

Figure 5.4.17 TEE Fill & Drain Test, Main Tube Cell #1 Pressure, TRAC-M with Level Tracking

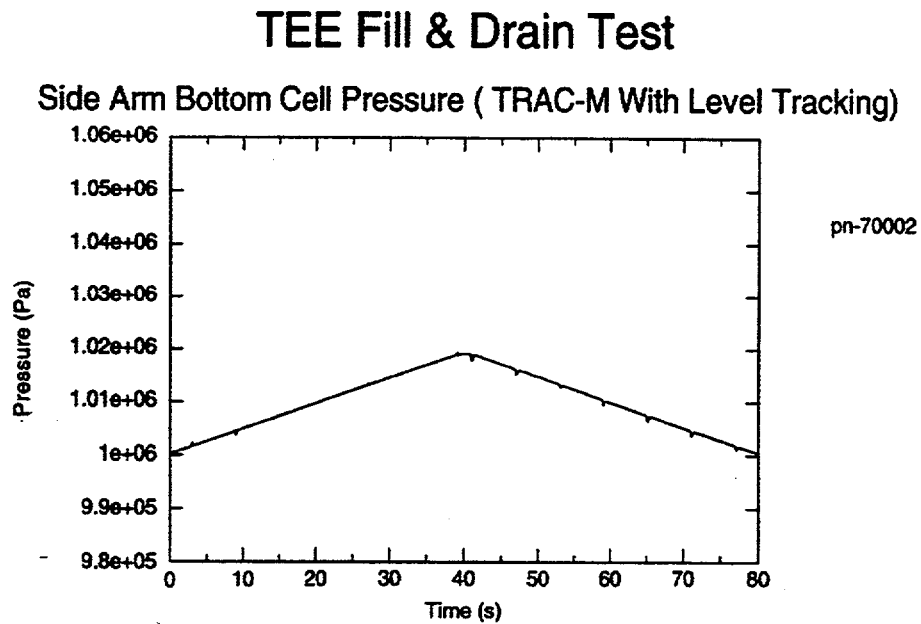


Figure 5.4.18 TEE Fill & Drain Test, Side Arm Bottom Cell Pressure, TRAC-M with Level Tracking

Two PIPE Fill & Drain Test

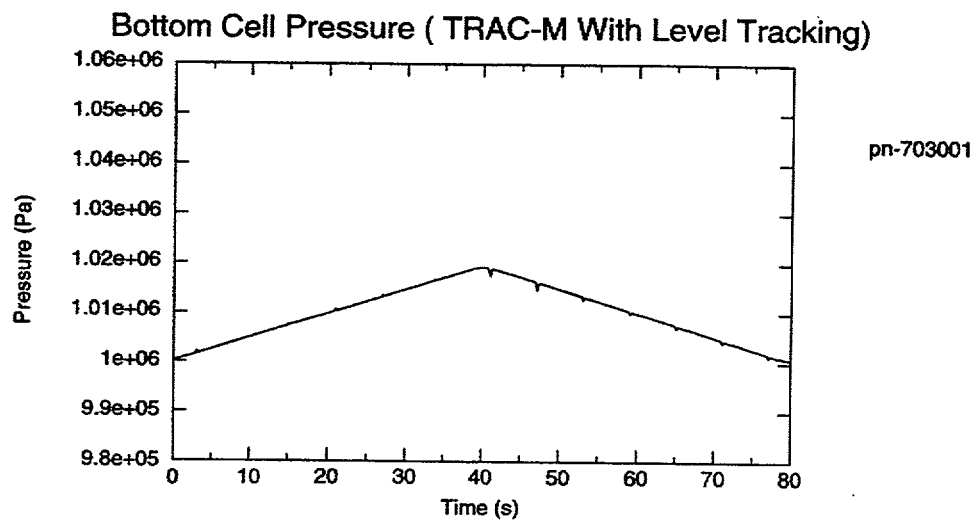


Figure 5.4.19 Two-Pipe Fill & Drain Test, Bottom Cell Pressure, TRAC-M with Level Tracking

Single Ring Vessel Fill & Drain Test

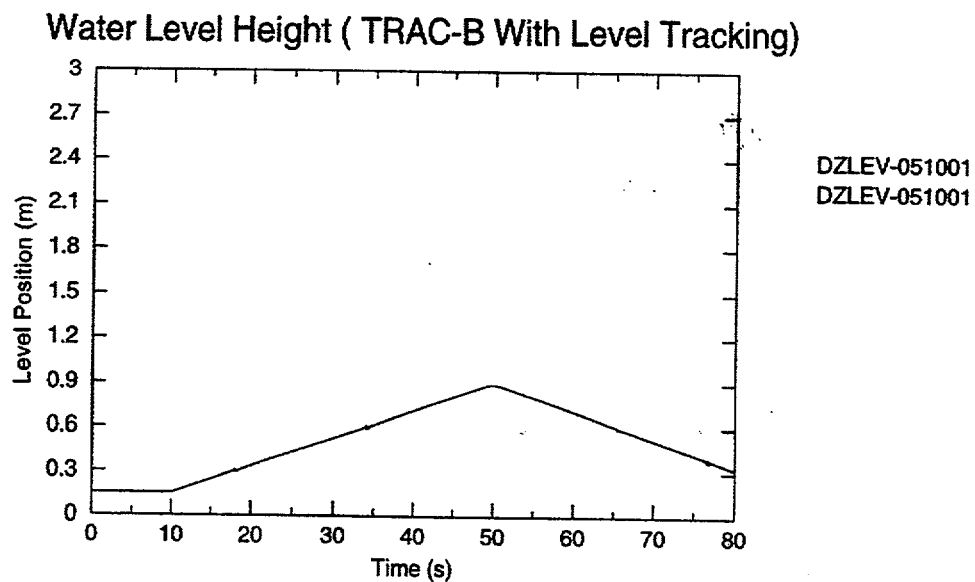


Figure 5.4.20 Single-Ring Vessel Fill & Drain Test, Water Level Height, TRAC-B with Level Tracking

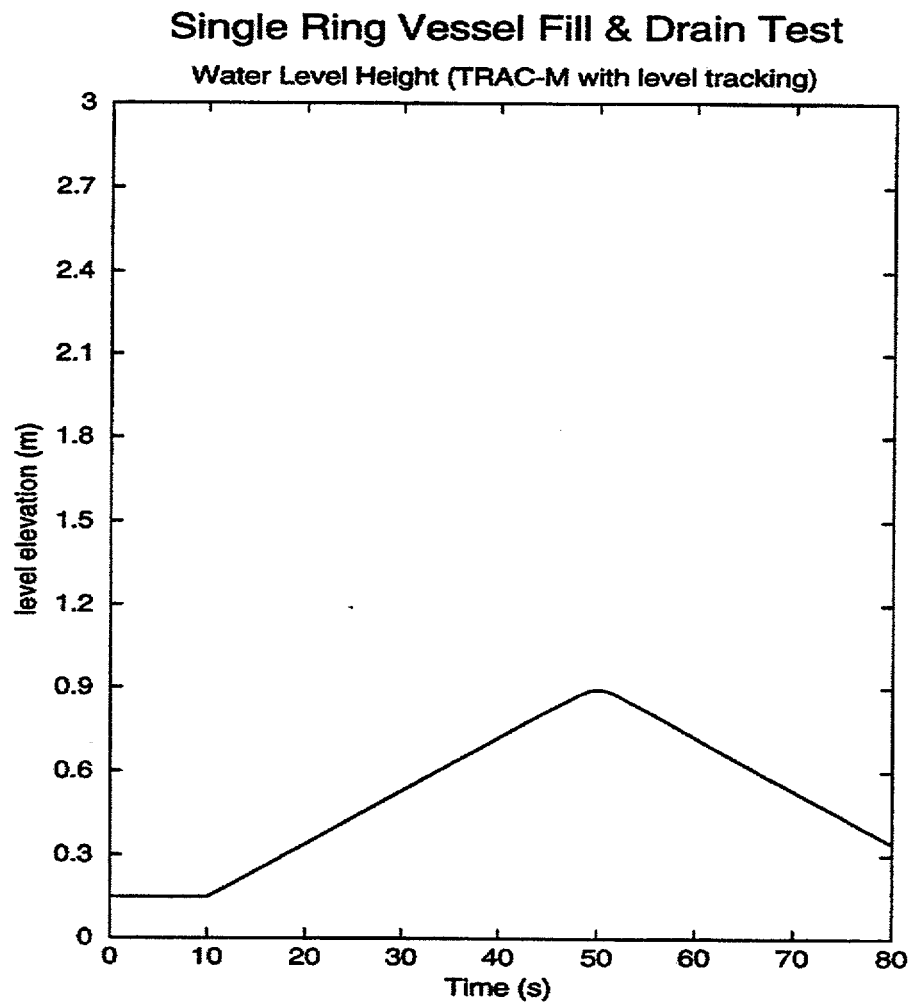


Figure 5.4.21 Single-Ring Vessel Fill & Drain Test, Water Level Height, TRAC-M with Level Tracking

Single Ring Vessel Fill & Drain Test

Bottom Cell Pressure (TRAC-B With Level Tracking)

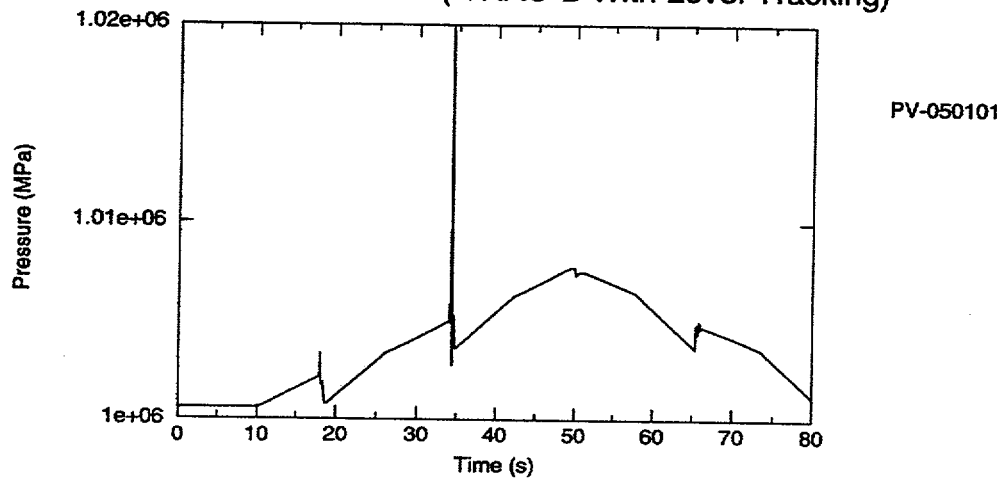


Figure 5.4.22 Single-Ring Vessel Fill & Drain Test, Bottom Cell Pressure, TRAC-B with Level Tracking

Single Ring Vessel Fill & Drain Test

Bottom Cell Pressure (TRAC-B Without Level Tracking)

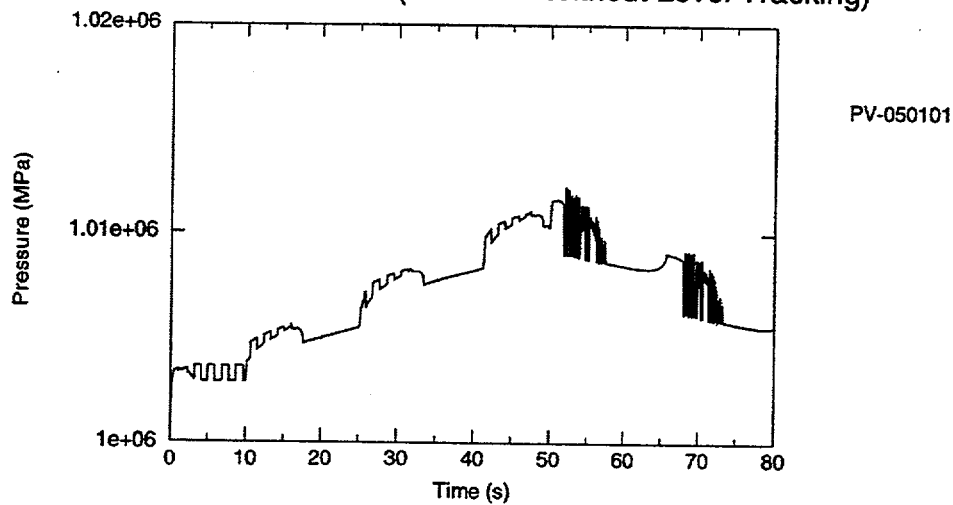


Figure 5.4.23 Single-Ring Vessel Fill & Drain Test, Bottom Cell Pressure, TRAC-B without Level Tracking

Single Ring Vessel Fill & Drain Test

Bottom Cell Pressure (TRAC-M With Level Tracking)

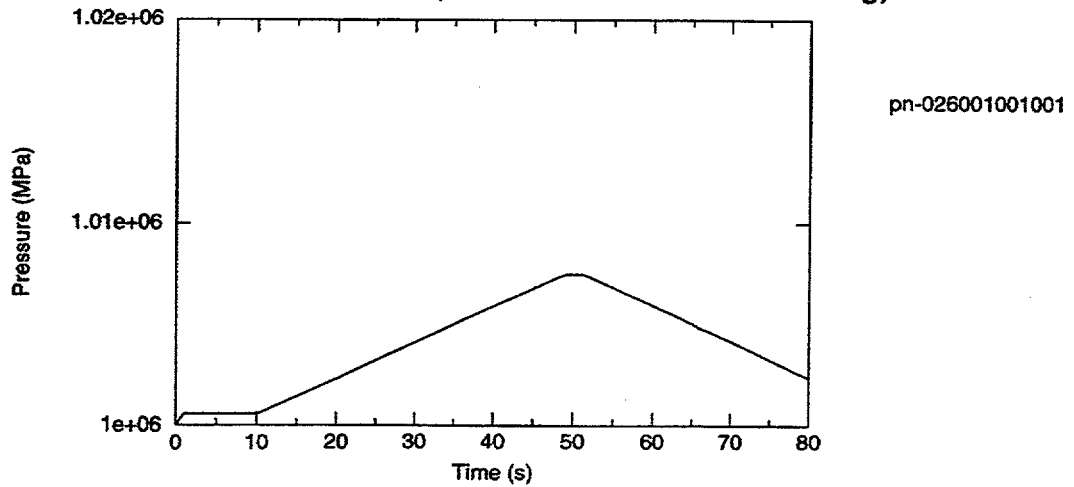


Figure 5.4.24 Single-Ring Vessel Fill & Drain Test, Bottom Cell Pressure, TRAC-M with Level Tracking

Single Ring Vessel Fill & Drain Test

Bottom Cell Pressure (TRAC-M Without Level Tracking)

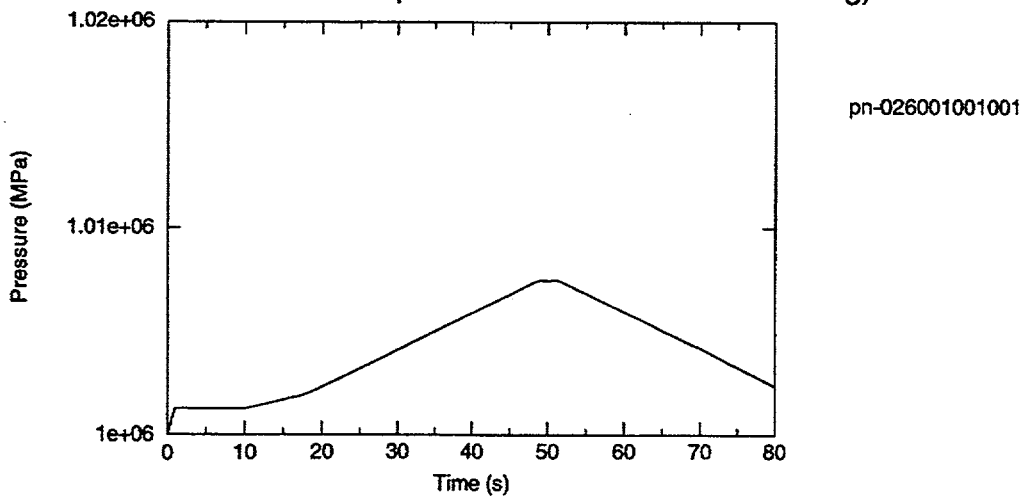


Figure 5.4.25 Single-Ring Vessel Fill & Drain Test, Bottom Cell Pressure, TRAC-M without Level Tracking

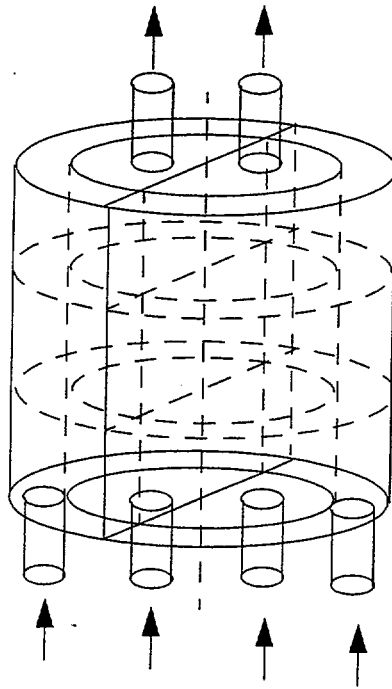
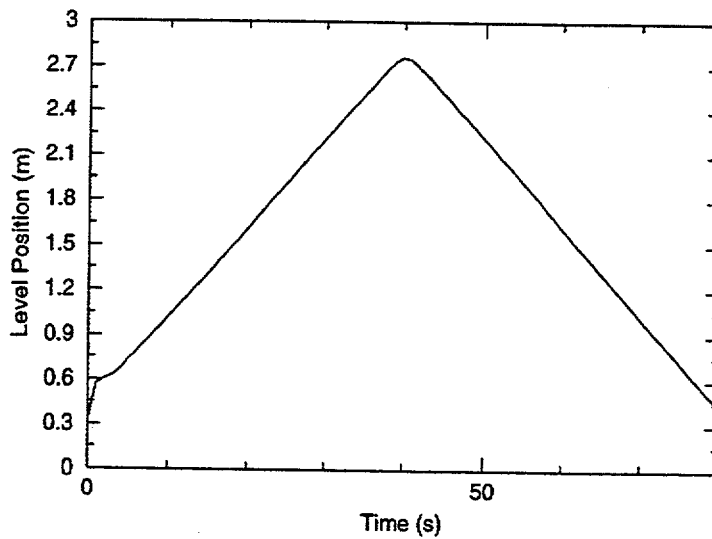


Figure 5.4.26 Multiple-Ring/Azimuthal Vessel Component Fill and Drain Test Problem

Two Ring Vessel Fill & Drain Test

Water Level Height (TRAC-M With Level Tracking)



dzlev-026008001001

Figure 5.4.27 Two-Ring Vessel Fill & Drain Test, Water Level Height, TRAC-M with Level Tracking

Two Vessel Fill & Drain Test

Water Level Height (TRAC-M With Level Tracking)

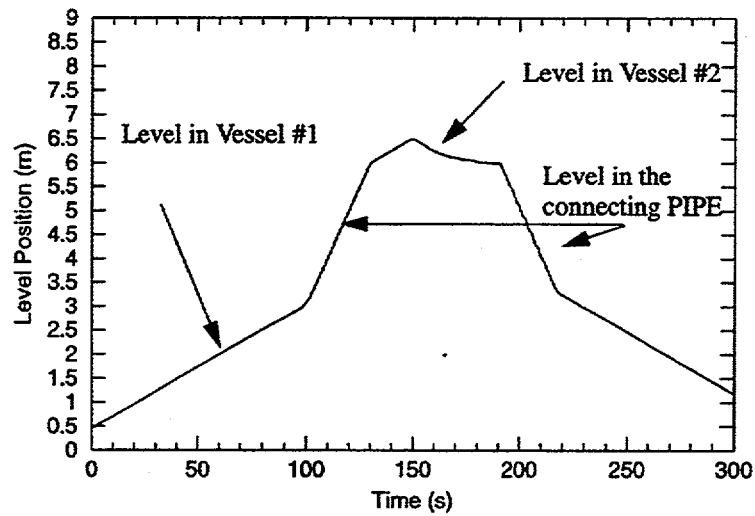


Figure 5.4.28 Two-Vessel Fill & Drain Test, Water Level Height, TRAC-M with Level Tracking

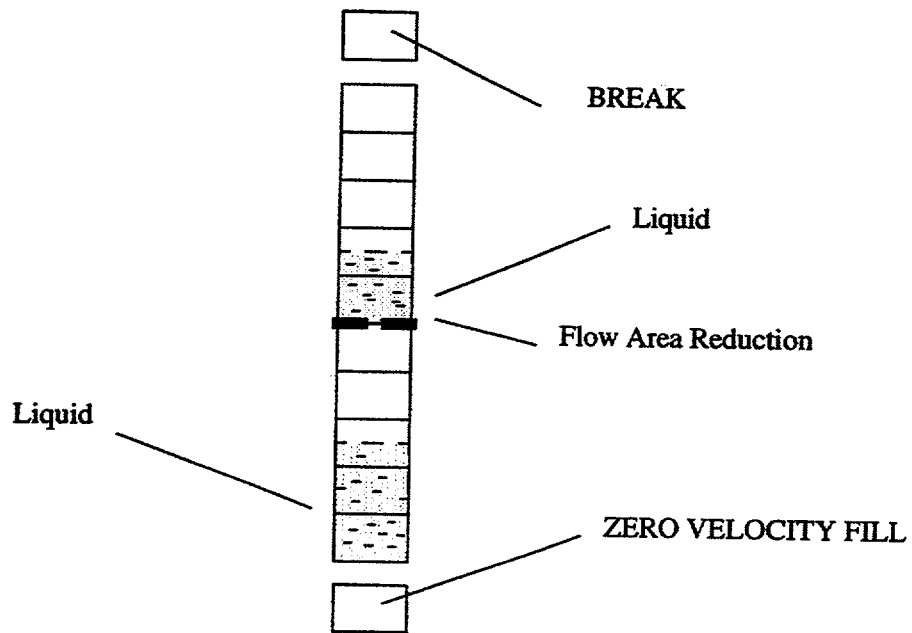


Figure 5.4.29 Inverted Void Fraction Profile Level Tracking Test

Valve Inverted Void Profile

Water Level Height (TRAC-M With Level Tracking)

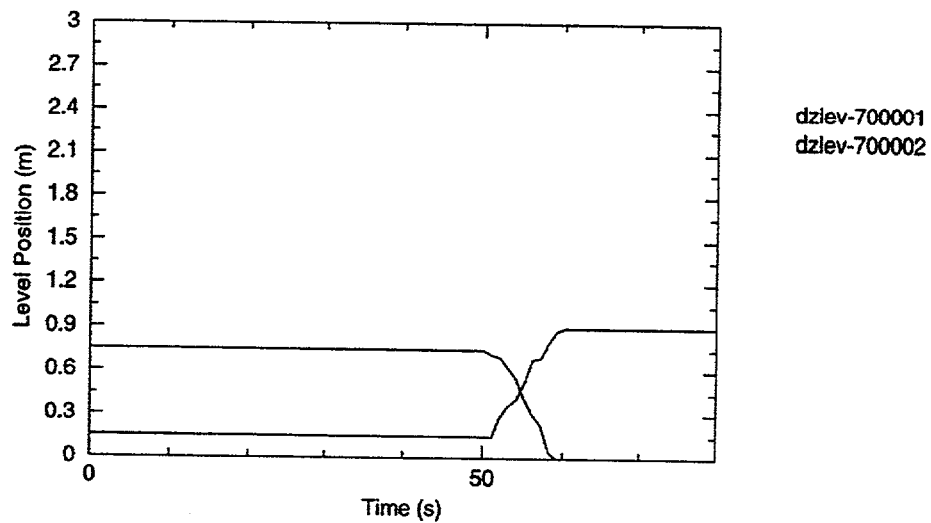


Figure 5.4.30 Inverted Void Profile Test, Water Level Height, TRAC-M with Level Tracking

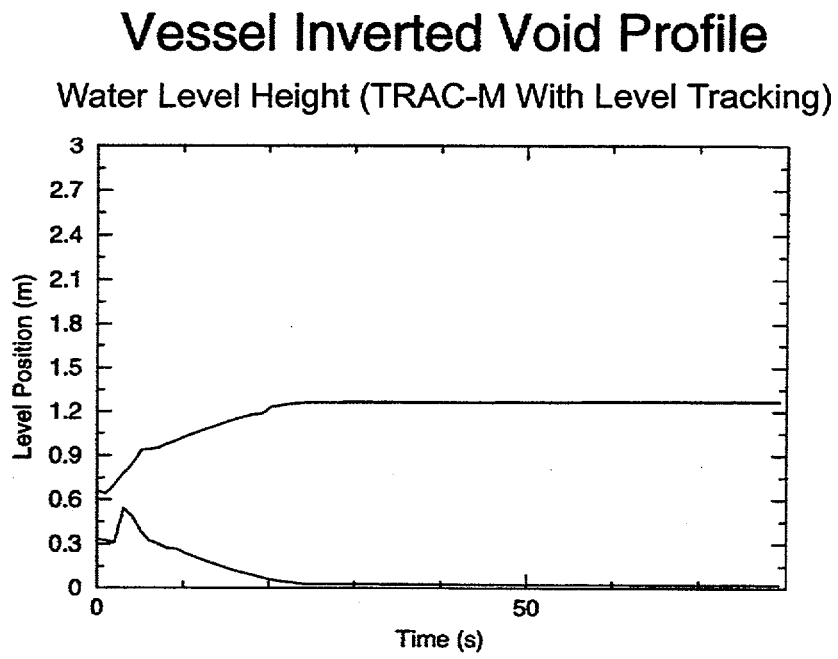


Figure 5.4.31 Inverted Void Profile in Vessel Test, Water Level Height, TRAC-M with Level Tracking

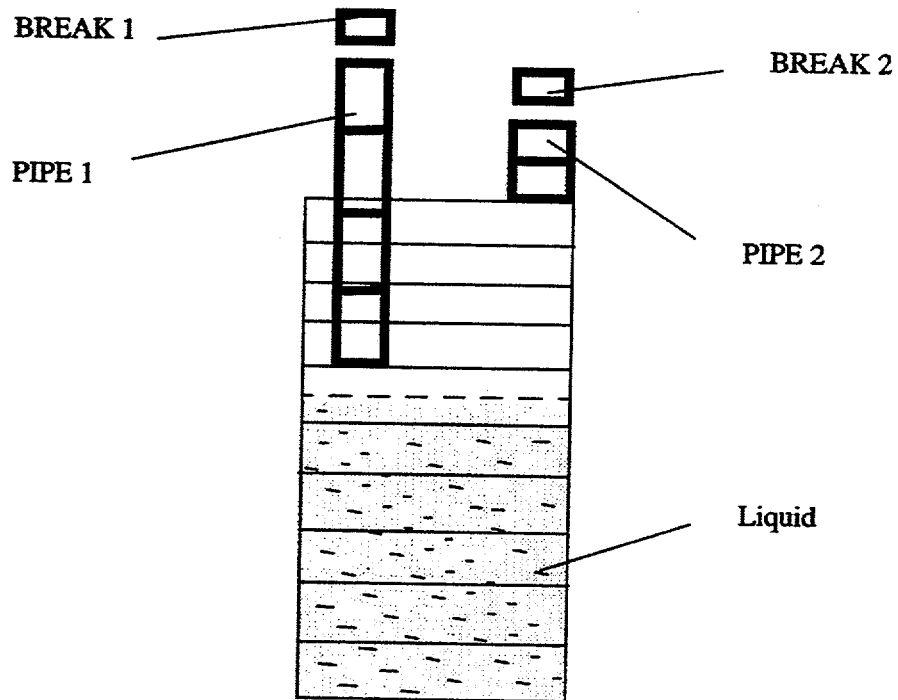


Figure 5.4.32 3-D to 1-D Above-Level Suction Test

Vessel Above Level Suction Test

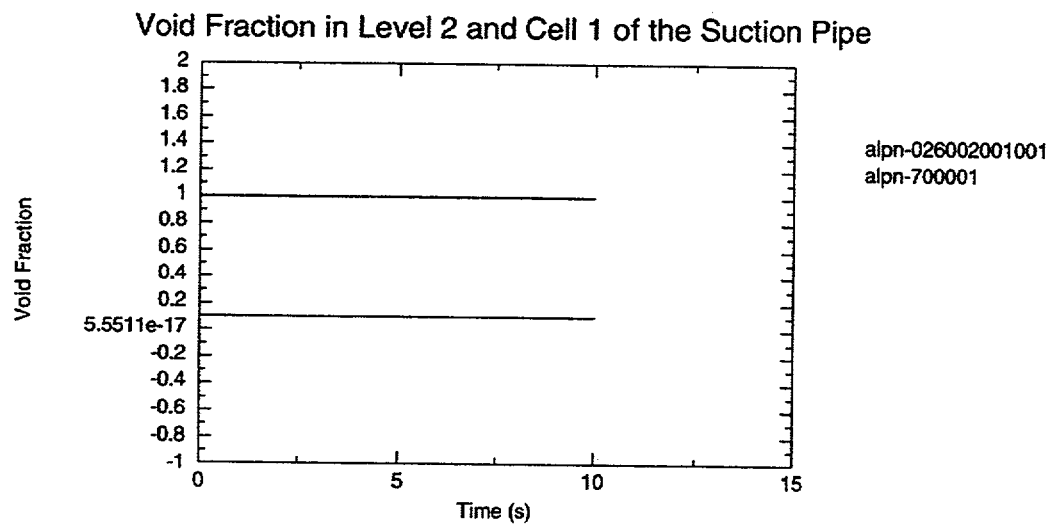


Figure 5.4.33 Vessel Above-Level Suction Test, Void Fraction

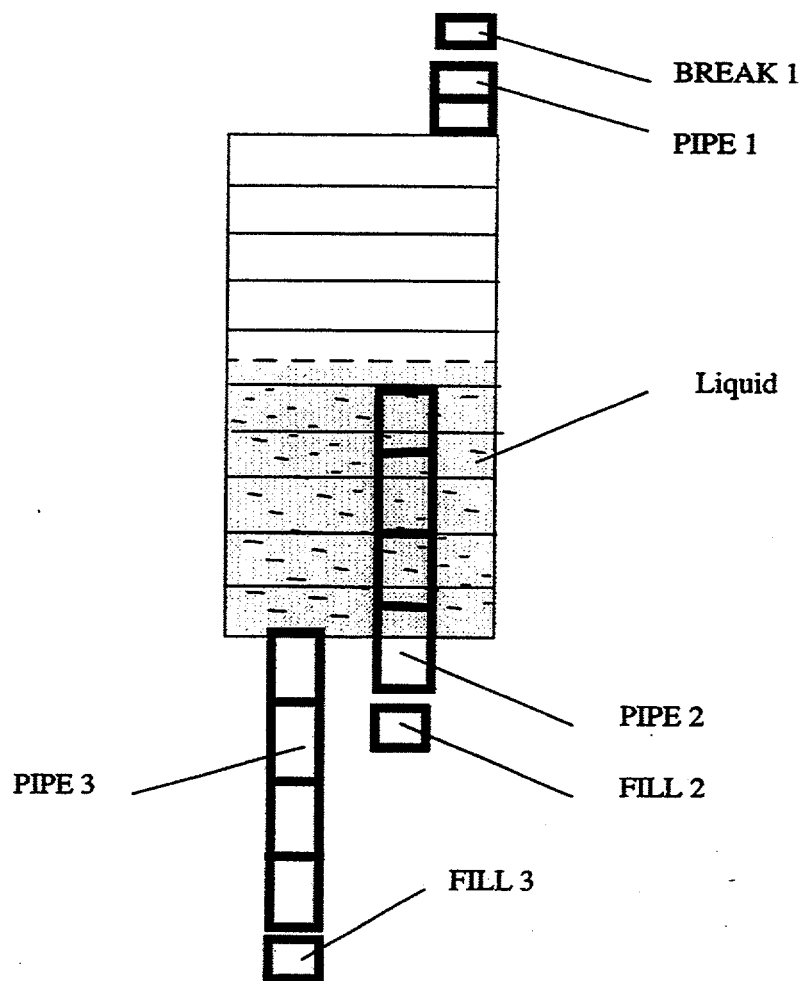


Figure 5.4.34 3-D to 1-D Below-Level Suction Test

Vessel Below Level Suction Test Void Fraction in Level 6 and Cell 1 of the Suction Pipe

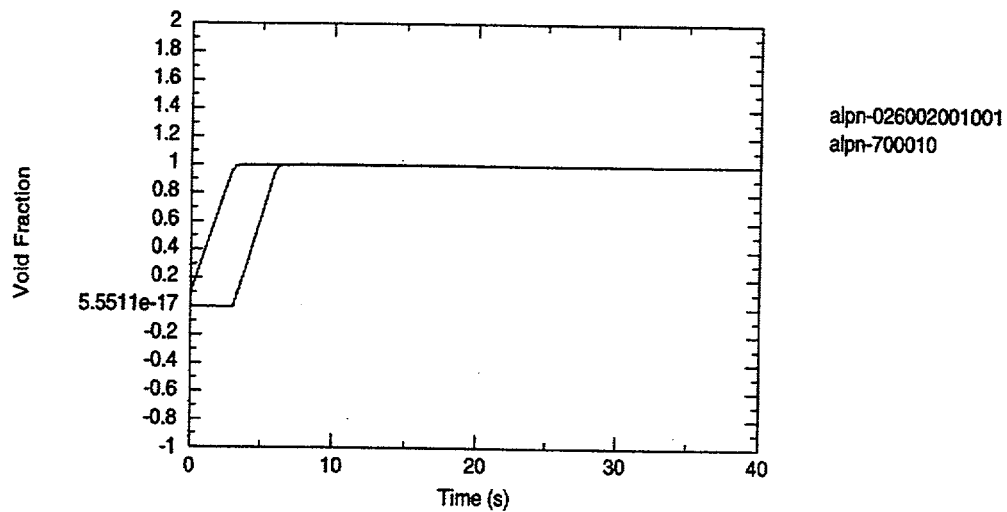


Figure 5.4.35 Vessel Below-Level Suction Test, Void Fractions in Level 6 of the Vessel and Cell #1 of the Pipe

Single PIPE Cold Water Flooding Test

Steam Supply Rate (TRAC-B Without Level Tracking)

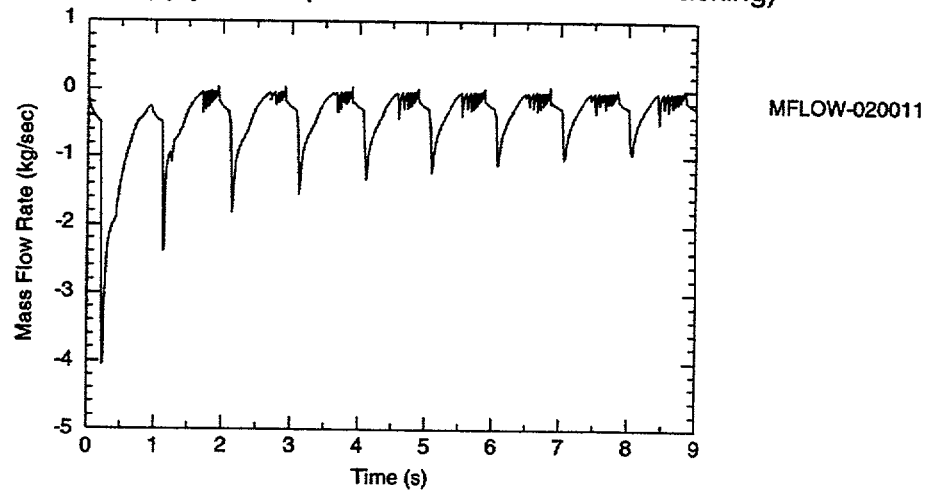
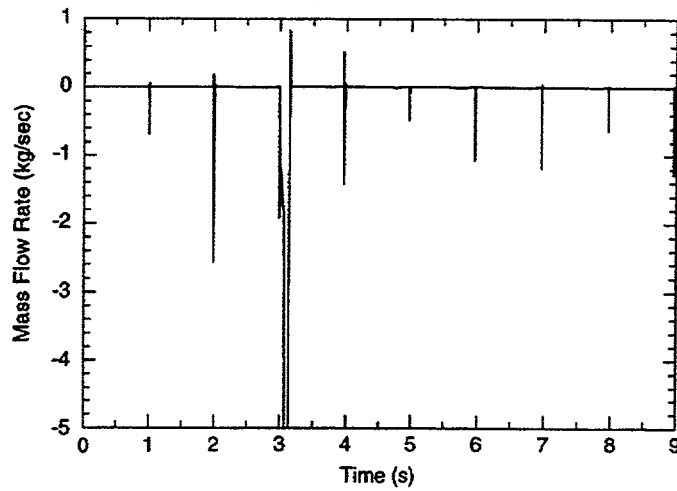


Figure 5.4.36 Single-Pipe Cold Water Flooding Test, Steam Flow Rate TRAC-B without Level Tracking

Single PIPE Cold Water Flooding Test

Steam Supply Rate (TRAC-B With Level Tracking)

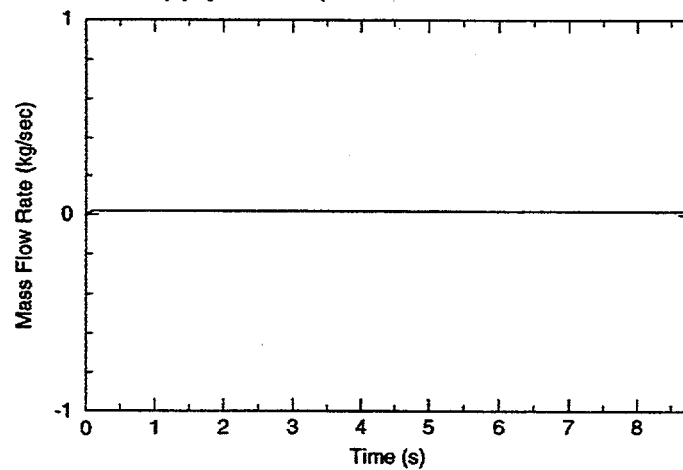


MFLOW-020011

Figure 5.4.37 Single-Pipe Cold Water Flooding Test, Steam Flow Rate, TRAC-B with Level Tracking

Single PIPE Cold Water Flooding Test

Steam Supply Rate (TRAC-M With Level Tracking)



mmvm-700011

Figure 5.4.38 Single-Pipe Cold Water Flooding Test, Steam Flow Rate, TRAC-M with Level Tracking

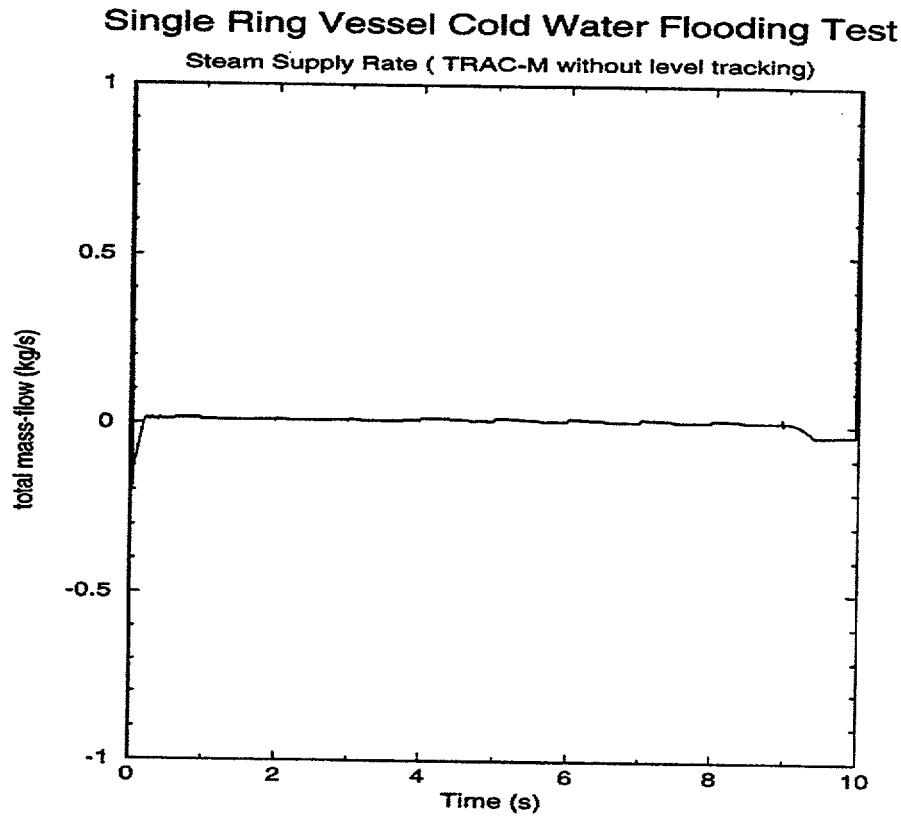


Figure 5.4.39 Single-Ring Vessel Cold Water Flooding Test, Steam Flow Rate, TRAC-M without Level Tracking

Single Ring Vessel Cold Water Flooding Test

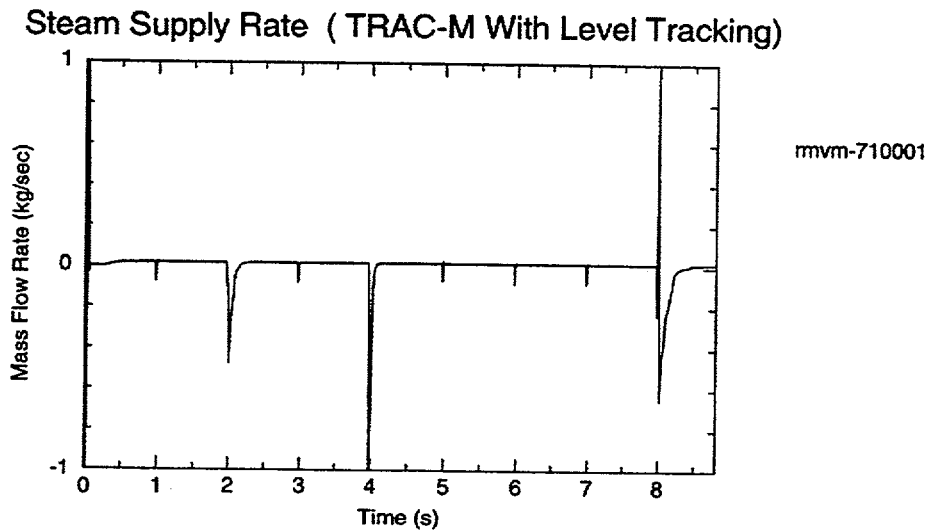
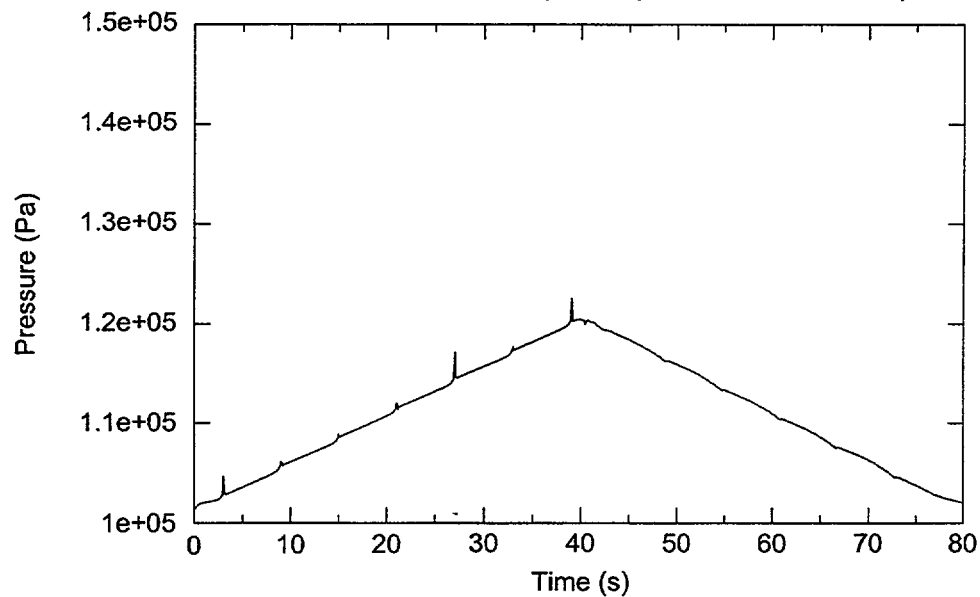


Figure 5.4.40 Single-Ring Vessel Cold Water Flooding Test, Steam Flow Rate, TRAC-M with Level Tracking

PIPE Fill & Drain Restart Test

Cell 1 Pressure (Complete Transient)

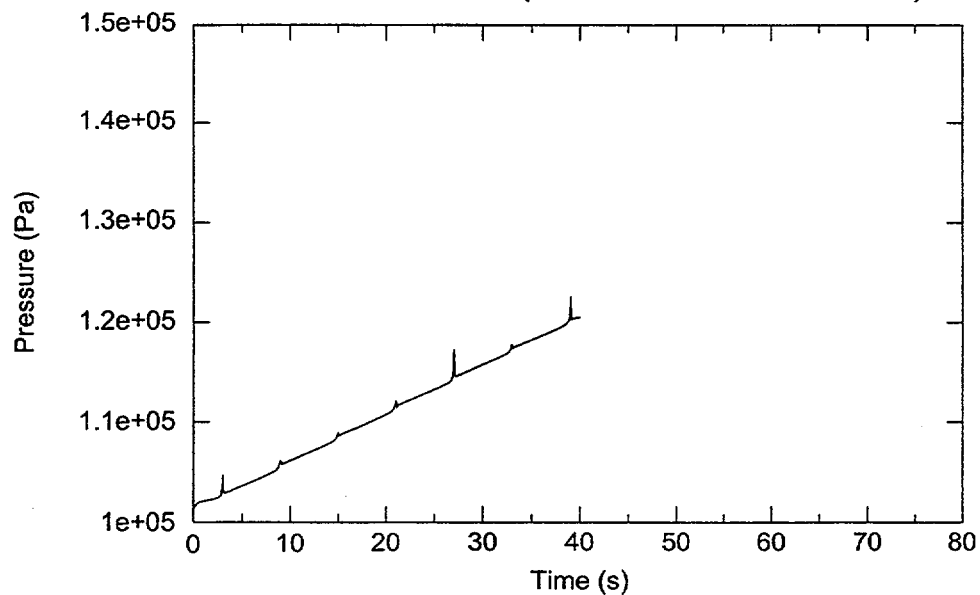


pn-70001

Figure 5.4.41 Complete PIPE Fill and Drain Transient

PIPE Fill & Drain Restart Test

Cell 1 Pressure (First Half of Transient)



pn-70001

Figure 5.4.42 First Half of the Transient

PIPE Fill & Drain Restart Test

Cell 1 Pressure (Second Half of Transient)

