e1=exp(-e11)
31    continue
      if(e22.gt.78.) go to 32
      e2=exp(-e22)
32    continue
      if(e33.ge.78.) go to 30
      e3=exp(-e33)
30    sum(j)=(e3/an+e2/an-2*e1/an)/an+sum(j)
      g(j)=sum(j)*alpha/delt
20    h(j)=h0*((j-1)**1.5+(j+1)**1.5-2.*(j**1.5))
      do 40 j=1,jmax
40    v(j)=a1*(p(j)**b)+c-t0
      data tstr/1.1586063/
      do 43 jlim=1,jmax
      sewal(jlim)=0.
      seliq(jlim)=0.
      do 42 j=1,jlim
      if(j.eq.jlim) go to 41
      sewal(jlim)=v(j)*g(jlim-j)+sewal(jlim)
42    seliq(jlim)=v(j)*h(jlim-j)+seliq(jlim)
41    sewal(jlim)=v(jlim)*g0+sewal(jlim)
      seliq(jlim)=v(jlim)*h0+seliq(jlim)
43    continue
      qswp(m,1)=sewal(1)/delt
      qsl(1)=seliq(1)/delt
       do 3 j=2,jmax
      qswp(m,j)=(sewal(j)-sewal(j-1))/delt
3     qsl(j)=(seliq(j)-seliq(j-1))/delt
      do 4 j=1,jmax
      qsw(j)=0.
      do 4 m=1,max
4     qsw(j)=qswp(m,j)+qsw(j)
      write(6,104) (m,difus(m),areas(m),denss(m),cps(m),thick(m),m=1,max)
104   format(1x,'mat. no.=',i3,'difus=',e10.4,'area=',f7.2,'dens=',
     1f7.1,'cps=',f7.1,'thick=',e10.3)
      izero=0
      write(6,105) izero,p0
      write(6,105) (j,p(j),j=1,jlim)
105   format(1x,'p(',i3,')=',1pe12.6)
11     k=1
      delt=dt
      itime=int(rtime/delt)+1
      qs=-(qsw(itime)+qsl(itime))
      qw=qsl(itime)
       return
      end
```

APPENDIX H

DETERMINATION OF THE DOWNCOMER WATER LEVEL AS A FUNCTION OF VOLUME

Chapter 5 describes how the change in liquid volume in the downcomer, $\Delta V2$, is calculated at each time step. Following that calculation, the current downcomer liquid volume is updated via:

$$V2^{n+1} = V2^{n} + \Delta V2 \ .$$

In order to determine the downcomer water level, xl2, it is necessary to establish a functional relationship:

$$xl2 = f(V2)$$

between the level and the volume. This relationship is determined by the downcomer geometry. In this work, an idealized downcomer geometry is utilized with the purpose of accommodating most steam generator designs. This idealized downcomer geometry is shown in Fig. H-1 divided into ten regions. In each of the ten regions a different functional relationship prevails between the level and the volume. The ten functional relationships are given in subroutine wlvl next

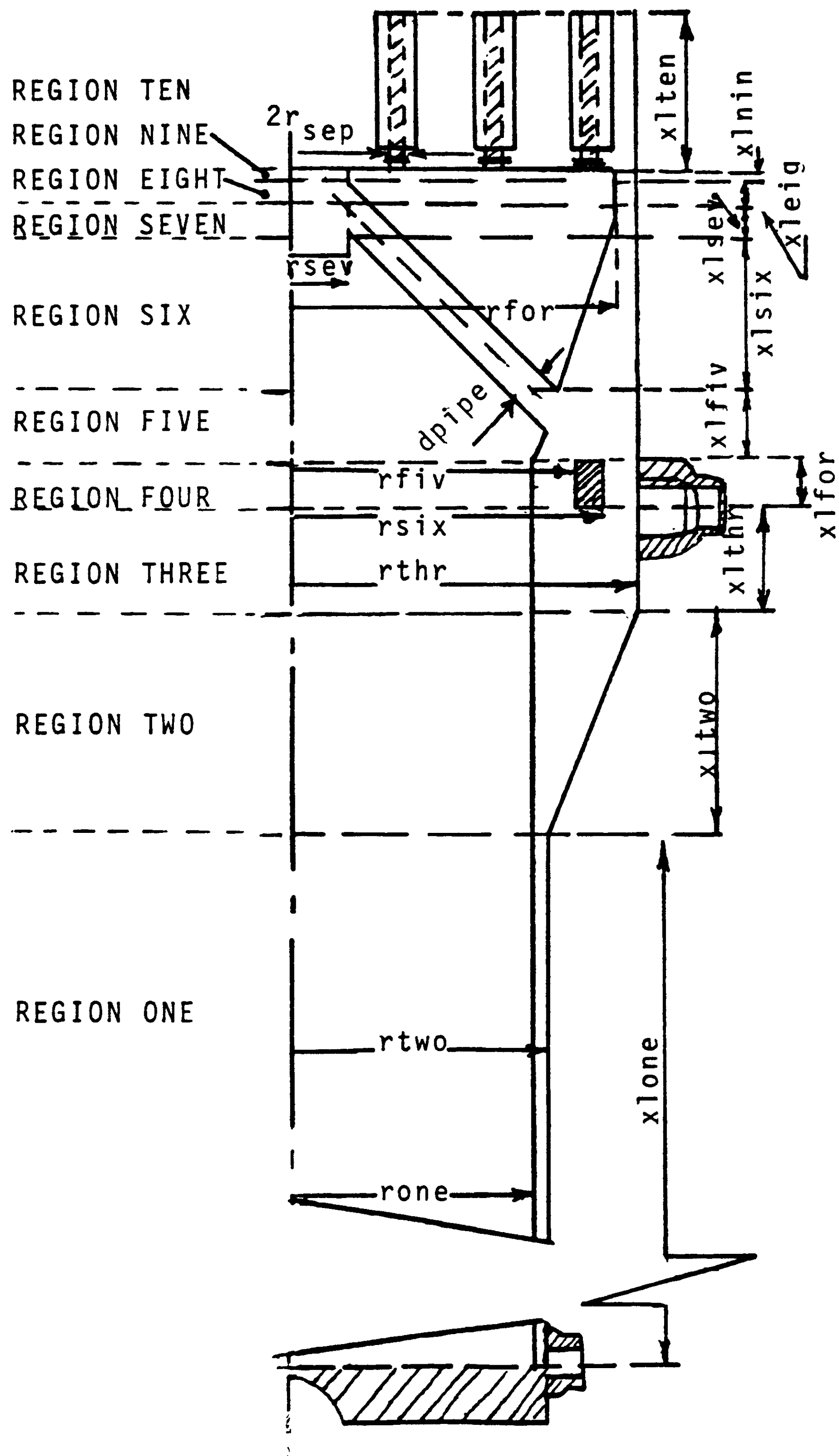


Figure H-1. Geometric variables in subroutine wlvl.