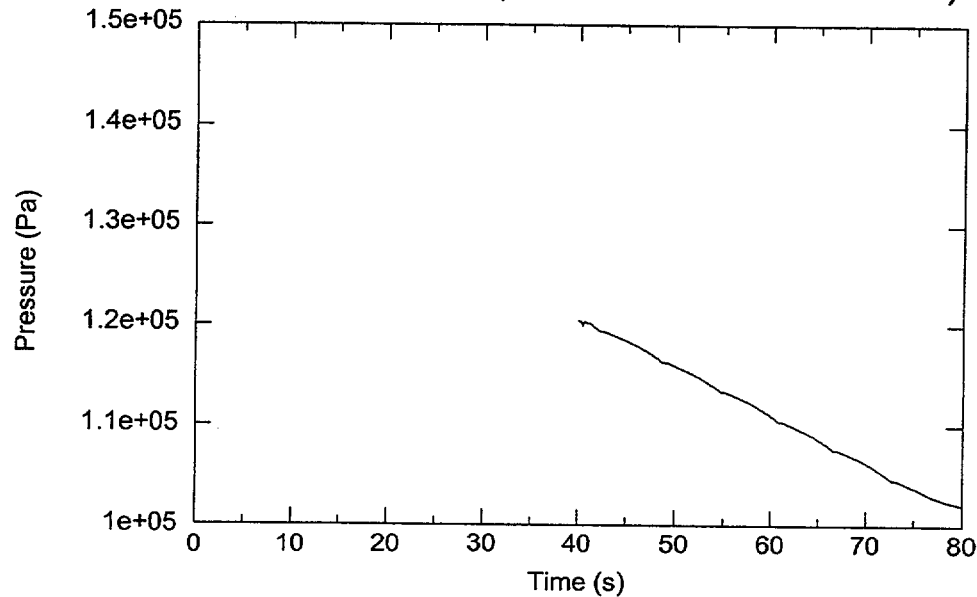


Figure 5.4.43 Second Half of the Transient

Single Ring Vessel Fill & Drain Restart Test

Bottom Cell Pressure (Complete Transient)

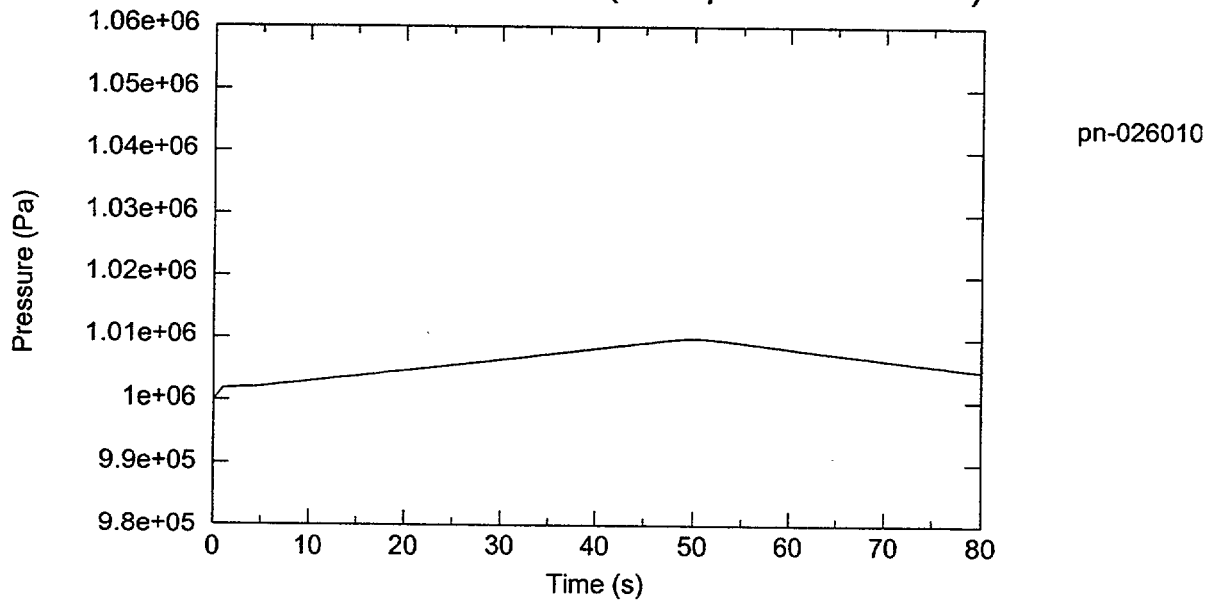


Figure 5.4.44 Complete Single-Ring Vessel Fill and Drain Transient

Single Ring Vessel Fill & Drain Restart Test

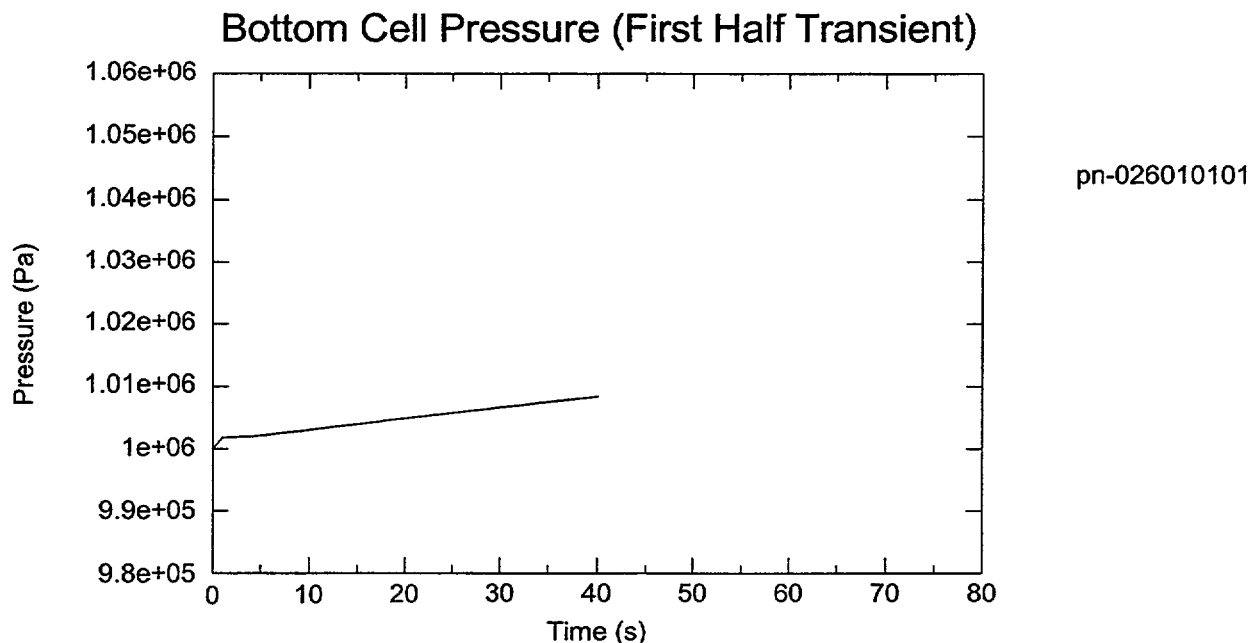


Figure 5.4.45 First Half of the Single-Ring Vessel Fill and Drain Transient

Single Ring Vessel Fill & Drain Restart Test

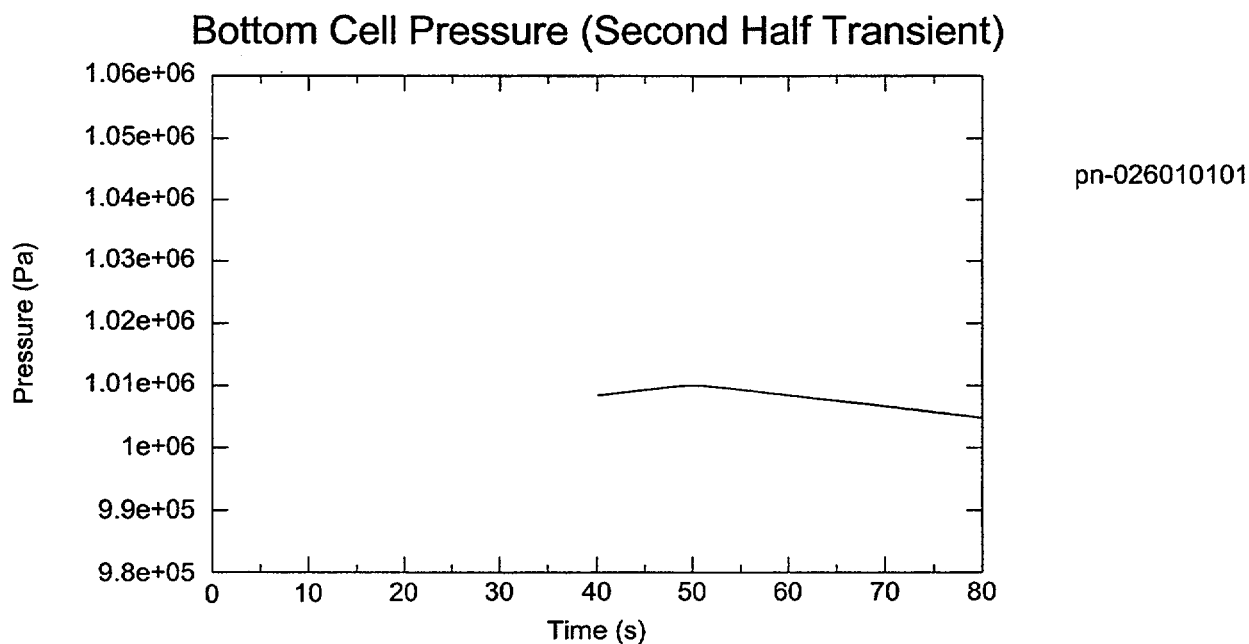


Figure 5.4.46 Second Half of the Single-Ring Vessel Fill and Drain Transient

5.5 BWR Control System

Both the TRAC-PF1 and the TRAC-BF1 control system models have been developed on the basis of similar design concepts, and have evolved from the control system model of TRAC-BD1. Nonetheless, significant differences between the two control system models have emerged as a result of different developmental paths. In order to preserve the functionality of the TRAC-BF1 control system model that model will be consolidated into TRAC-M. This section first compares the functionality of the two control system models, and then identifies a detailed list of consolidation items and defines the software requirements for incorporating these new features.

The four basic building blocks that comprise the TRAC-M control system include (1) signal variables, (2) control blocks, (3) component action tables, and (4) trips. As shown in Fig. 5.5.1, the signal variables serve as the interface between the control blocks and the rest of the TRAC-M building blocks. A TRAC-M user can select different types of signal variables from 104 choices, and can establish the input signals for the control blocks, trips, and component action tables. At the current stage of development, the TRAC-M signal variables only perform unidirectional data passing from the TRAC-M component and kinetics database to the control blocks. The calculated control block output data are directly referenced by the component action table, which is currently a part of the TRAC-M component data.

Unlike TRAC-M, TRAC-BF1 employs 16 input/output (I/O) variables for the interface between the control blocks and the component and kinetics database, as shown in Fig. 5.5.2. These I/O variables provide complete bidirectional data communication, and they also eliminate the intermediate step of the calculation that is performed in TRAC-M using the component action tables. Although TRAC-M has 104 types of signal variables, the following 9 of the 16 TRAC-BF1 I/O variables cannot be replaced by the existing TRAC-M signal variables:

- control rod position for 1-D kinetics
- control rod reactivity for the point kinetics
- total reactor power
- core average Boron concentration
- 1-D component cell center mixture enthalpy
- vessel downcomer water level
- turbine torque
- turbine speed
- feedwater heater shell-side control area and water level

Both codes have two different types of control blocks, including the user selected control block and the built-in steady-state controllers. As the control system model of these two codes

evolved from the TRAC-BD1 control system model, the original 61 TRAC-BD1 control blocks have been preserved in the TRAC-M code. In addition, five more control blocks (i.e, Time Delay, Proportional plus Integral (PI) controller, Proportional plus Integral plus Derivative (PID) controller, and two functional blocks) have been implemented into TRAC-M. Thus, no new user-selected control blocks from TRAC-BF1 need to be incorporated to TRAC-M. However, TRAC-BF1 has three built-in steady state controllers (the level controller, the pressure controller, and the core flow controller) that are specifically designed for BWR applications. These three steady-state controllers cannot be represented by the four existing TRAC-M constrained steady-state controllers and, therefore, need to be implemented.

When the control system model was first implemented into TRAC-BD1, each control block was assigned a unique ID number by the user. The block number need not be consecutive, but they do determine the order of the execution and, consequently, may affect the simulated results. The users are responsible for giving the correct order of the control block numbers, as determined by the logical relationships among these blocks. This feature may cause some user inconvenience—and, sometimes, poor simulation results. Improvements were, therefore, made to the TRAC-BF1 control system using an auto-sorting scheme, which automatically sorts the control blocks into three categories, including state variable control blocks, algebraic variable blocks, and blocks that belong to implicit loops. This scheme assigns an internal execution sequence number to each control block, and eliminates the requirement for users to sequentially number the control blocks for the correct order of execution.

TRAC-M still relies on the user-defined control block numbers to determine the execution order. In order to simplify the process of adding new control blocks into the TRAC-M model, a multiple pass card was provided to the user to override the execution order defined by the control block number (Ref. 5.5.1). However, TRAC-M still relies on user judgement to sort the control block at the input stage. Thus, the existing TRAC-BF1 control block auto-sorting scheme needs to be implemented into TRAC-M.

Control system model stability and accuracy were not a concern for either TRAC-B or TRAC-P until large T/H time step sizes could be used with the Courant-violating fast numerical scheme for TRAC-B and the SETS method for TRAC-M. A control system for the time step control scheme was built into TRAC-BF1 to allow the control system to advance several small computational steps within one T/H time step. The maximum allowable control system time step size is either one-half of the shortest time constant occurring in any of the state variable control blocks, or one-tenth of the shortest delay time occurring in any logic delay (LDLY) control blocks. With the control system time step sub-cycle and the implicit treatment of the control blocks within an implicit loop, TRAC-BF1 is able to preserve the accuracy of the control system calculation while the T/H system time step size is large.

The development of the TRAC-PF1 control system differs from that in TRAC-BF1. Instead of using a smaller control system time step size, the state-transition method analytic solution has been used to evaluate the Laplace-Transform Ordinary Differential Equation and the PI/PID-controller control blocks. This method is unconditionally stable and even more accurate than the semi-implicit numerics. However, the disadvantage of using this approach is that it relies on the user to choose the maximum time step size when the LDLY control block is used. At this point, a more reasonable approach is to merge both methods. This includes utilizing the TRAC-BF1

control system sub-cycling scheme to treat the LDLY control block, while also using the TRAC-M state-transition method analytic solution to achieve improved accuracy.

A complex control system usually requires that control blocks be linked together to form an implicit loop. TRAC-M does not explicitly treat an implicit loop, and relies on the multiple-pass card to define the computation sequence of the control blocks within an implicit loop. TRAC-BF1 treats all of the control blocks within an implicit loop as a unit, and places them in a sequence list. When the input to an implicit loop becomes available, the code simultaneously solves for all of the control block outputs implicitly. In the case of nonlinear control block functions, an iteration scheme is used. If convergence has not been obtained in a maximum of 20 iterations, the non-converged solution is treated as though it were converged. This special treatment of the implicit loop is considered better than that of TRAC-M. Thus, it will be merged with the basic TRAC-M control system model.

In summary, there are significant differences between the TRAC-M and TRAC-BF1 control system models. In order to preserve the TRAC-BF1 control system functionality, the following parts of the TRAC-BF1 control system model need to be integrated into TRAC-M.

- 1.0 Control block auto-sorting scheme
- 2.0 Implicit loop solution scheme
- 3.0 Control system time step sub-cycle
- 4.0 Three BWR steady-state controllers
- 5.0 Nine control system I/O signal variables
- 6.0 Input/output/dump/restart/graphics capability

The software requirement specifications for implementing these features into TRAC-M are discussed in detail in the following section.

5.5.1 Requirements

There are 6 major software requirement specifications and 10 sub-requirements that support the consolidation of the TRAC-BF1 control system model into TRAC-M. These requirements are detailed in the following paragraphs, and summarized in 5.5.1 following the discussion.

Requirement CNSYS 1.0: Control Block Auto-Sorting Scheme

The TRAC-BF1 auto-sorting algorithm automatically determines the execution sequence of the control blocks on the basis of their logic status within a control system. After this algorithm is implemented into TRAC-M, it should sort the blocks into an optimal execution order, and renumber the control blocks to reflect their sequence of execution. The user-assigned control block numbers should be retained for graphing and editing purposes, but they will not be used as part of the internal control system calculation.

The implemented auto-sorting algorithm will sort the control blocks in the following manner:

1. State variable blocks will be located in the control system input deck, and placed at the top of the sequence list, making certain that the blocks appear above their input in this list. Here, the state variable blocks include the following:

DINL --- double integrator with XOUT limited

INT --- integrator

INTM --- integrator with mode control

LAG --- first order lag

LINT --- limited integrator

LLAG --- lead-lag transfer function

SOTF --- second order transfer function

In addition, PI and PID controllers in TRAC-M will also be sorted as state variable blocks.

2. Algebraic variable blocks will then be added to the sequence list below the state variable blocks when all of their input blocks are already on the list. This may require several passes through the control system input deck.
3. In the event that all of the algebraic blocks cannot be sorted, the existence of an "implicit loop" will be noted. The implicit loops will then be placed as a unit in the sequence list, so that control blocks supplying inputs to the loop reside above the loop, and blocks requiring input from the loop reside below the loop.

Requirement CNSYS 2: Implicit Loop Solution Scheme

The algebraic control blocks in an implicit loop identified by the auto-sorting scheme will be solved as a unit using the TRAC-BF1 implicit loop solution scheme. The unit will be solved simultaneously for all the control block outputs in the implicit loop. The simultaneous solution implemented into TRAC-M will be performed in the following manner:

χ_1, \dots, χ_M are control block outputs in an implicit loop containing M control blocks. For the i th control block, the output could be represented in a general form where

$$\chi_i^{n+1} = f(\chi_i^{n+1}, \dots, \chi_M^{n+1}) \quad (5.5.1)$$

For linear control blocks, a linearization could be performed, and the solution of Equation 5.5.1 could be written as:

$$\chi_i^{n+1} = \chi_i^n + \sum_{j=1}^M \frac{df}{d\chi_j} (\chi_j^{n+1} - \chi_j^n) \quad (5.5.2)$$

This yields a set of simultaneous equations that will be solved by matrix inversion to yield an exact solution for χ_i^{n+1} , $i = 1, M$.

For nonlinear block functions, Equation 5.5.2 will be solved in the following iterative form:

$$\bar{\chi}_i^{n+1} = f(\chi_i^{n+1}, \dots, \chi_M^{n+1}) + \sum_{j=1}^M \frac{df}{d\chi_j} (\chi_j^{n+1} - \chi_j^n) \quad (5.5.3)$$

where '-' indicates new iterate values, and ' ' indicates previous iterate values. Iterations continue until fractional changes between successive iterate output values are less than 1.E-6 for all control blocks in the implicit loop. If the convergence has not been obtained in a maximum of 20 iterations, the non-converged solution will be treated as though it were converged, and a warning message will be issued.

The TRAC-BF1 matrix inversion subroutines (SGNDE and SGNSL) will also be implemented to solve Equation 5.5.2.

Requirement CNSYS 3: Control System Time Step Control

With the introduction of the state-transition analytic method solution, the accuracy of the TRAC-M Laplace transform control blocks is no longer limited by the control system time step size. Thus, TRAC-M will eliminate the criterion used in TRAC-BF1 to limit the control system time step size by setting it to half of the shortest time constant occurring in any of the state variable control blocks. Only the limitation imposed by the resolution of discontinuous transient events will be implemented into TRAC-M. The maximum control system time step size should be one-tenth of the shortest delay time occurring in any logical delay (LDLY) control block.

After the implementation, if the TRAC-M control system time step logic determines that the maximum allowable control system time step size is greater than or equal to the T/H time step size, the T/H time step size will be used for the control system time step size. If the maximum allowable control system time step size is less than the T/H time step size, the T/H time step size will be divided into the smallest number of equal intervals, such that the interval size is less than or equal to the maximum control system time step size. This interval will then be used as the control system time step size. In this manner, the control system calculation may be divided into several sub-steps while it catches up with the T/H calculation. At the end of this series of sub-steps, the control system calculation will be at the same time level as the T/H calculation.

Requirement CNSYS 4: BWR Steady-State Controllers

The three TRAC-BF1 BWR steady-state controllers that will be implemented into TRAC-M include the (1) BWR three-element water level controller, (2) core flow controller, and (3) BWR pressure controller. The internal computational schemes of these controllers will be invisible to the users. Only the I/O interfaces will be defined on the basis of input component ID and nodalization.

Requirement CNSYS 4.1: Vessel Water Level Controller

The level controller will be provided with the dynamic water level position inside a given component, along with the current feedwater line and steam line mass flow rates. The output will consist of the mass flow rate of the FILL component, which provides the inlet flow of the feedwater system. The following parameters will need to be provided to the controller during the steady-state calculation:

- water level set point (m)
- initial feedwater flow rate (kg/s)
- ID of the component (and the θ zone ID if it is a vessel) in which the collapsed water level will be calculated
- ID and the cell face number of the component in which the feedwater line mass flow rate will be detected
- ID and the cell face number of the component in which the steam line mass flow rate will be detected
- ID of the FILL component that provides the feedwater line mass flow rate

The values of the controller internal variables will be preserved during the regular dump, and the restart capability will be activated when this controller is used. After the steady state is achieved, the user may use the calculated FILL mass flow rate at the initiation of the “null” transient.

Requirement CNSYS 4.2: Flow Control System

This controller will perform the BWR core flow steady-state calculation by controlling the recirculation pump motor torque. It will detect the flow rate from a CHAN or JETP component, and calculate the appropriate pump motor torque. To do so, this controller will require the following input parameters:

- mass flow set point (kg/s)
- initial rated recirculation pump torque
- ID number of the component in which the flow rate will be detected, as well as the user’s identification of the flow detection location within the component
- ID number of the PUMP component in which the motor torque is to be controlled

Requirement CNSYS 4.3: Pressure Control System

This controller will detect the main steam line pressure and adjust the main steam line valve area to achieve the pressure set point. The controller requires the following input parameters:

- pressure set point
- time zero valve area fraction
- ID and the cell face number of the component in which the steam line pressure is to be detected
- ID number of the VALVE component in which the valve area is to be controlled

Requirement CNSYS 5: New Signal Variables

Eight TRAC-BF1 control system I/O variables will be added into TRAC-M in the form of signal variables. The functional requirements for each new signal variable are provided in the following sub-requirements.

Requirement CNSYS 5.1: Mixture Enthalpy

The current version of TRAC-M (V 3.044) does not have the cell-centered mixture enthalpy calculated for either 1-D or 3-D components. Two cell center arrays need to be incorporated into the 1-D and the 3-D component general database. A new subroutine will be called at the end of the PREP stage to calculate the cell center mixture enthalpy for all of the hydro cells. The same subroutine will also be called at the initiation stage before the control blocks and the signal variables are initialized. Given these coding changes, a new type of signal variable will be introduced to represent the cell-centered mixture enthalpy. This signal variable will use the same referencing scheme developed at the Pennsylvania State University to access the T/H database.

Requirement CNSYS 5.2: Vessel Downcomer Water Level

The vessel downcomer water level is used as the input signal to the BWR water level control system during both normal operation and accident conditions. A new two-dimensional array will be added to the 3-D vessel database to store the collapsed water level of each azimuthal sector. The data stored in the array will be updated at the end of the PREP stage during each time step. During the initialization stage, the array will be updated before the signal variables and control blocks are updated. Similar to the mixture enthalpy, the vessel level data in each radial zone will be accessed by a new type of signal variable.

Requirement CNSYS 5.3: Turbine Torque and Speed

The turbine torque or speed can be used as the input variable for the control system to adjust the turbine valve area and achieve the desired torque or speed output. Two new signal variables will be introduced to retrieve the turbine torque and speed data from the turbine component-specific data module. The input to these two signal variables will simply be the turbine component ID number.

Requirement CNSYS 5.4: Feedwater Heater Shell-Side Level and Control Area

The TRAC-BF1 control system is able to detect the feedwater heater component shell-side water level, and based on the level signal, adjusts the feedwater heater drain cooler outlet area to

simulate the feedwater heater shell-side water level control system. Two new signal variables will be introduced to represent the heater shell-side water level and control area. The signal variable representing the control area will use one control block output as the input, and will then be referenced by the feedwater heater component action table.

Requirement CNSYS 5.5: Core Average Boron Concentration

The TRAC-BF1 point kinetics model provides the control system with the core average boron concentration as one of the input variables. A new type of signal variable will be introduced into TRAC-M to perform a similar function (i.e., retrieving the core average boron concentration from the point kinetics module). The interface between the signal variable and the kinetics data array needs to be established. When the CHAN component is implemented, the point kinetics model will need to be modified to calculate the core average boron concentration when CHAN components are modeled. The implementation of a new signal variable will not be completed until the point kinetics model of the TRAC-M code is modified for BWR applications.

Requirement CNSYS 5.6: Point Kinetics Model Control Rod Reactivity

Two new signal variables will be implemented into the TRAC-M code as another interface between the point kinetics module and the control system. One signal variable will retrieve the user-defined control rod reactivity from the point kinetics model, while the other will pass a control block output to the point kinetics module as the additive control rod reactivity. In this way, the control system can simulate the control rod movement using the point kinetics model.

Requirement CNSYS 5.7: Control Rod Position

Similar to the two signal variables discussed above, a new signal variable will be introduced into TRAC-M to pass the control block output value to the spatial kinetics model as the control rod position. The user will define the control rod group ID as one input parameter of the signal variable. Then, the signal variable will retrieve one control block output as the calculated control rod group core insertion fraction, which should be a value between 0.0 and 1.0. This signal variable will be referenced by the TRAC Data Mapping Routine (TDMR) for the 3-D kinetics model and also the 1-D kinetics model. The user-defined control rod group ID number should be consistent with the definition of the 3-D kinetics user input.

Requirement CNSYS 5.8: Total Reactor Power

A new signal variable will be introduced into TRAC-M to use the total core power calculated by the spatial kinetics simulator (currently PARCS for 3-D analysis). The signal variable will retrieve the total core power from the TDMR module, and pass it to the control system as an input variable.

Requirement CNSYS 5.9: Pump Motor Torque

Unlike the pumps used for PWR primary loop circulation, the BWR recirculation pumps are driven by the motor-generator set, which provides varying pump motor torque. The recirculation pump impeller speed can be adjusted by the plant control system by varying the pump motor torque. In order to model this special feature, the TRAC-BF1 PUMP component was developed

with an option for the control system to control the pump motor torque. The current TRAC-M PUMP component does not allow the user to specify the motor torque. Instead, it provides the option to directly control the impeller angular speed.

The TRAC-M PUMP component torque equation will be modified to add the user-defined pump motor torque. The code will allow the user to define a component action table that provides the run-time pump motor torque.

Requirement CNSYS 6: Input/Output/Dump/Restart/Graphics Capability

All of the BWR steady-state controllers and the new signal variables are required to be implemented in a manner that is consistent with the existing controllers and signal variables, with a consistent I/O format. The dump and restart capabilities will also be activated for these variables, and the XTV module will be modified to accommodate the new controllers and signal variables.

5.5.2 Verification Testing and Assessment

As identified in Table 5.5.1, there are 6 major software requirements and 10 sub-requirements that must be implemented to consolidate the BWR control system functionality into TRAC-M. A total of 14 test problems are designed to test the code against these requirements. These test problems are summarized in Table 5.5.2, and discussed in the following paragraphs.

Test Problem 1

This test problem is designed to test the TRAC-M control system for Requirements CNSYS 1, 2, and 3. These three requirements include the control block auto-sorting, implicit loop solution scheme, and control system time step control. Two input decks will be developed for TRAC-M and TRAC-BF1. These two input decks will have identical T/H components and control systems. There will be only three T/H components consisting of a FILL, a PIPE and a BREAK. The PIPE component will have 10 vertical cells. The cell center pressure of cell #1 will be used as a dummy input variable for the control system. During the transient, the liquid originally residing in the PIPE will remain stagnant with no time dependent variable changes in the FILL and BREAK components.

The control systems for the two input decks will be identical, with each consisting of two independent loops. Fig. 5.5.3 shows a diagram of the control system. The first control block loop has only three control blocks, and is used to test the control system auto-sorting scheme when two independent control systems are modeled. The second control block of this loop represents a LDLY block. The delay time of this block will be adjusted during the testing to evaluate the implemented automatic control system time step sub-cycle algorithm required by Requirement CNSYS 3. The second control block loop has 21 control blocks that form three implicit loops, while one implicit loop includes a state variable control block. The control block ID numbers were randomly assigned to each control block to test the auto-sorting algorithm required by Requirement CNSYS 1. The results of this test problem will be compared with those of TRAC-BF1. The TRAC-M code should produce the same control block evaluation sequence as that of TRAC-BF1.

In addition, the new TRAC-M sorting scheme will be tested using the same input deck with "ICsSort" set to 2. When ICsSort is set to 2, the new auto-sorting algorithm is used. The result of the control block sorting will be examined based on the logical relationships among the control blocks.

Two test input decks will be prepared to represent the control system. The first, named "CnsysM.inp," is designed to test TRAC-M. The second, named "CnsysB.inp" is designed to generate the comparable TRAC-BF1 control block sorting results. The relevant portions of the control system sorting results are listed in Tables 5.5.3 and 5.5.4, which substantiate the conclusion that the two codes generate identical control block evaluation sequence lists. This proves that the implemented control block sorting algorithm reproduces the TRAC-BF1 control block sorting capability, and satisfies Requirement CNSYS 1.0.

The TRAC-BF1 type of control block sorting has a potential problem for creating an evaluation sequence that is inconsistent with the control block solution scheme. Consequently, a new control block sorting scheme was developed during the control system consolidation effort. A new test problem, "Cnsy1M.inp," was therefore developed to test the new auto-sorting scheme by examining the sorting sequence on the basis of the logical relationship among the control blocks. The results (shown in Table 5.5.5) demonstrate that the new sorting scheme generates the correct evaluation sequence for the control elements.

Test Problem 2

This test problem is designed to test the incorporation of the TRAC-BF1 implicit loop spontaneous solution scheme into TRAC-M. An algebraic implicit loop consisting of two control blocks is shown in Fig. 5.5.4. Both TRAC-BF1 and TRAC-M will be used to calculate the output of the "ADD" control block. The results generated by both codes should be in a good agreement.

Two input decks ("CnImpLoopB.inp" and "CnImpLoopM.inp") were prepared for TRAC-BF1 and TRAC-M. The calculated control block output by these two codes are shown in Fig. 5.5.5. It should be noted that both TRAC-M and TRAC-BF1 produced identical results. This shows that the incorporation of the implicit loop solution scheme meets Requirement CNSYS 2.0.

Although both TRAC-BF1 and TRAC-M produced the same results, the results shown in Fig. 5.5.5 are incorrect. The sum of two sine functions should always be less than 2.0. It is recommended that the error in the implemented TRAC-M control system implicit loop solution scheme be corrected.

Test Problem 3

This test problem will be used to test the control system time step sub-cycle algorithm using the same approach used in Test Problem 1. The logical delay time of the LDLY control block will be altered from 1.0 second to 0.1 second, forcing the code to perform control system time step sub-cycling. Although a smaller time step size may affect the control block output downstream of the LDLY control block, the output of other control blocks should remain the same. The output values of control block 650 (see Fig. 5.5.3) will be examined with different time step sizes.

Two TRAC-M input decks were derived from Test Problem 1. The first input deck, called "CnSysDelt1.inp," has 1.0 s defined for CBCONT1, while the second input deck, "CnSysDelt2.inp," has 0.1 s defined for CBCONT1. Since both input decks specify the maximum time step as 0.1 s, the first input deck will never need the control system time step sub-cycle. The second input deck, however, will lead the control system evaluation scheme into the time step sub-cycling. Fig. 5.5.6 shows the output of control block CB650 for these two cases (i.e., C1=1.0 or 0.1 s). It should be noted that the two outputs match, verifying the appropriate implementation of the control system time step sub-cycle algorithm. Therefore, the code meets Requirement CNSYS 3.0.

Test Problem 4

This test problem is designed to test the lumped flow controller of TRAC-M. As shown in Fig. 5.5.7, seven control blocks will be used to represent the control blocks that are lumped by flow controller 155. A sine wave signal, $11350(1.0+0.1\sin(0.1t))$, will be input to these two independent control systems, and the output values of the two control systems should be identical to meet Requirement CNSYS 4.2.

This test problem verifies the implementation of the lumped flow controller (control block type 203). Each control block of the lumped flow controller has been explicitly represented in the input deck, "ConFlow.inp," while the lumped controller appears as a single control block, CB155. It is expected that the output of CB155 should be identical to the output of the explicit flow control system. Fig. 5.5.8 shows the output from these two control systems, which are identical, thus verifying the correct implementation of the lumped control block. Therefore, the code meets Requirement CNSYS 4.2.

Test Problem 5

This test problem is designed to test the lumped level controller of TRAC-M. As shown in Fig. 5.5.9, 10 control blocks will be used to represent the control blocks that are lumped by level controller 155. A sine wave signal, $13(1.0+0.1\sin(0.1t))$, will be input to these two independent control systems, and the output values of the two control systems should be identical to meet Requirement CNSYS 4.1.

This test problem verifies the implementation of the lumped vessel downcomer water level controller (control block type 202), which is shown in Fig. 5.5.9. Each control block of the lumped level controller has been explicitly represented in the input deck, "ConLevel.inp," while the lumped controller appears as a single control block, CB155. It is expected that the output of CB155 should be identical to the output of the explicit level control system. Fig. 5.5.10 shows the output from these two control systems, which are identical, thus verifying the correct implementation of the lumped control block. Therefore, the code meets Requirement CNSYS 4.1.

Test Problem 6

This test problem is designed to test the lumped pressure controller of TRAC-M. As shown in Fig. 5.5.11, seven control blocks will be used to represent the control blocks that are lumped by

pressure controller 155 (control block type 204). A sine wave signal, $6.82E+6 (1.0+0.1\sin(0.1))$, will be input to these two independent control systems, and the output values of the two control systems should be identical to meet Requirement CNSYS 4.3.

Each control block of the lumped pressure controller has been explicitly represented in the input deck, "ConPress.inp," while the lumped controller appears as a single control block, CB155. It is expected that the output of CB155 should be identical to the output of the explicit level control system. Fig. 5.5.12 shows the output from these two control systems, which are identical, thus verifying the correct implementation of the lumped control block. Therefore, the code meets Requirement CNSYS 4.3.

Test Problem 7

This test problem is designed to assess the proper implementation of control rod reactivity, and to verify that the code meets Requirement CNSYS 5.6. As the first step, a constant control block will be used to define an input value of the signal variable which is a type "17." The signal variable output value should match the constant defined for the referenced control block.

A test input deck, "ConRodA.inp," has been developed to verify the proper implementation of this new signal variable type. The fourth signal variable is a type 17, and the referencing control block is -680, which is a constant control block with a constant value of 0.5. The output of this input deck indicates that the signal variable has a value of 0.5, which is equal to the output value of control block CB680, thus verifying the correct data passing between the control system and the new signal variable.

Although this test has verified the correct data passing between the control block and the signal variable, further testing will be needed when the signal variable is connected to the point kinetics model, which will be modified through the power module separation task.

Test Problem 8

This test problem is designed to assess the proper implementation of the control rod position, and to verify that the code meets Requirement CNSYS 5.6. Although the test results ultimately rely on running the coupled TRAC-M/PARCS code package, as the first step, this test problem will only verify the proper data passing between the signal variable and the control block. A constant control block will be used to define an input value of the signal variable which is a type "16." The signal variable output value should match the constant defined for the referenced control block.

A test input deck, "ConRodP.inp," has been developed to verify the proper implementation of the new signal variable type 16. The fourth signal variable of the input deck has a signal variable type 16, and the referencing control block is -680, which is a constant control block with a constant value of 0.5. The output of this input deck indicates that the signal variable has the value of 0.5, which is equal to the output value of control block CB680, thus verifying the correct data passing between the control system and the new signal variable.

Additional testing will be needed when this signal variable type is linked with the spatial kinetics code (PARCS) through the TDMR. It is expected that this new type of signal variable will enable a user to simulate the control of the control rod group position in a reactor core.

Test Problem 9

This test problem is designed to assess the proper implementation of two signal variables, the cell center mixture enthalpy and the vessel collapsed water level required by Requirements CNSYS 5.1 and 5.2. Since the cell center mixture enthalpy signal variable is designed to retrieve data from both 1-D and 3-D components, this test problem employs a 2-ring, 2-azimuthal sector, and 10-axial level vessel component, and 6 pipe components, as shown in Fig. 5.5.13. Each azimuthal sector of the vessel component is isolated from the others, except at the tenth level. Initially, each sector is filled with water with different water column heights. Zero flow boundary conditions are defined for all of the PIPE components that are connected to the bottom of the VESSEL component. After the steady state is reached, the signal variables retrieving the collapsed water level should match the hand calculation, as well as the mixture enthalpy signal variables.

This test problem is designed to verify implementation of two new types of signal variables, types 105 and 106. A signal variable of type 105 is designed to retrieve the cell center mixture enthalpy from both the 1-D and the 3-D hydraulic components. A signal variable of type 106 is designed to retrieve the collapsed vessel downcomer water level in a specified vessel azimuthal sector. The TRAC-M test input deck, "SigHmLvl.inp," uses a PIPE component (PIPE-26), a VESSEL component (VESSEL-26), and 13 signal variables to test the code. Since the fluid mass in the vessel remains constant, the collapsed water level in each azimuthal sector will not change, and the verification can easily be achieved by comparing the initial water level with the steady-state collapsed water level outputs. It is expected that the signal variable output value should equal the initial water level height.

Table 5.5.6 lists the definitions and final output of all signal variables. The hand-calculated water level and cell center mixture enthalpy, the collapsed water level, and the cell center mixture enthalpy indicated by the signal variables are also tabulated in Table 5.5.6. It should be noted that TRAC-M reproduced the results of the hand calculation well, verifying that the newly implemented signal variable types, 105 and 106, can be used to retrieve the collapsed water level and the mixture enthalpy data from the TRAC-M T/H database. Thus, the code meets Requirements CNSYS 5.1 and 5.2.

Test Problem 10

This test problem is designed to assess the introduction of the pump motor torque variable into the PUMP component, as required by Requirement CNSYS 5.9. The test problem consists of a PUMP component and two BREAK components, and is designed to test the new pump motor torque feature, which is introduced into the TRAC-M pump model to model BWR recirculation pumps. The input deck, "PumpTorq.inp," contains PUMP component 700 and BREAK components 701 and 702. A CONSTANT type of control block, -680, is used to provide the PUMP component with the desired pump motor torque. The PUMP component type "ipmpty" is set to 3, and a pump motor torque table, "pmpmt," is provided. In order to test the pump motor torque feature, all of the coefficients for the friction torque calculation (TFR series) are set equal to 0.0. The resulting friction torque becomes zero and, if the pump motor torque is correctly implemented, the pump impeller hydraulic torque should be equal to the motor torque when the steady-state condition is achieved.

The test problem was run to 800 s, and the steady-state condition was achieved. Control block -680 set the pump motor torque to 500 N*M. From the output file, the pump impeller hydraulic torque is shown to be 499.175 N*M, verifying the appropriate implementation of the pump motor torque. Thus, the code meets Requirement CNSYS 5.9.

Test Problem 11

This test problem is designed to test the new signal variables that will be developed to retrieve the turbine torque and speed from the turbine component (Requirement CNSYS 5.3). The test problem will be built on the basis of the single turbine component test problem used in Section 5.7, Fig. 5.7.3. As shown in Fig. 5.7.3, the inlet boundary condition will be replaced by a FILL component, which will be provided with a time-dependent steam mass flow rate calculated by the control system. The signal variable for the turbine torque or speed will be referenced by a control block, and the control system will calculate the correct mass flow rate through the turbine to achieve the desired turbine torque or speed. The code should be able to pass the consistent turbine torque and speed data from the component database to the signal variable. Testing will be performed in a future version of the code and the results will be considered acceptable if the plots of these two new signal variables match the turbine torque and speed.

Test Problem 12

The capability to control the feedwater heater drain cooler outlet flow area, as required by Requirement CNSYS 5.4, will be examined using this test problem. The input deck has been developed on the basis of the feedwater heater test problem in Section 5.1. The two signal variables to be used include the feedwater heater shell-side water level and the drain cooler outlet flow area, which is calculated by the control system. These two signal variables should be able to pass the data between the feedwater heater components and the control system. The control system is designed to stabilize the shell-side water level by adjusting the drain cooler outlet flow area. Test results shown in Section 5.1 verify that Requirement CNSYS 5.4 has been met.

Test Problem 13

This test problem is designed to evaluate the code against Requirements CNSYS 5.7 and 5.8. The input deck for the Westinghouse four-loop test problem and the corresponding PARCS and 1-D kinetics input decks will be modified to enable the movement of control rod groups. The new signal variable that retrieves the total core power information from the spatial kinetics model will pass the time-dependent total core power from the TDMR to the control system. After the control system compares the total core power with the set point, it will then calculate the proper control rod group position in the core. A new core power control system will be built to control the core power by adjusting the control rod positions. A power maneuver transient will be simulated to decrease the power from its initial value of 3250 MW to 3000 MW. The control system should be capable of adjusting the control rod position to achieve the desired power level.

This test problem also requires the proper implementation of the control rod position modeling capability in the TDMR, GI, PARCS, and 1-D kinetics modules, along with proper control rod mapping for either BWR or PWR applications. This test problem will be performed in a future version of the code.

Test Problem 14

This test problem is designed to test the aggregate performance of three BWR steady-state controllers, as required by Requirements CNSYS 4.1, 4.2, and 4.3. In addition, this test problem will be used to test whether the code meets Requirements CNSYS 5.1, 5.2, and 5.6. This test problem will model the entire BWR vessel internal flow field from the feedwater injection to the turbine stop valve. The input deck will include CHAN components for the reactor core, two JETP components, and two PUMP components representing the recirculation loops and the 3-D vessel component modeling of the lower plenum, upper plenum, and steam separators and dryers.

After the base input deck is developed, the steady-state level controller will be activated to evaluate the code against Requirement CNSYS 4.1. The water level controller will collect three signals from the system, including the feedwater line mass flow rate, steam line mass flow rate, and vessel downcomer water level position, which will be made available through a new signal variable required by Requirement CNSYS 5.2. The output of this steady-state controller is the feedwater mass flow rate imposed on the feedwater line FILL component. The results will be compared with the calculations without the level controller. With the level controller activated, the code should be able to stabilize the downcomer water level much faster than without the level controllers activated.

The flow controller will be activated to adjust the recirculation pump speed and accelerate the achievement of the desired core flow. A comparison will be made between the runs with and without the steady-state flow controller. The code should converge with a stable total core flow equal to the set point and, therefore, satisfy Requirement CNSYS 4.2.

Based on this input deck, the pressure controller will also be activated to use the steam dome pressure from the vessel component and correct the turbine valve opening area. It is expected that the controller will accelerate the achievement of the desired steam dome pressure. The results will be evaluated against Requirement CNSYS 4.3.

This test problem will also be used to test the code against requirements CNSYS 5.5 and CNSYS 5.6, with the TRAC-M point kinetics model modified through another independent task. A new signal variable representing the core average boron concentration, and another new signal variable retrieving the calculated control rod reactivity, will pass the boron concentration and the control rod reactivity to the point kinetics model. A boron injection and a control rod insertion transient will be initiated to test the code. It is expected that the total core power will decrease as a result of the introduction of additional negative reactivity.

5.5.3 Conclusions

A total of 14 test problems were used to verify the correct implementation of the new control system features. The first 10 test problems show that the control system is correctly implemented in TRAC-M, Version 3690. The remaining four test problems will be executed when other relevant tasks are completed and new versions are created. The remaining work includes the following four tasks:

- testing of the turbine component
- testing of the power separation module when it is completed in a future version

- testing of TDMR control rod position data passing when completed in a future version
- testing of the control system in the Browns Ferry TRAC-M LOCA input deck, when its development is completed

REFERENCES

- 5.5.1 Spore, J.W., et al., "TRAC-PF1/MOD2: Volume I. Theory Manual," U.S. Nuclear Regulatory Commission, NUREG/CR-5673, July 21, 1993.

Table 5.5.1 Control System Consolidation Software Requirement Specifications

Requirement #		Requirement Descriptions
CNSYS 1.		Control Block Auto-Sorting
CNSYS 2.		Implicit Loop Solution Scheme
CNSYS 3.		Control System Time Step Control
CNSYS 4.	CNSYS 4.1	Water Level Controller
	CNSYS 4.2	Flow Controller
	CNSYS 4.3	Pressure Controller
CNSYS 5.	CNSYS 5.1	Mixture Enthalpy Signal Variable
	CNSYS 5.2	Vessel Downcomer Level Signal Variable
	CNSYS 5.3	Turbine Torque and Speed Signal Variable
	CNSYS 5.4	Feedwater Heater Shell Side Water Level and Control Area Signal Variable
	CNSYS 5.5	Core Average Boron Concentration Signal Variable
	CNSYS 5.6	Control Rod Reactivity Signal Variables For Point Kinetics Model
	CNSYS 5.7	Control Rod Position Signal Variable For Spatial Kinetics Model
	CNSYS 5.8	Total Reactor Power From Spatial Kinetics Model
	CNSYS 5.9	Pump Motor Torque
CNSYS 6.		Input/Output/Dump/Restart/Graphics

Table 5.5.2 Control System Test Problem List

Requirements	Requirement Descriptions	Test
CNSYS 1.	Control Block Auto-Sorting	1
CNSYS 2.	Implicit Loop Solution Scheme	2
CNSYS 3.	Control System Time Step Control	3
CNSYS 4.1	Water Level Controller	5
CNSYS 4.2	Flow Controller	4
CNSYS 4.3	Pressure Controller	6
CNSYS 5.1	Mixture Enthalpy Signal Variable	9
CNSYS 5.2	Vessel Downcomer Level Signal Variable	9
CNSYS 5.3	Turbine Torque and Speed Signal Variable	11
CNSYS 5.4	Feedwater Heater Shell Side Water Level and Control Area Signal Variable	12
CNSYS 5.5	Core Average Boron Concentration Signal Variable	
CNSYS 5.6	Control Rod Reactivity Signal Variables For Point Kinetics Model	7
CNSYS 5.7	Control Rod Position Signal Variable For Spatial Kinetics Model	8
CNSYS 5.8	Total Reactor Power From Spatial Kinetics Model	13
CNSYS 5.9	Pump Motor Torque	10
CNSYS 6.	Input/Output/Dump/Restart/Graphics	1-13

Table 5.5.3 TRAC-BF1 Control Block Sorting Results

CONTROL BLOCK NUMBER SEQ	CONTROL BLOCK TYPE USER	CONTROL BLOCK 1ST INPUT NUMBER	CONTROL BLOCK 2ND INPUT NUMBER	CONTROL BLOCK 3RD INPUT NUMBER	CONTROL BLOCK OUTPUT I/O NO.	CONTROL BLOCK 1ST CONSTANT	CONTROL BLOCK 2ND CONSTANT	CONTROL BLOCK GAIN	CONTROL BLOCK MAXIMUM VALUE	CONTROL BLOCK MINIMUM VALUE	CONTROL BLOCK INITIAL VALUE	CONTROL BLOCK NAME
1	155	23	12	0	0	0.000D+00	0.000D+00	1.000D+00	1.000D+03	-1.000D+03	0.000D+00	fw w
2	510	26	16	0	0	1.000D+00	1.000D+00	1.000D+00	1.000D+03	-1.000D+03	0.000D+00	totc
3	600	26	4	0	0	1.000D+00	1.000D+00	1.000D+00	1.000D+03	-1.000D+03	0.000D+00	totc
4	650	23	17	0	0	0.000D+00	0.000D+00	1.000D+00	1.000D+03	-1.000D+03	0.000D+00	totc
5	300	49	-1	0	0	0.000D+00	0.000D+00	1.000D+00	1.000D+03	-1.000D+03	0.000D+00	totd
6	380	43	-2	0	0	0.000D+00	0.000D+00	1.000D+00	1.000D+03	-1.000D+03	0.000D+00	not
7	400	49	-1	0	0	0.000D+00	0.000D+00	7.070D-01	1.000D+03	-1.000D+03	0.000D+00	sen
8	580	9	0	0	0	1.000D-01	0.000D+00	1.000D+00	1.000D+03	-1.000D+03	1.000D-01	cons
9	680	9	0	0	0	5.000D-01	0.000D+00	1.000D+00	1.000D+03	-1.000D+03	5.000D-01	cons
10	100	39	3	9	0	1.000D+00	1.000D+00	1.000D+00	1.000D+03	-1.000D+03	0.000D+00	sen
11	200	39	5	9	0	1.000D+00	1.000D+00	1.000D+00	1.000D+03	-1.000D+03	0.000D+00	fw w
12	280	27	6	0	0	1.000D+00	1.000D-02	1.000D+00	1.000D+03	-1.000D+03	0.000D+00	fw w
13	800	59	11	2	0	1.000D+00	1.000D+00	1.000D+00	1.000D+03	-1.000D+03	0.000D+00	totc
14	90	39	13	9	0	1.000D+00	1.000D+00	1.000D+00	1.000D+03	-1.000D+03	0.000D+00	totc
15	150	12	11	0	0	0.000D+00	0.000D+00	1.000D+00	1.000D+03	-1.000D+03	0.000D+00	fw w
16	610	54	13	8	0	1.000D+00	1.000D+00	1.000D+00	1.000D+03	-1.000D+03	0.000D+00	totc
17	700	59	15	7	0	1.000D+00	1.000D+00	1.000D+00	1.000D+03	-1.000D+03	0.000D+00	totc
18	420	54	1021	8	0	1.000D+00	1.000D+00	1.000D+00	1.000D+03	-1.000D+03	0.000D+00	totc
19	50	54	1022	8	0	1.000D+00	1.000D+00	1.000D+00	1.000D+03	-1.000D+03	0.000D+00	totc
20	80	59	10	1019	0	1.000D+00	1.000D+00	1.000D+00	1.000D+03	-1.000D+03	0.000D+00	totc
21	520	59	14	1023	0	1.000D+00	1.000D+00	1.000D+00	1.000D+03	-1.000D+03	0.000D+00	totc
22	60	12	1020	0	0	1.000D+00	1.000D+00	1.000D+00	1.000D+03	-1.000D+03	0.000D+00	totc
23	320	12	1018	0	0	1.000D+00	1.000D+00	1.000D+00	1.000D+03	-1.000D+03	0.000D+00	totc
24	210	39	20	9	0	0.000D+00	0.000D+00	1.000D+00	1.000D+50	-1.000D+50	0.000D+00	totc

Table 5.5.4 TRAC-M Control System Sorting Results

Control System Evaluation Sequence					
Control Element	ID	Sequence Number	Input (1)	Input (2)	Input (3)
Type					
Signal variable	1	1			
Signal variable	2	2			
Trip #	100	3			
Signal variable	3	4			
Control block	-155	5	-280	0	0
Control block	-510	6	-610	0	0
Control block	-600	7	-650	0	0
Control block	-650	8	-700	0	0
Control block	-300	9	1	0	0
Control block	-380	10	3	0	0
Control block	-400	11	-300	0	0
Control block	-580	12	0	0	0
Control block	-680	13	0	0	0
Control block	-100	14	-600	-680	0
Control block	-200	15	-300	-680	0
Control block	-280	16	-380	0	0
Control block	-800	17	-200	-510	0
Control block	-90	18	-800	-680	0
Control block	-150	19	-200	0	0
Control block	-610	20	-800	-580	0
Control block	-700	21	-400	-150	0
Control block	-420	22	-10520	-580	0
Control block	-50	23	-10060	-580	0
Control block	-80	24	-100	-10050	0
Control block	-520	25	-90	-10320	0
Control block	-60	26	-10080	0	0
Control block	-320	27	-10420	0	0
Control block	-210	28	-80	-680	0
Trip #	1	29			

Table 5.5.5 TRAC-M Control Block Sorting Results with ICsSort=2

Control System Evaluation Sequence					
Control Element Type	ID	Sequence Number	Input (1)	Input (2)	Input (3)
Signal variable	1	1			
Signal variable	2	2			
Trip #	100	3			
Signal variable	3	4			
Control block	-300	5	1	0	0
Control block	-380	6	3	0	0
Control block	-400	7	1	0	0
Control block	-580	8	0	0	0
Control block	-680	9	0	0	0
Control block	-200	10	-300	-680	0
Control block	-280	11	-380	0	0
Control block	-150	12	-200	0	0
Control block	-155	13	-280	0	0
Control block	-700	14	-400	-150	0
Control block	-650	15	-700	0	0
Control block	-600	16	-650	0	0
Control block	-100	17	-600	-680	0
Control block	-80	18	-100	-10050	0
Control block	-60	19	-10080	0	0
Control block	-50	20	-10060	-580	0
Control block	-210	21	-80	-680	0
Control block	-800	22	-200	-510	0
Control block	-90	23	-800	-680	0
Control block	-610	24	-800	-580	0
Control block	-510	25	-610	0	0
Control block	-520	26	-90	-10320	0
Control block	-420	27	-10520	-580	0
Control block	-320	28	-10420	0	0
Trip #	1	29			

Table 5.5.6 Test Problem 9 Signal Variable Definitions and Results

Signal Variable ID	Signal Variable Type	Component ID	Cell ID	Hand Calculation	TRAC-M Output
2	105 (mixture enthalpy)	26	4002	3.77E+05 (J/kg)	3.77103E+05 (J/kg)
3	106 (collapsed water level)	26	1000	0.4 (m)	0.4 (m)
4	106	26	2000	0.7 (m)	0.7 (m)
5	106	26	3000	1.0 (m)	1.0 (m)
6	106	26	4000	1.3 (m)	1.3 (m)
7	105	26	4001	2.39E+05 (J/kg)	2.389715E+05 (J/kg)
8	105	26	4010	3.75E+05 (J/kg)	3.748023E+05 (J/kg)
9	105	26	1001	1.13E+05 (J/kg)	1.133712E+05 (J/kg)
10	105	26	2001	1.55E+05 (J/kg)	1.552393E+05 (J/kg)
11	105	700	1	1.13E+05 (J/kg)	1.133718E+05 (J/kg)
12	105	700	2	1.55E+05 (J/kg)	1.552393E+05 (J/kg)
13	105	700	3	1.97(J/kg)	1.971062E+05 (J/kg)
14	105	700	4	2.39E+05 (J/kg)	2.389714E+05 (J/kg)

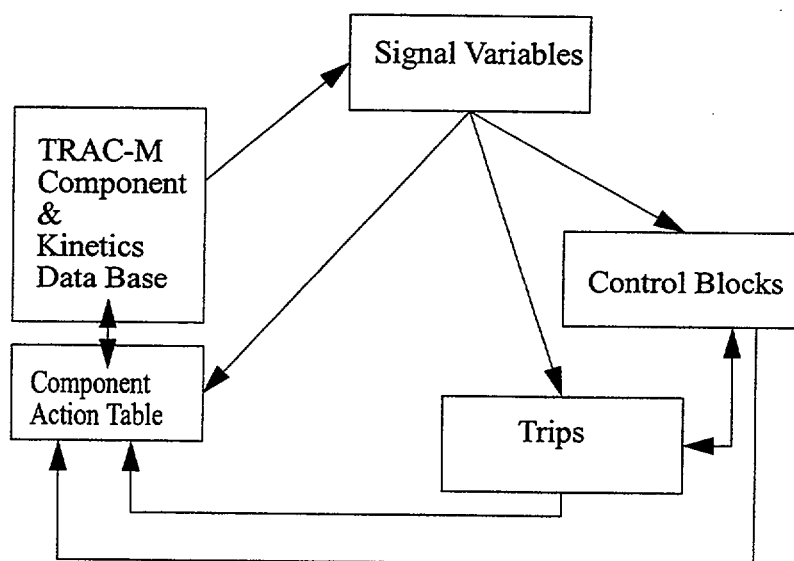


Figure 5.5.1 TRAC-M Control System Data Passing Flow Diagram

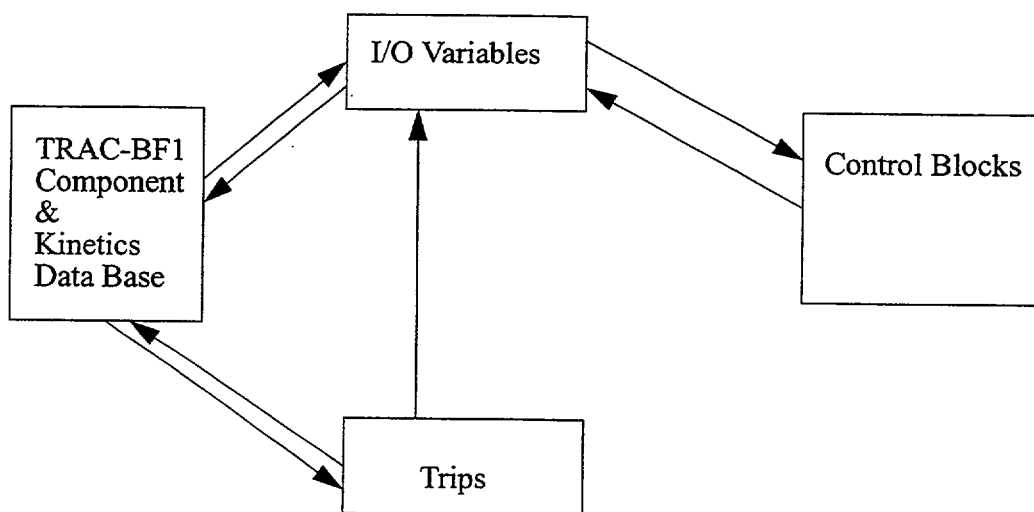


Figure 5.5.2 TRAC-BF1 Control System Data Passing Flow Diagram

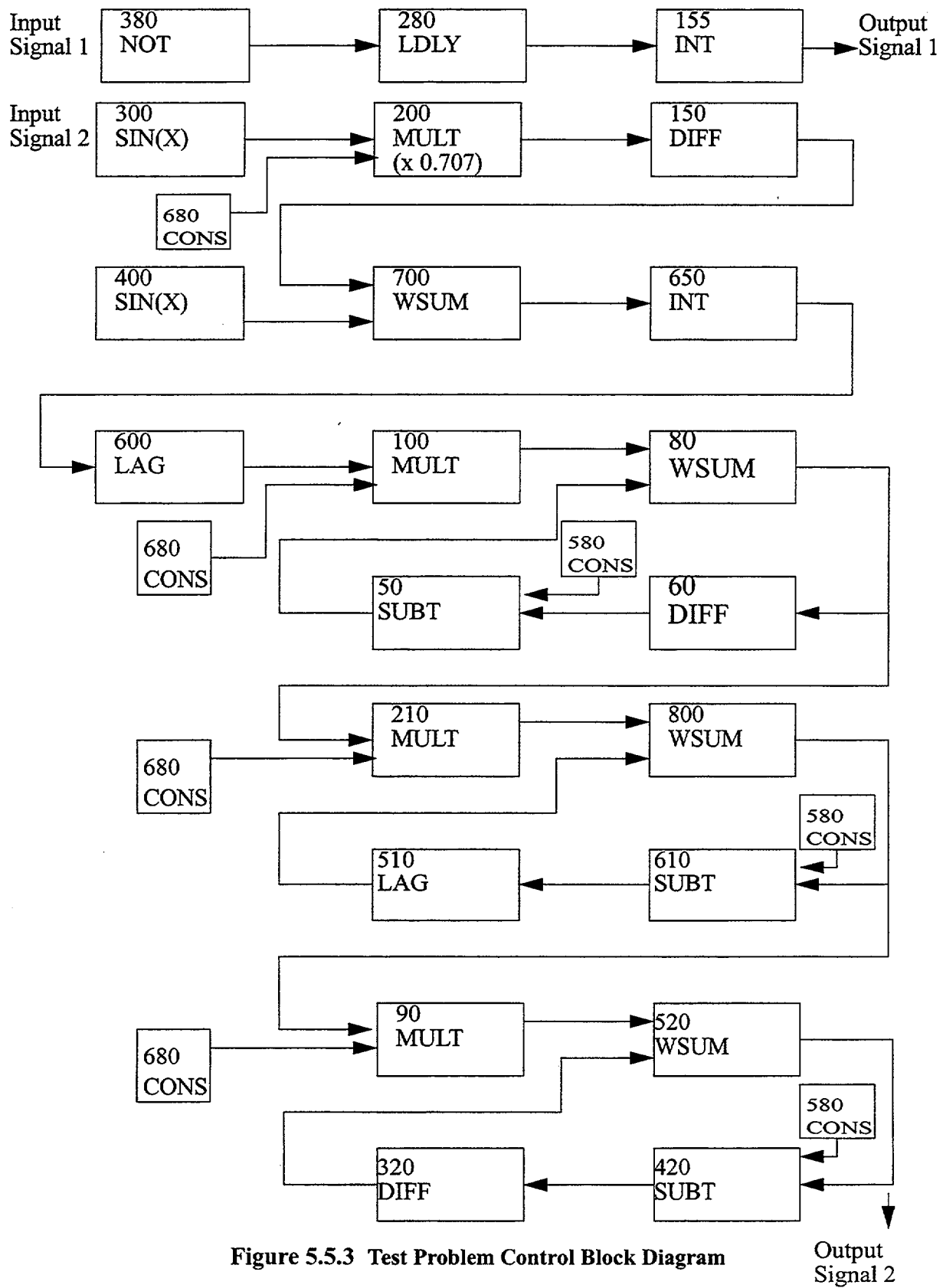


Figure 5.5.3 Test Problem Control Block Diagram

$$Z = \sin(Z) + \sin(0.1t)$$

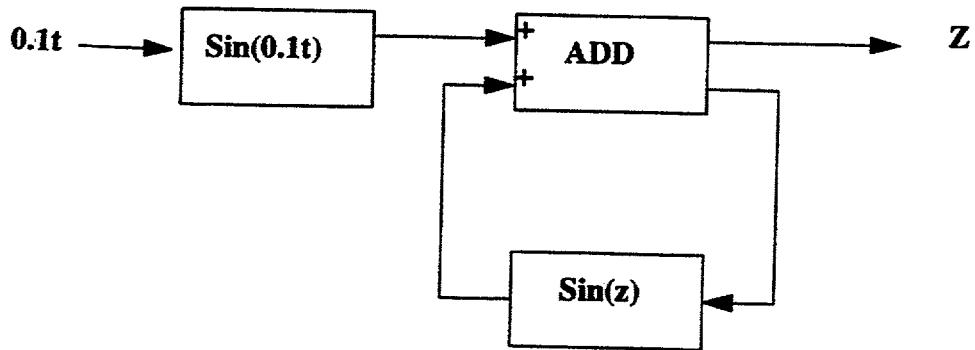


Figure 5.5.4 Implicit Loop Solution Test Problem

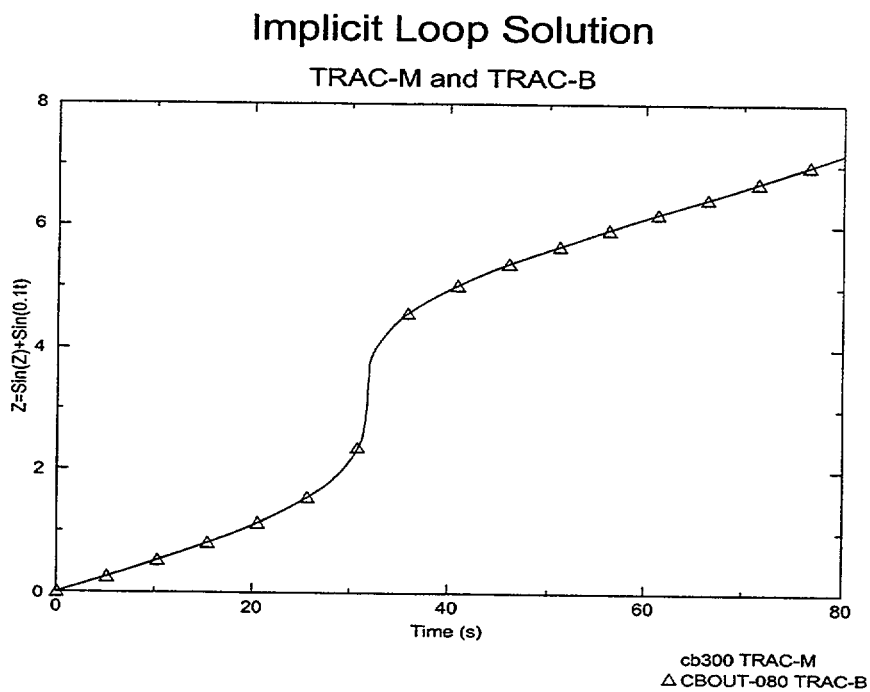


Figure 5.5.5 TRAC-M and TRAC-BF1 Implicit Loop Solution Scheme

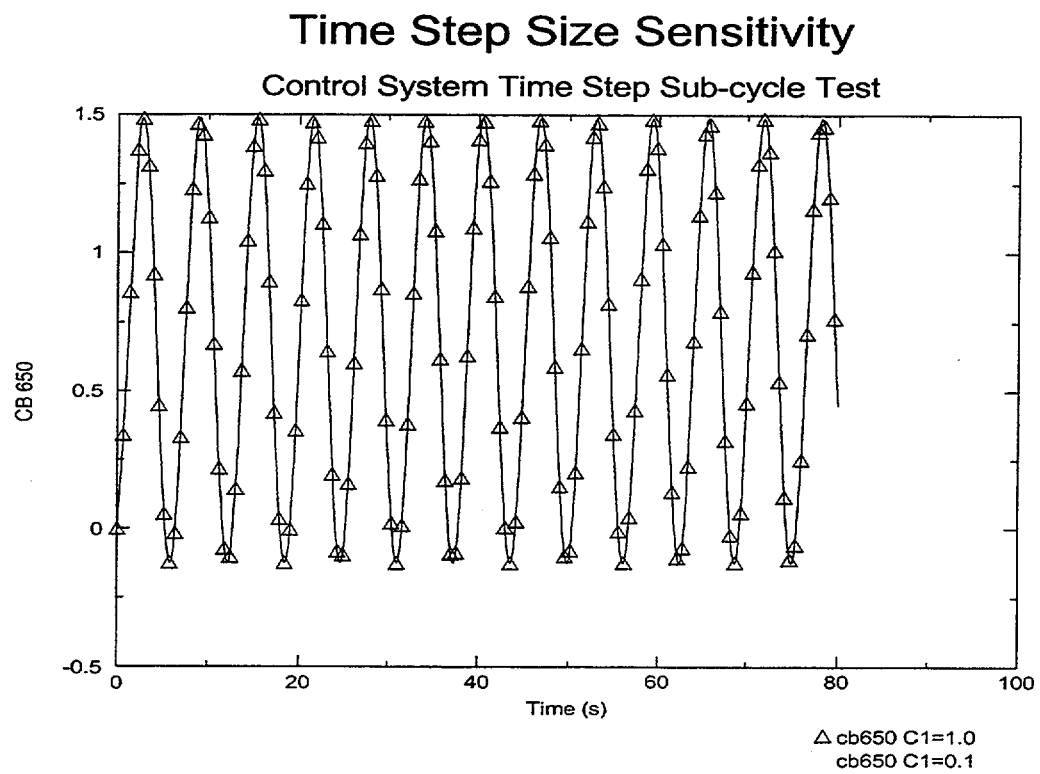


Figure 5.5.6 TRAC-M Control System Time Step Size Sensitivity Test

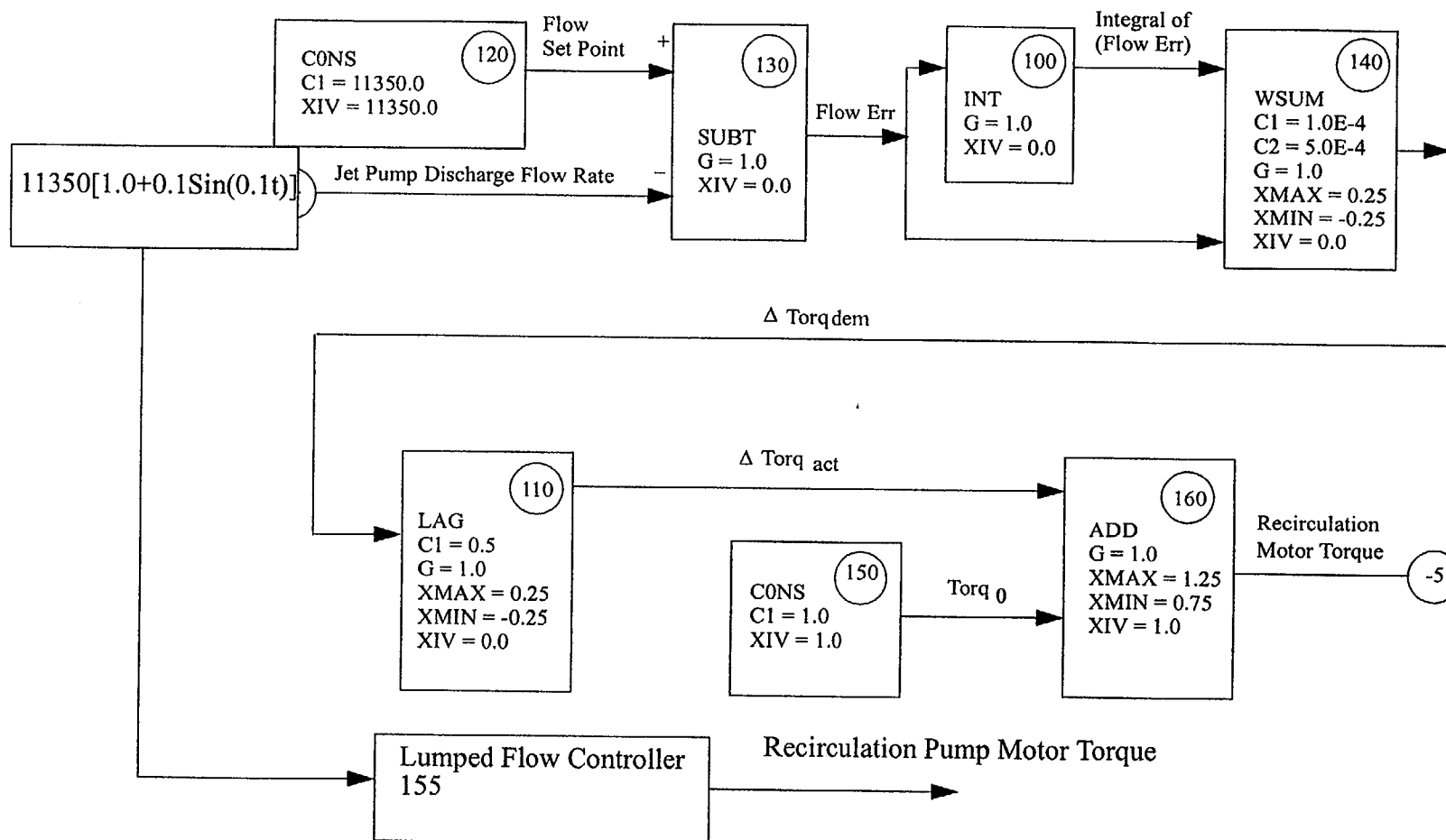


Figure 5.5.7 Lumped Flow Controller Test Problem

Flow Controller Test

Comparison Between Lumped Controller and Separated Controller

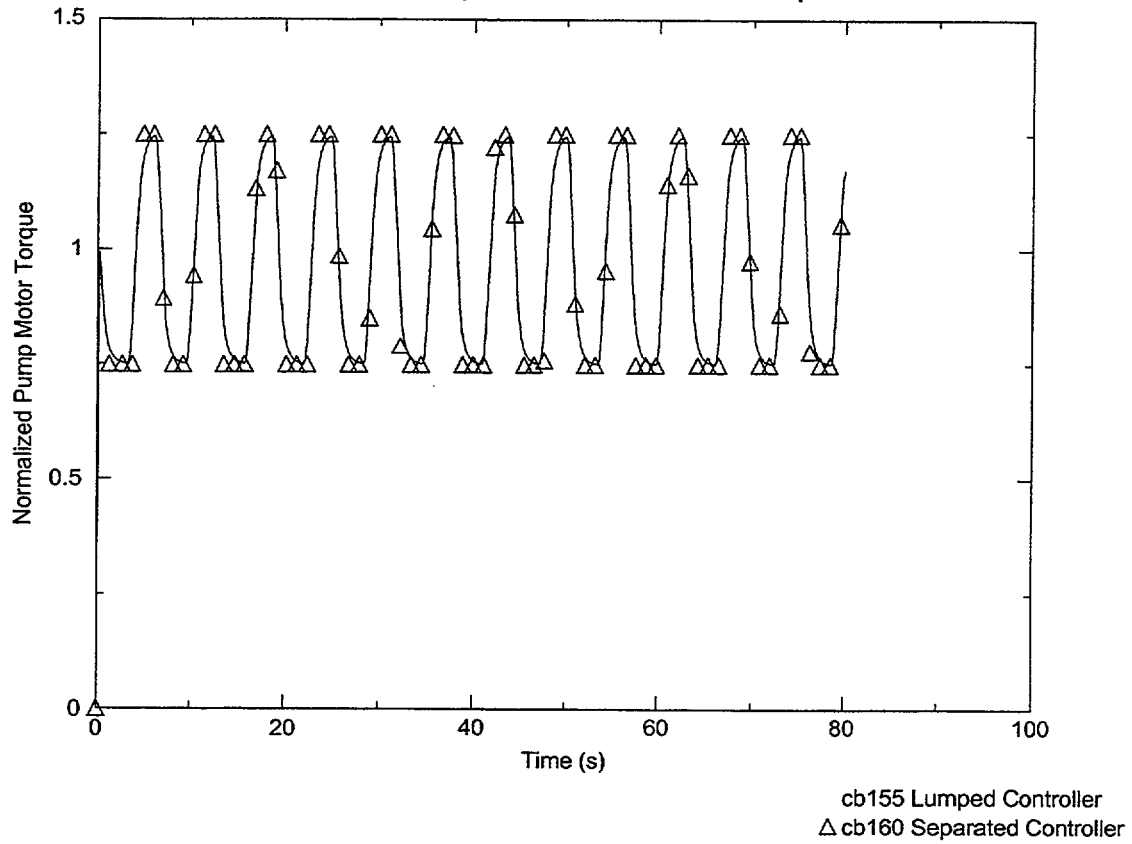


Figure 5.5.8 Lumped Flow Controller Test Results

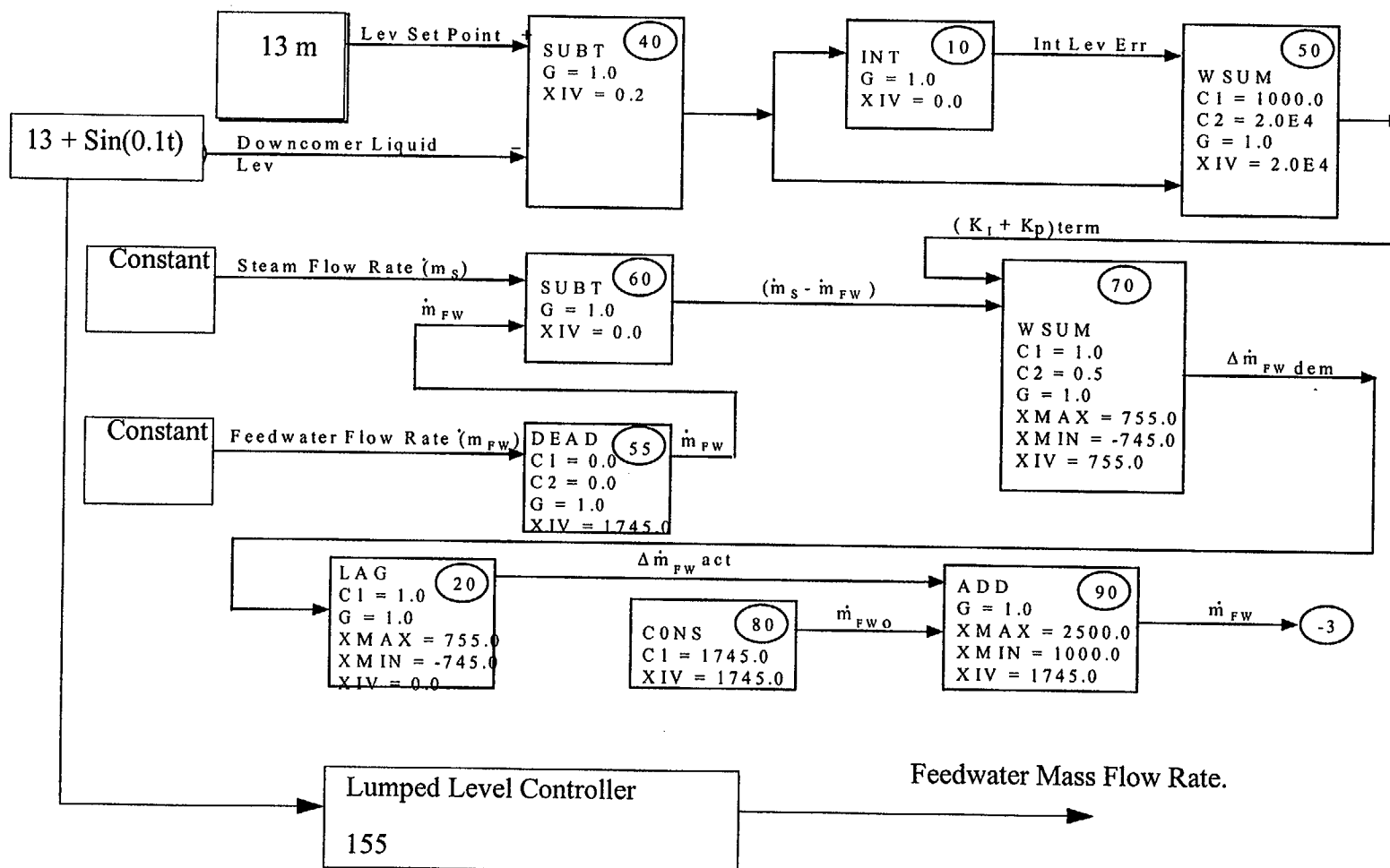


Figure 5.5.9 Lumped Level Controller Test Problem

Level Controller Test

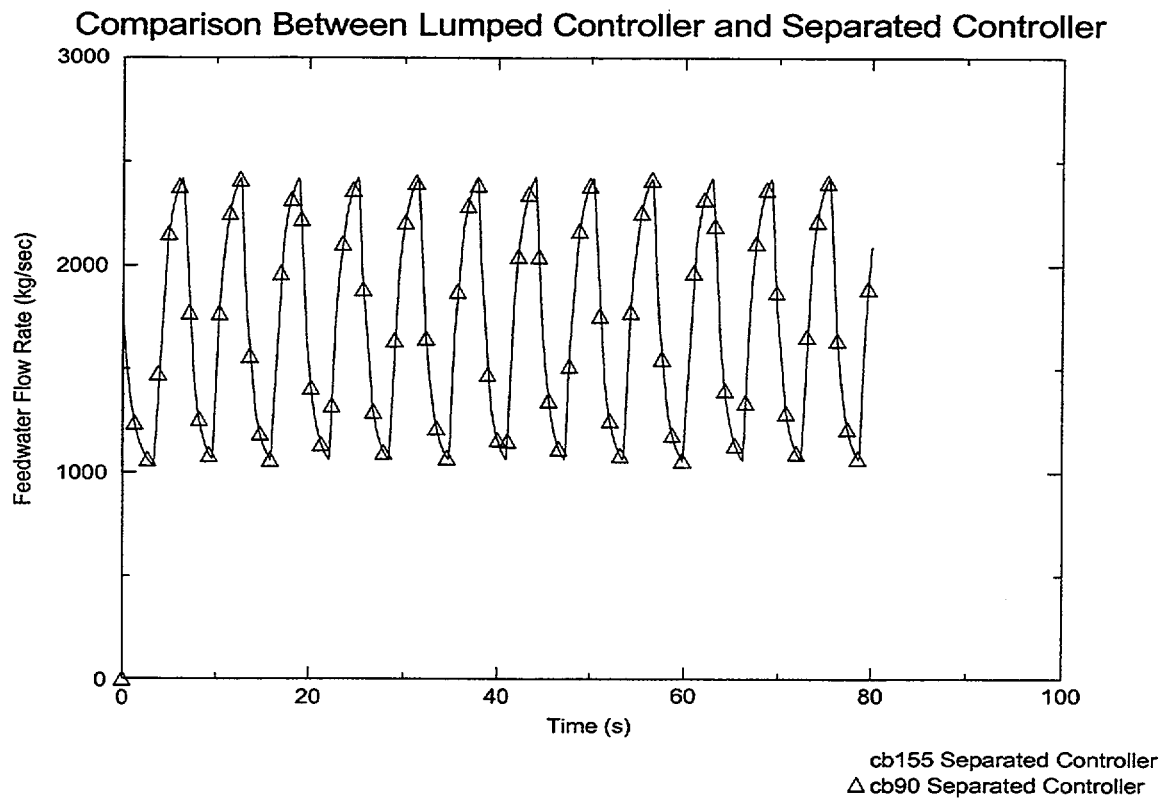


Figure 5.5.10 Lumped Level Controller Test Results

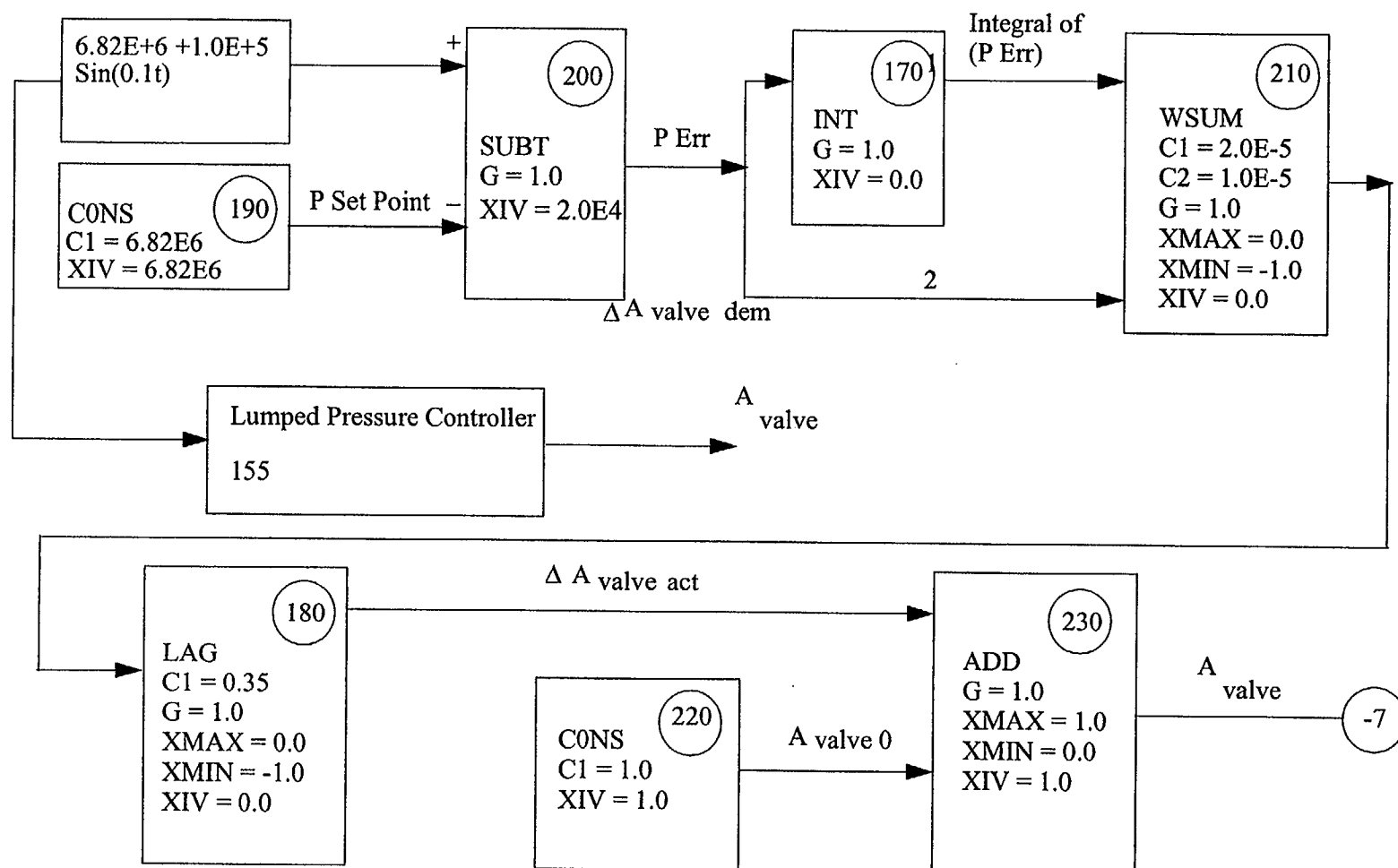


Figure 5.5.11 Lumped Pressure Controller Test Problem

Pressure Controller Test

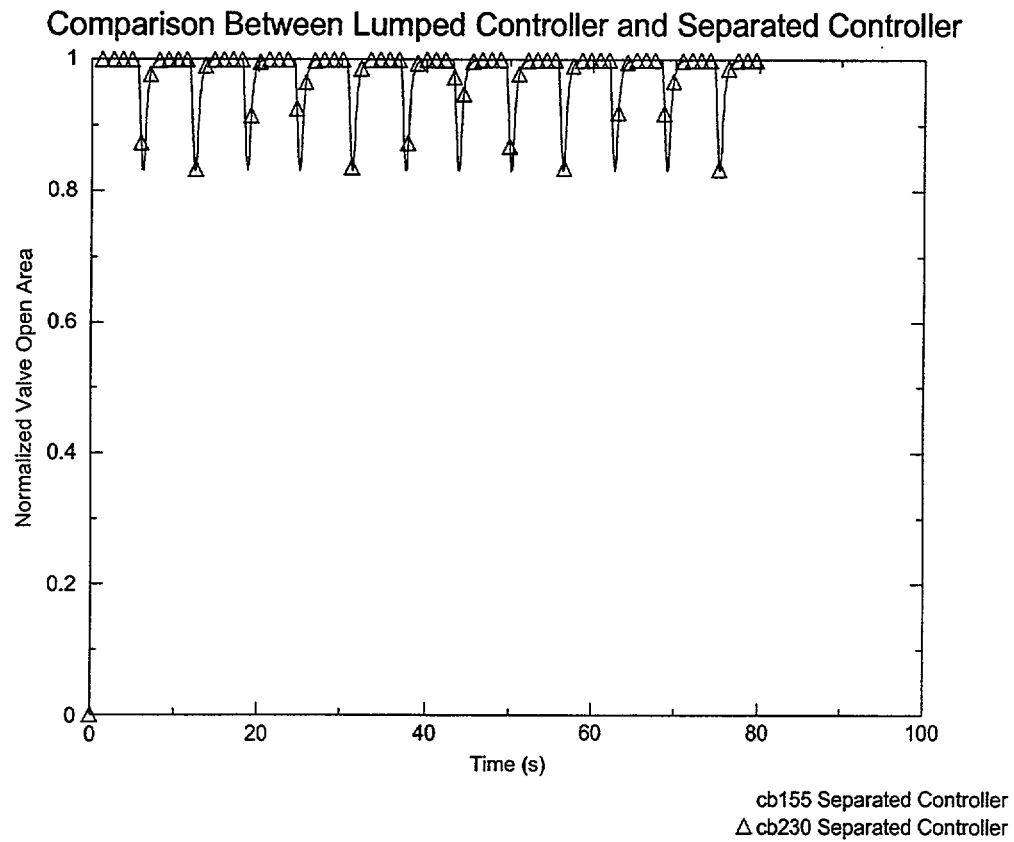


Figure 5.5.12 Lumped Pressure Controller Test Results

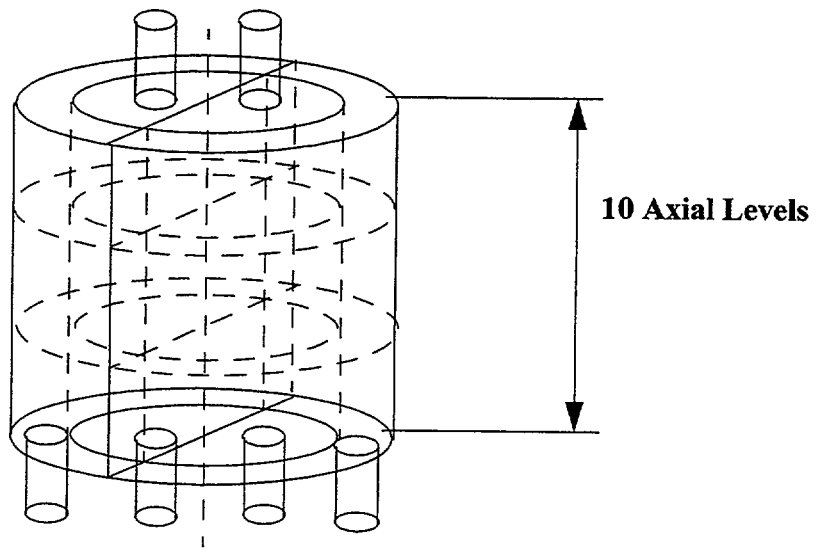


Figure 5.5.13 Mixture Enthalpy and Collapsed Water Level Test Problem

5.6 Single Junction (SJC) Component

The single junction component (SJC) allows certain one-dimensional (1-D) components to be specified on input as a single flow interface that acts as a junction between two fluid cells in two other TRAC components. The primary reason for adding this capability to TRAC-M is to simplify the translation of RELAP plant input models to TRAC-M input models. RELAP allows the definition of single flow interface components on input. Input models including this feature cannot be readily translated to TRAC-M input models. An added benefit is that the SJC capability allows more flexibility in defining networks of 1-D components.

1-D fluid components (PIPEs, PUMPs, etc.) are modeled in TRAC-M as 1-D arrays of mass and energy control volumes ("cells") that are connected with transport interfaces ("cell faces"), at which fluid velocities are defined. The cell faces at each end of a component connect to the cell faces at the ends of other 1-D components, or cell faces defined for the PLENUM, VESSEL, or boundary condition components. These connections between components are called "junctions" in TRAC-M. The cell face defined by a junction is shared by the two adjoining components, and must be defined identically in each component's input (i.e., 1-D component networking in TRAC-M is component-based, rather than junction-based). By defining a component consisting of only a single cell face, the SJC allows junction-based networking to be mimicked with TRAC-M input.

However, junction-based networking is only mimicked on input. The component-based paradigm is such an important feature of the TRAC-M software architecture that introducing a true SJC would significantly increase the complexity of the coding. Instead, the SJC functionality is implemented as an attribute of some already existing 1-D component types. The SJC does not actually exist *per se* at either the input level or the software implementation level, but its functionality is reproduced in the TRAC-M component-based coding.

The SJC can replace junctions between the ends of 1-D components, and can also provide extra side junctions to cells in 1-D components. This capability provides the increased flexibility in modeling networks of 1-D components (as mentioned above). Previously, the only way to specify side-junction connectivity was with the TEE component, which allows one component internal side junction per component, or with the PLENUM component, which allows any number of junctions to a single cell. The SJC permits any number of side junctions to any cell in some 1-D component types.

The 1-D component types that can be specified on input as having the SJC attribute will be called the single-junction attribute component set (SJACS). This set consists of PIPE, PUMP, and VALVE components. The 1-D component types that can be specified on input as having additional side junctions (provided by SJACS components) will be called the multiple-side junction attribute component set (MSJACS). This set also consists of PIPE, PUMP, and VALVE components. In other words, a PUMP component can be specified on input as consisting of a single-cell face, and another PUMP component can be specified as having an additional side junction to one of its cells. The same component cannot be specified as having both a single face and additional side junctions, because there would be no cell for which the additional side junctions can connect. In other words, the same component cannot be both an SJACS and an MSJACS component.

A representative component network is shown in Fig. 5.6.1. A PIPE component is specified to have the multiple-side junction attribute (and so is an MSJACS PIPE), and its side junction connects to another PIPE component that has the single-junction attribute (and so is an SJACS PIPE). This SJACS PIPE connects to a normal PIPE, which connects to a BREAK boundary condition component. The right end of the MSJACS PIPE connects to an SJACS VALVE, which also connects to a BREAK boundary condition component.

The SJC will be implemented such that the default mode will not invoke any SJC functionality (i.e., old input files will run as before). The inclusion of a single optional flag in the input file will allow reading of all new SJC-required input.

The SJC will make use of the existing fluids component networking solution, with as few modifications as possible. In other words, the SJC will appear to the network solver as a 1-D fluids component. It will also use the standard TRAC-M routines that operate on cell faces, with as few changes as possible.

5.6.1 Requirements

The following requirements are designed to ensure that the SJC is correctly implemented in TRAC-M. These requirements compare the results of calculations that use SJACS components with similar calculations that do not use them.

Requirement SJC1

A TRAC-M calculation in which an SJACS component is used as a junction between the ends of two 1-D components should produce results that are identical to those generated by a calculation in which the 1-D components are connected without the SJACS component.

Requirement SJC2

A TRAC-M calculation in which an SJACS component is used as a junction between the end of a 1-D component and a VESSEL cell should produce results that are identical to those generated by a calculation in which the components are connected without using the SJACS component.

Requirement SJC3

A TRAC-M calculation in which an SJACS component is used as an additional side junction, connecting an MSJACS component with the end of a 1-D component, should produce results that are identical to those generated by a calculation in which a TEE component is used without an SJACS component.

Requirement SJC4

A TRAC-M calculation in which an SJACS component is used as an additional side junction, connecting an MSJACS component with a VESSEL cell, should produce results that are similar to those generated by a calculation in which a TEE component is used where no momentum is transported from the TEE main leg to the side leg. The results should be similar, but not identical, because the single cell used in the TEE side leg is not included in the SJACS component.

Requirement SJC5

A TRAC-M calculation in which an SJACS VALVE component is used should produce results that are similar to those generated by a calculation in which a non-SJACS VALVE component is used. The underlying finite-differencing model produced by the differing component input models are identical, but the order of operations in the setup of the network equations may be different. This difference in the order of operations means that the results may be only similar, not identical.

Requirement SJC6

A TRAC-M calculation in which an SJACS PUMP component is used should produce results that are similar to those generated by a calculation in which a non-SJACS PUMP component is used. The underlying finite-differencing models produced by the differing component input models are identical, but the order of operations in the setup of the network equations may be different. This difference in the order of operations means that the results may be only similar, not identical.

Requirement SJC7

A TRAC-M calculation in which SJACS PIPE components are used to create additional side junctions to a single-cell MSJACS component should produce results that are similar to those generated by a calculation using a standard PLENUM component with the same flow network. The results should be similar, but not identical, because the PLENUM component uses different routines to set up and solve the finite-difference equations.

5.6.2 Verification Testing and Assessment

This section presents the results of the verification testing and assessment of the integration of the SJC component. The results show that the integration of the SJC component has been performed correctly.

5.6.2.1 End-Junction Replacement Tests and Their Results

Three tests demonstrate that Requirement SJC1 is met by comparing the results of a simple input model that uses no SJC components with an equivalent input model that does. The non-SJC input file models two five-cell PIPE components that are connected end to end, with a FILL component as the lower boundary condition and a BREAK component as the upper boundary condition as shown in Figs. 5.6.2, 5.6.3 and 5.6.4. The PIPE components are initially filled with steam moving at 1 m/s, and the FILL provides steam continuously at 1 m/s. The SJC input file models an SJACS component inserted between the two PIPE components. The results of the non-SJC and SJC input files should be identical, as measured by the modified trcdif file.