to labels 1 to 10 respectively. The subroutine is listed at the end of this Appendix. For convenience of interpretation, the nomenclature in Fig. H-1 coincides with the Fortran nomenclature. Since the programming is straightforward and the subroutine is short, the equations are not reproduced here.

It should be noted that, as indicated in Fig. H-1, the levels are referenced to the top of the passage from downcomer to evaporator rather than the tube sheet. The reference point is arbitrary and there is no particular reason for the current choice. Levels mentioned in the text have been, as noted, converted to the reference system of the data base being used for comparison with this work.

Table H-1 lists the input parameters required for a calculation. The parameters must be set directly into subroutine geomet, which is listed following wlvl, with the exception of 'xfeed', the feedwater ring level, and 'vfeed', the downcomer volume corresponding to 'xfeed'. These quantities have default values set at 'sum3' and 'v2thr', respectively, in subroutine wlvl. The above nomenclature is easily illustrated by the following example. The variable 'V2one' is the total volume of region one, 'v2two' is the total volume up to 'xlone' + 'xltwo'. i.e.:

v2two = v2one + volume of region two.

Table H-1.

Input for the calculation of downcomer water level as a function of volume.

PARAMETER	EXPLANATION	WHERE TO INPUT
xfeed	Level of feedwater ring.	Subroutine wlvl
vfeed	Downcomer volume corresponding to xfeed.	Subroutine wlvl
rone, rtwo, rthr, rfor, rfiv, rsix, rsev	Steam generator characteristic radii measured from central axis. All are shown in Fig. H-1.	Subroutine geomet
xlone, xltwo, xlthr, xlfor, xlfiv, xlsix, xlsev, xleigh, xlnin, xlten	Characteristic lengths corresponding to the heights of the ten regions. All are shown in Fig. H-1.	Subroutine geomet
dpipe	Diameter of drain pipe, see Fig. H-1.	Subroutine geomet
npipe	Number of drain pipes.	Subroutine geomet
rsep	Radius of separator stand pipe, see Fig. H-1.	Subroutine geomet
nsep	Number of separators.	Subroutine geomet

In regions five, six, and seven the relation between level and volume is a third degree polynomial. The only real positive root in the range of interest is calculated using the Newton-Raphson procedure. This method is described briefly in Chapter 5 and in depth in Ref. (H3). Subroutine raph executes this procedure and it is listed following wlvl and geomet.

In regions one, two, three, four, eight, nine, and ten the relation between level and volume is linear. Hence, level determination is straightforward.

Listings of subroutines wlvl, geomet, and raph follow in that order.

```
      subroutine wlvl(v2,xl2,vfeed,xfeed)
c  determine working parameters for calculations
      data xlone,v2one,xltwo,a2,b2,c2,sum2,v2two,xlthr,
     1     v2thr,sum3,xlfor,v2for,d,e5,a5,b5,c5,sum4,xlast,apipe,e6,
     2     a6,b6,c6,a7,b7,c7,sum7,v2sev,f7,sum8,v2eig,
     3     f9,sum9,v2nin,f10,v2ten,v2six,v2fiv,k,xleig,xlnin,xlten/
     4     0.,0.,0.,0.,0.,0.,0.,0.,0.,
     5     0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,
     6     0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,
     7     0.,0.,0.,0.,0.,0.,0.,0,0.,0.,0./
      if(k.eq.1) go to 55
      call geomet(xlone,v2one,xltwo,a2,b2,c2,sum2,v2two,xlthr,
     1        v2thr,sum3,xlfor,v2for,d,e5,a5,b5,c5,sum4,xlast,apipe,
     2        e6,a6,b6,c6,a7,b7,c7,sum7,v2sev,f7,sum8,v2eig,
     3        f9,sum9,v2nin,f10,v2ten,v2six,v2fiv,k,xleig,xlnin,xlten)
55    k=1
      vfeed=v2thr
      xfeed=sum3
c  determine level region
      if(v2.gt.v2nin.and.v2.lt.v2ten) go to 10
      if(v2.gt.v2eig.and.v2.lt.v2nin) go to 9
      if(v2.gt.v2sev.and.v2.lt.v2eig) go to 8
      if(v2.gt.v2six.and.v2.lt.v2sev) go to 7
      if(v2.gt.v2fiv.and.v2.lt.v2six) go to  6
      if(v2.gt.v2for.and.v2.lt.v2fiv) go to 5
      if(v2.gt.v2thr.and.v2.lt.v2for)go to 4
      if(v2.gt.v2two.and.v2.lt.v2thr) go to 3
      if(v2.gt.v2one.and.v2.lt.v2two) go to 2
      if(v2.gt.0.000.and.v2.lt.v2one) go to 1
      if(v2.lt.0.) write(6,100)
      if(v2.gt.v2ten) write(6,200)
      return
c  determine the water level
1     xl2=v2*xlone/v2one
      return
2     d2=v2one-v2
      guess=xltwo/2.
      call raph(a2,b2,c2,d2,guess,x2)
      xl2=x2+xlone
      return
3     xl2=sum2+(v2-v2two)*xlthr/(v2thr-v2two)
      return
4     xl2=sum3+(v2-v2thr)*xlfor/(v2for-v2thr)
      return
5     d5=d+v2for-v2
      guess=(xlast+e5)/2.
       call raph(a5,b5,c5,d5,guess,x2)
      xl2=sum4+xlast-x2
      return
6     d6=d+v2for+apipe*e5-v2
      guess=(e5+e6)/2.
      call raph(a6,b6,c6,d6,guess,x2)
      xl2=sum4+xlast-x2
      return
```

```
7      d7=v2sev-v2
       guess=e6/2.
       call raph(a7,b7,c7,d7,guess,x2)
       xl2=sum4+xlast-x2
       return
8      xl2=sum7+((v2-v2sev)/(v2eig-v2sev))*xleig
       return
9      xl2=sum8+((v2-v2eig)/(v2nin-v2eig))*xlnin
       return
10     xl2=sum9+((v2-v2nin)/(v2ten-v2nin))*xlten
       return
100    format(1x,'Downcomer has dried out')
200    format(1x,'Water has covered steam dryers')
       end
       subroutine geomet(xlone,v2one,xltwo,a2,b2,c2,sum2,v2two,xlthr,
      1v2thr,sum3,xlfor,v2for,d,e5,a5,b5,c5,sum4,xlast,apipe,e6,
      2a6,b6,c6,a7,b7,c7,sum7,v2sev,f7,sum8,v2eig,
      3f9,sum9,v2nin,f10,v2ten,v2six,v2fiv,k,xleig,xlnin,xlten)
c  subroutine transforms sg geometrical info into working
c  parameters for water level determination as function of volume
c  geometrical info must be supplied in data cards directly  into
c  this routine
       data rone,rtwo,rthr,rfor,rfiv,rsix,rsev/
      1      1.9082,2.0034,2.9242,2.7686,0.,0.,0.4572/
       data    pi,dpipe,npipe,nsep,rsep/
      1   3.14159,0.1524,    6, 166,0.057/
       xlone=6.3135
       xltwo=1.8034
       xlthr=0.