5.6.2.1.1 Input Files tinm1p1-b and tinm1p1-s (Base Case)

The base case is represented by the tinm1p1-b and tinm1p1-s input files. A two-part nodding diagram (base-case model with no SJC and the SJC model) is shown in Fig. 5.6.2. The results from the two calculations are identical.

5.6.2.1.2 Input Files tinm1p2-b and tinm1p2-s (Reverse SJC Orientation)

Reversal of the orientation of the SJACS component should have no effect on the results. A two-part nodding diagram (base-case model with no SJC and the SJC model) is shown in Fig. 5.6.3. The results from the two calculations are identical.

5.6.2.1.3 Input Files tinm1p3-b and tinm1p3-s (Non-Zero Grav Input)

The modified trcdif file should still be identical with non-zero grav input (the direction cosine with respect to the vertical). A two-part nodding diagram (base-case model with no SJC and the SJC model) is shown in Fig. 5.6.4. The results from the two calculations are identical.

5.6.2.2 Composite Tee Tests and Their Results

Seven tests demonstrate that Requirement SJC3 is met by comparing the results of a simple input model that uses no SJC components with an equivalent input model that does. The non-SJC input contains a standard TEE component with a right-angle side leg, whereas the SJC input contains a composite TEE made up of two standard PIPE components and an SJACS PIPE component that links one end of one PIPE with a side junction in the other PIPE (defined on input as an MSJACS component) as shown in Figs. 5.6.5 through 5.6.10. The TEE component having a right angle side leg, combined with the flow direction in the various parametric cases, ensures that there is no momentum transported between the main leg and the side leg. This allows direct comparison of results with the SJC case, in which no momentum is transported by definition and, thus, the results should be identical.

5.6.2.2.1 Input Files tinm2p1-b and tinm2p1-s (Base Case)

Fill boundary conditions add vapor at 1 m/s from the ends of the main leg, whereas a BREAK boundary condition is used at the end of the side leg. A two-part nodding diagram (base-case model with no SJC and the SJC model) is shown in Fig. 5.6.5. The results from the two calculations are identical.

5.6.2.2.2 Input Files tinm2p2-b and tinm2p2-s (Reverse FILL Flow)

This parametric reverses the direction of the flow at the FILL components, so that vapor is continually removed at 1 m/s. A two-part nodding diagram (base-case model with no SJC and the SJC model) is shown in Fig. 5.6.6. The results from the two calculations are identical.

5.6.2.2.3 Input Files tinm2p3-b and tinm2p3-s (Reverse SJACS-Component Orientation)

This parametric reverses the orientation of the SJACS component that links the side junction in the MSJACS PIPE component to one end of the side-leg PIPE. Such reversal of orientation should cause no change in the computed results. A two-part nodding diagram (base-case model with no SJC and the SJC model) is shown in Fig. 5.6.7. The results from the two calculations are identical.

5.6.2.2.4 Input Files tinm2p4-b and tinm2p4-s (Non-Zero Grav Terms)

In this parametric case, the grav terms (direction cosines with the vertical) at the SJACS interface are non-zero. Both SJC and non-SJC input files should produce the same gravitational body force terms, and these terms should have identical results on the flow. A two-part nodding diagram (base-case model with no SJC and the SJC model) is shown in Fig. 5.6.8. The results from the two calculations are identical.

5.6.2.2.5 Input Files tinm2p5-b and tinm2p5-s (Non-Zero Elevation Terms)

In this parametric case, the elev terms (reference elevations) at the SJACS interface are nonzero. Both SJC and non-SJC input files should produce the same gravitational body force terms, and these terms should have identical results on the flow. A two-part nodding diagram (base-case model with no SJC and the SJC model) is shown in Fig. 5.6.9. The results from the two calculations are identical.

5.6.2.2.6 Input Files tinm2p6-b and tinm2p6-s (Restart)

This parametric case tests the restart of the base case. The results from the two calculations, with and without restart, are identical.

5.6.2.2.7 Input Files tinm2p7-b and tinm2p7-s (Single-Cell Main Leg)

This case is a composite TEE with a single-cell main leg. A two-part nodding diagram (base-case model with no SJC and the SJC model) is shown in Fig. 5.6.10. The results from the two calculations are identical.

5.6.2.3 Link to VESSEL, Static Check (Input Files tinm3p1-b and tinm3p1-s)

The VESSEL component is extracted from the w4loopn input file, and connected to a PIPE and BREAK component as boundary conditions for the non-SJC base case. An SJACS PIPE component is inserted between the VESSEL and the PIPE as a first test of the networking connectivity of the SJACS component to the VESSEL. There is nothing to drive the flow in this model; it is designed as a static check. Therefore, the flow is expected to decrease rapidly to insignificance after an initial perturbation. This test demonstrates the static component of Requirement SJC2. A two-part nodding diagram (base-case model with no SJC and the SJC model) is shown in Fig. 5.6.11. The results from the two calculations are almost identical, and the differences are very small.

5.6.2.4 Link to VESSEL, Simulated BWR CHAN Component

The w4loopn VESSEL used above is used again in an initial test of a composite BWR CHAN component. For the base case, a three-cell main leg of a TEE component is used to connect axial level 2 to axial level 6, with a single-cell side leg used to model the leakage path from cell 2 of the TEE main leg to axial level 4 of the VESSEL as shown in Fig. 5.6.12. For the SJC case, a three-cell PIPE component is used to connect axial level 2 to axial level 6, with an SJACS PIPE component used to model the leakage path from cell 2 of the PIPE component to axial level 4 of

the VESSEL. These two models produce very similar, but not identical, results because the single cell of the TEE side leg is not modeled in the SJC case. In both cases, simple boundary conditions are applied to the VESSEL using a FILL and a PIPE for cold leg inflow and a BREAK and PIPE for hot leg outflow. These tests (described below) demonstrate that Requirement SJC4 is met, as is the dynamic component of Requirement SJC2.

5.6.2.4.1 Input Files tinm4p1-b and tinm4p1-s (Static Check)

The hot leg fill is set to zero, and both cases are checked for no flow loops in the VESSEL. In both cases, the velocities should approach zero after initial perturbations caused by an inaccurate initial static head input. A two-part nodding diagram (base-case model with no SJC and the SJC model) is shown in Fig. 5.6.12. The differences between the results of the two calculations are very small.

5.6.2.4.2 Input Files tinm4p2-b and tinm4p2-s (Leakage Path)

Flow is injected at the hot leg, and the leakage flow is compared for the SJC and non-SJC models. In the SJC model, the flow is taken at the SJACS cell face, whereas in the non-SJC model, the flow is taken at the edge of the single-cell side leg. The flows are expected to be similar, but not identical, because of the unavoidable difference in model geometry.

A two-part nodding diagram (base-case model with no SJC and the SJC model) is shown in Fig. 5.6.13. The differences between the results of the two calculations are very small. The relative difference in leakage flows is illustrated in Fig. 5.6.14. This test shows that Requirement SJC4 is met.

5.6.2.5 SJACS Pump Component (Input Files tinm5p1-b and tinm5p1-s)

The non-SJC model has a five-cell PIPE and a five-cell PUMP placed end to end, with BREAK components as boundary conditions as shown in Fig. 5.6.15. The PUMP momentum source is at the second cell face by definition, and is set to produce a flow from an initial condition of stagnant vapor. The SJC model has a six-cell PIPE, an SJACS PUMP, and a four-cell PIPE placed end to end, which creates an identical finite-difference scheme. The results should be similar, but not identical, because of possible round-off error in the setup of the network equations, which would occur as a result of the different nodding of the input components.

A two-part nodding diagram (base-case model with no SJC and the SJC model) is shown in Fig. 5.6.15. The differences between the results of the two calculations are very small. The relative difference in pump flows is illustrated in Fig. 5.6.16. This test demonstrates that Requirement SJC6 is met.

5.6.2.6 SJACS VALVE Component

The base input model is similar to that used for the PUMP test. A five-cell PIPE and a five-cell VALVE are placed end to end, with BREAK components as boundary conditions as shown in Fig. 5.6.17. The VALVE variable flow area is set to the second cell face using input variable IVPS. The BREAKs provide a pressure difference that drives the flow from an initial stagnant condition, and the valve flow area is ramped from closed to full open during the first second. The

SJC model has a six-cell PIPE, an SJACS VALVE, and a four-cell PIPE placed end to end, which creates an identical finite-difference scheme. The results should be similar, but not identical, because of the possible round-off error in the setup of the network equations, which is caused by the different nodding of the input components. The tests described below demonstrate that Requirement SJC5 is met.

5.6.2.6.1 Input Files `tinm6p1-b` and `tinm6p1-s` (Base Case)

This is the base case for the SJACS VALVE component. A two-part nodding diagram (base-case model with no SJC and the SJC model) is shown in Fig. 5.6.17. The differences between the results of the two calculations are very small. The relative difference in valve flows is illustrated in Fig. 5.6.18.

5.6.2.6.2 Input Files `tinm6p2-b` and `tinm6p2-s` (VALVE Adjacent to BREAK)

In this parametric case, the VALVE is placed directly adjacent to the right-side BREAK, and the VALVE flow area is ramped from full open to closed between calculation times of 1 and 2 seconds. A two-part nodding diagram (base-case model with no SJC and the SJC model) is shown in Fig. 5.6.19. The differences between the results of the two calculations are very small. The relative difference in valve flows is illustrated in Fig. 5.6.20.

5.6.2.7 Test `tfpipe2` Modification

The results of the standard `tfpipe2` heated-wall blowdown test problem, are compared with a model in which an SJACS PIPE component is inserted to the left of the BREAK, replacing the cell face that had been modeled by the adjacent PIPE as shown in Fig. 5.6.21. `Tfpipe2` is a simple test that exercises two-phase flow and choked flow capabilities. (The previous problems all used initial and boundary conditions of single-phase vapor.) This test demonstrates that Requirement SJC1 is met under two-phase, choked-flow conditions. The outflow from the system is expected to be identical for the SJC and non-SJC cases.

5.6.2.7.1 Input files `tinm7p1-b` and `tinm7p1-s` (Base Case)

These files comprise the base case for the modification of Test `tfpipe2`. A two-part nodding diagram (base-case model with no SJC and the SJC model) is shown in Fig. 5.6.21. The results of the two calculations are identical.

5.6.2.7.2 Input files `tinm7p2-b` and `tinm7p2-s` (Reversed SJACS Orientation)

The orientation of the SJACS component is reversed (i.e., junctions in the input deck are reversed). A two-part nodding diagram (base-case model with no SJC and the SJC model) is shown in Fig. 5.6.22. To get NULL results, a line of coding was activated to `Gen1dTask::poster` that correctly sets the `negVapVel` bit for the SJACS component. This coding was placed in the source to execute this test, but it is normally commented out. It can be found by searching for pattern "`tinm7p2`" in file `Gen1dTaskM.f90`. With this change, the modified `trcdif` files for this case are identical. This test demonstrates that Requirement SJC1 is met.

5.6.2.7.3 Composite PLENUM (Input Files tinm8p1-b and tinm8p1-s)

The non-SJC model for this case has a PLENUM component connected to two PIPE components that have BREAK components on their other ends. The system is initially filled with vapor at 1 bar, except for the PLENUM, which is initialized at 2 bar. This creates a dual, symmetric, depressurizing shock tube. In the SJC model, a single-cell MSJACS PIPE with zero-velocity FILL components on each end replaces the PLENUM, and two SJACS PIPEs are connected as side-leg junctions. Identical PIPE and BREAK components are used to construct a system model that is functionally identical to the non-SJC model. Flows in the systems are expected to be similar, but not identical.

A two-part nodding diagram (base-case model with no SJC and the SJC model) is shown in Fig. 5.6.23. For the numerics to be as close as possible, some PLENUM-specific functionality was commented out in Gen1dCrunch::StbVel1D and Gen1dCrunch::tflds1. The locations of these six changes can be found by doing a pattern search for "tinm8p1" on file Gen1DCrunchM.f90. With these changes, the results of the two models are very similar, as shown in Fig. 5.6.24. This test demonstrates that Requirement SJC7 is met.

5.6.3 Conclusions

The results of these tests demonstrate that the SJC component is correctly constructed in TRAC-M(F90).

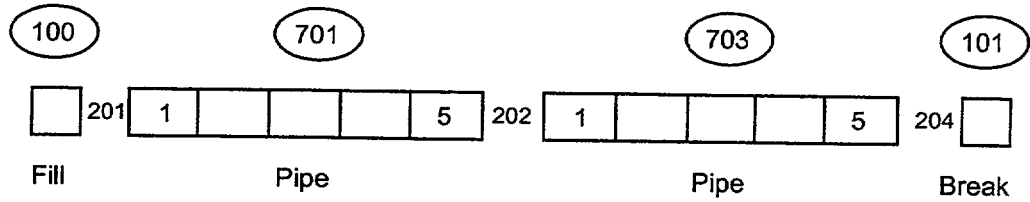


Diagram illustrating a pipeline with a break:

- Fill** (Stage 100): A single box labeled 201.
- Pipe** (Stage 701): A 5-stage pipeline with values 1, an empty box, an empty box, an empty box, and 5. Labeled 202.
- Pipe** (Stage 703): A 5-stage pipeline with values 1, an empty box, an empty box, an empty box, and 5. Labeled 204.
- Break** (Stage 101): A single box.

Page 5-159

(a) Base Case (No SJC)



(b) SJC Case

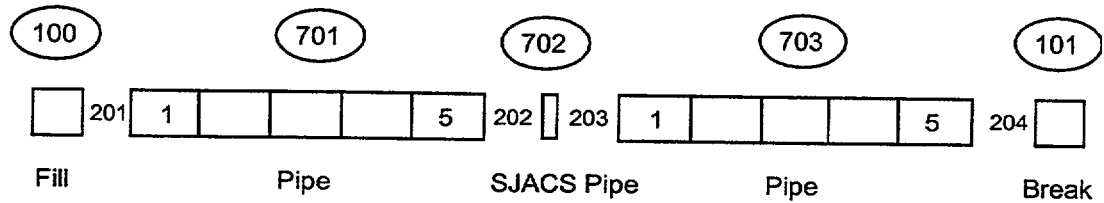
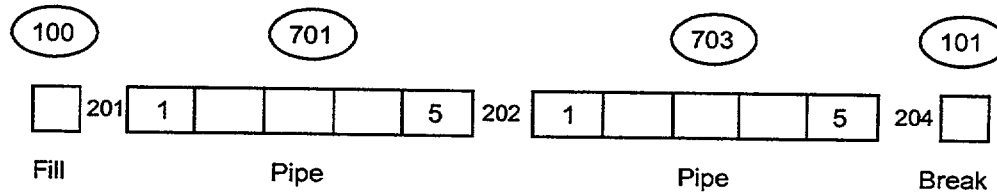


Figure 5.6.3 Noding Diagram for End-Junction Replacement Test Cases - Parametric 2 (Reverse Orientation)

(a) Base Case (No SJC)



(b) SJC Case

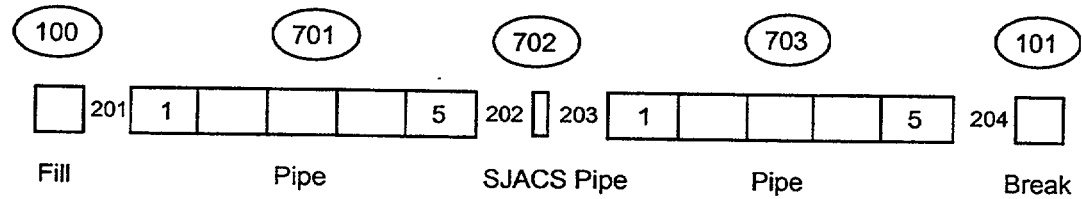
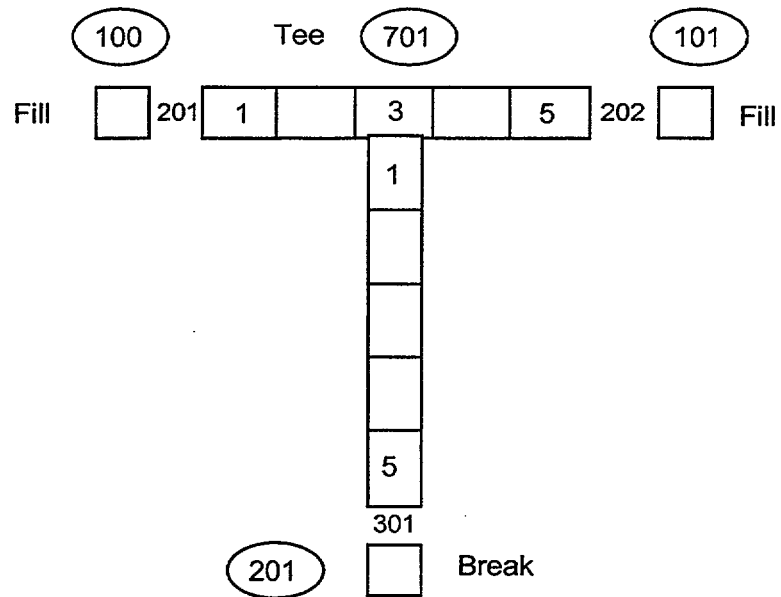


Figure 5.6.4 Noding Diagram for End-Junction Replacement Test Cases - Parametric 3 (Non-Zero Gravity)

(a) Base Case (No SJC)



(b) SJC Case

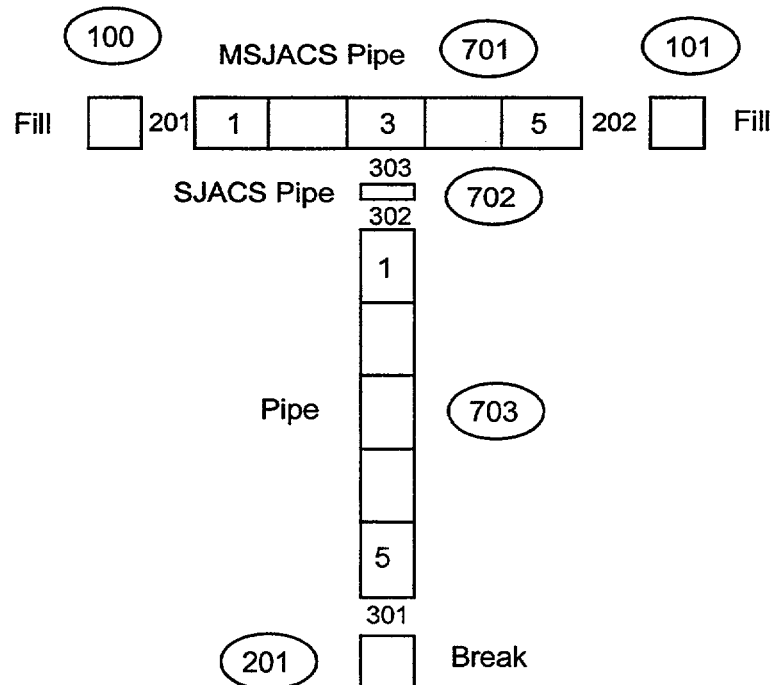
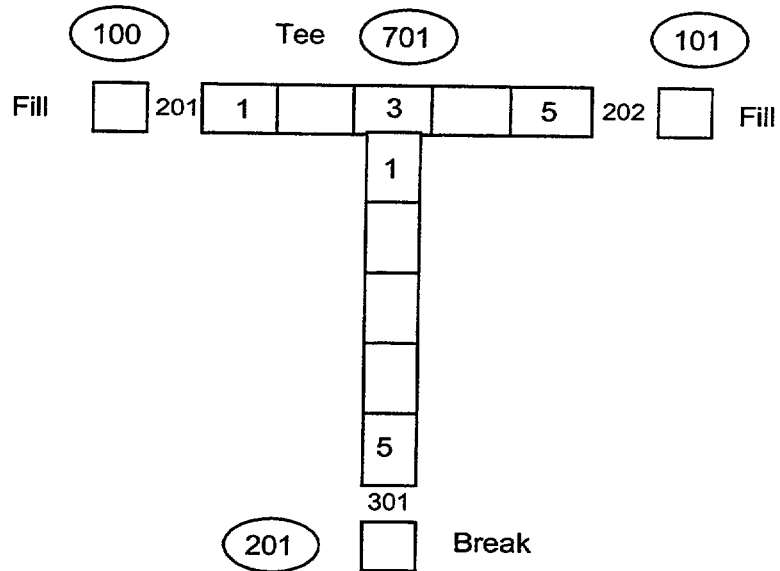


Figure 5.6.5 Noding Diagrams for Composite TEE Test Cases - Parametric 1 (Base Case)

(a) Base Case (No SJC)



(b) SJC Case

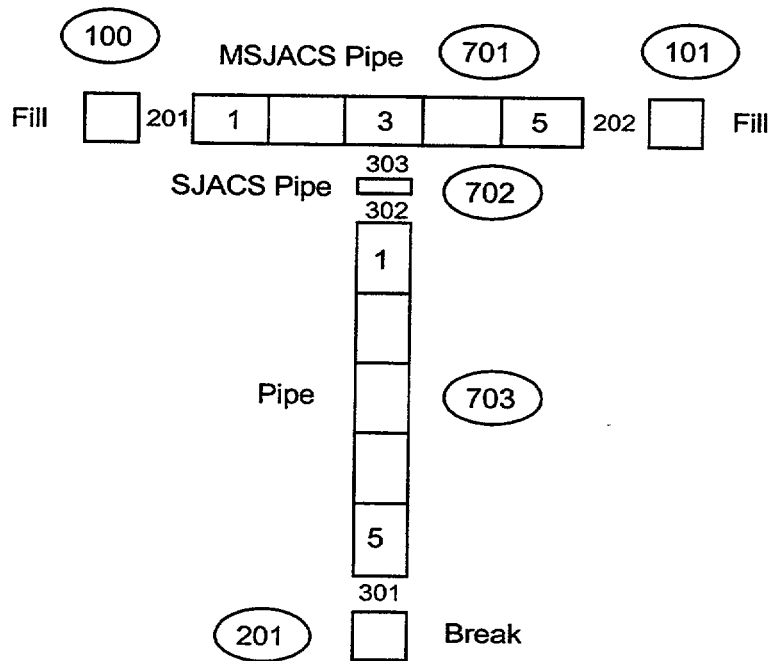
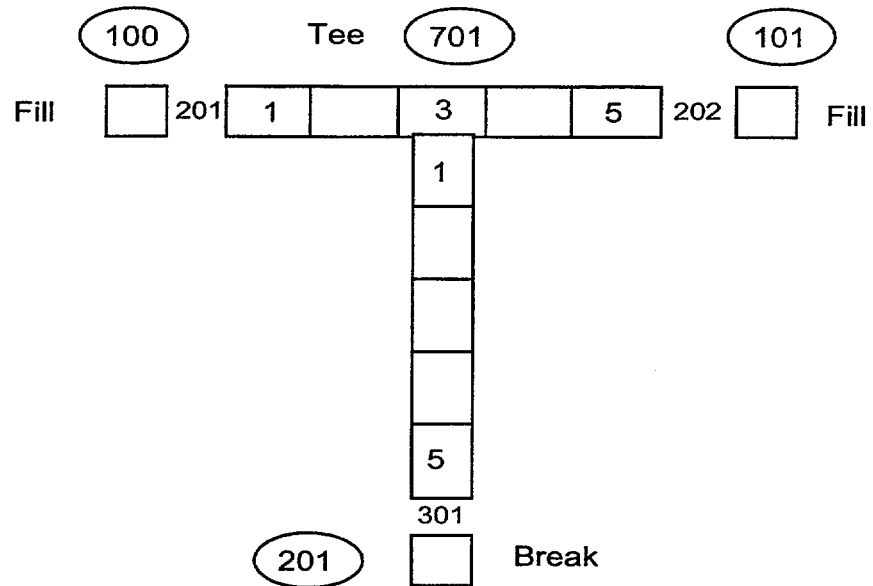
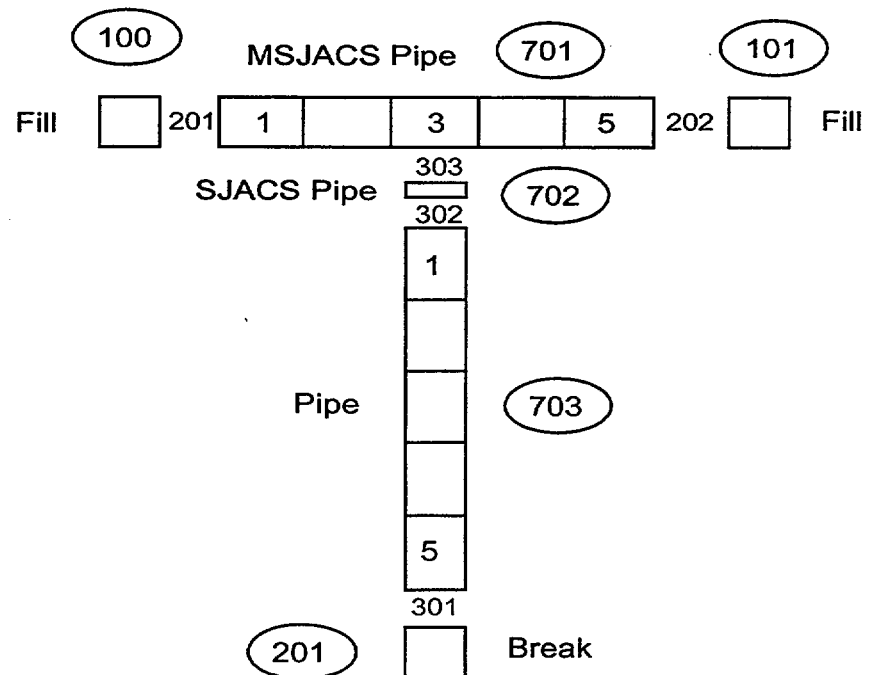


Figure 5.6.6 Noding Diagrams for Composite TEE Test Cases - Parametric 2 (In-Flow at FILLS)

(a) Base Case (No SJC)

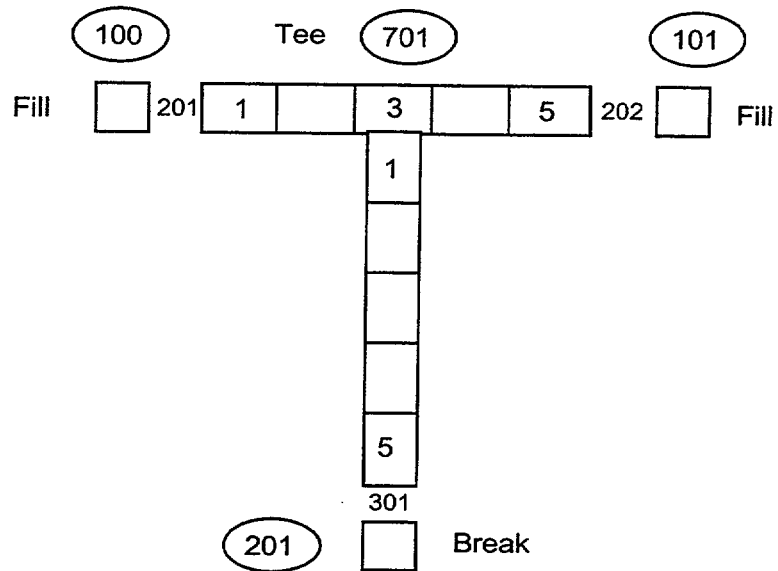


(b) SJC Case



**Figure 5.6.7 Noding Diagrams for Composite TEE Test Cases - Parametric 3
(Reversed SJC Orientation)**

(a) Base Case (No SJC)



(b) SJC Case

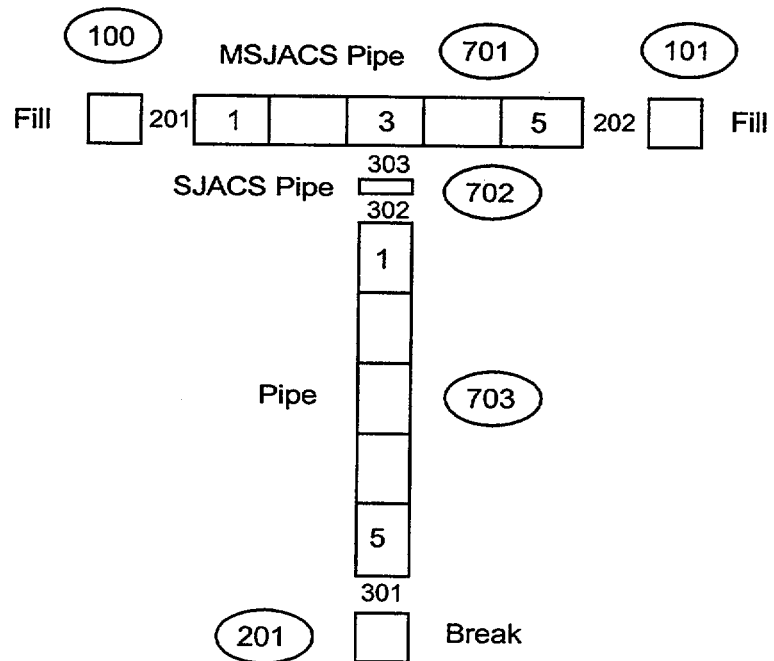
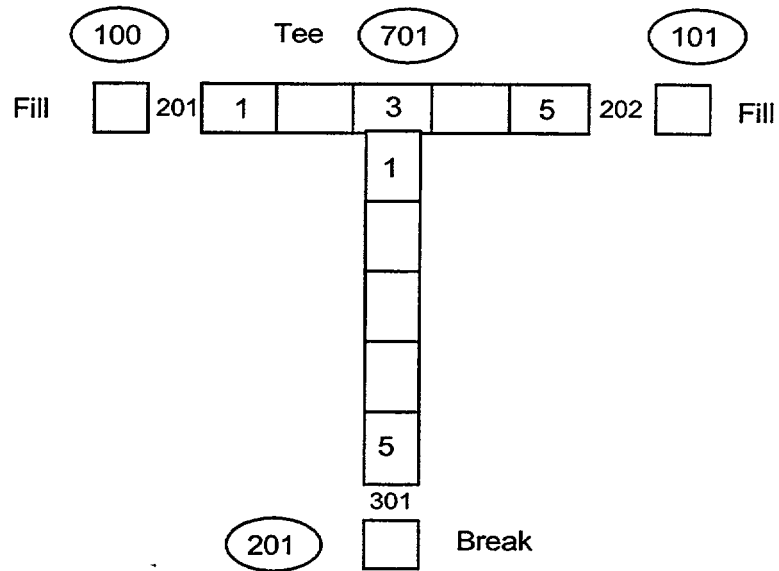


Figure 5.6.8 Noding Diagrams for Composite TEE Test Cases - Parametric 4 (Non-Zero Gravity Terms)

(a) Base Case (No SJC)



(b) SJC Case

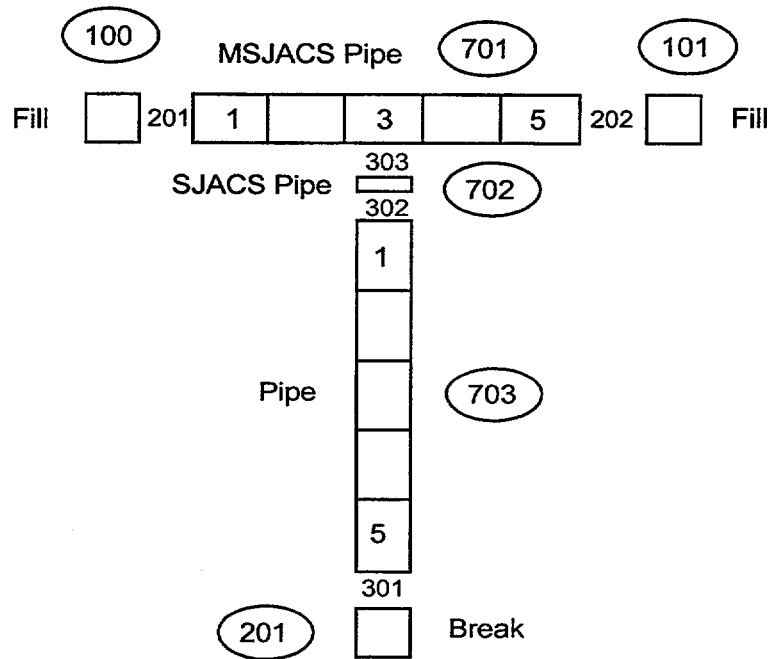
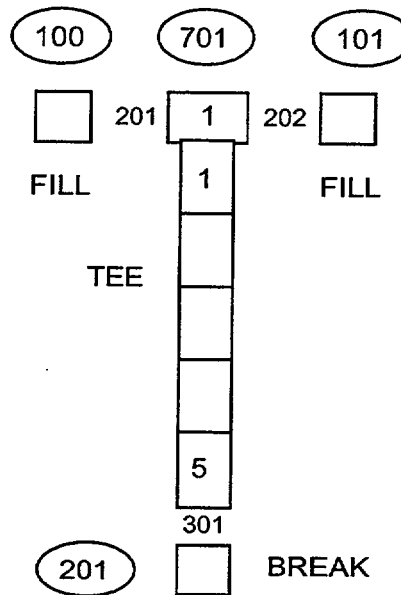


Figure 5.6.9 Noding Diagrams for Composite TEE Test Cases - Parametric 5 (Non-Zero Elevation Terms)

(a) Base Case (No SJC)



(b) SJC Case

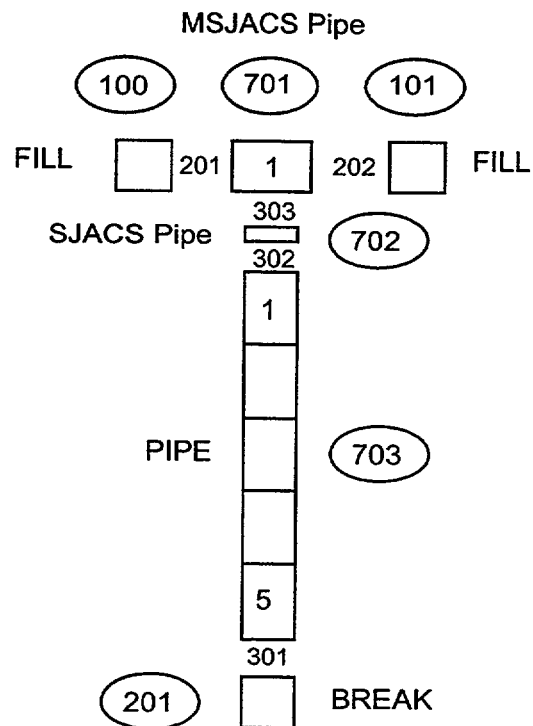
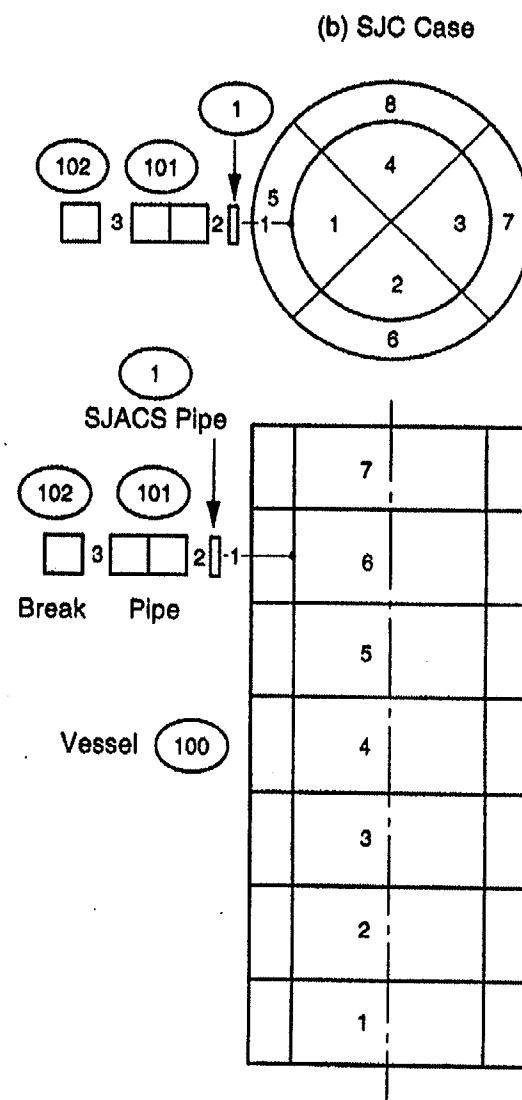
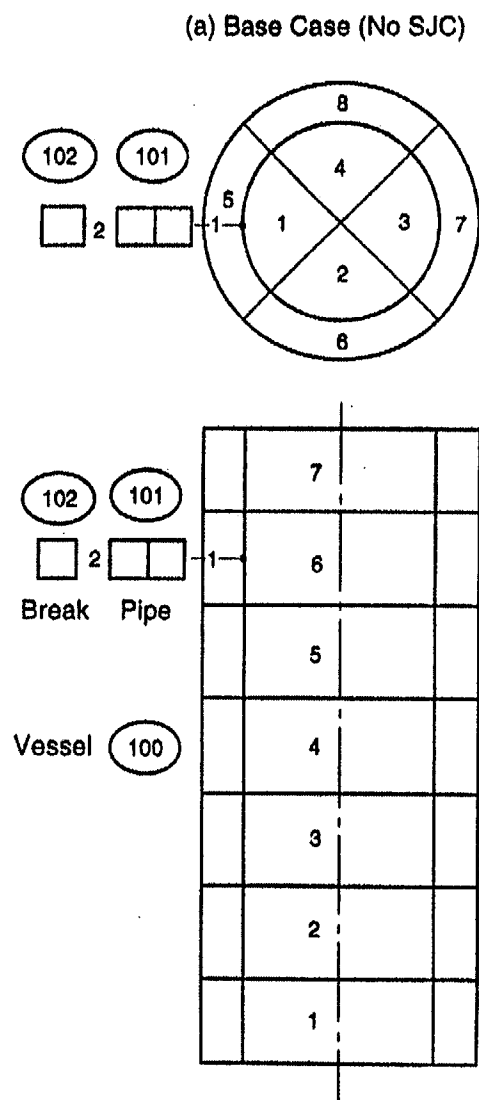


Figure 5.6.10 Noding Diagrams for Composite TEE Test Cases with Single-Cell Main Leg - Parametric 7

Figure 5.6.11 Noding Diagrams for Link-to-VESSEL Test Cases - Parametric 1 (Base Case)



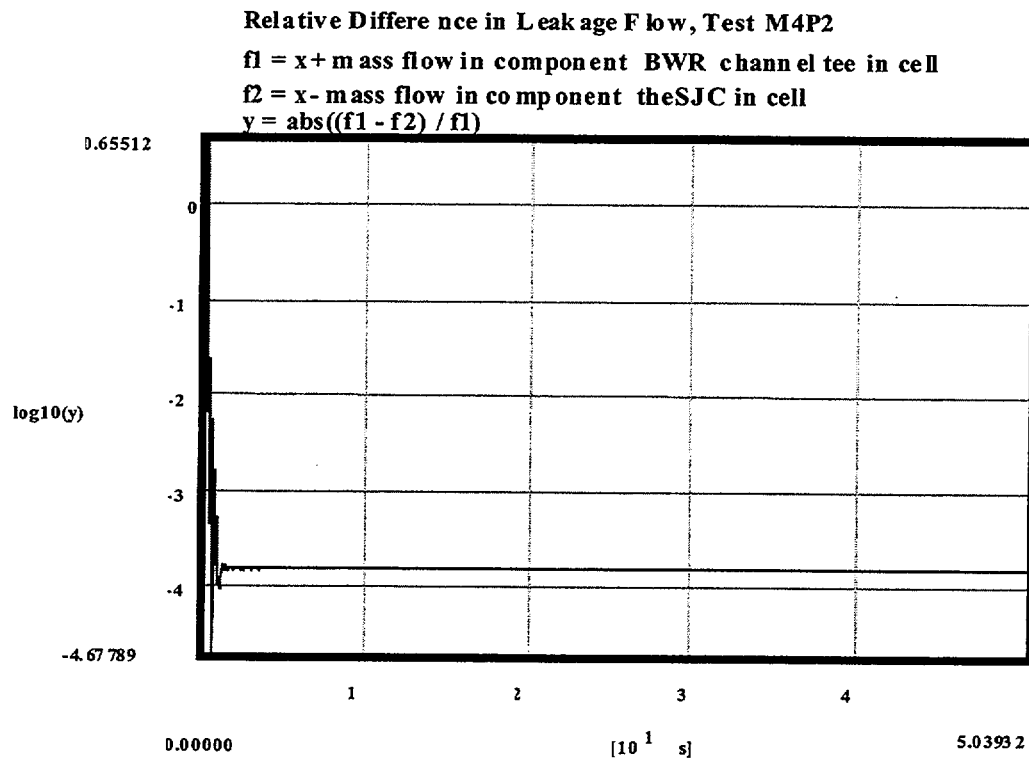
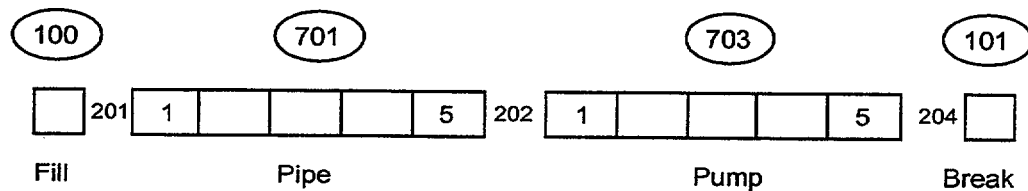


Figure 5.6.14 Relative Difference in Leakage Flows for the SJC and Non-SJC Models

(a) Base Case (No SJC)



(b) SJC Case

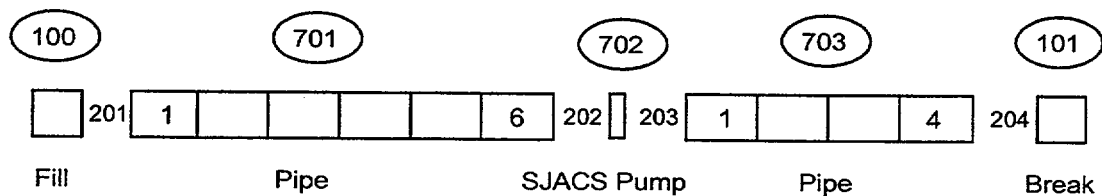


Figure 5.6.15 Noding Diagrams for Pump Test Cases - Parametric 1 (Base Case)

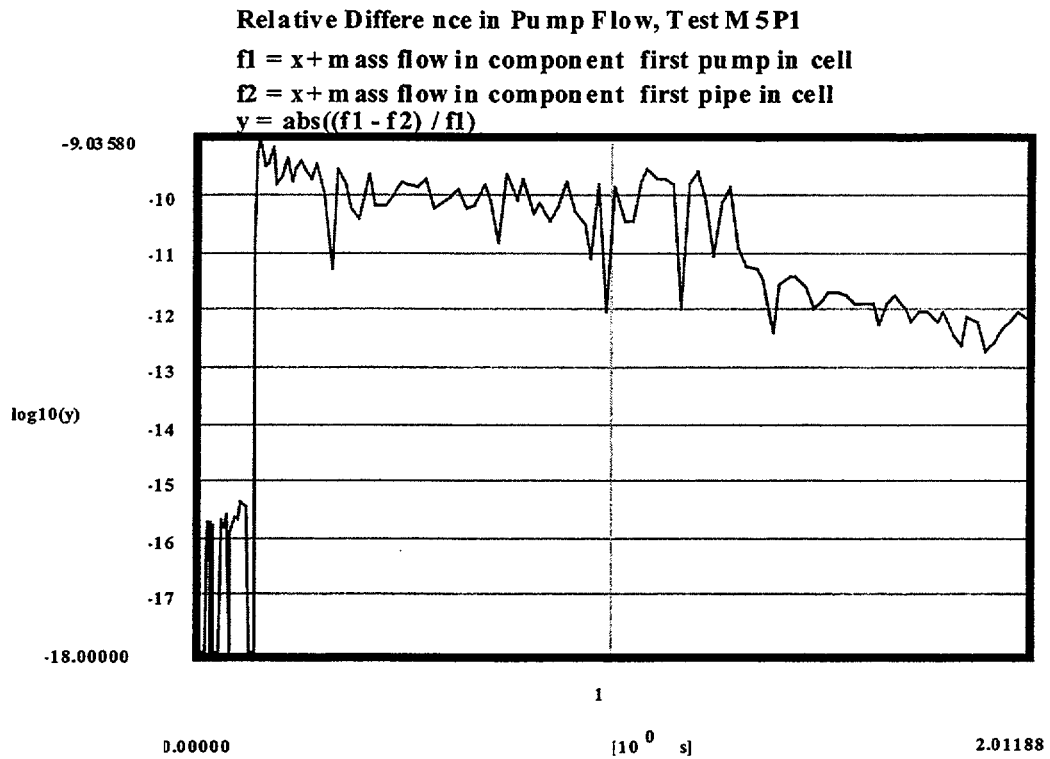
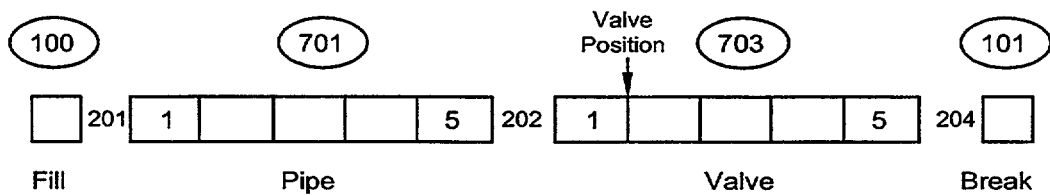


Figure 5.6.16 Relative Difference in Pump Flows for the SJC and Non-SJC Models

(a) Base Case (No SJC)



(b) SJC Case

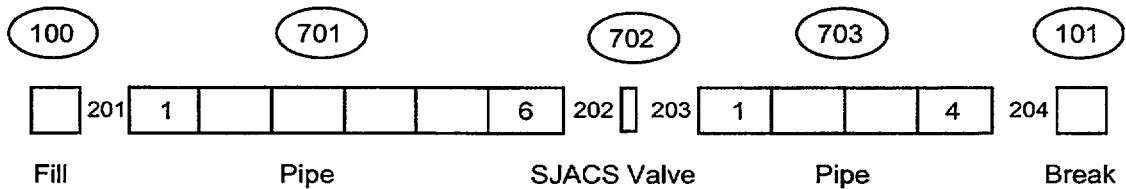


Figure 5.6.17 Noding Diagrams for VALVE Test Cases

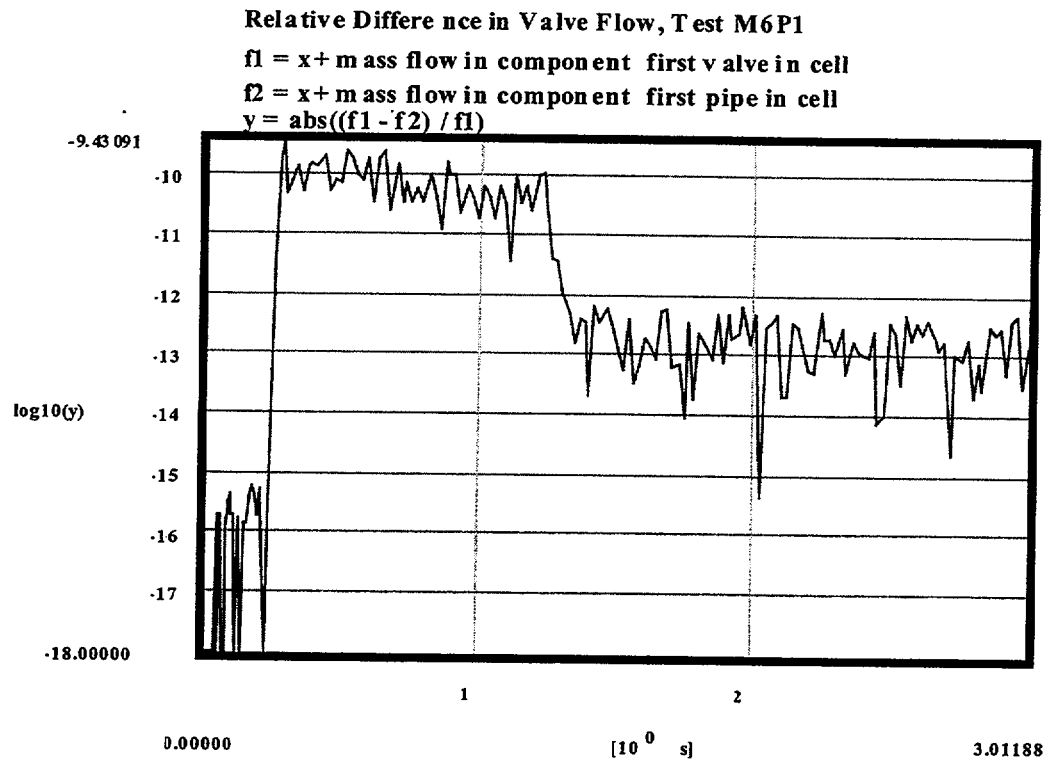
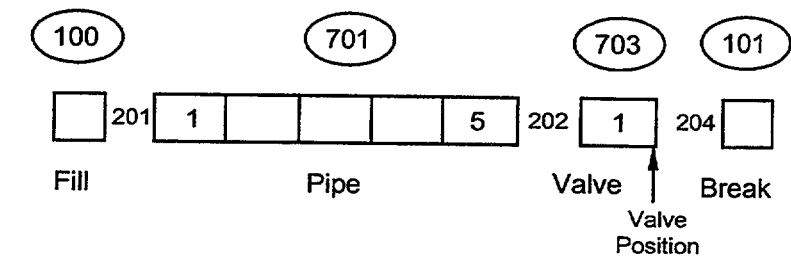


Figure 5.6.18 Relative Difference in Valve Flows for the SJC and Non-SJC Models
(a) Base Case (No SJC)



(b) SJC Case

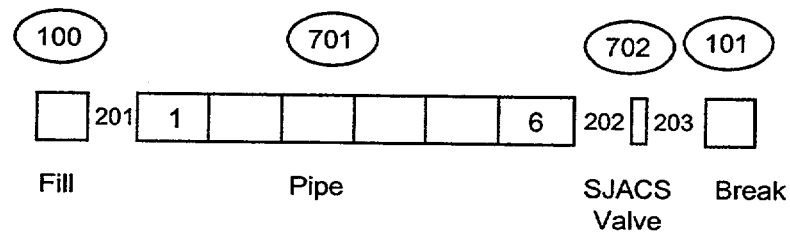


Figure 5.6.19 Noding Diagrams for VALVE Test Cases Next to BREAK - Parametric 2
(VALVE next to BREAK)

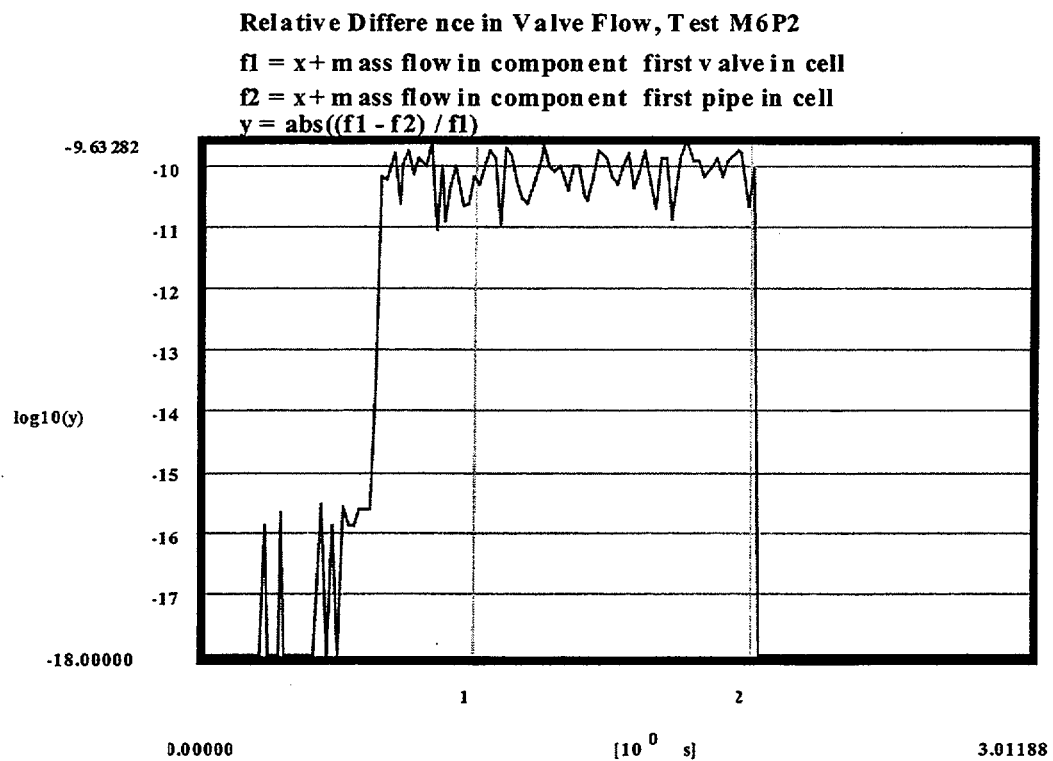


Figure 5.6.20 Relative Difference in Valve Flows for the SJC and Non-SJC Models

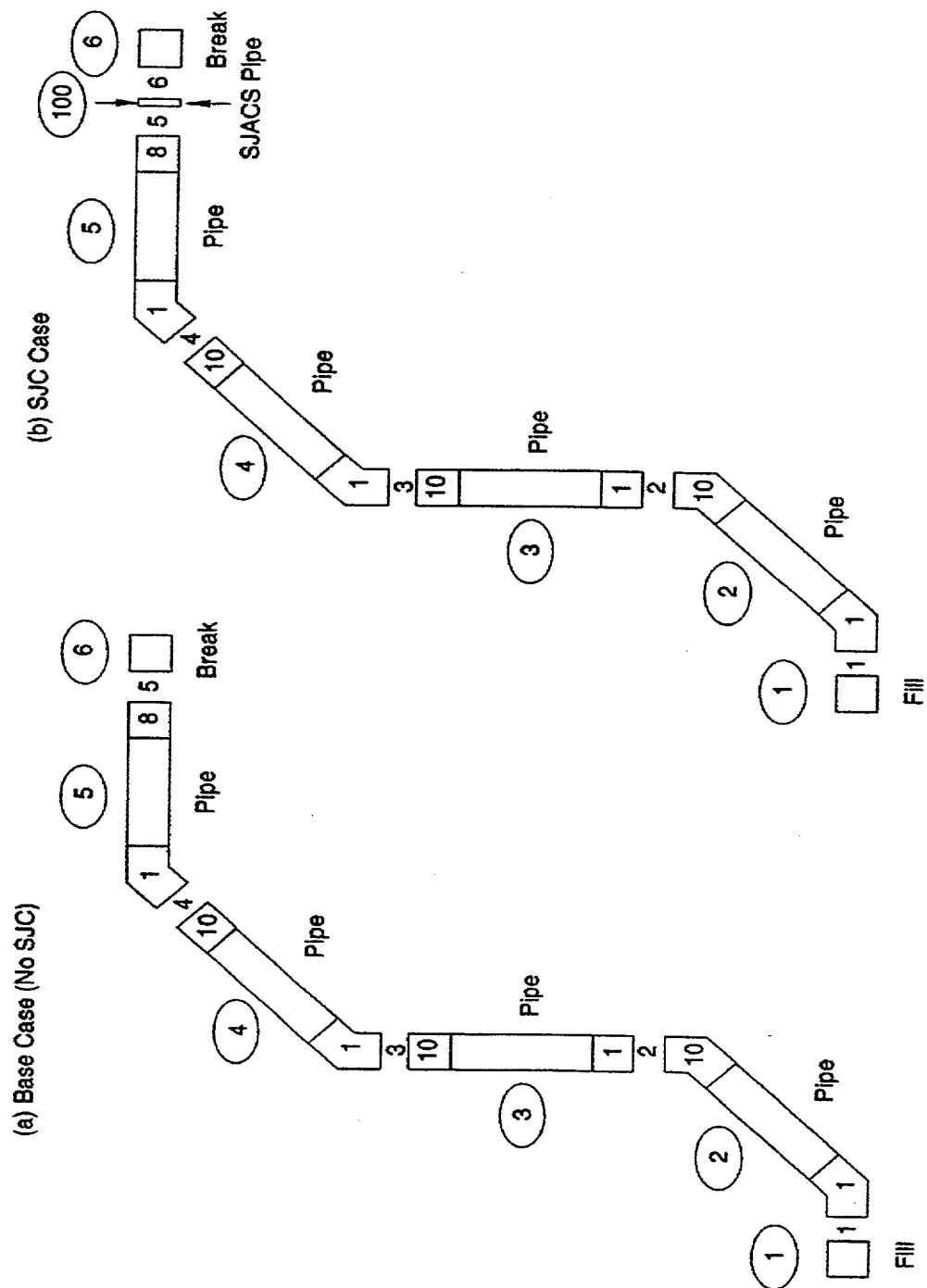


Figure 5.6.21 Noding Diagrams for TFPIPE2 Test Cases - Parametric 1 (Base Case)

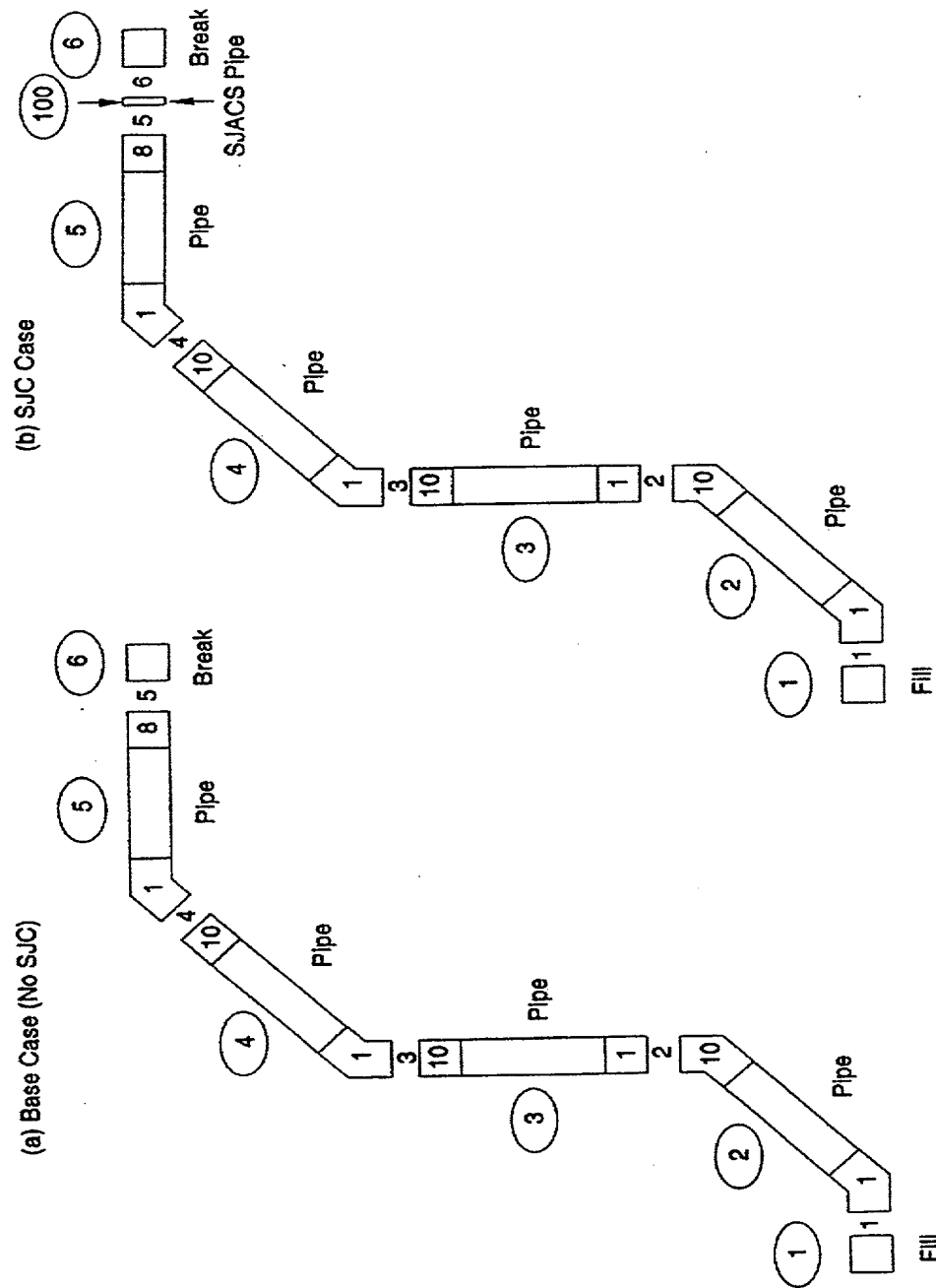
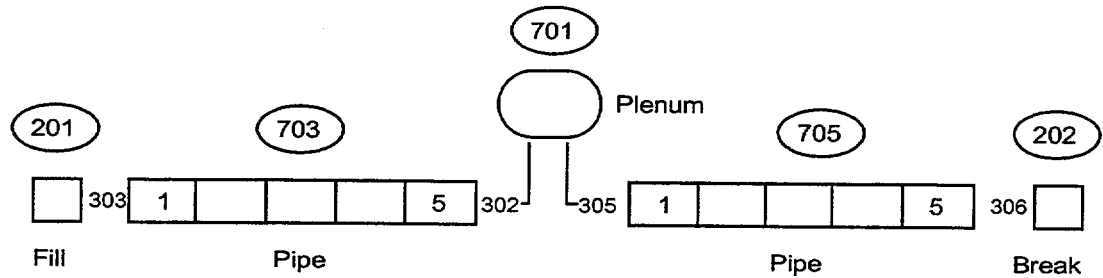
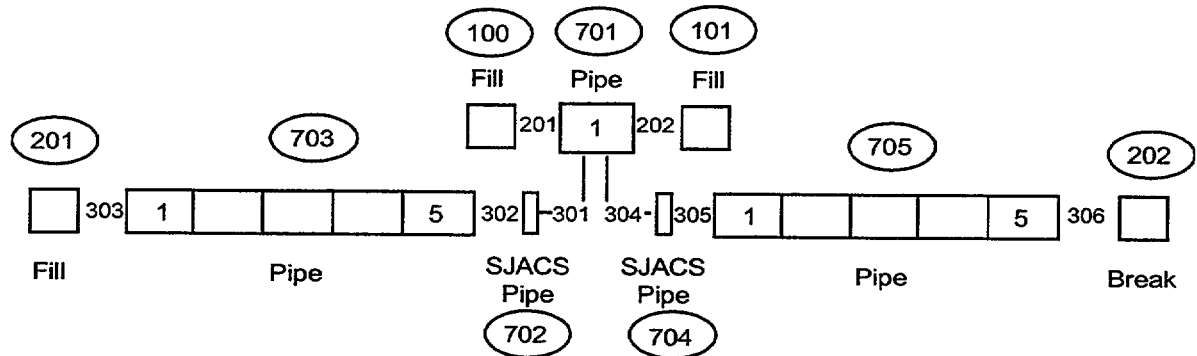


Figure 5.6.22 Noding Diagrams for TFPIPE2 Test Cases - Parametric 2 (Reversed Orientation)

(a) Base Case (No SJC)



(b) SJC Case



**Figure 5.6.23 Noding Diagrams for PLENUM Test Cases - Parametric 1 (Base Case)
(Component 701 is MSJACS PIPE)**

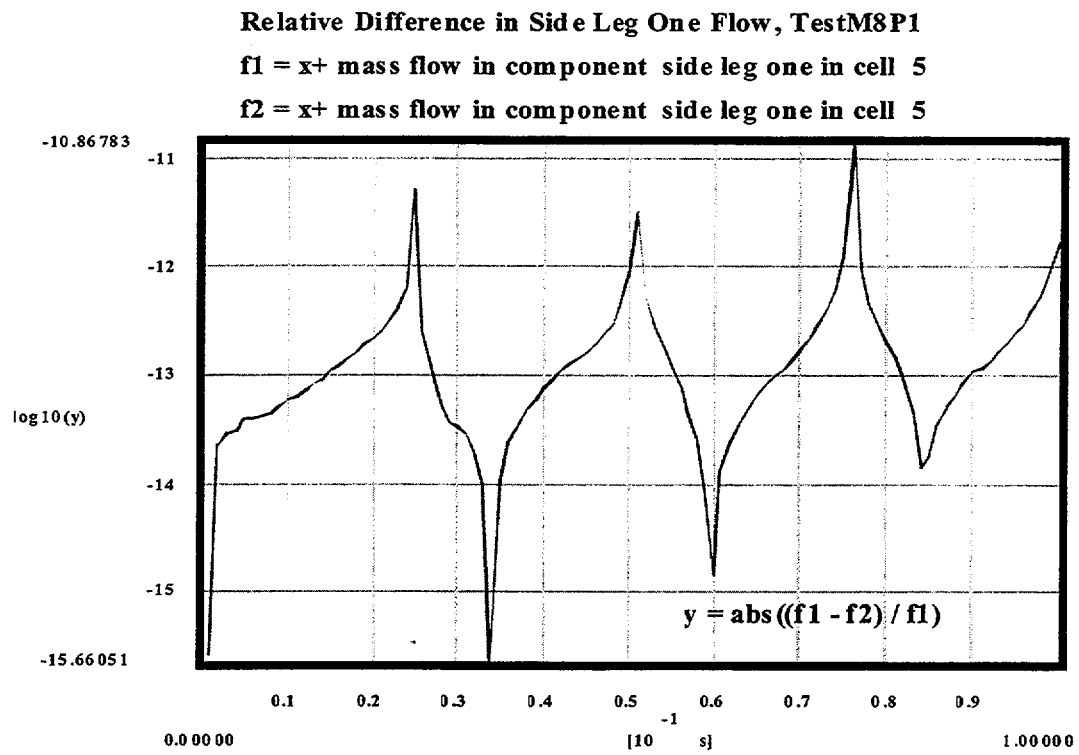


Figure 5.6.24 Relative Difference in Side-Leg Flows for the SJC and Non-SJC Models

5.7 Turbine (TURB) Component

Turbines are used in reactor systems to convert the kinetic and thermal energy of steam to mechanical energy for the turbine rotor blades. The turbine blades, which provide resistance to the steam flow, rotate and provide mechanical energy to turn the generator shaft. Turbines can also be used to provide mechanical energy for rotating shafts to drive pumping systems. The turbine (TURB) component from TRAC-B is integrated into TRAC-M.

5.7.1 Requirements

This section discusses the requirements that must be considered during the design of the TURB component in order to achieve full functionality. A few fundamental equations are given to aid in understanding the requirements. The nomenclature is given in Table 5.7.1. A summary of the requirements appears in Table 5.7.2.

Requirement TURB 1.0: Hydraulics

The turbine component model in TRAC-M will be based on the TRAC-M TEE component model, as is the TURB component in TRAC-B. Thus, the TURB component will use the TEE component procedures and modules for performing standard tasks such as input, output, dump, restart, and graphics. The schematic of a TURB component is shown in Fig. 5.7.1. This new component in TRAC-M will use the generic two-fluid equations and numerical scheme for one-dimensional components, together with the existing constitutive relations of TRAC-M for wall drag, interfacial shear, and heat transfer (which are also used by the TEE component). The requirement is that the new TRAC-M TURB component and its predecessor, the TRAC-M TEE component, must produce identical solutions when all turbine-specific models are excluded.

Requirement TURB 2.0: Additional Mass and Energy Terms

This requirement specifies that terms must be added to the fluid mass and energy equations of TRAC-M to account for energy removal and work performance and phase changes from single-phase fluid (superheated steam) to saturation conditions upon exit from the turbine nozzle (Ref. 5.7.1).

Turbines employed in nuclear power plants consist of two types, an impulse turbine or a reaction turbine. Both turbines have nozzles and rotors. In principle, the impulse steam turbine consists of a casing containing stationary steam nozzles, and a rotor with moving or rotating buckets. For impulse turbines, all of the pressure drop occurs in the nozzle row, with very little pressure drop occurring across the rotor blades. The steam passes through the stationary nozzles and is directed at high velocity against the rotor blades, causing the rotor to turn at high speed. There is a conversion of heat energy to kinetic energy as the heat energy from decreasing the steam enthalpy is converted to kinetic energy from the increased steam velocity, as the steam enters the expansion nozzle. For impulse turbines, the pressure ratio (downstream pressure/upstream pressure) is nearly uniform throughout the superheat range.

By contrast, a reaction turbine has alternating rows of fixed and rotating blades. The steam expands first in the stationary or fixed blades, where it gains velocity as it drops in pressure. The steam then enters the moving blades where its direction is changed, and it imparts an impulse to

the moving blades. This sequence is repeated as the steam passes through fixed and moving blades.

The thermodynamic analysis is more complex for a reaction turbine than for an impulse turbine, but the overall energy conversion process is similar for both types of turbines (or stages). Both types experience an energy and pressure drop, and the conversion of flow energy to mechanical energy. The use of "lumping" turbine stages together in series, along with the energy and pressure drop characteristics, is the basis for modeling both stage types with one component.

The behavior of a turbine process path on an entropy-enthalpy plane is shown in Fig. 5.7.2. An ideal reversible turbine would follow the expansion path from point 1 to point 2. For an actual irreversible turbine, however, the expansion path will go from point 1 to point 3, yielding an actual enthalpy decrease that is somewhat less than the ideal enthalpy decrease.

The turbine efficiency (Ref. 5.7.2) is defined as

$$\eta = \frac{h_1 - h_3}{h_1 - h_2} \quad (5.7.1)$$

For an ideal isentropic expansion of a perfect gas and using the perfect gas law, it can be shown that the enthalpy decrease across the turbine nozzle (Ref. 5.7.2) is

$$\Delta h_{ideal} = \frac{\kappa}{(\kappa - 1)\rho_1} \frac{P_1}{\rho_1} \left[1 - \left(\frac{P_2}{P_1} \right)^{\frac{(\kappa - 1)}{\kappa}} \right] \quad (5.7.2)$$

where ρ is the mixture density, and κ is the ratio of the specific heats. In Equation 5.7.2, the pressures, P_1 and P_2 are the cell-centered pressures in Cells 1 and 2 of Fig. 5.7.1. For the discrete nodalization shown in Fig. 5.7.1, the actual turbine work (Ref. 5.7.1) is

$$w_{turb} = \eta \Delta h_{ideal} \quad (5.7.3)$$

and the total power extracted from the steam and delivered to the rotor is

$$Q_{turb} = \dot{m} \eta \Delta h_{ideal} \quad (5.7.4)$$

with \dot{m} as the mixture mass flowrate.

The term w_{turb} will be included in TRAC-M by using a steady-state stagnation enthalpy balance equation, consistent with TRAC-B (Ref. 5.7.1). The stagnation enthalpy equation is based on the

total energy equation, which takes into account the kinetic energy terms associated with velocities at the entrance (b1) and exit (b2) of the turbine nozzle in Fig. 5.7.1. The stagnation enthalpy balance equation is obtained by dividing the total mixture energy equation by the mixture mass flow. Referring to Fig. 5.7.1, the steady-state stagnation enthalpy balance equation for the nozzle is

$$\frac{1}{2}(V_{b1})^2 + h_{b1} = \frac{1}{2}(V_{b2})^2 + h_{b2} + w_{turb} \quad (5.7.5)$$

The mixture velocity V_{b1} of Equation 5.7.5 is found by dividing the mixture mass flowrate at the left face of cell 1 by the mixture density and face area at the entrance (b1) of the nozzle. The velocity, V_{b2} , is found by assuming that the mixture mass flowrate entering junction b at b1 of Fig. 5.7.1 is equal to the mixture mass flowrate leaving junction b at b2 as

$$\dot{m}_{b1} = \dot{m}_{b2} \quad (5.7.6)$$

or

$$(\rho VA)_{b1} = (\rho VA)_{b2} \quad (5.7.7)$$

with the velocity at b2 calculated from Equation 5.7.7 as

$$V_{b2} = \frac{(\rho VA)_{b1}}{(\rho A)_{b2}} \quad (5.7.8)$$

The area at b1 is the left face area of cell 1, and the area at b2 is the area input for junction b. The TRAC-B turbine model also assumes that the exit conditions of steam from the turbine membrane (turbine exit) are at saturation (Ref. 5.7.1), as recommended by Salisbury (Ref. 5.7.3).

An exit flow quality at station b2 (see Fig. 5.7.1) is calculated in TRAC-B from Equation 5.7.5 by using the mixture enthalpy relation at saturation, which is

$$h_{b2} = x_{b2}(h_{g,b2})^{sat} + (1 - x_{b2})(h_{l,b2})^{sat} \quad (5.7.9)$$

in Equation 5.7.5 to get

$$x_{b2} = \frac{\left(\frac{1}{2}(V_{b1})^2 + h_{b1} - \frac{1}{2}(V_{b2})^2 - w_{turb} - (h_{1,b2})^{sat}\right)}{(h_{g,b2})^{sat} - (h_{1,b2})^{sat}} \quad (5.7.10)$$

The gas and liquid phasic mass flowrate entering at b1 of junction b are summed to give a total mixture mass flowrate

$$\dot{m}_{b1} = \dot{m}_{g,b1} + \dot{m}_{l,b1} \quad (5.7.11)$$

The exit flow quality from Equation 5.7.10 is used to partition the mixture mass flowrate exiting junction b at b2 into phasic mass flowrates of

$$\begin{aligned} \dot{m}_{g,b2} &= x_{b2} \dot{m}_{b1} \\ \dot{m}_{l,b2} &= (1 - x_{b2}) \dot{m}_{b1} \end{aligned} \quad (5.7.12)$$

This method conserves mass for flow from cell 1 to cell 2 of Fig. 5.7.1. The phasic mass flowrates in cell 2 are determined from Equation 5.7.12 which is based on the assumption that cell 2 is at saturation. This partitioning is used in conservation of mass equation. The term w_{turb} is also added in the TRAC-M energy equation. These additional terms are the same as those used in TRAC-B. It is required that these terms are implemented correctly in TRAC-M.

Requirement TURB 3.0: Turbine Momentum Equation

The complicated nature of a steam turbine precludes a first principles momentum model for the purpose of system transient calculations. Thus, the turbine complexity results in the use of a lumped parameter model, wherein the features of the different stages of a turbine can be “lumped” across junction b of Fig. 5.7.1, with inlet b1 and outlet b2.

An important requirement is the use of an appropriate turbine blade assembly momentum equation (turbine nozzle velocity equation) with the TRAC-M phasic momentum equations at the position of the turbine nozzle in Fig. 5.7.1 for computing a nozzle friction factor.

This requirement will give the correct pressure drop across the blade assembly. The standard TRAC-M momentum equation set will use a friction factor based on the turbine nozzle equation of

$$V_{noz} = \left[\left(\frac{\kappa}{\kappa - 1} \right) \frac{P_1}{\rho_1} \left(r_1^{\frac{2}{\kappa}} - r_1^{\frac{(\kappa + 1)}{\kappa}} \right) \right]^{\frac{1}{2}} \quad (5.7.13)$$

with

$$r_1 = \frac{P_2}{P_1} \quad (5.7.14)$$

as the downstream pressure ratio.

A steady-state differenced form of the mixture momentum equation

$$V_{b2} \frac{(V_{b2} - V_{b1})}{\Delta x_1} = - \frac{1}{\rho} \frac{(P_2 - P_1)}{\Delta x_b} - \frac{f}{D_H} |V| V \quad (5.7.15)$$

is used to compute the nozzle friction factor, which is added to the friction term for the nozzle junction, giving the correct pressure drop for the turbine. This is done for forward flow from cell 1 to cell 2 of Fig. 5.7.1. TRAC-B friction factor is used in TRAC-M. It is required that the nozzle velocities calculated by TRAC-M are comparable to those of TRAC-B within 10%. Since TRAC-M uses SETS numerics, it should also use larger time steps.

TRAC-M will use its own set of momentum equations, but the effects of lumping turbine stages in series have to be included in the pressure ratio term (Ref. 5.7.1) of Equation 5.7.14. Lumping turbines in series allows for a decreased computational demand on the computer code.

Requirement TURB 4.0: Critical Flow

TRAC-B requires the use of a modification to Equation 5.7.13 to model critical flow (Ref. 5.7.1) at the turbine nozzle. The modification uses the critical flow value for the downstream pressure ratio of Equation 5.7.14 in Equation 5.7.13 to compute the critical nozzle velocity. The critical pressure ratio is obtained by finding the value of r_1 that maximizes the nozzle mass flowrate.

This is done in TRAC-B to take advantage of lumping turbine stages in series for the nozzle junction only, and the TRAC-B choked-flow model is used at all other junctions (cell faces) of the TRAC-B turbine component.

In the TRAC-M turbine component, it is required that the TRAC-B method for turbine nozzle junction choking will be available as an option for the user. The user will also have the choice of using the default TRAC-M choked-flow model at the nozzle junction, as part of this requirement. The nozzle critical flow calculated by TRAC-M shall be within 5% of hand calculations.

Requirement TURB 5.0: Side Arm Drain

An optional side arm drain is required to remove liquid from the main turbine flow stream, and divert it to the side arm of the turbine. This will allow the turbine component to simulate liquid drains to remove condensate. The TRAC-B side arm drain correlation is inversely proportional to the size of the time step (Ref. 5.7.1). This model is known to oscillate based on earlier tests conducted with the TRAC-B turbine model. As a result, the model will be replaced with a simple steady-state correlation from the PEPSE code (Ref. 5.7.4) that is proportional to the amount of moisture contained in cell 2 of Fig. 5.7.1. The PEPSE side arm drain model is being used since it does not have a built-in time step dependence, as does TRAC-B. The PEPSE side arm drain model has also been demonstrated to work without oscillations. It is required that the PEPSE side arm drain model is correctly implemented.

Requirement TURB 6.0: Turbine System

A model is required to simulate the dynamics of multiple stages and the turbine rotor assembly. This model and the relevant equations will be taken from TRAC-B (Ref. 5.7.2). The rotor model will track the turbine rotor speed, which is required for the calculation of the turbine efficiency and for the detection of over-speed conditions. The turbine speed is calculated by integration of the equation

$$I \frac{d\Omega}{dt} = T + T_f + T_b \quad (5.7.16)$$

where T is the rotor torque supplied by steam flow, T_f is the frictional rotor torque, and T_b is the bearing torque. It is required that TRAC-M shall simulate multi-stage systems, dynamics of the rotor and its associated efficiency calculation and the results shall be within 10% of the results calculated by TRAC-B.

5.7.2 Verification Testing and Assessment

Test TURB 1.0: Hydraulics

The purpose of this test is to verify that the implementation of the TURB component gives the same result as the TEE component when the TURB models are excluded. This test will also be performed from steady-state conditions using the restart feature of TRAC-M for the TURB component's input, output, and graphics features. The acceptance criteria is that identical results must be obtained using both the TEE and TURB components when the TURB is used as a TEE,

and that the model must be able to restart from steady state with input, output, and graphics information.

The TRAC-M TURB component should have the same flow rate and pressure distribution as a TEE component when the NOTBWR variable is set to TRUE. To verify this, TURB and TEE components were compared for the model in Fig. 5.7.3 with NOTBWR set to TRUE.

The test started from single-phase steam conditions, and executed for 100 s. All components contained single-phase steam at a pressure of 4.7 Mpa, a temperature of 533 K, and a void fraction of 1.0. The cell pressure results of the simulation for each case are shown in Fig. 5.7.4.

The TURB hydraulics test was successful because the vapor exit velocities are identical for both cases, using the TURB component with NOTBWR as TRUE and the TEE. This verifies that Requirement TURB 1.0 is met for the turbine.

Test TURB 2.0: Additional Mass and Energy Terms

The TRAC-B turbine model requires the addition of terms to the mass and energy equations to reflect the work lost by converting the kinetic energy of steam flow to mechanical energy for the turbine blades. This same requirement applies to the TRAC-M TURB component, which is based on the TRAC-M TEE component.

The purpose of this test is to demonstrate that the additional mass and energy terms added to the TRAC-M TEE component are correctly implemented, and that mass and energy are conserved. These terms in TRAC-M will be compared to the terms of the TRAC-B mass and energy equations, and to steady-state hand calculations for the simple one-stage turbine model of Fig. 5.7.3. This verification test will not yield a one-to-one comparison, since the two codes use different constitutive models, and the numerical results may differ slightly, but will verify that the terms have been correctly implemented and that mass and energy are conserved as specified by Requirement TURB 2.0.

It should be noted that, along with different constitutive models, the TRAC-M code uses a mean mass continuity equation for single-phase flow when the gas void fraction has values of zero or one. The TRAC-M code also has a predictor and stabilizer phasic momentum equation step, which introduces additional numerical damping into the TRAC-M solution. The TRAC-B code uses the full six equation phasic set at all times during the solution procedure, by retaining a small amount of liquid or steam for the junction void fraction.

The acceptance criteria are that the additional TRAC-M turbine component mass and energy flux terms will produce the same general trends as the TRAC-B turbine mass and energy flux terms, and that the calculated mass and energy conservation will be (at a minimum) as good as those generated by TRAC-B for the simple problem illustrated in Fig. 5.7.3.

The conserving character of the model was tested using the nodalization of Fig. 5.7.3 for both TRAC-B and TRAC-M TURB components. For implementation of the additional mass and energy terms in TRAC-M, the mass and energy flows at steady state must balance. This simply states that the mass and energy flow in must equal the mass and energy flow out. This is the original test that was performed for TRAC-B (Ref. 5.7.1). This test uses the total energy balance equation

$$\dot{m} \left[h + \frac{V^2}{2} \right]_{in} = \dot{m} \left[h + \frac{V^2}{2} \right]_{out} + \dot{m} w_{turb} \quad (5.7.17)$$

Table 5.7.3 shows the values of the left and right sides of Equation 5.7.17, with the respective energy error for TRAC-M. Ref. 5.7.1 noted that the TRAC-B implicit turbine model conserved mass and energy better than the explicit turbine model, but additional errors arose as the number of stages were increased. This was attributed to an erroneous flow quality calculation performed in TRAC-B, which has been corrected in TRAC-M.

The conservation of mass character of the turbine component in TRAC-B and TRAC-M was also tested by running a constant fill problem for the model in Fig. 5.7.3. The results are shown in Fig. 5.7.5. The mass flowrate for the TRAC-M turbine model is the same at the inlet and the outlet, where the mass flowrate for the TRAC-B turbine is larger at the outlet as a consequence of allowing the flow quality to have numerical values greater than one. This test was performed using the semi-implicit method in TRAC-B. The results using the Courant violating numerics are shown in Fig. 5.7.6 on a reduced scale. The results show that mass and energy are conserved in TRAC-M while there are some errors in TRAC-B. Figs. 5.7.5 and 5.7.6 verify that requirement TURB 2.0 has been satisfied.

Test TURB 3.0: Turbine Momentum Equation

The TRAC-B turbine component indirectly relies on the use of the ideal gas turbine nozzle equation for calculation of a friction factor (form loss) across the turbine nozzle. This test will compare the calculated nozzle velocities of the TRAC-M turbine component to those of the TRAC-B turbine component, using the basic model of Fig. 5.7.3, and will compare the allowable time step sizes for the turbine model of both codes. The acceptance criteria is that the TRAC-M turbine model will compute nozzle velocities that are comparable to those of TRAC-B (within 10%), and will allow larger time steps than its TRAC-B predecessor. The computed nozzle velocities for the two codes are shown in Table 5.7.4.

A plot of the nozzle velocities at steady-state conditions is shown in Fig. 5.7.7. The nozzle velocities can be adjusted closer together by tuning the friction parameters of the code. The error is shown in Table 5.7.4, as calculated without tuning the code models.

An analysis of the effects of the time step on the turbine model was also conducted using the TRAC-B and TRAC-M models for the simple one-stage system shown in Fig. 5.7.3. Break conditions and system geometry were identical for both input decks. The only significant difference between the TRAC-B and TRAC-M models was an additional friction factor specified at the central turbine junction in the TRAC-M deck. This factor was necessary to achieve roughly the same steady-state pressure distribution in the turbine for both codes.

Initially, the maximum time step was arbitrarily set to 1.0 s and the problem end time was set to 100 s for both models. TRAC-B imposed a Courant limit, which allowed a maximum time step of only about 0.1 s; therefore, increasing the time step beyond 0.1 s would not affect the results. However, TRAC-M did run at a time step of 1.0 s. The results shown in Fig. 5.7.8 illustrate the pressure traces in the second cell of the turbine for maximum time steps of 0.1 s for TRAC-B and 1.0 s for TRAC-M.

The steady-state results are very similar beyond 0.5 s. The Courant-limited TRAC-B results illustrate a pressure oscillation at the beginning of the transient that quickly damped out. This oscillation did not occur with TRAC-M. This figure illustrates that the TRAC-M turbine model allows larger time steps than TRAC-B, since the same results were achieved although TRAC-M ran the problem with a time step nearly 10 times greater than TRAC-B.

The sensitivity of the TRAC-M model to time step size was tested by varying the maximum time step between 0.1 and 5.0 s. Fig. 5.7.9 illustrates the TRAC-M results from maximum time steps of 0.1 s, 1.0 s, and 5.0 s, compared to the TRAC-B results from a maximum time step of 0.1 s. TRAC-M results are not sensitive to selection of maximum time steps. The results of this section indicate that Requirement TURB 3.0 has been met.

Test TURB 4.0: Critical Flow

This test is designed to verify that Requirement TURB 4.0 is satisfied. The TRAC-M turbine component will have options for using the TRAC-B turbine nozzle critical flow model, based on the ideal gas flow equation, and the TRAC-M default choked-flow model.

The turbine nozzle critical flow model option in the TRAC-M turbine component will be checked by comparison to hand calculations for steady-state conditions. The acceptance criteria is that the calculated and computed values must agree to within 5%.

The relation describing the critical flow for an ideal gas in a nozzle is

$$V_{noz} = \left[\left(\frac{\kappa}{\kappa - 1} \right) \frac{P_1}{\rho_1} \left(r_{crit}^{\frac{2}{\kappa}} - r_{crit}^{\frac{(\kappa+1)}{\kappa}} \right) \right]^{\frac{1}{2}} \quad (5.7.18)$$

where κ is the ratio of specific heats, P_1 is the upstream pressure, ρ_1 is the upstream density, and r_{crit} is the critical pressure ratio, which is

$$r_{crit} = \left(\frac{2}{\kappa + 1} \right)^{\kappa(\kappa - 1)} \quad (5.7.19)$$

For the simple model of Fig. 5.7.3, the following values for the upstream pressure and density are found from the output as

$$P_1 = 5.42237 \text{ MPa}$$

$$\rho_1 = 26.57 \text{ kg/m}^3$$

$$\kappa = 1.1$$

Using Equation 5.7.19,

$$r_{crit} = 0.58467$$

and

$$V_{crit} = \left[2.0 \left(\frac{1.1}{0.1} \right) \frac{(5.42237e6)}{26.57} \{ (0.58467)^{1.818181} - (0.58467)^{1.909090} \} \right]^{\frac{1}{2}}$$

$$V_{crit} = 283.8 \text{ m/s}$$

The TRAC-M calculated value is 283.86 m/s. The error is

$$\text{percent error} = \frac{(283.8 - 283.86)}{283.86} 100 = -0.021137$$

This error is negligible, and indicates that the correct critical flow is calculated for the nozzle. Thus, the results satisfy Requirement TURB 4.0.

Test TURB 5.0: Side Arm Drain

This test uses the model shown in Fig. 5.7.10. 100% separation will be established for the side-arm junction of the low-pressure turbine to ensure that a gas exits the model at junction 3, while liquid exits the side arm at junction 7. This test will establish that the steam and water are properly separated through the side arm junction, by examining the output for this junction to show that liquid alone exits the side arm junction. This demonstrates that the acceptance criteria are met.

In Fig. 5.7.11, the liquid velocity at the drain inlet is compared as calculated by the TRAC-M and TRAC-B separation model. Based on Fig. 5.7.11, the TRAC-B separation model does not correctly model liquid flow out of the drain. This may be attributable to the dependence of the TRAC-B separation model on the inverse of the time step. In Fig. 5.7.11, the TRAC-B and TRAC-M flow areas were set to 1.0 square meters. In Fig. 5.7.12, the results are compared between the two codes for a low-flow case and on a reduced vertical scale. The liquid velocity calculated by TRAC-M is more stable throughout the transient because the model is not dependent on the value of the time step, as is TRAC-B which contains a $1/\Delta t$ dependency. These results verify that Requirement TURB 5.0 has been satisfied.

Test TURB 6.0: Turbine System

This requirement represents the dynamics of the turbine rotor assembly (Ref. 5.7.1), and will be tested using the model illustrated in Fig. 5.7.10 connected to a rotor assembly. The results will be compared to TRAC-B for the same model, for the calculated rotor power and rotor speed for transient and steady-state conditions. The model shown in Fig. 5.7.10 has two stages, which demonstrate multiple-stage modeling capability.

The two code models should show the same trends in behavior and generally equivalent numerical values (within 10%), which is the success criterion for this test. This criterion is chosen because the experimental data may not have less than a 10% accuracy.

The results of calculations from both codes for pressures, nozzle velocities, and vapor temperatures are shown in Figs. 5.7.13 through 5.7.15. Note that the TRAC-B results shown in Fig. 5.7.15 show oscillations in the vapor temperature. The results are indicative of the mass and energy errors noted in earlier multistage turbine models. The TRAC-M results verify that Requirement TURB 6.0 has been met.

5.7.3 Conclusions

These results show that the TURB model implemented in TRAC-M meets the stated requirements. The TURB model in TRAC-M is an improvement over the TRAC-B model.

REFERENCES

- 5.7.1 Wade, N.L., et al., "TRAC-BF1/MOD1 Models and Correlations," U.S. Nuclear Regulatory Commission, NUREG/CR-4391, June 1992.
- 5.7.2 Taylor, D., et al., "TRAC-BD1/MOD1: An Advanced Best-Estimate Computer Program for Boiling-Water Reactor Transient Analysis," Volume 1, Model Description, U.S. Nuclear Regulatory Commission, NUREG/CR-3633, April 1984.
- 5.7.3 Salisbury, J.K., *Steam Turbines and Their Cycles*, Kreiger Publishing, 1950.
- 5.7.4 Minner, G.L., et al., "PEPSE: Volume II, Theoretical Description, Nuclear Turbine Stage Groups," ScienTech, Inc., 1996.

Table 5.7.1 Nomenclature Used in Equations Related to TURB Component Design

Variable	Value
η	turbine efficiency
h	enthalpy (J/kg)
P	pressure (MPa)
ρ	density (kg/m ³)
κ	ratio of specific heats
w_{turb}	turbine work term (J/kg)
V_{b1}	velocity at nozzle entrance (m/s)
V_{noz}	velocity at the nozzle
V_{crit}	critical nozzle velocity
f	friction factor (dimensionless)
D_H	hydraulic diameter (m)
I	rotor moment of inertia (Nm)
Ω	rotor angular speed (radians/sec)
T	torque (Nm)
\dot{m}_{b1}	mixture mass flowrate (kg/s) at b1 of junction b in Fig. 5.7.1
$\dot{m}_{g,b1}$	gas mass flowrate (kg/s) at b1 of junction b in Fig. 5.7.1
$\dot{m}_{l,b1}$	liquid mass flowrate (kg/s) at b1 of junction b in Fig. 5.7.1
$\dot{m}_{g,b2}$	gas mass flowrate (kg/s) at b2 of junction b in Fig. 5.7.1
$\dot{m}_{l,b2}$	liquid mass flowrate (kg/s) at b2 of junction b in Fig. 5.7.1
r_{crit}	critical pressure ratio
Δx_1	length (m) of Cell 1 in Fig. 5.7.1
Δx_b	length (m) of junction b in Fig. 5.7.1

Table 5.7.2 TURB Requirements

Requirement	Requirement Statement
TURB 1.0	The TRAC-M turbine model and TRAC-M TEE must yield identical answers when the turbine options are turned off, and the TURB component will use the TEE component procedures and modules to perform standard tasks (such as input, output, dump, restart, and graphics).
TURB 2.0	The TRAC-M turbine model will include the additional terms that are necessary for the mass and energy equations to simulate a turbine, based on the TRAC-B TURB component. They shall be correctly implemented.
TURB 3.0	The turbine nozzle momentum equation will provide the correct pressure drop and stage lumping capabilities for a turbine. The nozzle velocity calculated by TRAC-M shall be within 10% of that calculated by TRAC-B.
TURB 4.0	The TRAC-B nozzle choked-flow model and the use of the TRAC-M default choked-flow model for the turbine nozzle will be available as options. The TRAC-M calculated choked-flow shall be within 5% of hand calculations.
TURB 5.0	The TURB component will simulate liquid drains for the side arm of the TEE.
TURB 6.0	The TURB component will simulate multistage systems and the dynamics of a rotor and its associated efficiency calculation. The results shall be within 10% of those calculated by TRAC-B.