       xlfor=0.
       xlfiv=1.0593
       xlsix=1.0414
       xlsev=0.
       xleig=0.
       xlnin=0.381
       xlten=0.99695
       xlast=xlfiv+xlsix+xlsev
       v2one=pi*(rtwo**2.-rone**2.)*xlone
       a2=pi/3.*((rthr-rtwo)/xltwo)**2.
       b2=pi*rtwo*(rthr-rtwo)/xltwo
       c2=pi*(rtwo**2.-rone**2.)
       v2two=a2*xltwo**3.+b2*xltwo**2.+c2*xltwo+v2one
       v2thr=pi*(rthr**2.-rone**2.)*xlthr+v2two
       v2for=pi*(rthr**2.-rsix**2.+rfiv**2.-rone**2.)*xlfor+v2thr
       a5=pi/3.*((rfor-rone)/xlast)**2.
       b5=pi*rfor*(rone-rfor)/xlast
       c5=(rfor**2.-rthr**2.)*pi
       d=pi*xlast*((rthr**2.-rfor**2.)+(rfor-(rfor-rone)/3.)*(rfor-rone))
       e5=xlast-xlfiv
       v2fiv=a5*e5**3.+b5*e5**2.+c5*e5+d+v2for
       a6=a5
       b6=b5
       apipe=npipe*pi*(dpipe**2./4.)*sqrt(((rone-rsev)/xlsix)**2.+1.)
       c6=c5-apipe
```

```
      e6=e5-xlsix
      v2six=a6*e6**3.+b6*e6**2.+c6*e6+d+v2for+apipe*e5
      a7=a5
      b7=b5
      f7=pi*rsev**2.
      c7=c6-f7
      v2sev=d+v2for+apipe*(xlsix+e6)+f7*e6
      v2eig=f7*xleig+v2sev
       f9=pi*(rthr**2.-rfor**2.+rsev**2.)
      v2nin=f9*xlnin+v2eig
      f10=pi*(rthr**2.-nsep*rsep**2.)
      v2ten=f10*xlten+v2nin
      sum1=xlone
      sum2=sum1+xltwo
         sum3=sum2+xlthr
      sum4=sum3+xlfor
      sum5=sum4+xlfiv
      sum6=sum5+xlsix
      sum7=sum6+xlsev
      sum8=sum7+xleig
      sum9=sum8+xlnin
      return
      end
      subroutine raph (a,b,c,d,guess,x2)
      nmax=5000
      x1=guess
1     fx1=a*x1**3.+b*x1**2.+c*x1+d
      fpx1=3.*a*x1**2.+2.*b*x1+c
      x2=x1-fx1/fpx1
      x=abs(fx1)
      n=n+1
      if(x.lt.0.0001.or.n.gt.nmax) return
      x1=x2
      go to 1
      end
```

BIOGRAPHICAL NOTE

Hugo C. da Silva Jr. was born July 28, 1954 in Rio de Janeiro, Brazil, of Rebecca Fay Silva, a chemist from Lake Charles, Louisiana, and Hugo Cardoso da Silva, an electrical engineer from Rio. He lived the early years of his life partly in Rio and partly in Parkersburg, West Virginia, where English became his first language.

From 1965 to 1971, the author attended the Colégio de São Bento of Rio de Janeiro where he was graduated valedictorian of his class.

In January 1972 he passed the entrance examination to the Instituto Technológico de Aeronáutica in Sao José dos Campos. In December 1976, he was graduated there, with a Bachelor of Science degree in Mechanical Engineering. The author spent the year of 1976 completing an engineering internship program in Sao José dos Campos, first in a General Motors Diesel Engine Plant and subsequently at Engesa, a manufacturer of heavy special traction vehicles.

In 1977 the author initiated a Master's program in Mechanical Engineering, oriented toward nuclear power generation, at the Catholic University of Rio de Janeiro, where he obtained the Science Master degree in January 1979.

The author entered the graduate school at the Massachusetts Institute of Technology in February 1979. In February 1984 he was awarded the degree of Doctor of Philosophy in Nuclear Engineering.

The author is an associate member of Sigma Xi and a student member of the American Nuclear Society. He speaks both English and Portuguese as a native and is fluent in Spanish and French.