Table 5.7.3 Energy Error in TRAC-M

	Left Side of Eq. 5.7.17	Right Side of Eq. 5.7.17	%Error
TRAC-M	5.657e+9 (J/s)	5.672e+9 (J/s)	0.268

Table 5.7.4 Nozzle Velocity Comparison

	Nozzle Velocity (m/s)
TRAC-B	83.4
TRAC-M	77.1

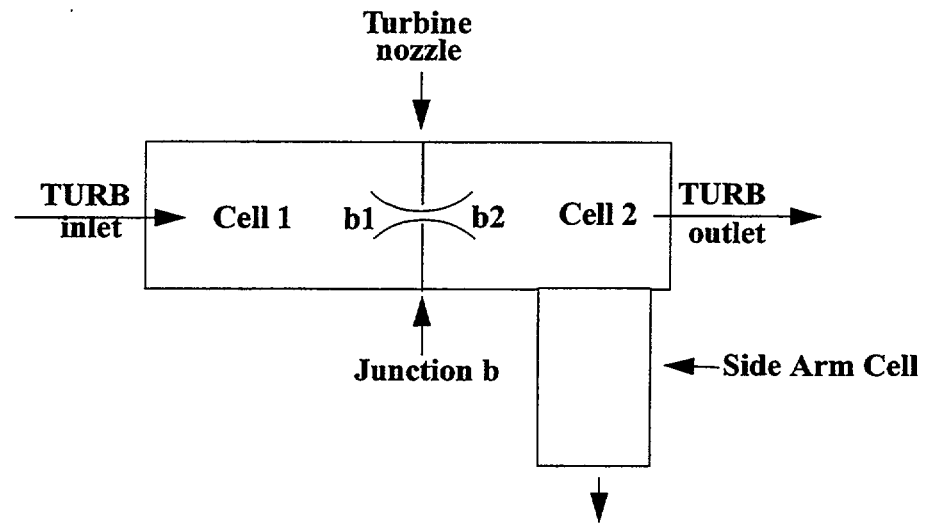


Figure 5.7.1 Turbine Nodalization

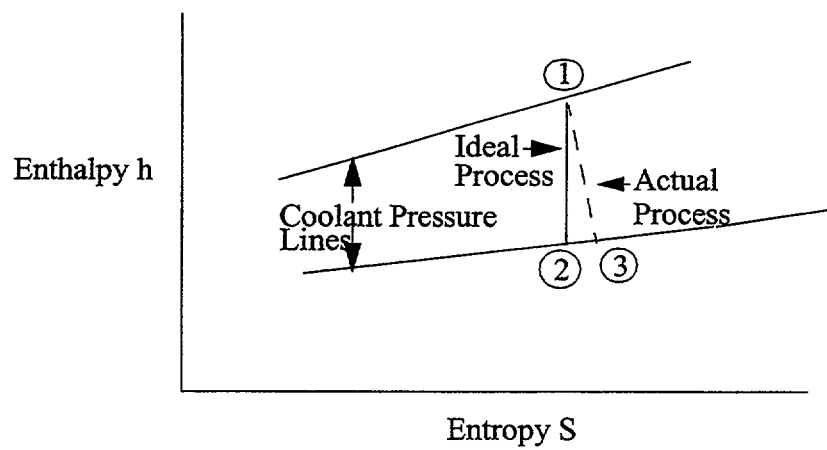


Figure 5.7.2 Entropy-Enthalpy Process Diagram

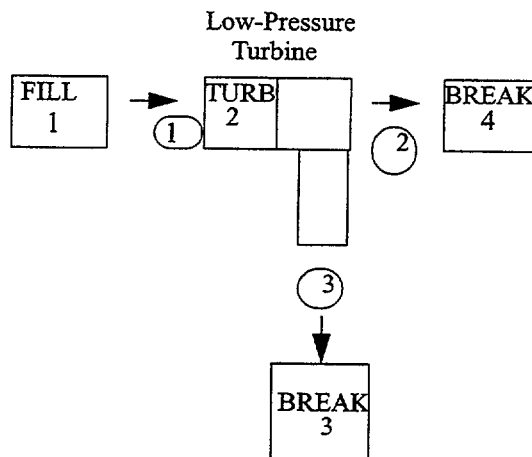


Figure 5.7.3 Simple Turbine Test Model

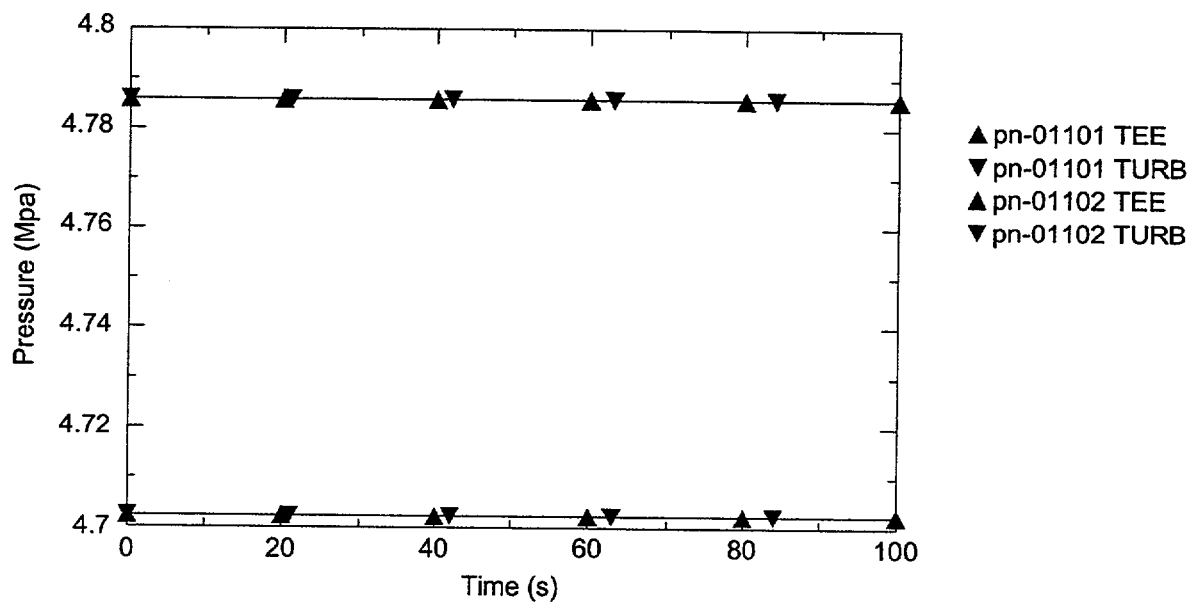


Figure 5.7.4 Pressure Comparison for TURB1

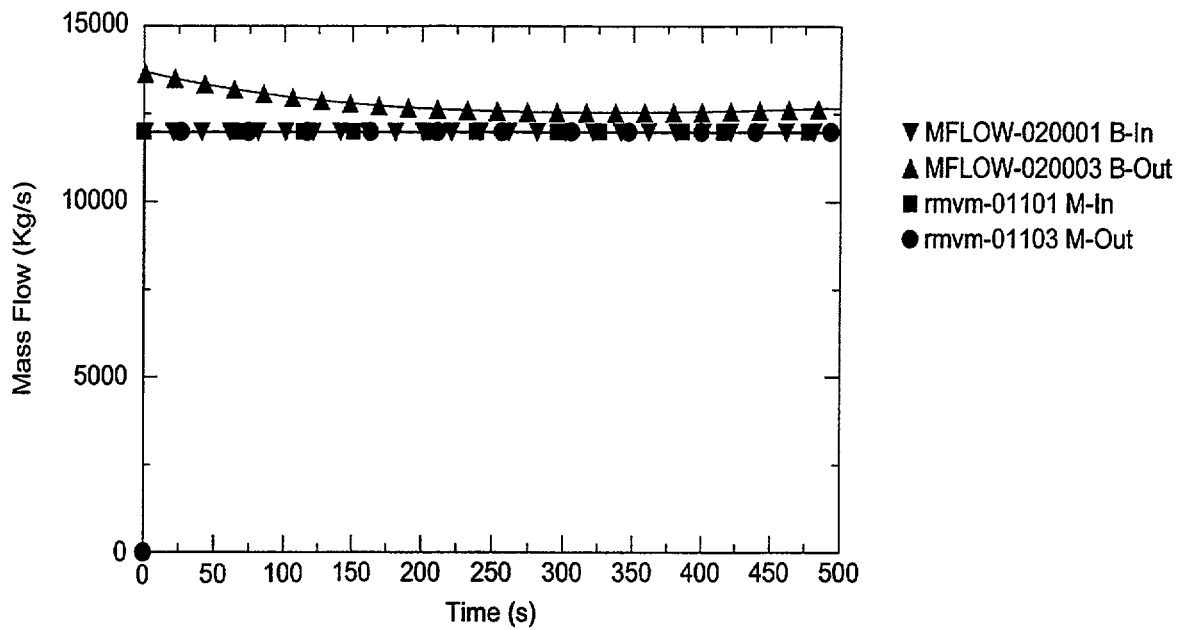


Figure 5.7.5 Mass Conservation Test (Semi-Implicit)

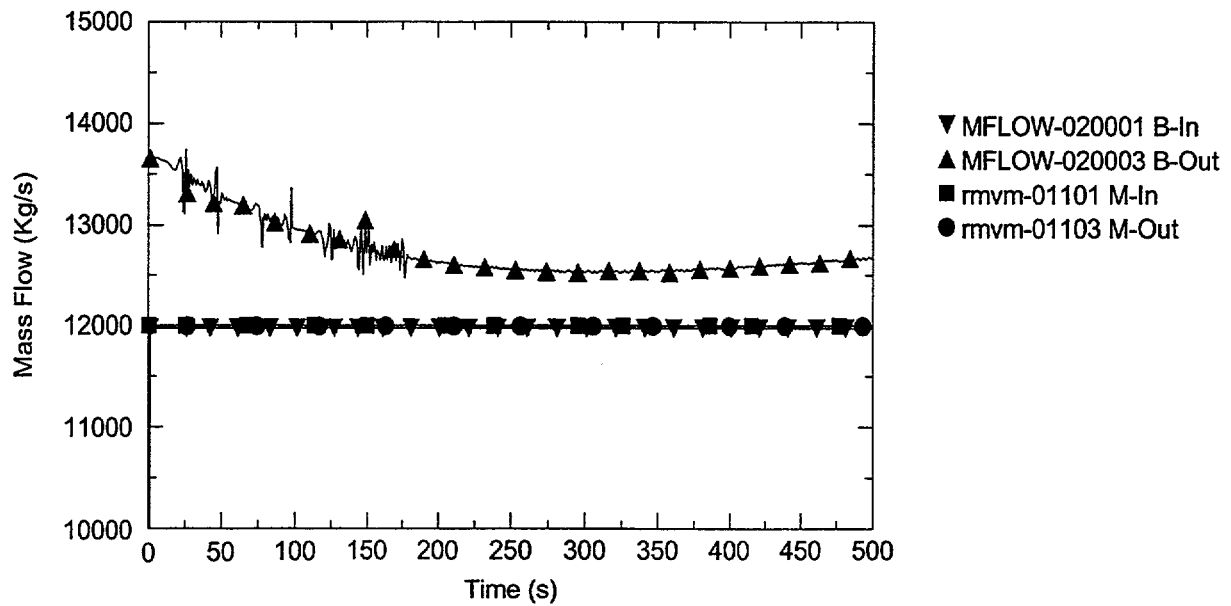


Figure 5.7.6 Mass Conservation Test with Courant Violating Numerics

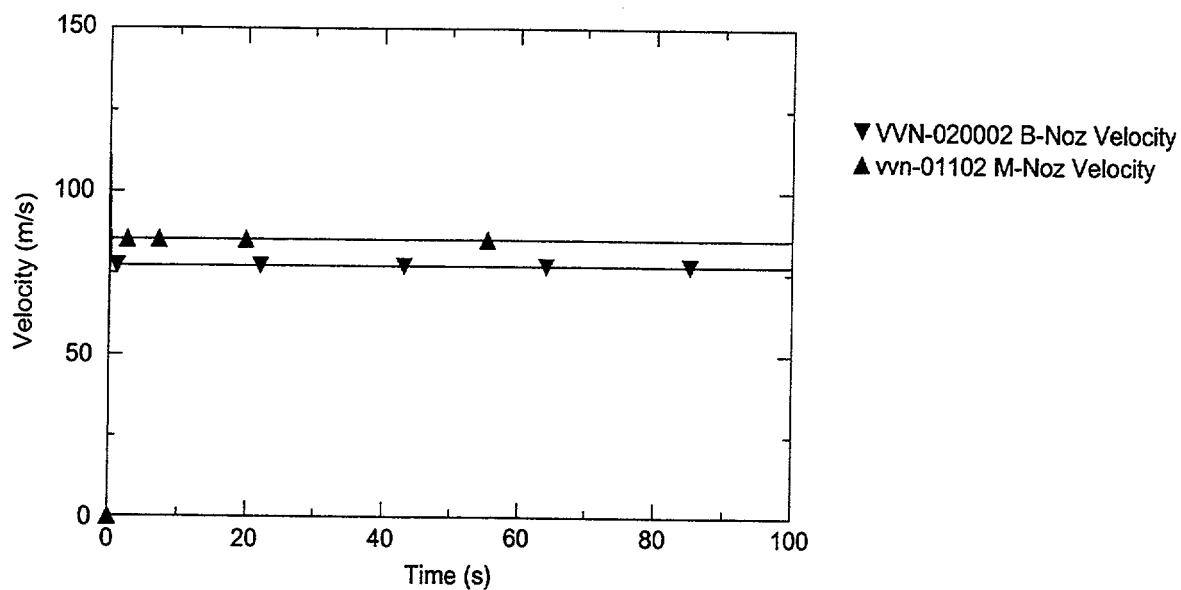


Figure 5.7.7 TRAC-B and TRAC-M Nozzle Velocities

Pressure in second cell of turbine

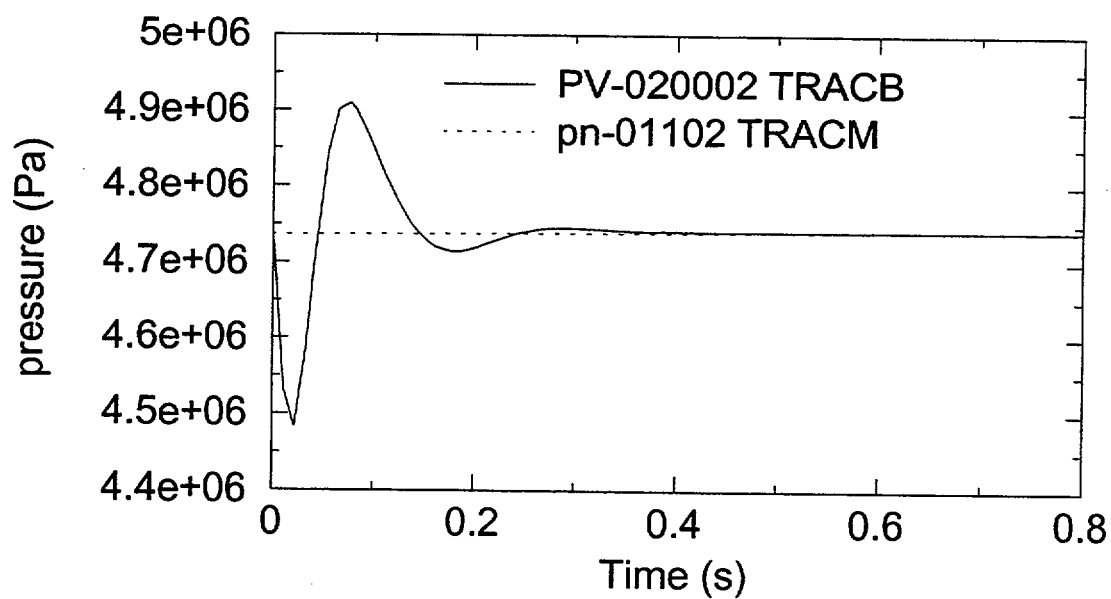
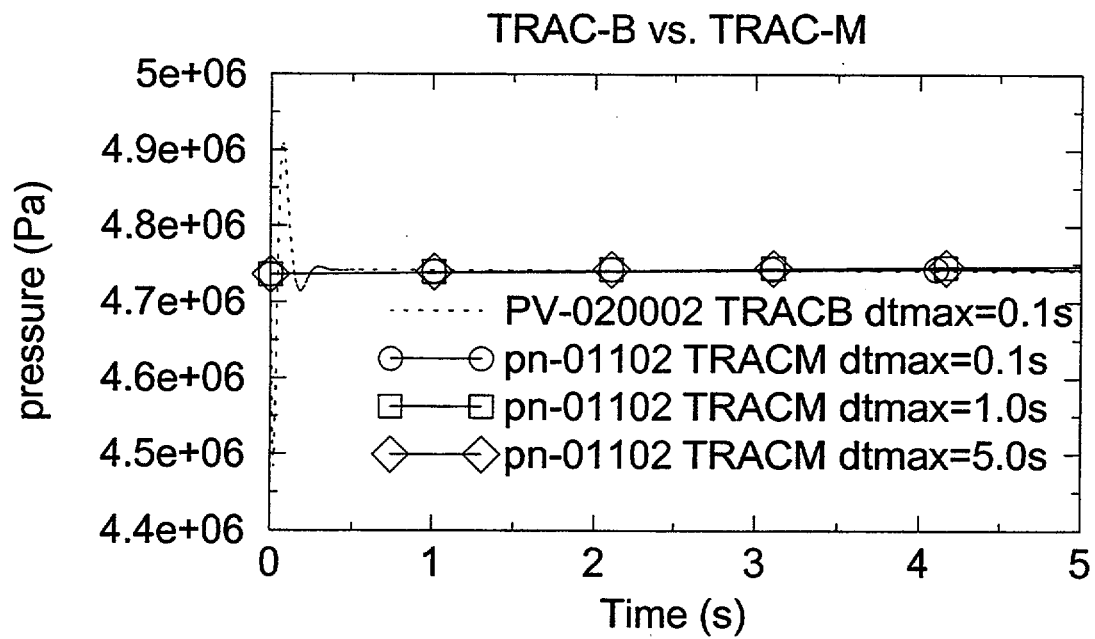


Figure 5.7.8 Pressure Response in Second Cell of Turbine

Time Step Sensitivity



**Figure 5.7.9 Pressure Response for TRAC-B and TRAC-M
with 0.1, 1.0, and 5.0 Second Time Steps**

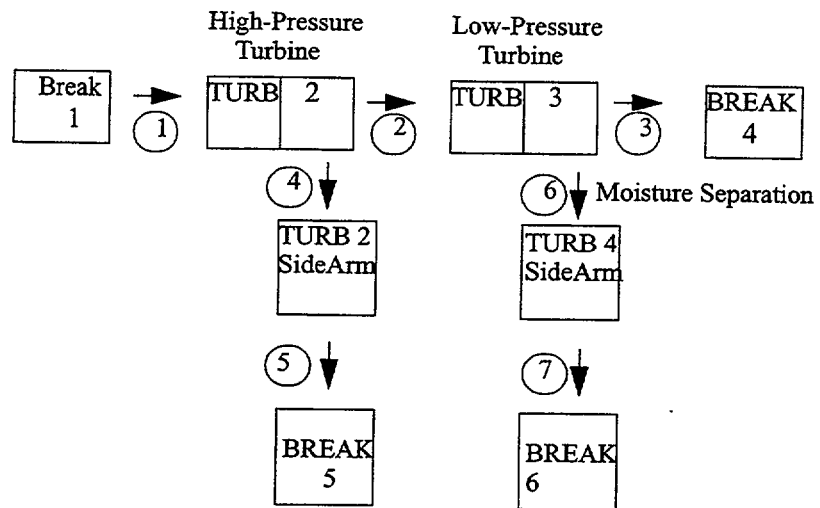


Figure 5.7.10 Turbine System Model

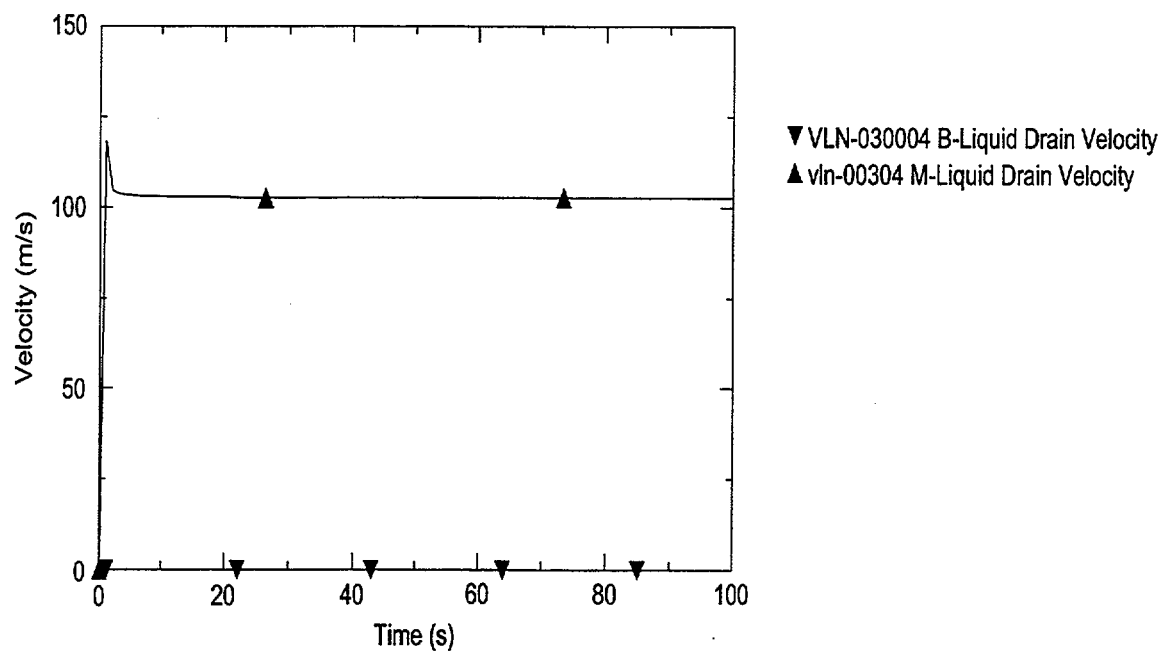


Figure 5.7.11 Comparison of TRAC-M and TRAC-B Liquid Drain Velocities

Liquid Velocity at Drain Inlet

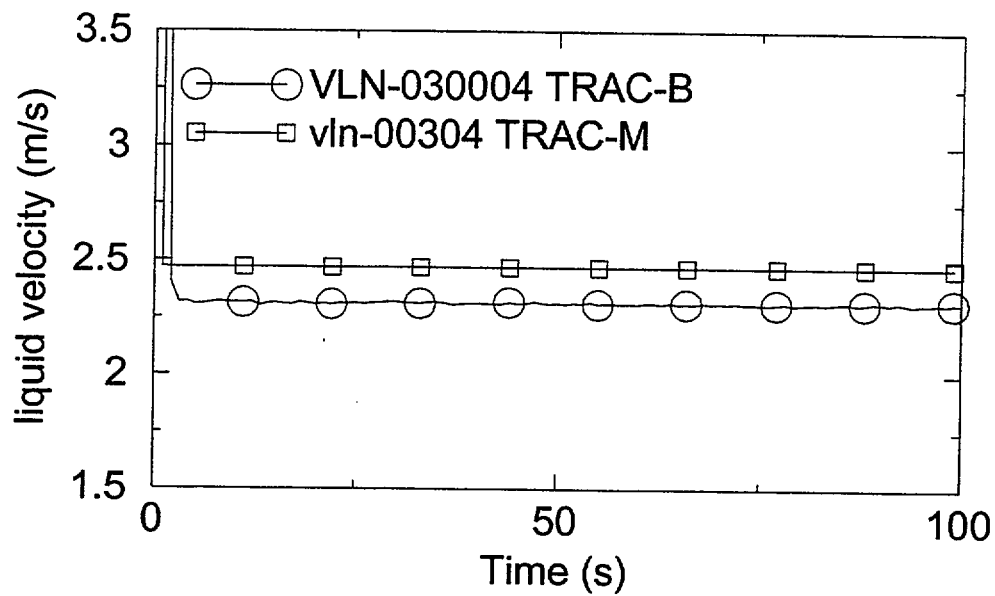


Figure 5.7.12 Comparison of Drain Models for Low Flow

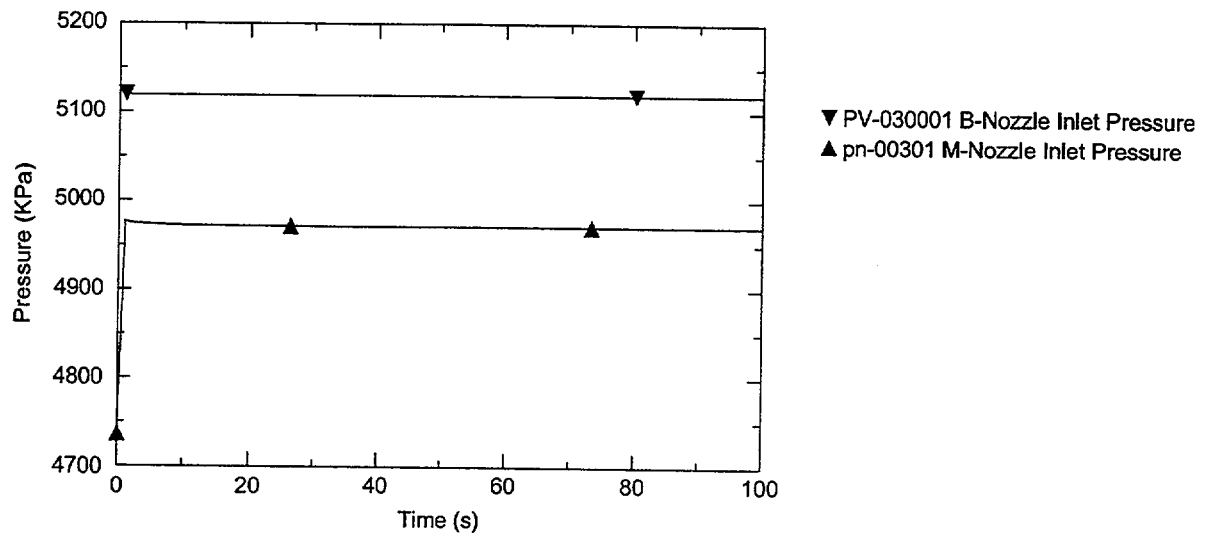


Figure 5.7.13 System Pressure Comparison

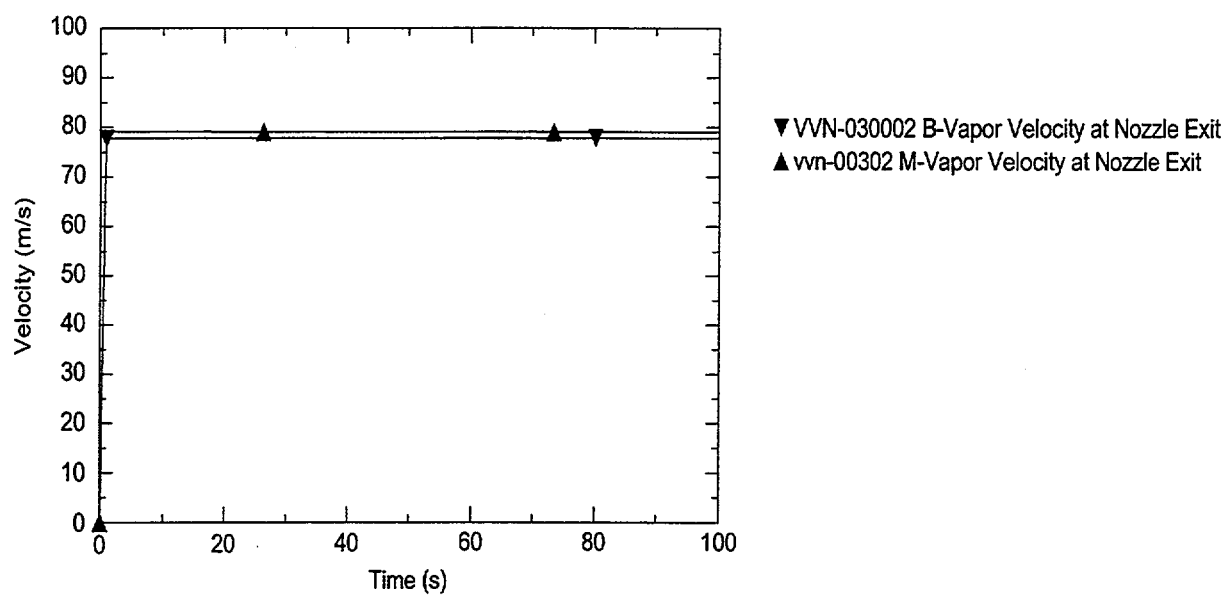


Figure 5.7.14 System Velocity Comparisons

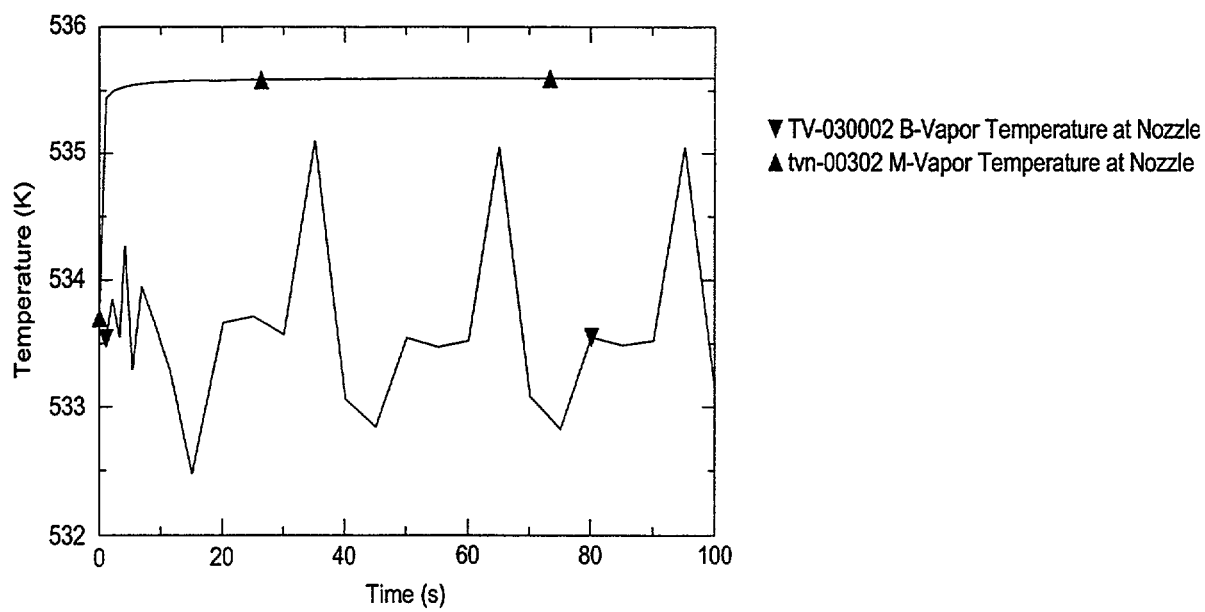


Figure 5.7.15 System Vapor Temperature Comparison

6. Assessment of Integration of Spatial Kinetics

The spatial kinetics capability based on the Purdue Advanced Reactor Core Simulator (PARCS) code has been integrated to TRAC-M. The integration is accomplished by implementing a linkage between the two codes. At each time step, fuel and thermal hydraulic data are transferred from TRAC to the PARCS code, and the power data are transferred from the PARCS code to TRAC. The new coding provides the capability to map the data. This chapter presents the assessment of this mapping functionality.

The linkage is composed of three separate codes, including the General Interface (GI), T/H Data Map (THDM), and Neutronics Data Map (NDM). In future versions of TRAC-M(F90), the GI and NDM will be integrated into the PARCS code to form a spatial kinetics module, and THDM will be integrated with the T/H modules. There are two reasons for this approach. The first is that the present linkage requires substantial user input, which burdens the user. The second is that the performance of the code with all of these mapping functions becomes too slow. The assessment of the mapping capability in future versions will be performed as new versions are completed.

6.1 Description of Linkage

The implementation of a thermal-hydraulic/neutronic interface for the reactor vessel is depicted in Fig. 6.1.1. The design is based on an internal integration approach, which implies that the neutronics code, which will maintain both point and spatial kinetics solution capabilities, solves for the neutron behavior only, and the system code solves for both the system and the core thermal-hydraulics. It should be noted that the design shown in Fig. 6.1.1 is simply representational; details of calculation flow control will be discussed in Section 6.1.4.

This concept utilizes independent input processing for the T/H and neutronic codes, which ensures that modifications to existing input decks will be minimal. In this version, the input for the interface is processed separately by the code-specific mapping routines. Code-specific mapping and calculational control information will be discussed in Sections 6.1.3 and 6.1.4, respectively. In future versions, the user will not provide this input; it will be provided within the code.

The basic design of the interface includes an immutable generalized interface (GI) unit, which provides the mapping of property data between T/H zones, neutronic nodes, and heat structure (HS) components. Memory mapping between the interface module and the T/H and neutronic modules is accomplished with customized mapping routines that are specific to the T/H and neutronic codes used in the coupled code system. These routines and the variables to be transferred will be discussed in the following sections.

6.1.1 Variable Transfer

The coupling of the neutronics module to the T/H module is accomplished by incorporating T/H feedback effects into the cross-sections. For an explicit coupling approach, space-dependent fluid and conduction property data (such as fuel and moderator temperature and moderator density) from the end of the time step are passed through the interface to the neutronics module. The coupling of the T/H module to the neutronics module is achieved with the node-averaged

fission, fission product decay, and actinide decay powers (Q_f , Q_{fp} , Q_{ac}), which are used in the heat transfer calculation. For transient control in the coupled code, *time-dependent information*, such as the logic for control rod movement and the time step data, are also passed through the interface.

In an effort to fully generalize the treatment of the fuel temperature feedback, the space-dependent *average fuel temperature* (\bar{T}_f) and *fuel temperature profile* in the pin ($T_f(r_0 \rightarrow r_n)$) are transferred to the interface, as shown in Fig. 6.1.1. This provides flexibility for the neutronics codes, which vary in the method of calculating doppler temperature for feedback into the cross-sections. To describe the profile, information concerning the pin geometry and discretization is passed through the interface to the neutronics code at the beginning of the calculation. The treatment of the moderator density feedback is generalized to allow the use of mixture density and/or void density correlations. This is accomplished by transferring the space-dependent *liquid and vapor densities*, along with the *void fraction* (ρ_l , ρ_v , α), from the T/H module to the interface. Finally, the space-dependent *moderator temperature* (T_m), *pressure* (P), and *Boron concentration* (B) are also passed through the interface. It should be noted that all property data mapped by this interface utilize SI units.

In practice, the variables described above are sufficient to couple a T/H code to a neutronics code. Nonetheless, flexibility is incorporated to allow the transfer and mapping of any space-dependent information. With regard to output editing and generation of restart files, it is assumed that the T/H and neutronics codes will maintain independent control. However, it is necessary to ensure that the frequency of output and restart updates is synchronized for the coupled code. Therefore, sufficient control information should be passed through the interface and processed by either of the code-specific mapping routines.

6.1.2 General Interface Structure and Data Transfer

The structure of the GI unit is dependent on the nature of the coupling used to link the T/H, neutronics, and interface modules. In general, this module contains three functional units, as shown in Fig. 6.1.2. In the initialization stage, all necessary mapping and geometry information is transferred to the GI, where it is stored for use in the two subsequent variable mapping units. It should be noted that this general structure also applies to the code-specific T/H and neutronic data mapping routines.

As discussed in Section 6.1.3, the mapping functions to be passed to the GI are represented as two permutation matrices, one for each mapping direction. These two permutation matrices, which are stored using the Coordinate Storage Format (Ref. 6.1.1) are processed by either the T/H or neutronic data mapping routines, and are sent to the GI during initialization. For the unit controlling the mapping of T/H fluid and conduction data to neutronic nodes, a vector of T/H zone-wise and component-wise variables are passed to the GI. Conversely, for the unit controlling the mapping of neutronic data to heat structure components and T/H zones, a vector containing all neutronic nodal data is passed to the GI.

For a multiple-process coupling strategy, as discussed in Section 6.1.4, the transfer of this data through the interface routines is performed with *send* and *receive* functions, and essentially two

pieces of information are transferred. The first involves information that is actually used by GI, and consists of the permutation matrices and the vectors to be mapped. The second consists of any additional information that is used for calculational coherency between the T/H and neutronic processes. As shown in Fig. 6.1.3, this information is packed into a data structure consisting of characters, logicals, integers, and reals, which is used to transfer initial control information through the interface, as well as time-dependent information during the calculation (such as control rod and time step logic).

Most of the information stored in these buffers is needed only by the T/H and neutronic processes. However, the following locations are set aside for access by the GI:

- Initial Thermal-Hydraulic Control Buffer:

- logical(1): Indication of an error in the T/H code
- logical(2): Indication of a data error in the T/H-Specific Data Map Routine
- logical(3): Indication of a PVM error in the T/H-Specific Data Map Routine
- logical(4): Indication of a data error in the General Interface
- logical(5): Indication of a PVM error in the General Interface
- logical(6): Indication of normal calculation termination
- logical(7): Indication of whether T/H-Specific Data Map Routine is sending the permutation matrices

- Initial Neutronic Control Buffer:

- logical(1): Indication of an error in the Neutronic code
- logical(2): Indication of a data error in the Neutronic-Specific Data Map Routine
- logical(3): Indication of a PVM error in the Neutronic-Specific Data Map Routine
- logical(4): Indication of a data error in the General Interface
- logical(5): Indication of a PVM error in the General Interface
- logical(6): Indication of normal calculation termination
- logical(7): Indication of whether Neutronic-Specific Data Map Routine is sending the permutation matrices

- Time-Dependent Thermal-Hydraulic Control Buffer:

- logical(1): Indication of an error in the T/H code
- logical(2): Indication of a data error in the T/H-Specific Data Map Routine
- logical(3): Indication of a PVM error in the T/H-Specific Data Map Routine
- logical(4): Indication of a data error in the General Interface
- logical(5): Indication of a PVM error in the General Interface
- logical(6): Indication of normal calculation termination

- Time-Dependent Neutronic Control Buffer:

- logical(1): Indication of an error in the Neutronic code
- logical(2): Indication of a data error in the Neutronic-Specific Data Map Routine

- logical(3): Indication of a PVM error in the Neutronic-Specific Data Map Routine
- logical(4): Indication of a data error in the General Interface
- logical(5): Indication of a PVM error in the General Interface
- logical(6): Indication of normal calculation termination

6.1.3 Design of the Mapping Function

The function used by the GI to map zone-wise and component-wise data to neutronic nodes is represented as a single permutation matrix, P , composed of variable-dependent submatrices of either size n -by- m or size n -by- m' , where n is the number of neutronic nodes, m is the number of T/H zones, and m' is the number of heat structure components. In practice, a different submatrix is employed for each of the variables passed from the T/H code to allow for separate averaging techniques. The submatrices for the fuel temperatures, which correspond to heat structure components, have dimension n -by- m' , and the submatrices for the zone-wise T/H property data have dimension n -by- m . In an effort to simplify the input requirements, the user is given the option to assign an input permutation submatrix to multiple state variables. In addition, the mapping function between neutronic nodes and both heat structure components and T/H zones is represented as a single permutation matrix, P' . This permutation matrix is composed of submatrices of size m' -by- n for mapping powers (i.e., fission, decay, and gamma) to components, and submatrices of size m -by- n , which are used to map a fraction of the node-wise powers directly to the T/H zones.

As mentioned earlier, these matrices are processed from user input in the code-specific T/H or NDM routines. An error checking module is provided for the input permutation matrices; however, the primary responsibility is on the user to ensure their consistency with the input T/H, neutronic, and heat structure nodalization. Details regarding the error checking module will be discussed in Section 6.1.5. Note that in future versions, the user will not provide these inputs.

The use of the permutation matrix, P , in mapping zone-wise and component-wise T/H variables to neutronic nodes is given by

$$P x^{HS+T/H} = x^{Neut.} ; x^{HS+T/H} \in \mathcal{R}^{im'+jm} ; x^{Neut.} \in \mathcal{R}^{(i+j)n} \quad (6.1.1)$$

where the dimension of the vector, $x^{HS+T/H}$, is based on i component-wise variables and j zone-wise variables, and the dimension of $x^{Neut.}$ is based on $i+j$ total variables. Conversely, the use of the permutation matrix, P' , in mapping neutronic nodal powers to corresponding heat structure components and T/H zones is given by

$$P' x^{Neut.} = x^{HS+T/H} ; x^{HS+T/H} \in \mathcal{R}^{im'+jm} ; x^{Neut.} \in \mathcal{R}^{(i+j)n} \quad (6.1.2)$$

where $x^{HS+T/H}$ consists of i component-wise powers for the heat structures, and j powers for the fractional deposition in T/H zones, which is used to account for direct heating to the coolant.

The form of these permutation matrices is easily demonstrated with the simple T/H and neutronic planar nodalizations shown in Fig. 6.1.4.

The corresponding permutation for a single T/H variable would be

$$x^{Neut} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} x^{T/H} ; \quad \begin{matrix} x^{Neut} \in \mathbb{R}^{16} \\ x^{T/H} \in \mathbb{R}^4 \end{matrix} \quad (6.1.3)$$

This example assumes that the boundaries of the T/H zones and neutronic nodes are congruent, which is the case for most practical applications. However, when the boundaries are not congruent, resulting in overlapping T/H and neutronic regions, the user is responsible (in this version) for incorporating appropriate weighting factors (e.g., volume fractions) into the permutation matrix. The general form of the permutation matrix is now given as

$$P_i = \sum_{k(i) \in j} w_{k(i)}^i e_{k(i)} ; \quad e_{k(i)} \in \mathbb{R}^{(l,j)} ; \quad \text{for } i=1 \dots n \quad (6.1.4)$$

where $k(i)$ designates the T/H zone(s) belonging to neutronic node i , and $e_{k(i)}$ is a row vector with 1 in the $k(i)$ -th position, and zeros everywhere else. The dimension represented by j corresponds to either m or m' , depending on the variable-dependent submatrix. The weighting factor, $w_{k(i)}^i$, is a scalar that represents the weighting of the $k(i)$ -th T/H zone on the i -th neutronic node. The restriction on the weighting factor, is given as

$$\sum_{k(i) \in m} w_{k(i)}^i = 1.0 \quad (6.1.5)$$

It should be noted that for P , Equation 6.1.5 represents the summation of elements in row i , and for P' , Equation 6.1.5 represents the summation of elements in column i . An example that would require the use of weighting factors is shown in Fig. 6.1.5, which requires a planar mapping from cylindrical T/H coordinates to Cartesian neutronic coordinates.

By taking the weighting factors for this example to be simple area fractions, the permutation sub-matrix for a single T/H variable would be of the form

$$P = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0.5 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.71 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.29 \\ 0.71 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.29 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.5 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.71 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.29 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.5 & 0.5 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.5 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.71 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.29 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.71 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.29 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.5 & 0.5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.5 & 0.5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.71 & 0 & 0 & 0 & 0 & 0 & 0 & 0.29 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.5 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.71 & 0 & 0 & 0 & 0 & 0 & 0 & 0.29 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.71 & 0 & 0 & 0 & 0 & 0 & 0 & 0.29 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.5 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (6.1.6)$$

6.1.4 Calculational Control

The calculational control for the T/H, neutronic, and interface modules is accomplished with independent processes for each module. For this **multiple-process** approach, the coupling of the modules is achieved with the use of the Parallel Virtual Machine (PVM) package, which is distributed with the GI software. Specifically, the entry points into and out of the interface are managed with *sends* and *receives*, as shown in Fig. 6.1.6. This coupling strategy has the advantage of requiring minimal coding modifications to the T/H and neutronic modules. However, the user is required to manage an additional coding package for PVM.

In the calculational procedure, the T/H and neutronic modules independently perform input processing, which also includes calls to the code-specific data mapping routines to process the input required by the GI (e.g., the permutation matrices). Once the permutation matrices have been processed in the code-specific data mapping routines, these matrices are passed to the initialization unit of the GI routine to be stored for future use. Following the input processing, the neutronics code calculates the steady-state power distribution based on the initial T/H condition of the core. Using this power distribution, either the steady-state initialization or the transient calculation can be performed. The decision of whether to use point or spatial kinetics during the steady-state initialization is left to the user. However, to avoid a non-initialized core condition, the kinetics used in the transient calculation should be consistent with that employed for the steady-state initialization.

Once the input has been processed and each module has been initialized, the flow through the interface routines is the same regardless of whether a steady-state or transient calculation is being performed. As shown in Fig. 6.1.6, this flow proceeds in two directions, dictated by the PVM *sends* and *receives*.

Following the advancement of the heat conduction and hydrodynamic calculation, the zone-wise and component-wise variables stored in the T/H memory structure, denoted by (A) in Fig. 6.1.1, are packed into a single vector in the T/H Data Map (THDM) routine, converted to the units utilized by the interface, and sent to the GI module to be unpacked and stored in the memory structure denoted by (AB) . Once this data has been received, the GI module performs all data mapping from both T/H zones and heat structure components to neutronic nodes. As described in Section 6.1.3, this is accomplished with a single matrix-vector multiply. The vector of T/H data corresponding to neutronic nodes is then sent to the Neutronics Data Map (NDM) routine, where it is unpacked and stored in the neutronics memory structure, denoted by (B) . At this point, all necessary unit conversion needed for consistency in the neutronics module are performed.

Upon the completion of the neutronics calculation, the neutronic powers are packed into a single vector by the NDM routine, and all necessary unit conversion and/or normalization is performed. This vector of nodal neutronic data is then sent to the GI module, where it is mapped to T/H zones and heat structure components. The resulting vector of neutronic powers corresponding to zones and components is sent to the THDM routine, where it is unpacked and stored in the T/H memory structure. Any additional normalizations or conversions are performed at this point.

6.1.5 Time Step Control

Time step control for the coupled code is handled by the THDM routine, and is based on the control information passed through the interface from the T/H and neutronics modules. Because of the difference in characteristic times between the T/H and neutronics during severe transients (e.g., super-prompt critical events or events involving a phasic transition), the flexibility exists to allow either module to subcycle the time step. However, following a time step subcycling, it is necessary to re-synchronize the time steps, and this is controlled by the THDM routine. In addition, if the time step size sent to the neutronics is such that the change in the core condition is too large, logic control is provided for the neutronics code to reject the time step data and request a new T/H calculation at a finer time step size. As an example, this logic control is depicted in Fig. 6.1.7, where the time step size used by the T/H module is too large (1), and the kinetics sends logic information back to the T/H code to re-perform the calculation at time step $t^{n+1/2}$ (2).

6.1.6 Programming Standard for Interface Routines

The design of the GI routine, which adheres to the FORTRAN-90 standard, is modular in nature and maintains portability across all computer platforms. In addition, an error-checking module is included to help ensure that the input permutation matrices are consistent with specified requirements. Specifically, the dimension of the input matrices is compared to the dimension of the input vectors, the restriction given in Equation 6.1.5 is checked, and the elements of the matrices and vectors are compared against pre-determined upper and lower bounds. This error routine does not verify that the relation between node numbers and zone/component numbers is consistent with the models input to the T/H and neutronics codes.

6.1.7 Interface Summary

The coupling of T/H and neutronics codes described here is accomplished through the use of an immutable GI unit. To accommodate this design, it is necessary to utilize code-specific data mapping routines for both the T/H and neutronics codes. This allows the two modules to be coupled with only minor programming modifications to each.

The interface design employs a basic internal coupling strategy, and the calculational control is explicit in nature. However, the design is flexible enough to accommodate more sophisticated thermal-hydraulic/neutronic coupling strategies (e.g., implicit or semi-implicit). In addition, this coupled code will have the capability to accurately predict a wide range of transient scenarios, including BWR stability problems.

REFERENCES

- 6.1.1 Saad, Y., *Numerical Methods for Large Eigenvalue Problems*, Manchester University Press, Manchester, UK., pp. 40–41, 1992.

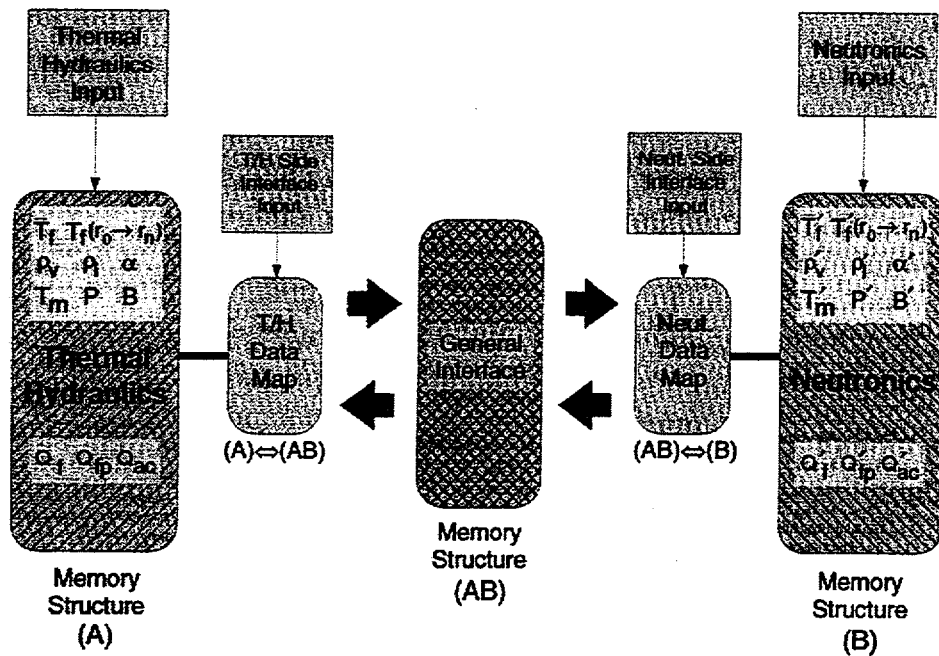


Figure 6.1.1 Diagram of Interface Implementation

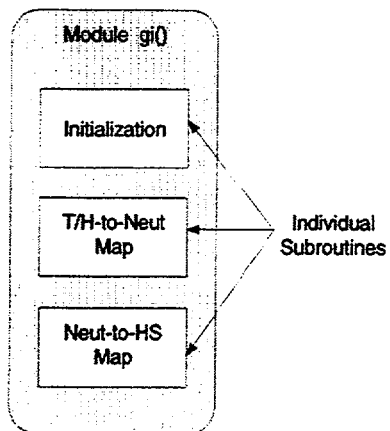


Figure 6.1.2 Structure of General Interface Module



Figure 6.1.3 Auxiliary Data Structure

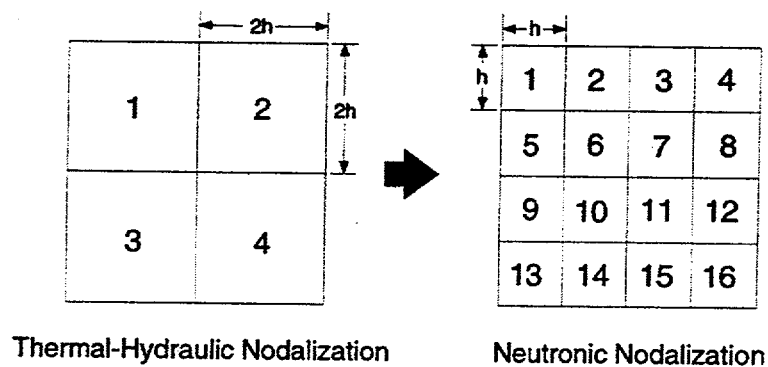


Figure 6.1.4 Cartesian Thermal-Hydraulic to Neutronic Mapping

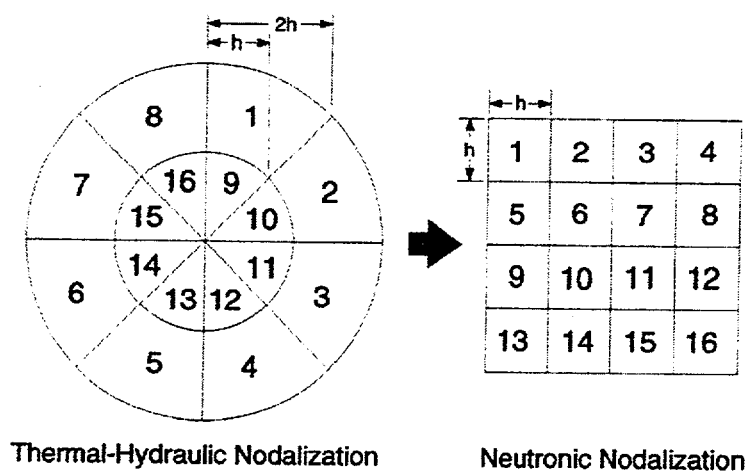


Figure 6.1.5 Cylindrical to Cartesian Mapping

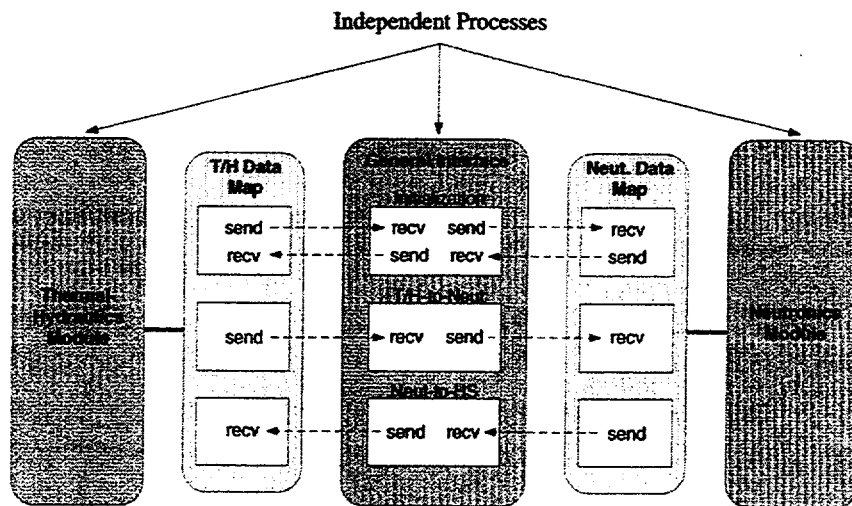


Figure 6.1.6 Coupling with PVM

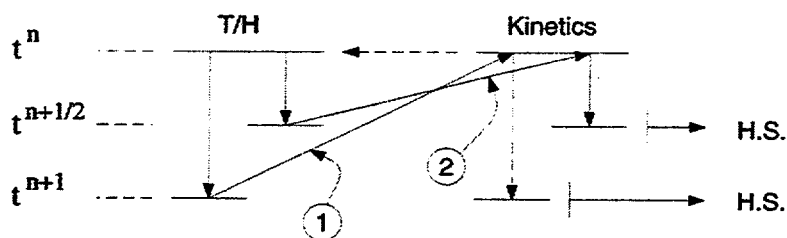


Figure 6.1.7 Example of Kinetics Controlling Time Step

6.2 Requirements

This section discusses the functional requirements for the General Interface (GI). These requirements are separated into the following categories:

- A. Requirements for correct interface initialization
- B. Requirements for thermal-hydraulics to neutronics mapping
- C. Requirements for neutronics to thermal-hydraulics mapping
- D. Requirements for error checking
- E. Requirements for termination of calculations

A test plan is designed to ensure that these requirements are met. Meeting these requirements ensures that the linkage between TRAC-M and the PARCS code is correctly implemented. These requirements and the associated test plan and acceptance criteria are as follows:

A. Interface initialization requirements:

- 1. The data structure containing T/H control information will be received from the THDM routine, and sent to the NDM routine.
- 2. The data structure containing neutronic control information will be received from the NDM routine and sent to the THDM routine.
- 3. Both floating point permutation matrices will be received from the THDM routine, and stored in the memory of the GI.
- 4. Both floating point permutation matrices will be received from the NDM routine and stored in the memory of the GI.

The determination to perform Requirement A.3 or A.4 will be based on the first value of the logical control buffers sent from the T/H and neutronic routines. Specifically, a value of TRUE in the seventh location of the T/H logical buffer will result in Requirement A.3 being performed. Conversely, a value of TRUE in the seventh location of the neutronic logical buffer will result in Requirement A.4 being performed.

B. Thermal-hydraulics to neutronics mapping requirements:

- 1. A vector will be received from the T/H routine containing all of the space-dependent T/H data.
- 2. The matrix-vector multiplication will be performed.

3. The resulting vector will be sent to the NDM routine.
4. The control data will be received from the T/H routine and transmitted to the neutronics routine.

C. Neutronics to thermal-hydraulics mapping requirements:

1. A vector will be received from the neutronics routine containing all of the space-dependent neutronics data.
2. The matrix-vector multiplication will be performed.
3. The resulting vector will be sent to the THDM routine.
4. The control data will be received from the neutronics routine and transmitted to the T/H routine.

D. Error-checking requirements for each of the previous requirements:

Test cases will be perturbed such that the indicated parameter triggers the appropriate message.

Initialization

1. The indication of which process (thermal-hydraulic or neutronic) is sending the permutation matrices is inconsistent [Fatal].
2. The matrix elements are outside the specified range [Fatal].
3. The weighting factors in the permutation matrix are inaccurate [Fatal].

Thermal-Hydraulics to Neutronics Mapping

1. The matrix and vector dimensions are inconsistent [Fatal].
2. Negative elements exist in the vector [Fatal].

Neutronics to Thermal-Hydraulics Mapping

1. The matrix and vector dimensions are inconsistent [Fatal].
2. Negative elements exist in the vector [Fatal].

E. Termination of the General Interface based on the following conditions:

1. The GI terminates normally as a result of logical information sent from both the T/H and neutronics code.

2. The GI terminates based on a logical fault signal sent from either the T/H or neutronics code.
3. The GI terminates based on an error detected in the General Interface.

Note: PVM error checking will not be tested for this Quality Assurance Test Plan (QATP).

The test cases are chosen to evaluate the full scope of functionality provided by the GI, as described in Section 6.2.1, along with the corresponding permutation matrices in Section 6.2.2 and input and output vectors in Sections 6.2.3 and 6.2.4. In addition, the transfer of the control buffers through the GI will be tested using the data shown in Sections 6.2.5 and 6.2.6.

In order to meet the error-checking requirements, permutations of the first test case, described in Sections 6.2.1 through 6.2.5, will be performed as follows:

Initialization

- 1a. The seventh element in the initial logical buffer for the neutronic process will be set to "F" (False).
- 1b. The seventh element in the initial logical buffer for the thermal-hydraulic process will be set to "T" (True).
- 2a. The first non-zero element in the thermal-hydraulic to neutronic matrix will be assigned a value of 2.0.
- 2b. The first non-zero element in the neutronic to thermal-hydraulic matrix will be assigned a value of 2.0.
- 2c. The first non-zero element in the thermal-hydraulic to neutronic matrix will be assigned a value of -1.0.
- 2d. The first non-zero element in the neutronic to thermal-hydraulic matrix will be assigned a value of -1.0.
3. The parametrics performed in 2a through 2d will also trigger error in weighting factors.

Thermal-Hydraulics to Neutronics Mapping

- 1a. The dimension of the unpermuted vector of thermal-hydraulic and heat structure data will be decreased by 1 during the first time-dependent calculation.
- 1b. The dimension of the unpermuted vector of thermal-hydraulic and heat structure data will be decreased by 1 during the second time-dependent calculation. This test is performed because the GI only allocates space for the unpermuted vector once (at the first time step). Thus, if the dimension of the vector sent from the T/H process is different on subsequent time steps than the size of the previously allocated vector, an error should occur.

2. The first element in the unpermuted vector will be assigned a value of -1.0.

Neutronics to Thermal-Hydraulics Mapping

- 1a. The dimension of the unpermuted vector of neutronic data will be decreased by 1 during the first time-dependent calculation.
- 1b. The dimension of the unpermuted vector of neutronic data will be decreased by 1 during the second time-dependent calculation. The reason for this test is as described above.
2. The first element in both the unpermuted and permuted vectors will be assigned a value of -1.0.

Detection of faults originating in the thermal-hydraulics and neutronics codes will also be tested by assigning the first, second, and third logicals in the initial and time-dependent control buffers a value of "T". Finally, normal process termination will be tested based on a value of "T" for the sixth logical in the second time-dependent control buffer of both the thermal-hydraulics and neutronics code.

6.2.1 Description of Test Cases

The test matrix shown in Table 6.2.1, is used to test the functionality of the GI, where the number and type of independent solution variables (corresponding to the submatrices and subvectors) meet Requirements A through E. These cases will employ two different geometries for the T/H zones (i.e., Cartesian and cylindrical) in order to exercise alternative mapping matrix structures.

Case 1

This is the base test case, as illustrated in Fig. 6.2.1, where the four heat structures are tied to four parallel T/H pipes. A one-to-one mapping exists between the T/H zones, heat structure components, and neutronic nodes.

Case 2

This case represents a finer neutronic nodalization compared to that for the T/H zones and heat structure components. In addition, for the single heat structure, 17 radial nodes in the fuel pin are utilized to test the case of a well-described temperature profile. Note that, for this case, the T/H code will construct and send the permutation matrices to the GI.

Case 3

This case represents a finer T/H and heat structure nodalization compared to that for the neutronics. Here, the 16 parallel pipes are arranged in a Cartesian 4x4 grid, which corresponds to a 2x2 grid for the neutronic nodes, such which each neutronic node contains four T/H zones and four heat structure components.

Case 4

This case tests a mapping from a cylindrical T/H nodalization to a Cartesian neutronic nodalization. Here, 16 parallel T/H pipes, each with a single heat structure component, are arranged in a cylindrical grid, and mapped to 16 neutronic nodes arranged in a 4x4 grid, as demonstrated in Fig. 6.1.5. This mapping is not one-to-one.

Case 5

This case tests the scenario in which there are multiple heat structure components per T/H zone. A one-to-one correspondence exists between the heat structures and the neutronic nodes, which are arranged in a 4x4 grid. The T/H zones are arranged in a 2x2 grid, and each T/H zone contains four heat structures. The mapping between T/H zones and neutronic nodes is demonstrated in Fig. 6.1.4.

Case 6

This case is used to test 3-D mapping. Each T/H pipe, with a corresponding heat structure, is mapped to eight neutronic nodes (i.e., 2x2x2 layout).

Case 7

This annulus test case is set up to test the situation where a single heat structure is mapped to several T/H zones, as illustrated in Fig. 6.2.2.

6.2.2 Permutation Submatrices in Coordinate Storage Format

The submatrices for the seven cases listed in Section 6.2.1 are shown below. For each test case, the number and type of variables corresponding to each submatrix is as follows:

- | | | | |
|-----|---------------------------------|----|------------------------------------------------------------|
| (A) | Thermal-Hydraulic to Neutronic: | 5; | Mod. Temp., Liq. Dens., Vap. Dens., Void Fr., Boron Conc. |
| (B) | Heat Structure to Neutronic: | 3; | Avg. Fuel Temp., Fuel Centerline Temp., Fuel Surface Temp. |
| (C) | Neutronic to Thermal-Hydraulic: | 3; | Fission Power, FP Decay Power, Actinide Decay Power |
| (D) | Neutronic to Heat Structure: | 3; | Fission Power, FP Decay Power, Actinide Decay Power |

The entire permutation matrix consists of diagonal blocks corresponding to each submatrix. The order of the submatrices for each mapping direction is as follows:

Thermal-Hydraulic / Heat Structure to Neutronic:

diag [(A₁), (A₂), (A₃), (A₄), (A₅), (B₁), (B₂), (B₃)]

Neutronic to Thermal-Hydraulic / Heat Structure:

diag [(C₁), (C₂), (C₃), (D₁), (D₂), (D₃)]

It should be noted that the ROW and COL vectors shown for the test cases correspond to the local submatrices. The indices for the global permutation matrices required by the GI should be computed based on the described structure.

Case 1:

Thermal-Hydraulic to Neutronic

VAL = [1.0 1.0 1.0 1.0]
 ROW = [1 2 3 4]
 COL = [1 2 3 4]

Heat Structure to Neutronic

VAL = [1.0 1.0 1.0 1.0]
 ROW = [1 2 3 4]
 COL = [1 2 3 4]

Neutronic to Thermal-Hydraulic

VAL = [1.0 1.0 1.0 1.0]
 ROW = [1 2 3 4]
 COL = [1 2 3 4]

Neutronic to Heat Structure

VAL = [1.0 1.0 1.0 1.0]
 ROW = [1 2 3 4]
 COL = [1 2 3 4]

Case 2:

Thermal-Hydraulic to Neutronic

VAL = [1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0]
 ROW = [1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16]
 COL = [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]

Heat Structure to Neutronic

VAL = [1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0]
 ROW = [1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16]
 COL = [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]

Neutronic to Thermal-Hydraulic

VAL = [1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0]
 ROW = [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]
 COL = [1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16]

Neutronic to Heat Structure

VAL = [1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0]
 ROW = [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]
 COL = [1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16]

Case 3:

Thermal-Hydraulic to Neutronic

VAL = [.25 .25 .25 .25 .25 .25 .25 .25 .25 .25 .25 .25 .25 .25 .25 .25]
 ROW = [1 1 2 2 1 1 2 2 3 3 4 4 3 3 4 4]
 COL = [1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16]

Heat Structure to Neutronic

VAL = [.25 .25 .25 .25 .25 .25 .25 .25 .25 .25 .25 .25 .25 .25 .25 .25]
 ROW = [1 1 2 2 1 1 2 2 3 3 4 4 3 3 4 4]
 COL = [1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16]

Neutronic to Thermal-Hydraulic

VAL = [.25 .25 .25 .25 .25 .25 .25 .25 .25 .25 .25 .25 .25 .25 .25 .25]
ROW = [1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16]
COL = [1 1 2 2 1 1 2 2 3 3 4 4 3 3 4 4]

Neutronic to Heat Structure

VAL = [.25 .25 .25 .25 .25 .25 .25 .25 .25 .25 .25 .25 .25 .25 .25 .25]
ROW = [1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16]
COL = [1 1 2 2 1 1 2 2 3 3 4 4 3 3 4 4]

Case 4:

Thermal-Hydraulic to Neutronic

VAL = [.50 .50 .71 .29 .71 .29 .50 .50 .71 .29 .50 .50 .50 .50 .71 .29 ,
.71 .29 .50 .50 .50 .50 .71 .29 .50 .50 .71 .29 .71 .29 .50 .50]
ROW = [1 1 2 2 3 3 4 4 5 5 6 6 7 7 8 8 ,
9 9 10 10 11 11 12 12 13 13 14 14 15 15 16 16]
COL = [7 8 8 16 1 9 1 2 7 15 15 16 9 10 2 10 ,
6 14 13 14 11 12 3 11 5 6 5 13 4 12 3 4]

Heat Structure to Neutronic

VAL = [.50 .50 .71 .29 .71 .29 .50 .50 .71 .29 .50 .50 .50 .50 .71 .29 ,
.71 .29 .50 .50 .50 .50 .71 .29 .50 .50 .71 .29 .71 .29 .50 .50]
ROW = [1 1 2 2 3 3 4 4 5 5 6 6 7 7 8 8 ,
9 9 10 10 11 11 12 12 13 13 14 14 15 15 16 16]
COL = [7 8 8 16 1 9 1 2 7 15 15 16 9 10 2 10 ,
6 14 13 14 11 12 3 11 5 6 5 13 4 12 3 4]

Neutronic to Thermal-Hydraulic

VAL = [.50 .50 .71 .29 .71 .29 .50 .50 .71 .29 .50 .50 .50 .50 .71 .29 ,
.71 .29 .50 .50 .50 .50 .71 .29 .50 .50 .71 .29 .71 .29 .50 .50]
ROW = [7 8 8 16 1 9 1 2 7 15 15 16 9 10 2 10 ,
6 14 13 14 11 12 3 11 5 6 5 13 4 12 3 4]
COL = [1 1 2 2 3 3 4 4 5 5 6 6 7 7 8 8 ,
9 9 10 10 11 11 12 12 13 13 14 14 15 15 16 16]

Neutronic to Heat Structure

VAL = [.50 .50 .71 .29 .71 .29 .50 .50 .71 .29 .50 .50 .50 .50 .71 .29 ,
.71 .29 .50 .50 .50 .50 .71 .29 .50 .50 .71 .29 .71 .29 .50 .50]
ROW = [7 8 8 16 1 9 1 2 7 15 15 16 9 10 2 10 ,
6 14 13 14 11 12 3 11 5 6 5 13 4 12 3 4]
COL = [1 1 2 2 3 3 4 4 5 5 6 6 7 7 8 8 ,
9 9 10 10 11 11 12 12 13 13 14 14 15 15 16 16]

Case 5:

Thermal-Hydraulic to Neutronic

VAL = [1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0]
ROW = [1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16]
COL = [1 1 2 2 1 1 2 2 3 3 4 4 3 3 4 4]

Heat Structure to Neutronic

VAL = [1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0]
ROW = [1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16]
COL = [1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16]

Neutronic to Thermal-Hydraulic

VAL = [1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0]
ROW = [1 1 2 2 1 1 2 2 3 3 4 4 3 3 4 4]
COL = [1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16]

Neutronic to Heat Structure

VAL = [1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0]
ROW = [1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16]
COL = [1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16]

Case 6:

Thermal-Hydraulic to Neutronic

VAL = [1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 ,
1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0]
ROW = [1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 ,
17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32]
COL = [1 1 2 2 1 1 2 2 3 3 4 4 3 3 4 4 ,
1 1 2 2 1 1 2 2 3 3 4 4 3 3 4 4]

Heat Structure to Neutronic

VAL = [1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 ,
1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0]
ROW = [1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 ,
17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32]
COL = [1 1 2 2 1 1 2 2 3 3 4 4 3 3 4 4 ,
1 1 2 2 1 1 2 2 3 3 4 4 3 3 4 4]

Neutronic to Thermal-Hydraulic

VAL = [1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 ,
1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0]
ROW = [1 1 2 2 1 1 2 2 3 3 4 4 3 3 4 4 ,
1 1 2 2 1 1 2 2 3 3 4 4 3 3 4 4]
COL = [1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 ,
17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32]

Neutronic to Heat Structure

VAL = [1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 ,
1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0]
ROW = [1 1 2 2 1 1 2 2 3 3 4 4 3 3 4 4 ,
1 1 2 2 1 1 2 2 3 3 4 4 3 3 4 4]
COL = [1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 ,
17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32]

Case 7:

Thermal-Hydraulic to Neutronic

VAL = [f (1-f)] ; f=0.23
ROW = [1 1]
COL = [1 2]

Heat Structure to Neutronic

VAL = [1]
ROW = [1]
COL = [1]

Neutronic to Thermal-Hydraulic

VAL = [f (1-f)] ; f=0.23
ROW = [1 2]
COL = [1 1]

Neutronic to Heat Structure

VAL = [1]
ROW = [1]
COL = [1]

6.2.3 Input Vectors After Permutation

The number and types of variables (subvectors) used for each test case were shown in the description of the permutation matrices. The order of the variables in the entire input vector is consistent with that for the matrix. The elements of the input vectors for each mapping direction are shown below for each test case, where *vec_{th}* is the unpermuted vector of T/H and heat structure data, and *vec_{cn}* is the unpermuted vector of neutronic data. An example of the order of subvectors is shown in Case 1, and applies to Cases 2 through 7. It should be noted that two time-dependent calculations are being performed, and the input vectors shown below are the same during both time steps.

Case 1:

Thermal-Hydraulic / Heat Structure to Neutronic

(5 T/H vars, 4 T/H zones, 3 HS vars, 4 HS components: 32 elements)

	Mod. Temp.	Liq. Dens.	Vap. Dens.	Void Fr.	
<i>vec_{th}</i> ^T =	[1.0 2.0 3.0 4.0 3.0 3.0 3.0 3.0 1.0 1.0 1.0 1.0 0.5 0.5 1.0 0.0 ,				
	Boron Conc.	Avg. Fuel Temp.	Center Temp.	Surface Temp.	
	4.0 2.0 4.0 3.0 5.0 5.0 5.0 5.0 6.0 6.0 6.0 6.0 4.0 4.0 4.0 4.0]				

Neutronic to Thermal-Hydraulic / Heat Structure

(3 T/H vars, 4 Neut. nodes, 3 HS vars, 4 Neut. nodes: 24 elements)

	<-- Mapped to Coolant -->												
	Fission Power	FP Power	Actinide Power										
<i>vec_{cn}</i> ^T =	[3.0 3.0 3.0 3.0 4.0 4.0 4.0 4.0 5.0 5.0 5.0 5.0 ,												
	<-- Mapped to Fuel -->												
	Fission Power	FP Power	Actinide Power										
	6.0 6.0 6.0 6.0 7.0 7.0 7.0 7.0 8.0 8.0 8.0 8.0]												

Case 2:

Thermal-Hydraulic / Heat Structure to Neutronic

(5 T/H vars, 1 T/H zone, 3 HS vars, 1 HS component: 8 elements)

vec_{th}^T = [1.0 3.0 1.0 0.5 4.0 5.0 6.0 4.0]

Neutronic to Thermal-Hydraulic / Heat Structure

(3 T/H vars, 16 Neut. nodes, 3 HS vars, 16 Neut. nodes: 96 elements)

vecn^T = [3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 ,
4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 ,
5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 ,
6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 ,
7.0 7.0 7.0 7.0 7.0 7.0 7.0 7.0 7.0 7.0 7.0 7.0 7.0 7.0 7.0 7.0 ,
8.0 8.0 8.0 8.0 8.0 8.0 8.0 8.0 8.0 8.0 8.0 8.0 8.0 8.0 8.0 8.0]

Case 3:

Thermal-Hydraulic / Heat Structure to Neutronic

(5 T/H vars, 16 T/H zones, 3 HS vars, 16 HS components: 128 elements)

vecth^T = [1.0 2.0 3.0 4.0 1.0 2.0 3.0 4.0 1.0 2.0 3.0 4.0 1.0 2.0 3.0 4.0 ,
3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 ,
1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 ,
0.5 0.5 1.0 0.0 0.5 0.5 1.0 0.0 0.5 0.5 1.0 0.0 0.5 0.5 1.0 0.0 ,
4.0 2.0 4.0 3.0 4.0 2.0 4.0 3.0 4.0 2.0 4.0 3.0 4.0 2.0 4.0 3.0 ,
5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 ,
6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 ,
4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0]

Neutronic to Thermal-Hydraulic / Heat Structure

(3 T/H vars, 4 Neut. nodes, 3 HS vars, 4 Neut. nodes: 24 elements)

vecn^T = [3.0 3.0 3.0 3.0 4.0 4.0 4.0 4.0 5.0 5.0 5.0 5.0 6.0 6.0 6.0 6.0 ,
7.0 7.0 7.0 7.0 8.0 8.0 8.0 8.0]

Case 4:

Thermal-Hydraulic / Heat Structure to Neutronic

(5 T/H vars, 16 T/H zones, 3 HS vars, 16 HS components: 128 elements)

vecth^T = [1.0 2.0 3.0 4.0 1.0 2.0 3.0 4.0 1.0 2.0 3.0 4.0 1.0 2.0 3.0 4.0 ,
3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 ,
1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 ,
0.5 0.5 1.0 0.0 0.5 0.5 1.0 0.0 0.5 0.5 1.0 0.0 0.5 0.5 1.0 0.0 ,
4.0 2.0 4.0 3.0 4.0 2.0 4.0 3.0 4.0 2.0 4.0 3.0 4.0 2.0 4.0 3.0 ,
5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 ,
6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 ,
4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0]

Neutronic to Thermal-Hydraulic / Heat Structure

(3 T/H vars, 16 Neut. nodes, 3 HS vars, 16 Neut. nodes: 96 elements)

```
vecnT = [3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 ,
          4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 ,
          5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 ,
          6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 ,
          7.0 7.0 7.0 7.0 7.0 7.0 7.0 7.0 7.0 7.0 7.0 7.0 7.0 7.0 7.0 7.0 ,
          8.0 8.0 8.0 8.0 8.0 8.0 8.0 8.0 8.0 8.0 8.0 8.0 8.0 8.0 8.0 8.0]
```

Case 5:

Thermal-Hydraulic / Heat Structure to Neutronic

(3 T/H vars, 4 T/H zones, 3 HS vars, 16 HS components: 68 elements)

```
vecthT = [1.0 2.0 3.0 4.0 3.0 3.0 3.0 3.0 1.0 1.0 1.0 1.0 0.5 0.5 1.0 0.0 ,
          4.0 2.0 4.0 3.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 ,
          5.0 5.0 5.0 5.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 ,
          6.0 6.0 6.0 6.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 ,
          4.0 4.0 4.0 4.0]
```

Neutronic to Thermal-Hydraulic / Heat Structure

(3 T/H vars, 16 Neut. nodes, 3 HS vars, 16 Neut. nodes: 96 elements)

```
vecnT = [3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 ,
          4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 ,
          5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 ,
          6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 ,
          7.0 7.0 7.0 7.0 7.0 7.0 7.0 7.0 7.0 7.0 7.0 7.0 7.0 7.0 7.0 7.0 ,
          8.0 8.0 8.0 8.0 8.0 8.0 8.0 8.0 8.0 8.0 8.0 8.0 8.0 8.0 8.0 8.0]
```

Case 6:

Thermal-Hydraulic / Heat Structure to Neutronic

(5 T/H vars, 4 T/H zones, 3 HS vars, 4 HS components: 32 elements)

```
vecthT = [1.0 2.0 3.0 4.0 3.0 3.0 3.0 3.0 1.0 1.0 1.0 1.0 0.5 0.5 1.0 0.0 ,
          4.0 2.0 4.0 3.0 5.0 5.0 5.0 5.0 6.0 6.0 6.0 6.0 4.0 4.0 4.0 4.0]
```

Neutronic to Thermal-Hydraulic / Heat Structure

(3 T/H vars, 32 Neut. nodes, 3 HS vars, 32 Neut. nodes: 192 elements)

```

vecnT = [3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 ,
          3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 ,
          4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 ,
          4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 ,
          5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 ,
          5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 ,
          6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 ,
          6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 ,
          7.0 7.0 7.0 7.0 7.0 7.0 7.0 7.0 7.0 7.0 7.0 7.0 7.0 7.0 7.0 7.0 ,
          7.0 7.0 7.0 7.0 7.0 7.0 7.0 7.0 7.0 7.0 7.0 7.0 7.0 7.0 7.0 7.0 ,
          8.0 8.0 8.0 8.0 8.0 8.0 8.0 8.0 8.0 8.0 8.0 8.0 8.0 8.0 8.0 8.0 ,
          8.0 8.0 8.0 8.0 8.0 8.0 8.0 8.0 8.0 8.0 8.0 8.0 8.0 8.0 8.0 8.0]

```

Case 7:

Thermal-Hydraulic / Heat Structure to Neutronic
(5 T/H vars, 2 T/H zones, 3 HS vars, 1 HS component: 13 elements)

```

vecthT = [1.0 2.0 3.0 3.0 1.0 1.0 0.5 1.0 4.0 2.0 5.0 6.0 4.0]

```

Neutronic to Thermal-Hydraulic / Heat Structure
(3 T/H vars, 1 Neut. node, 3 HS vars, 1 Neut. node: 6 elements)

```

vecnT = [3.0 4.0 5.0 6.0 7.0 8.0]

```

6.2.4 Output Vectors After Permutation

The output vector resulting from the permutation is shown below for each test case, where vecthp is the permuted vector of T/H and heat structure data, and vecnp is the permuted vector of neutronic data. These vectors were obtained by performing the matrix-vector multiply electronically with the Matrix Laboratory (MATLAB). Visual verification of the input vectors, input matrices, and output vectors was performed.

Case 1:

Thermal-Hydraulic / Heat Structure to Neutronic
(5 T/H vars, 4 Neut. nodes, 3 HS vars, 4 Neut. nodes: 32 elements)

```

vecthpT = [1.0 2.0 3.0 4.0 3.0 3.0 3.0 3.0 1.0 1.0 1.0 1.0 0.5 0.5 1.0 0.0 ,
           4.0 2.0 4.0 3.0 5.0 5.0 5.0 5.0 6.0 6.0 6.0 6.0 4.0 4.0 4.0 4.0]

```

Neutronic to Thermal-Hydraulic / Heat Structure
(3 T/H vars, 4 T/H zones, 3 HS vars, 4 HS components: 24 elements)

```

vecnpT = [3.0 3.0 3.0 3.0 4.0 4.0 4.0 4.0 5.0 5.0 5.0 5.0 6.0 6.0 6.0 6.0 ,
           7.0 7.0 7.0 7.0 8.0 8.0 8.0 8.0]

```

Case 2:

Thermal-Hydraulic / Heat Structure to Neutronic

(5 T/H vars, 16 Neut. nodes, 3 HS vars, 16 Neut. nodes: 128 elements)

$\text{vecthp}^T =$ [1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 ,
3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 ,
1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 ,
0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 ,
4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 ,
5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 ,
6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 ,
4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0]

Neutronic to Thermal-Hydraulic / Heat Structure

(3 T/H vars, 1 T/H zone, 3 HS vars, 1 HS component: 6 elements)

$\text{vecnp}^T =$ [48.0 64.0 80.0 96.0 112.0 128.0]

Case 3:

Thermal-Hydraulic / Heat Structure to Neutronic

(5 T/H vars, 4 Neut. nodes, 3 HS vars, 4 Neut. nodes: 32 elements)

$\text{vecthp}^T =$ [1.5 3.5 1.5 3.5 3.0 3.0 3.0 3.0 1.0 1.0 1.0 1.0 0.5 0.5 0.5 0.5 ,
3.0 3.5 3.0 3.5 5.0 5.0 5.0 5.0 6.0 6.0 6.0 6.0 4.0 4.0 4.0 4.0]

Neutronic to Thermal-Hydraulic / Heat Structure

(3 T/H vars, 16 T/H zones, 3 HS vars, 16 HS components: 96 elements)

$\text{vecnp}^T =$ [0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 ,
0.75 0.75 0.75 0.75 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 ,
1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.25 1.25 1.25 1.25 ,
1.25 1.25 1.25 1.25 1.25 1.25 1.25 1.25 1.25 1.25 1.25 1.25 ,
1.5 1.5 1.5 1.5 1.5 1.5 1.5 1.5 1.5 1.5 1.5 1.5 ,
1.5 1.5 1.5 1.5 1.75 1.75 1.75 1.75 1.75 1.75 1.75 1.75 ,
1.75 1.75 1.75 1.75 1.75 1.75 1.75 1.75 2.0 2.0 2.0 2.0 ,
2.0 2.0 2.0 2.0 2.0 2.0 2.0 2.0 2.0 2.0 2.0 2.0]

Case 4:

Thermal-Hydraulic / Heat Structure to Neutronic

(5 T/H vars, 16 Neut. nodes, 3 HS vars, 16 Neut. nodes: 128 elements)

```
vec thpT = [3.5 4.0 1.0 1.5 3.0 3.5 1.5 2.0 2.0 1.5 3.5 3.0 1.5 1.0 4.0 3.5 ,
             3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 ,
             1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 ,
             0.5 0.0 0.5 0.5 1.0 0.5 0.5 0.5 0.5 0.5 0.5 1.0 0.5 0.5 0.0 0.5 ,
             3.5 3.0 4.0 3.0 4.0 3.5 3.0 2.0 2.0 3.0 3.5 4.0 3.0 4.0 3.0 3.5 ,
             5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 ,
             6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 ,
             4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0]
```

Neutronic to Thermal-Hydraulic / Heat Structure

(3 T/H vars, 16 T/H zones, 3 HS vars, 16 HS components: 96 elements)

```
vec npT = [3.63 3.63 3.63 3.63 3.63 3.63 3.63 3.63 2.37 2.37 2.37 2.37 ,
            2.37 2.37 2.37 2.37 4.84 4.84 4.84 4.84 4.84 4.84 4.84 4.84 ,
            3.16 3.16 3.16 3.16 3.16 3.16 3.16 3.16 6.05 6.05 6.05 6.05 ,
            6.05 6.05 6.05 6.05 3.95 3.95 3.95 3.95 3.95 3.95 3.95 3.95 ,
            7.26 7.26 7.26 7.26 7.26 7.26 7.26 7.26 4.74 4.74 4.74 4.74 ,
            4.74 4.74 4.74 4.74 8.47 8.47 8.47 8.47 8.47 8.47 8.47 8.47 ,
            5.53 5.53 5.53 5.53 5.53 5.53 5.53 5.53 9.68 9.68 9.68 9.68 ,
            9.68 9.68 9.68 9.68 6.32 6.32 6.32 6.32 6.32 6.32 6.32 6.32]
```

Case 5:

Thermal-Hydraulic / Heat Structure to Neutronic

(5 T/H vars, 16 Neut. nodes, 3 HS vars, 16 Neut. nodes: 128 elements)

```
vec thpT = [1.0 1.0 2.0 2.0 1.0 1.0 2.0 2.0 3.0 3.0 4.0 4.0 3.0 3.0 4.0 4.0 ,
             3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 ,
             1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 ,
             0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 1.0 1.0 0.0 0.0 1.0 1.0 0.0 0.0 ,
             4.0 4.0 2.0 2.0 4.0 4.0 2.0 2.0 4.0 4.0 3.0 3.0 4.0 4.0 3.0 3.0 ,
             5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 ,
             6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 ,
             4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0]
```

Neutronic to Thermal-Hydraulic / Heat Structure

(3 T/H vars, 4 T/H zones, 3 HS vars, 16 HS components: 60 elements)

```
vec npT = [12.0 12.0 12.0 12.0 16.0 16.0 16.0 16.0 20.0 20.0 20.0 20.0 ,
            6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 ,
            6.0 6.0 6.0 6.0 7.0 7.0 7.0 7.0 7.0 7.0 7.0 7.0 ,
            7.0 7.0 7.0 7.0 7.0 7.0 7.0 7.0 8.0 8.0 8.0 8.0 ,
            8.0 8.0 8.0 8.0 8.0 8.0 8.0 8.0 8.0 8.0 8.0 8.0]
```

Case 6:

Thermal-Hydraulic / Heat Structure to Neutronic

(5 T/H vars, 32 Neut. nodes, 3 HS vars, 32 Neut. nodes: 256 elements)

```
vec thpT = [1.0 1.0 2.0 2.0 1.0 1.0 2.0 2.0 3.0 3.0 4.0 4.0 3.0 3.0 4.0 4.0 ,  
            1.0 1.0 2.0 2.0 1.0 1.0 2.0 2.0 3.0 3.0 4.0 4.0 3.0 3.0 4.0 4.0 ,  
            3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 ,  
            3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 ,  
            1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 ,  
            1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 ,  
            0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 1.0 1.0 0.0 0.0 1.0 1.0 0.0 0.0 ,  
            0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 1.0 1.0 0.0 0.0 1.0 1.0 0.0 0.0 ,  
            4.0 4.0 2.0 2.0 4.0 4.0 2.0 2.0 4.0 4.0 3.0 3.0 4.0 4.0 3.0 3.0 ,  
            4.0 4.0 2.0 2.0 4.0 4.0 2.0 2.0 4.0 4.0 3.0 3.0 4.0 4.0 3.0 3.0 ,  
            5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 ,  
            5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 ,  
            6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 ,  
            6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 ,  
            4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 ,  
            4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0]
```

Neutronic to Thermal-Hydraulic / Heat Structure

(3 T/H vars, 4 T/H zones, 3 HS vars, 4 HS components: 24 elements)

```
vec npT = [24.0 24.0 24.0 24.0 32.0 32.0 32.0 32.0 40.0 40.0 40.0 40.0 ,  
           48.0 48.0 48.0 48.0 56.0 56.0 56.0 56.0 64.0 64.0 64.0 64.0]
```

Case 7:

Thermal-Hydraulic / Heat Structure to Neutronic

(5 T/H vars, 1 Neut. node, 3 HS vars, 1 Neut. node: 8 elements)

```
vec thpT = [1.77 3.0 1.0 0.885 2.46 5.0 6.0 4.0]
```

Neutronic to Thermal-Hydraulic / Heat Structure

(3 T/H vars, 2 T/H zones, 3 HS vars, 1 HS components: 9 elements)

```
vec npT = [0.69 2.31 0.92 3.08 1.15 3.85 6.0 7.0 8.0]
```

6.2.5 Initial Control Buffers

The data used to test the transfer of the initial control buffers is shown below for each test case. All of the data shown represent “dummy” values, except the following:

- The first element in the logical buffer is used to communicate a calculation error in the respective code (neutronic or T/H).
- The second element in the logical buffer is used to communicate a data error in the code-specific data map routine.

- The third element in the logical buffer is used to communicate a PVM error in the code-specific data map routine.
- The fourth element in the logical buffer is used by the GI to communicate a data error to the respective code.
- The fifth element in the logical buffer is used by the GI to communicate a PVM error to the respective code.
- The sixth element in the logical buffer is used to communicate a normal calculation termination.
- The seventh element in the logical buffer tells the GI which code is sending the permutation matrices (i.e., the code sending the matrices will have a value of "T" in this location).

Case 1:

	<u>Neutronics Code</u>	<u>Thermal-Hydraulics Code</u>
char*6	["test01", "test02"]	["test03", "test04"]
logical	[F, F, F, F, F, F, T]	[F, F, F, F, F, F, F]
integer*2	[0, 2, 1]	[1, 0, 2]
integer*4	[888]	[999]
real*4	[1.0, 2.0, 3.0]	[4.0, 5.0, 6.0]
real*8	[10.0, 20.0, 30.0]	[40.0, 50.0, 60.0]

Case 2:

	<u>Neutronics Code</u>	<u>Thermal-Hydraulics Code</u>
char*6	["test01", "test02"]	["test03", "test04"]
logical	[F, F, F, F, F, F, F]	[F, F, F, F, F, F, T]
integer*2	[0, 2, 1]	[1, 0, 2]
integer*4	[888]	[999]
real*4	[1.0, 2.0, 3.0]	[4.0, 5.0, 6.0]
real*8	[10.0, 20.0, 30.0]	[40.0, 50.0, 60.0]

Case 3:

	<u>Neutronics Code</u>	<u>Thermal-Hydraulics Code</u>
char*6	["test01", "test02"]	["test03", "test04"]
logical	[F, F, F, F, F, F, T]	[F, F, F, F, F, F, F]
integer*2	[0, 2, 1]	[999]
integer*4	[888]	[16, 16]
real*4	[1.0, 2.0, 3.0]	[4.0, 5.0, 6.0]
real*8	[10.0, 20.0, 30.0]	[40.0, 50.0, 60.0]

Case 4:

	<u>Neutronics Code</u>	<u>Thermal-Hydraulics Code</u>
char*6	["test01", "test02"]	["test03", "test04"]
logical	[F, F, F, F, F, F, T]	[F, F, F, F, F, F, F]
integer*2	[0, 2, 1]	[1, 0, 2]
integer*4	[888]	[999]
real*4	[1.0, 2.0, 3.0]	[4.0, 5.0, 6.0]
real*8	[10.0, 20.0, 30.0]	[40.0, 50.0, 60.0]

Case 5:

	<u>Neutronics Code</u>	<u>Thermal-Hydraulics Code</u>
char*6	["test01", "test02"]	["test03", "test04"]
logical	[F, F, F, F, F, F, T]	[F, F, F, F, F, F, F]
integer*2	[0, 2, 1]	[1, 0, 2]
integer*4	[888]	[999]
real*4	[1.0, 2.0, 3.0]	[4.0, 5.0, 6.0]
real*8	[10.0, 20.0, 30.0]	[40.0, 50.0, 60.0]

Case 6:

	<u>Neutronics Code</u>	<u>Thermal-Hydraulics Code</u>
char*6	["test01", "test02"]	["test03", "test04"]
logical	[F, F, F, F, F, F, T]	[F, F, F, F, F, F, F]
integer*2	[0, 2, 1]	[1, 0, 2]
integer*4	[888]	[999]
real*4	[1.0, 2.0, 3.0]	[4.0, 5.0, 6.0]
real*8	[10.0, 20.0, 30.0]	[40.0, 50.0, 60.0]

Case 7:

	<u>Neutronics Code</u>	<u>Thermal-Hydraulics Code</u>
char*6	["test01", "test02"]	["test03", "test04"]
logical	[F, F, F, F, F, F, T]	[F, F, F, F, F, F, F]
integer*2	[0, 2, 1]	[1, 0, 2]
integer*4	[888]	[999]
real*4	[1.0, 2.0, 3.0]	[4.0, 5.0, 6.0]
real*8	[10.0, 20.0, 30.0]	[40.0, 50.0, 60.0]

6.2.6 Time-Dependent Control Buffers

The data used to test the transfer of the time-dependent control buffers are shown below for each test case. Two time-dependent calculations are performed, and all of the data in the control buffers remain the same during both time steps, except for the sixth logical position in both the T/H and neutronic control buffers, which is used to indicate a normal calculation termination. All of the data shown in the buffers represent "dummy" values, except the following:

- The first element in the logical buffer is used to communicate a calculation error in the respective code (neutronic or T/H).
- The second element in the logical buffer is used to communicate a data error in the code-specific data map routine.
- The third element in the logical buffer is used to communicate a PVM error in the code-specific data map routine.
- The fourth element in the logical buffer is used by the GI to communicate a data error to the respective code.
- The fifth element in the logical buffer is used by the GI to communicate a PVM error to the respective code.
- The sixth element in the logical buffer is used to communicate a normal calculation termination.

Case 1:

	<u>Neutronics Code</u>	<u>Thermal-Hydraulics Code</u>
char*6	["test05", "test06"]	["test07", "test08"]
logical (1st)	[F, F, F, F, F, F]	[F, F, F, F, F, F]
logical (2nd)	[F, F, F, F, F, T]	[F, F, F, F, F, T]
integer*2	[3, 4, 5]	[6, 7, 8]
integer*4	[888]	[999]
real*4	[7.0, 8.0, 9.0]	[10.0, 11.0, 12.0]
real*8	[70.0, 80.0, 90.0]	[100.0, 110.0, 120.0]

Case 2:

	<u>Neutronics Code</u>	<u>Thermal-Hydraulics Code</u>
char*6	["test05", "test06"]	["test07", "test08"]
logical (1st)	[F, F, F, F, F, F]	[F, F, F, F, F, F]
logical (2nd)	[F, F, F, F, F, T]	[F, F, F, F, F, T]
integer*2	[3, 4, 5]	[6, 7, 8]
integer*4	[888]	[999]
real*4	[7.0, 8.0, 9.0]	[10.0, 11.0, 12.0]
real*8	[70.0, 80.0, 90.0]	[100.0, 110.0, 120.0]

Case 3:

	<u>Neutronics Code</u>	<u>Thermal-Hydraulics Code</u>
char*6	["test05", "test06"]	["test07", "test08"]
logical (1st)	[F, F, F, F, F, F]	[F, F, F, F, F, F]
logical (2nd)	[F, F, F, F, F, T]	[F, F, F, F, F, T]
integer*2	[3, 4, 5]	[6, 7, 8]
integer*4	[888]	[999]
real*4	[7.0, 8.0, 9.0]	[10.0, 11.0, 12.0]
real*8	[70.0, 80.0, 90.0]	[100.0, 110.0, 120.0]

Case 4:

	<u>Neutronics Code</u>	<u>Thermal-Hydraulics Code</u>
char*6	["test05", "test06"]	["test07", "test08"]
logical (1st)	[F, F, F, F, F, F]	[F, F, F, F, F, F]
logical (2nd)	[F, F, F, F, F, T]	[F, F, F, F, F, T]
integer*2	[3, 4, 5]	[6, 7, 8]
integer*4	[888]	[999]
real*4	[7.0, 8.0, 9.0]	[10.0, 11.0, 12.0]
real*8	[70.0, 80.0, 90.0]	[100.0, 110.0, 120.0]

Case 5:

	<u>Neutronics Code</u>	<u>Thermal-Hydraulics Code</u>
char*6	["test05", "test06"]	["test07", "test08"]
logical (1st)	[F, F, F, F, F, F]	[F, F, F, F, F, F]
logical (2nd)	[F, F, F, F, F, T]	[F, F, F, F, F, T]
integer*2	[3, 4, 5]	[6, 7, 8]
integer*4	[888]	[999]
real*4	[7.0, 8.0, 9.0]	[10.0, 11.0, 12.0]
real*8	[70.0, 80.0, 90.0]	[100.0, 110.0, 120.0]

Case 6:

	<u>Neutronics Code</u>	<u>Thermal-Hydraulics Code</u>
char*6	["test05", "test06"]	["test07", "test08"]
logical (1st)	[F, F, F, F, F, F]	[F, F, F, F, F, F]
logical (2nd)	[F, F, F, F, F, T]	[F, F, F, F, F, T]
integer*2	[3, 4, 5]	[6, 7, 8]
integer*4	[888]	[999]
real*4	[7.0, 8.0, 9.0]	[10.0, 11.0, 12.0]
real*8	[70.0, 80.0, 90.0]	[100.0, 110.0, 120.0]

Case 7:

	<u>Neutronics Code</u>	<u>Thermal-Hydraulics Code</u>
char*6	["test05", "test06"]	["test07", "test08"]
logical (1st)	[F, F, F, F, F, F]	[F, F, F, F, F, F]
logical (2nd)	[F, F, F, F, F, T]	[F, F, F, F, F, T]
integer*2	[3, 4, 5]	[6, 7, 8]
integer*4	[888]	[999]
real*4	[7.0, 8.0, 9.0]	[10.0, 11.0, 12.0]
real*8	[70.0, 80.0, 90.0]	[100.0, 110.0, 120.0]

6.2.7 Test Plan Verification and Acceptance Criteria

The test plan will be verified using a script to compare the produced output to the expected output. The acceptance criteria are as follows:

- Every element of the input vectors and permutation matrices is correct.
- Every element of the output vectors is correct. In addition, the output vectors computed in the GI are consistent with the vectors received in the T/H and neutronics codes.
- The initial and time-dependent control buffers sent from the T/H code are the same as those received by the GI and neutronics codes.
- The initial and time-dependent control buffers sent from the neutronics code are the same as those received by the GI and T/H codes.
- All error measures are properly triggered.
- Safe termination of the process is achieved.

Table 6.2.1 Test Case Matrix for General Interface

Case	Thermal-Hydraulic Geometry^(a)	Thermal-Hydraulic Zones	Neutronic Radial Nodes	Neutronic Axial Nodes^(b)	Heat Struct. Components
1 ^(c)	Cartesian	4	4	1	4
2	Cartesian	1	16	1	1
3	Cartesian	16	4	1	16
4	Cylindrical	16	16	1	16
5	Cartesian	4	16	1	16
6	Cartesian	4	16	2	4
7 ^(d)	Cylindrical	2	1	1	1

^(a) Neutronic geometry is always Cartesian

^(b) Number of thermal-hydraulic and heat structure axial nodes is always 1

^(c) Base test case corresponding to Fig. 6.2.1

^(d) Test case for annulus problem corresponding to Fig. 6.2.2

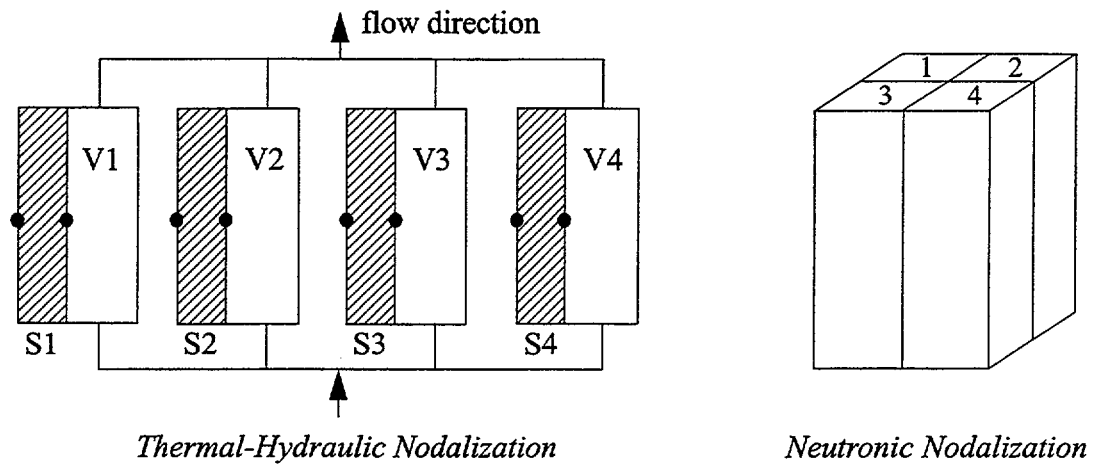


Figure 6.2.1 Problem Nodalization for Base Test Case

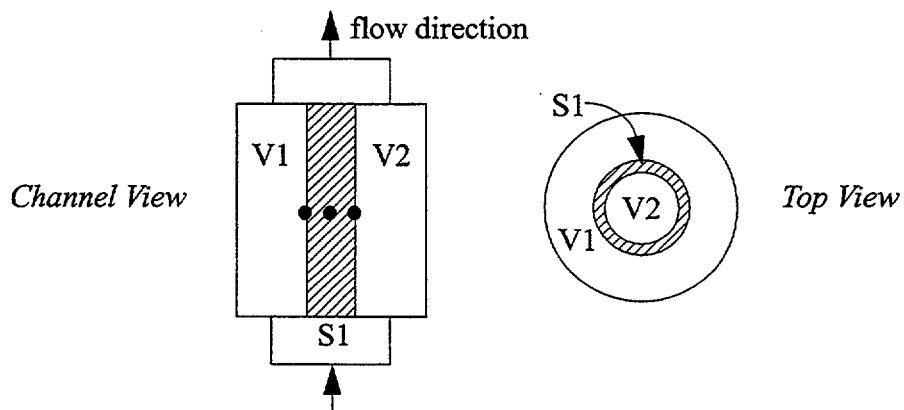


Figure 6.2.2 Thermal-Hydraulic Nodalization for Annulus Test Case

6.3 Verification Testing and Assessment

The General Interface (GI) contains three functional units and one error-checking unit. The three functional units correspond to (1) initialization, (2) T/H to neutronics mapping, and (3) neutronics to T/H mapping. The data transferred through the GI relates to T/H and neutronic control information and property data. In addition, the mapping between the T/H and neutronic spatial domains is represented by permutation matrices, which are constructed by either of the code-specific routines and sent to the GI during the initialization.

The test plan described for the General Interface in Section 6.2 was executed for each of the seven cases. Tables 6.3.1 through 6.3.7 list the test results for each of these cases, and indicate whether the execution was successful. Table 6.3.1 also includes results from the parametrics described in the test plan. In addition, comments are included to provide some detail concerning the results that were obtained.

Table 6.3.1 Matrix of Test Results for Case 1

Requirement	Successful ?	Comments
<i>Case 1: Base test case</i>		
A.1	true	The initial T/H control buffers shown for Case 1 (in Section 6.2.5) were received correctly from the T/H process and correctly sent to the neutronic process.
A.2	true	The initial neutronic control buffers shown for Case 1 (in Section 6.2.5) were correctly received from the neutronic process and sent correctly to the T/H process.
A.4	true	The permutation matrices were correctly received from the neutronic process, and were consistent with the data shown for Case 1 (in Section 6.2.2).
B.1	true	The unpermuted T/H vector shown for Case 1 (in Section 6.2.3) was correctly received from the T/H process.
B.2	true	The matrix-vector multiply was performed, and the resulting permuted vector was consistent with that shown for Case 1 (in Section 6.2.4).
B.3	true	The permuted T/H vector was correctly sent to the neutronic process.
B.4	true	The time-dependent T/H control buffers shown for Case 1 (in Section 6.2.6) were correctly received from the T/H process and correctly sent to the neutronic process.

Table 6.3.1 Matrix of Test Results for Case 1 (Continued)

Requirement	Successful ?	Comments
C.1	true	The unpermuted neutronic vector shown for Case 1 (in Section 6.2.3) was correctly received from the neutronic process.
C.2	true	The matrix-vector multiply was performed, and the resulting permuted vector was consistent with that shown for Case 1 (in Section 6.2.4).
C.3	true	The permuted neutronic vector was correctly sent to the T/H process.
C.4	true	The time-dependent neutronic control buffers shown for Case 1 (in Section 6.2.6) were correctly received from the neutronic process and correctly sent to the T/H process.
E.1	true	Normal termination was achieved as a result of the detection of a signal sent from the T/H and neutronic processes.
<i>Case 1a: Parametric on receipt of permutation matrices</i>		
A.3	true	The permutation matrices were correctly received from the T/H process, and were consistent with the data shown for Case 1 (in Section 6.2.2).
<i>Case 1b: Parametric for initialization error</i>		
D.1 (Initialization)	true	Both the neutronic and T/H processes attempted to send the permutation matrices. This resulted in both an error message being printed, and the safe termination of all three processes.
<i>Case 1c: Parametric for initialization error</i>		
D.1 (Initialization)	true	Neither the neutronic nor the T/H process attempted to send the permutation matrices. This resulted in both an error message being printed, and the safe termination of all three processes.
<i>Case 1d: Parametric for initialization error</i>		
D.2,3 (Initialization)	true	Permutation matrix elements were greater than 1.0. This resulted in both an error message being printed, and the safe termination of all three processes.

Table 6.3.1 Matrix of Test Results for Case 1 (Continued)

Requirement	Successful ?	Comments
<i>Case 1e: Parametric for initialization error</i>		
D.2,3 (Initialization)	true	Permutation matrix elements were less than 0.0. This resulted in both an error message being printed, and the safe termination of all three processes.
<i>Case 1f: Parametric for T/H-to-neutronics error</i>		
D.1 (T/H-to-Neut.)	true	Decreasing the dimension of the unpermuted T/H vector during the first mapping step resulted in both an error message being printed, and the safe termination of all three processes.
<i>Case 1g: Parametric for T/H-to-neutronics error</i>		
D.1 (T/H-to-Neut.)	true	Decreasing the dimension of the unpermuted T/H vector during the second mapping step resulted in both an error message being printed, and the safe termination of all three processes.
<i>Case 1h: Parametric for T/H-to-neutronics error</i>		
D.2 (T/H-to-Neut.)	true	Forcing elements of the vector data to be less than the 0.0 resulted in both an error message being printed, and the safe termination of all three processes.
<i>Case 1i: Parametric for neutronics-to-T/H error</i>		
D.1 (Neut.-to-T/H)	true	Decreasing the dimension of the unpermuted neutronic vector during the first mapping step resulted in both an error message being printed, and the safe termination of all three processes.
<i>Case 1j: Parametric for neutronics-to-T/H error</i>		
D.1 (Neut.-to-T/H)	true	Decreasing the dimension of the unpermuted neutronic vector during the second mapping step resulted in both an error message being printed, and the safe termination of all three processes.
<i>Case 1k: Parametric for neutronics-to-T/H error</i>		
D.2 (Neut.-to-T/H)	true	Forcing elements of the vector data to be less than the 0.0 resulted in both an error message being printed, and the safe termination of all three processes.

Table 6.3.1 Matrix of Test Results for Case 1 (Continued)

Requirement	Successful ?	Comments
<i>Case 1l: Parametric for process error</i>		
E.2	true	Setting the first, second, and third logicals in the initial neutronic control buffer to "TRUE" resulted in both an error message being printed, and the safe termination of all three processes.
<i>Case 1m: Parametric for process error</i>		
E.2	true	Setting the first, second, and third logicals in the initial T/H control buffer to "TRUE" resulted in both an error message being printed, and the safe termination of all three processes.
<i>Case 1n: Parametric for process error</i>		
E.2	true	Setting the first, second, and third logicals in the time-dependent neutronic control buffer to "TRUE" resulted in both an error message being printed, and the safe termination of all three processes.
<i>Case 1o: Parametric for process error</i>		
E.2	true	Setting the first, second, and third logicals in the time-dependent T/H control buffer to "TRUE" resulted in both an error message being printed, and the safe termination of all three processes.

Table 6.3.2 Matrix of Test Results for Case 2

Requirement	Successful ?	Comments
A.1	true	The initial T/H control buffers shown for Case 2 (in Section 6.2.5) were correctly received from the T/H process and correctly sent to the neutronic process.
A.2	true	The initial neutronic control buffers shown for Case 2 (in Section 6.2.5) were correctly received from the neutronic process and correctly sent to the T/H process.

Table 6.3.2 Matrix of Test Results for Case 2 (Continued)

Requirement	Successful ?	Comments
A.4	true	The permutation matrices were correctly received from the neutronic process, and were consistent with the data shown for Case 2 (in Section 6.2.2).
B.1	true	The unpermuted T/H vector shown for Case 2 (in Section 6.2.3) was correctly received from the T/H process.
B.2	true	The matrix-vector multiply was performed, and the resulting permuted vector was consistent with that shown for Case 2 (in Section 6.2.4).
B.3	true	The permuted T/H vector was correctly sent to the neutronic process.
B.4	true	The time-dependent T/H control buffers shown for Case 2 (in Section 6.2.6) were correctly received from the T/H process and correctly sent to the neutronic process.
C.1	true	The unpermuted neutronic vector shown for Case 2 (in Section 6.2.3) was correctly received from the neutronic process.
C.2	true	The matrix-vector multiply was performed, and the resulting permuted vector was consistent with that shown for Case 2 (in Section 6.2.4).
C.3	true	The permuted neutronic vector was correctly sent to the T/H process.
C.4	true	The time-dependent neutronic control buffers shown for Case 2 (in Section 6.2.6) were correctly received from the neutronic process and correctly sent to the T/H process.
E.1	true	Normal termination was achieved as a result of the detection of a signal sent from the T/H and neutronic processes.

Table 6.3.3 Matrix of Test Results for Case 3

Requirement	Successful ?	Comments
A.1	true	The initial T/H control buffers shown for Case 3 (in Section 6.2.5) were correctly received from the T/H process and correctly sent to the neutronic process.
A.2	true	The initial neutronic control buffers shown for Case 3 (in Section 6.2.5) were correctly received from the neutronic process and correctly sent to the T/H process.
A.4	true	The permutation matrices were correctly received from the neutronic process, and were consistent with the data shown for Case 3 (in Section 6.2.2).
B.1	true	The unpermuted T/H vector shown for Case 3 (in Section 6.2.3) was correctly received from the T/H process.
B.2	true	The matrix-vector multiply was performed, and the resulting permuted vector was consistent with that shown for Case 3 (in Section 6.2.4).
B.3	true	The permuted T/H vector was correctly sent to the neutronic process.
B.4	true	The time-dependent T/H control buffers shown for Case 3 (in Section 6.2.6) were correctly received from the T/H process and correctly sent to the neutronic process.
C.1	true	The unpermuted neutronic vector shown for Case 3 (in Section 6.2.3) was correctly received from the neutronic process.
C.2	true	The matrix-vector multiply was performed, and the resulting permuted vector was consistent with that shown for Case 3 (in Section 6.2.4).
C.3	true	The permuted neutronic vector was correctly sent to the T/H process.
C.4	true	The time-dependent neutronic control buffers shown for Case 3 (in Section 6.2.6) were correctly received from the neutronic process and correctly sent to the T/H process.
E.1	true	Normal termination was achieved as a result of the detection of a signal sent from the T/H and neutronic processes.

Table 6.3.4 Matrix of Test Results for Case 4

Requirement	Successful ?	Comments
A.1	true	The initial T/H control buffers shown for Case 4 (in Section 6.2.5) were correctly received from the T/H process and correctly sent to the neutronic process.
A.2	true	The initial neutronic control buffers shown for Case 4 (in Section 6.2.5) were correctly received from the neutronic process and correctly sent to the T/H process.
A.4	true	The permutation matrices were correctly received from the neutronic process, and were consistent with the data shown for Case 4 (in Section 6.2.2).
B.1	true	The unpermuted T/H vector shown for Case 4 (in Section 6.2.3) was correctly received from the T/H process.
B.2	true	The matrix-vector multiply was performed, and the resulting permuted vector was consistent with that shown for Case 4 (in Section 6.2.4).
B.3	true	The permuted T/H vector was correctly sent to the neutronic process.
B.4	true	The time-dependent T/H control buffers shown for Case 4 (in Section 6.2.6) were correctly received from the T/H process and correctly sent to the neutronic process.
C.1	true	The unpermuted neutronic vector shown for Case 4 (in Section 6.2.3) was correctly received from the neutronic process.
C.2	true	The matrix-vector multiply was performed, and the resulting permuted vector was consistent with that shown for Case 4 (in Section 6.2.4).
C.3	true	The permuted neutronic vector was correctly sent to the T/H process.
C.4	true	The time-dependent neutronic control buffers shown for Case 4 (in Section 6.2.6) were correctly received from the neutronic process and correctly sent to the T/H process.
E.1	true	Normal termination was achieved as a result of the detection of a signal sent from the T/H and neutronic processes.

Table 6.3.5 Matrix of Test Results for Case 5

Requirement	Successful ?	Comments
A.1	true	The initial T/H control buffers shown for Case 5 (in Section 6.2.5) were correctly received from the T/H process and correctly sent to the neutronic process.
A.2	true	The initial neutronic control buffers shown for Case 5 (in Section 6.2.5) were correctly received from the neutronic process and correctly sent to the T/H process.
A.4	true	The permutation matrices were correctly received from the neutronic process, and were consistent with the data shown for Case 5 (in Section 6.2.2).
B.1	true	The unpermuted T/H vector shown for Case 5 (in Section 6.2.3) was correctly received from the T/H process.
B.2	true	The matrix-vector multiply was performed, and the resulting permuted vector was consistent with that shown for Case 5 (in Section 6.2.4).
B.3	true	The permuted T/H vector was correctly sent to the neutronic process.
B.4	true	The time-dependent T/H control buffers shown for Case 5 (in Section 6.2.6) were correctly received from the T/H process and correctly sent to the neutronic process.
C.1	true	The unpermuted neutronic vector shown for Case 5 (in Section 6.2.3) was correctly received from the neutronic process.
C.2	true	The matrix-vector multiply was performed, and the resulting permuted vector was consistent with that shown for Case 5 (in Section 6.2.4).
C.3	true	The permuted neutronic vector was correctly sent to the T/H process.
C.4	true	The time-dependent neutronic control buffers shown for Case 5 (in Section 6.2.6) were correctly received from the neutronic process and correctly sent to the T/H process.
E.1	true	Normal termination was achieved as a result of the detection of a signal sent from the T/H and neutronic processes.

Table 6.3.6 Matrix of Test Results for Case 6

Requirement	Successful ?	Comments
A.1	true	The initial T/H control buffers shown for Case 6 (in Section 6.2.5) were correctly received from the T/H process and correctly sent to the neutronic process.
A.2	true	The initial neutronic control buffers shown for Case 6 (in Section 6.2.5) were correctly received from the neutronic process and correctly sent to the T/H process.
A.4	true	The permutation matrices were correctly received from the neutronic process, and were consistent with the data shown for Case 6 (in Section 6.2.2).
B.1	true	The unpermuted T/H vector shown for Case 6 (in Section 6.2.3) was correctly received from the T/H process.
B.2	true	The matrix-vector multiply was performed, and the resulting permuted vector was consistent with that shown for Case 6 (in Section 6.2.4).
B.3	true	The permuted T/H vector was correctly sent to the neutronic process.
B.4	true	The time-dependent T/H control buffers shown for Case 6 (in Section 6.2.6) were correctly received from the T/H process and correctly sent to the neutronic process.
C.1	true	The unpermuted neutronic vector shown for Case 6 (in Section 6.2.3) was correctly received from the neutronic process.
C.2	true	The matrix-vector multiply was performed, and the resulting permuted vector was consistent with that shown for Case 6 (in Section 6.2.4).
C.3	true	The permuted neutronic vector was correctly sent to the T/H process.
C.4	true	The time-dependent neutronic control buffers shown for Case 6 (in Section 6.2.6) were correctly received from the neutronic process and correctly sent to the T/H process.
E.1	true	Normal termination was achieved as a result of the detection of a signal sent from the T/H and neutronic processes.

Table 6.3.7 Matrix of Test Results for Case 7

Requirement	Successful ?	Comments
A.1	true	The initial T/H control buffers shown for Case 7 (in Section 6.2.5) were correctly received from the T/H process and correctly sent to the neutronic process.
A.2	true	The initial neutronic control buffers shown for Case 7 (in Section 6.2.5) were correctly received from the neutronic process and correctly sent to the T/H process.
A.4	true	The permutation matrices were received correctly from the neutronic process, and were consistent with the data shown for Case 7 (in Section 6.2.2).
B.1	true	The unpermuted T/H vector shown for Case 7 (in Section 6.2.3) was correctly received from the T/H process.
B.2	true	The matrix-vector multiply was performed, and the resulting permuted vector was consistent with that shown for Case 7 (in Section 6.2.4).
B.3	true	The permuted T/H vector was correctly sent to the neutronic process.
B.4	true	The time-dependent T/H control buffers shown for Case 7 (in Section 6.2.6) were correctly received from the T/H process and correctly sent to the neutronic process.
C.1	true	The unpermuted neutronic vector shown for Case 7 (in Section 6.2.3) was correctly received from the neutronic process.
C.2	true	The matrix-vector multiply was performed, and the resulting permuted vector was consistent with that shown for Case 7 (in Section 6.2.4).
C.3	true	The permuted neutronic vector was correctly sent to the T/H process.
C.4	true	The time-dependent neutronic control buffers shown for Case 7 (in Section 6.2.6) were correctly received from the neutronic process and correctly sent to the T/H process.
E.1	true	Normal termination was achieved as a result of the detection of a signal sent from the T/H and neutronic processes.

6.4 Conclusions

The primary purpose of the GI is to facilitate the coupling of any T/H code with any neutronics code. To meet this objective, the GI was designed with several functional requirements, as described in Section 6.1. The successful execution of the QATP, which was designed to test these functions, was discussed in Section 6.3. The results show that the GI has been successfully integrated with TRAC-M and the PARCS code.

7. Plant Transient Calculations

7.1 AP600 Small-Break LOCA Calculations

The AP600 reactor was jointly developed by the Westinghouse Electric Corporation, the U.S. Department of Energy, and the Electric Power Research Institute. This design utilizes passive safety systems that replace the traditional ECCS in a PWR.

7.1.1 AP600 System Description

The new passive system in the AP600 includes the following components:

- **Core Makeup Tanks (CMTs):** These two full-pressure tanks use gravitational potential to provide borated water to the vessel in a loss-of-coolant accident (LOCA). These tanks replace the traditional high-pressure injection system.
- **In-Containment Refueling Water Storage Tank (IRWST):** This very large tank is situated high in the containment to provide long-term gravity-fed cooling water to the core. This replaces the low-pressure safety injection system.
- **Automatic Depressurization System (ADS):** This system consists of valves arranged to actuate in four stages. The first three stages depressurize the system by blowing down the fluid in the pressurizer into the IRWST through a sparger. The fourth stage valves, which are connected to each of the two hot legs, blow down steam and liquid directly into the containment. This operation depressurizes the system pressure to almost containment pressure. This permits injection of water into the vessel from a huge source of water in the IRWST.
- **Passive Residual Heat Removal (PRHR) System:** This C-shaped heat exchanger, submerged in the IRWST, removes decay heat.

In addition to these new passive injection system components, the AP600 safety system also includes accumulators as in a standard PWR. A schematic of the AP600 system is illustrated in Fig. 7.1.1. Further details on the AP600 system are presented in Refs. 7.1.1 and 7.1.2.

7.1.2 Accident Scenario Investigated

The generic response of the AP600 to a small-break loss-of-coolant accident (SBLOCA) is as follows. Following the break initiation, the AP600 depressurizes as a result of the inventory that is lost through the break. Signals are reached that cause the reactor to scram, and trip the reactor coolant pumps. Soon after, an "S-signal" is generated, which actuates the flow through the CMTs and the PRHR system. The PRHR system provides a cooling path that parallels one of the steam generators. The "S-signal" is an AP600-specific safety signal. In this transient, it is actuated by the low pressurizer pressure. The "S-signal" actuates components for operation of passive safety systems (e.g., opens the isolation valve for CMT operation). First, CMTs operate in a single-phase recirculation (natural circulation) mode, and then they switch to a draining mode when the water at the top of the CMTs reaches saturation and flashes, thereby breaking the natural

circulation. The vessel inventory is replenished by gravity flow from the two CMTs and two accumulators. When the level in one CMT reaches a pre-defined setpoint, the first three stages of the ADS are initiated in a timed sequence. The fourth stage of the ADS is initiated when the liquid volume in a CMT reaches a set value. This brings the system pressure down to the point close to the containment pressure, where gravity injection from the IRWST into the vessel can begin.

7.1.3 TRAC-M Modeling

Extensive nodalization studies for the RELAP5 code were performed in Ref. 7.1.3. Three different decks were analyzed to accommodate different sensitivity analyses. The first deck contained a quasi-three-dimensional vessel nodalization, in which the core, plenums, and downcomer were nodalized using cross-flow junctions. The second deck contained a quasi-three-dimensional nodalization for the core and plenums, but the downcomer and the system had simplified nodalization. The third deck contained detailed system and downcomer noding, but a one-dimensional core and plenums. Sensitivity calculations indicated that numerically induced internal circulation flows could occur when quasi-three-dimensional noding is used to model the core and plenums. One of the conclusions in Ref. 7.1.3 is that the first two decks should not be used until this numerical problem is resolved in the RELAP5 code. (Readers who are interested in details of these calculations should refer to Ref. 7.1.3.)

The third deck, identified as DM1D, was used for the analysis presented in this report. The noding was one-dimensional, except for the downcomer, which used cross-flow junctions to simulate two-dimensional flows. The downcomer nodalization consisted of eight axially stacked annular rings, with eight azimuthal sectors in each ring. This was necessary to track axial and azimuthal boron and thermal gradients in the downcomer. Adequacy of the nodalization was tested against predictions obtained from the COMMIX code using a standalone model. The input deck used three parallel tubes to model the steam generator. RELAP5 has a thermal stratification model, so only 10 nodes were needed in each CMT. The break was modeled as a sharp-edged orifice.

The TRAC-M input deck used 3-D noding for the vessel and IRWST, and 1-D noding for all other components. The 3-D noding in the core had two azimuthal sections in order to simulate asymmetric flows if they occur. This noding was similar to that used in the RELAP5 noding used in the first and second decks mentioned above. Each of the two CMTs was simulated with 50 nodes, in order to overcome the lack of a thermal stratification model. The adequacy of the nodalization was tested using two standalone models utilizing 12 and 50 axial nodes. In these standalone models, a constant flow rate with a step function in temperature variation was imposed at the inlet of the CMT. The variation of temperature at the outlet compared to the step function. Dispersion of the temperature profile indicated the extent of numerical diffusion. The 12-node CMT model had a large dispersion, while the dispersion in the 50-node model was reasonable. An accurate prediction of temperature profile for the liquid flow, as it drains from the CMT, is important for accurate prediction of drainage by gravity and system behavior. Ref. 7.1.3 shows that if the thermal tracking model is not used, CMT refills may occur, and ADS actuation may be substantially delayed.

7.1.4 TRAC-M Calculations

Both TRAC-M(F77) and TRAC-M(F90), Versions 3590 and 3640, were used to perform SBLOCA calculations. The transient was 5000 s long, and it took several days and, generally, several restarts to perform calculations. In general, for long-running problems, the decks are designed to have several restarts. For this case, the first restart deck was designed to run 2000 s. Subsequent decks were designed to have a restart at each 1000 s. Both codes ran without iteration failure until 4000 s. Between 4000 and 5000 s, when the fourth-stage ADS was activated at a low pressure, there were numerous iteration problems, and calculations were dumped. This necessitated selection of smaller time steps and/or tightening of the convergence criteria and restarting the problem. This process certainly slowed down the performance of calculations. After several restarts, TRAC-M(F77) completed its calculations. However, calculations with TRAC-M(F90), Version 3580, at the NRC could not be extended beyond ~4900 s. The code was compiled in the DEC-Alpha platform, and time step size had been reduced to 5.0E-10 and the convergence criterion was tightened to 1.0E-06 while the recommended value for the convergence is 1.0E-04 or 1.0E-05. Calculations were not progressing reasonably. The deck was sent to the Los Alamos National Laboratory, and was successfully run there using Version 3690 of the code on a Windows NT platform until 5000 s. The compiler was Digital Visual FORTRAN Compiler. The code was compiled without optimization with the debugger on. It appears that the compiler differences did not permit completion of the run at the NRC.

Figs. 7.1.2 and 7.1.3 show comparisons of the predictions that were generated by TRAC-M(F77) and TRAC-M(F90), Version 3590, of the pressurizer pressure and break flow rates for the first 2000 s of the transient. As is evident from the figures, the agreement between the pressure calculations is "Excellent," and the agreement between break flow calculations is "Excellent" to "Reasonable," considering that TRAC-M(F90) contains some choke flow updates that TRAC-M(F77) code does not have.

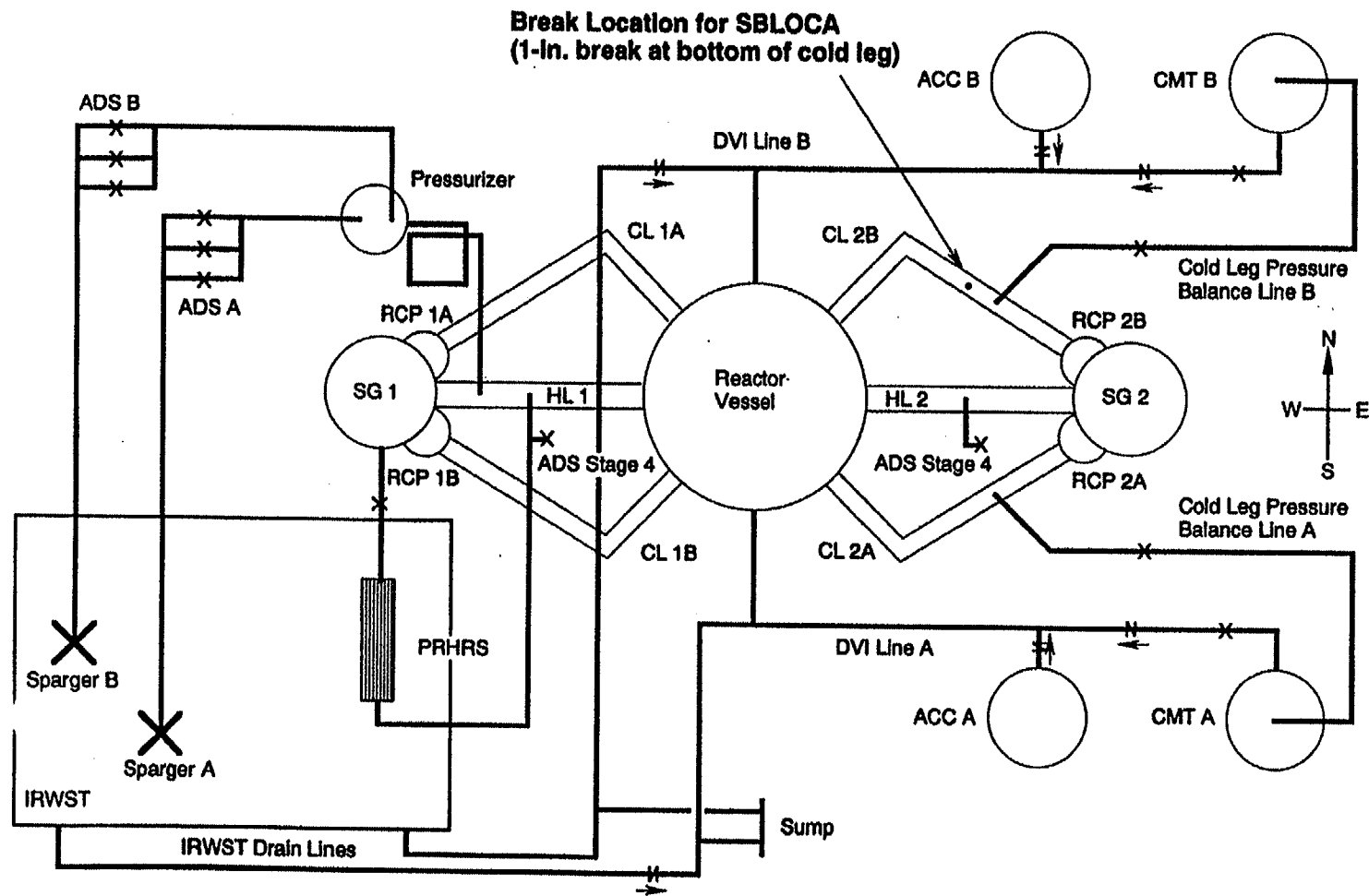
7.1.5 Conclusions

The conversion from TRAC-M(F77) to TRAC-M(F90) has been successful for this type of application. However, code robustness depends on compilers and optimization levels, particularly for FORTRAN 90, since this language is new.

REFERENCES

- 7.1.1 Kelly, J.M., *Thermal-Hydraulic Modeling Needs for Passive Reactors*, Proceedings of the OECD/CSNI Workshop on Transient Thermal-Hydraulic and Neutronic Codes Requirements, November 5-8, 1996, NUREG/CP-0159 or NEA/CSNI/R(97)4.
- 7.1.2 "AP600 Standard Safety Analysis Report," Westinghouse Electric Corporation, Rev. 1, January 1994 and Rev. 3, May 1995.
- 7.1.3 Burt, J., L. Shotkin and J. Staudenmeier, "Effect of RELAP5/MOD3.2 User Options on Calculated Results," *Nuclear Technology*, v. 119, September 1997.

Figure 7.1.1 Schematic of the AP600 System



AP600 SBLOCA

Pressurizer Pressure

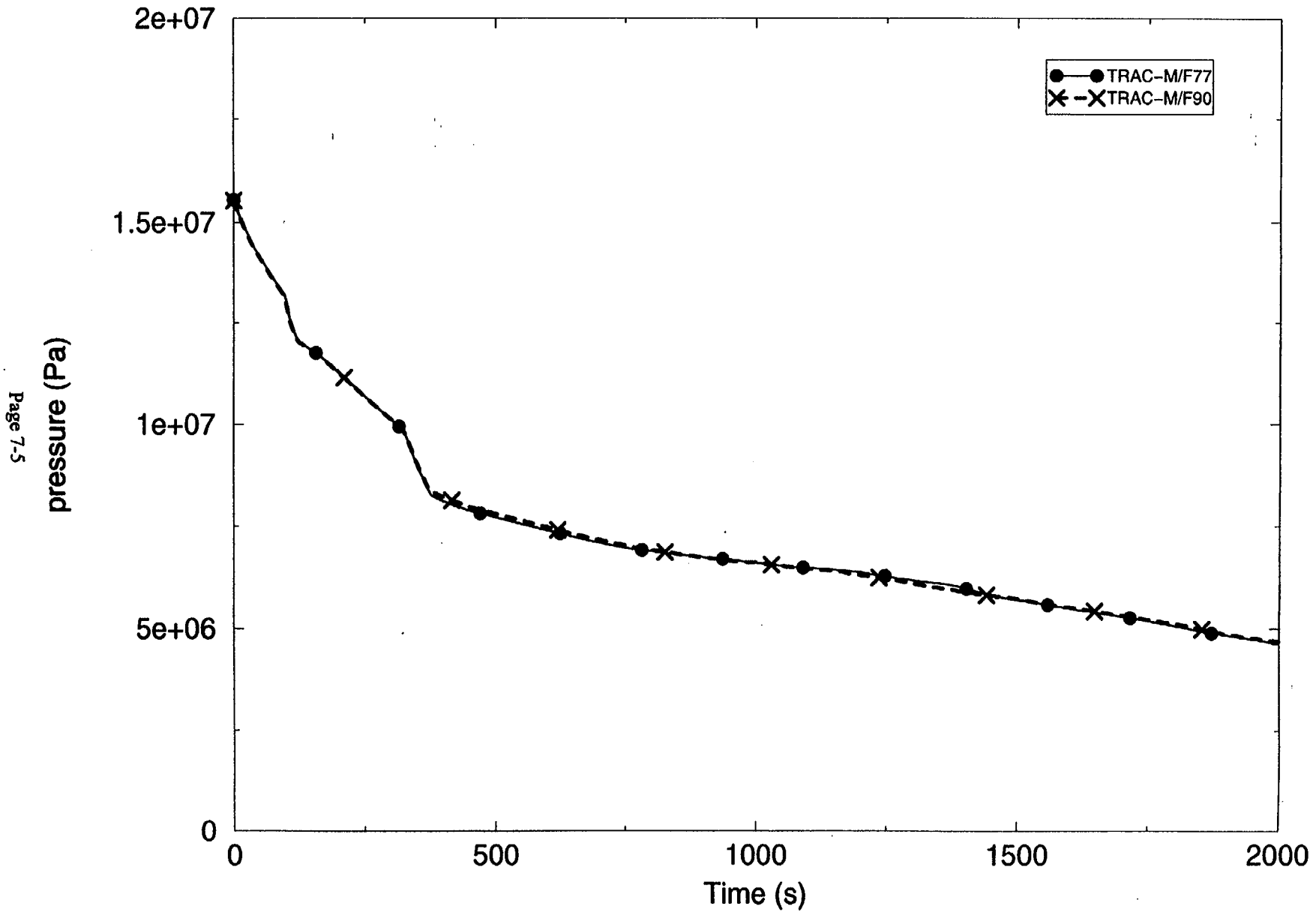


Figure 7.1.2 Comparison of Pressurizer Pressure Calculations

AP600 SBLOCA

Break Flow

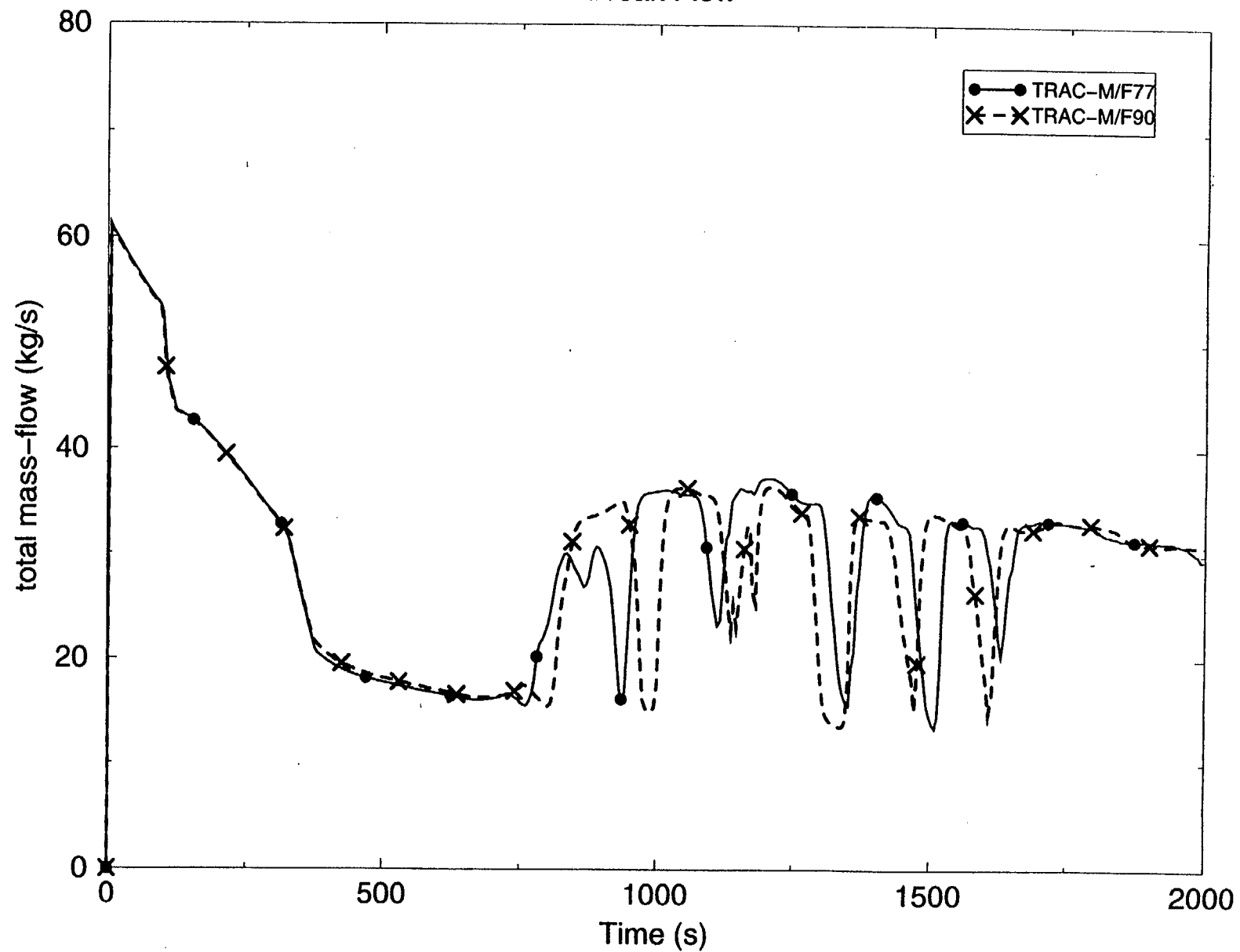


Figure 7.1.3 Comparison of Break Flow Rate Calculations

7.2 TRAC-M Spatial Kinetics Functionality: BWR Application

The spatial kinetics capability based on the PARCS code has been integrated to TRAC-M. The testing of the TRAC-M spatial kinetics capability for BWR functionality is performed using the OECD Nuclear Energy Agency (NEA) Peach Bottom Turbine Trip 2 (PBTT2) Benchmark problem. The turbine trip (TT) transient in a BWR is a pressurization event, in which the coupling between core phenomena and system dynamics plays an important role. The transient begins with a sudden turbine stop valve (TSV) closure. The pressure oscillation generated in the main steam piping propagates with relatively little attenuation into the reactor core. The oscillation induced by core pressure results in dramatic changes of the core void distribution and fluid flow. The core void collapses and, because of the feedback, the neutron flux spikes. The magnitude of the neutron flux transient taking place is strongly affected by the initial rate of pressure rise, and has a strong spatial variation. The correct simulation of the power response to the pressure pulse and subsequent void collapse requires a 3-D core modeling supplemented by 1-D simulation of the remainder of the reactor coolant system.

For the purposes of verifying the functionality of spatial kinetics in TRAC-M, preliminary results are presented here for both the steady-state and transient conditions of the PBTT2 transient. The results are compared to the measured plant data, as well as the results of the RAMONA code (Ref. 7.2.1). For the PBTT2 transient test, the dynamic measurements were taken with a high-speed digital acquisition system capable of sampling over 150 signals every 6 milliseconds. The following sections first describe the plant design and the PARCS neutronic model, as well as the mapping of neutronic parameters to TRAC T/H models. The results of the plant steady-state and transient calculations are summarized in Section 7.2.2.

7.2.1 Benchmark Model Specifications

The reference design for the benchmark was derived from Ref. 7.2.2 for the actual Peach Bottom reactor and operational data.

7.2.1.1 Core Geometry and Fuel Assembly Specifications

The radial geometry of the reactor core is shown in Fig. 7.2.1. Radially, the core is divided into cells that are 15.494 cm wide, each of which corresponds to one fuel assembly (FA), plus a radial reflector (shaded area of Fig. 7.2.1) of the same width. There are a total of 912 assemblies, 764 FAs, and 148 reflector assemblies. Axially, the reactor core is divided into 26 layers (24 FA layers plus bottom reflectors) with a constant height of 15.24 cm (including reflector nodes). The total active core height is 365.76 cm. The axial nodalization accounts for material changes in the fuel design and for exposure variation.

Table 7.2.1 provides geometric data for the FAs in Peach Bottom Cycle 2. There are six types of FAs and detailed geometric data for each is available in the benchmark specifications or in Ref. 7.2.2.

The core loading during the test included 576 fuel bundles of the original 7x7 type from cycle 1 (C1), and a reload of 188 8x8 fuel bundles. A schematic of the 8x8 reload fuel lattice is shown in Fig. 7.2.2.

7.2.1.2 PARCS Neutronics Modeling

The PARCS neutronics model employed two prompt and six delayed neutron groups. Each FA in the core was modeled using a separate neutronic node with unique neutronic feedback properties. Nineteen different fuel assembly types were contained within the core geometry, for which corresponding sets of cross-sections were provided. Each composition was defined by material properties (attributable to changes in the fuel design) and burnup. The radial distribution of these assembly types within the reactor geometry is shown in Fig. 7.2.2.

A complete set of diffusion coefficients and macroscopic cross-sections for scattering, absorption, and fission as a function of the moderator density and fuel temperature was defined for each composition. All cross-section data, along with a program for linear interpolation, were supplied at the benchmark ftp site.

7.2.1.3 TRAC-M Thermal-Hydraulic Model and Coupling of Neutronics to Thermal-Hydraulics

The basis for the TRAC-M T/H model is the design data for the Peach Bottom 2 reactor vessel and the core design specifications shown in Table 7.2.2.

The Peach Bottom steady-state input model was prepared in the TRAC-B format so that it could be exercised with both TRAC-B and TRAC-M. The reactor system was described with seven subsystems consisting of 30 TRAC components with 30 three-dimensional and 75 one-dimensional finite volumes. Further details on modeling and nodalization will be provided in a future report.

The Peach Bottom core contains a total of 764 bundles. Based on the location (inside vs. periphery), the array size (7x7 vs. 8x8), and the diameter of the inlet orifice (1.469 in. vs. 2.211 in.), the fuel bundles were grouped into seven CHAN components. There were two finite volumes below the lower tie plate, and a single volume above the upper tie. In all of these CHAN components, a uniform mesh, consisting of 24 volumes, was used along the heated length of the fuel. The mapping of fuel bundles to CHAN components is shown in Table 7.2.3 and Fig. 7.2.3.

7.2.2 Peach Bottom Benchmark Results with TRAC-M

This section presents results for the steady-state and transient solutions of the PBTT2 problem. It should be noted that the OECD Peach Bottom benchmark exercise will only begin in the first quarter of 2001, and the first results are not due until the end of the third quarter of 2001. Therefore, the results shown here are *preliminary* and intended to demonstrate only the functionality of TRAC-M with spatial kinetics for a BWR. Work is continuing on the benchmark problem, and will be completed by the OECD deadline. Final results will be published in a separate report.

7.2.2.1 PBTT2 Steady-State Initialization Results

Steady-state calculations were performed using TRAC-M, and the TRAC-M model was initialized at the initial conditions for PBTT2. The core power was at 61.6% of its nominal value, and the core flow was at 81% of rated plant conditions.

Table 7.2.4 compares the steady-state values predicted by TRAC-M against the data measured and/or calculated from the plant. The agreement between the measured data and predicted values is "Reasonable." The calculation of the void fraction at the core exit during its convergence to its final value is shown in Fig. 7.2.4. The tolerance on the exit void fraction was 1.0E-03, which was reached after 31 s of the null transient execution.

The converged axial power distribution calculated by TRAC-M for the initial steady-state conditions is shown in Fig. 7.2.5. The power distribution is compared to the plant process computer output, and the agreement is within "Reasonable" tolerance. The discrepancy in the predicted axial power shape can be attributed, in part, to limitations in the current modeling of the flow paths in the channel. The portion of the fluid in the core that flows between the assembly cans (also known as the bypass flow) is not modeled as a separate flow path. Because this "bypass" flow is generally subcooled, the neutron moderation and the power at the top of the core will be underpredicted. Future improvements in the T/H modeling will address this shortcoming by adding additional flow channels to the vessel.

Table 7.2.5 compares the TRAC-M predictions of some selected T/H steady-state parameters with the plant process computer data and RAMONA results. The k -effective (K_{eff}) predicted by TRAC-M is within the uncertainty predicted by the RAMONA code. The actual core K_{eff} is unity. The radial power distribution for the PBTT2 initial conditions is shown in Fig. 7.2.6.

7.2.2.2 PBTT2 Transient Results with TRAC-M

The transient for PBTT2 was calculated using TRAC-M starting from the initial conditions shown in the previous section. The turbine stop valve was closed, as indicated in the specifications. The sudden reduction in steam flow resulted in an increase in the upper plenum pressure. The upper plenum pressure predicted by TRAC-M is shown in Fig. 7.2.7. The plant measurement is also shown for comparison.

The increase in the core pressure leads to a reduction in the core void fraction, as well as an increase in the core power because of the negative void feedback in the core. The core average void fraction and power during the transient as calculated by TRAC-M are shown in Fig. 7.2.8.

The total core reactivity and power predicted by TRAC-M during the transient are compared to the measured plant data, as shown in Figs. 7.2.9 and 7.2.10, respectively. The slight difference occurs because the modeling of the bypass systems valves in this preliminary model is incomplete. Specifically, in the test, when the turbine stop valve closes, the turbine bypass valves open to allow a steam release and, therefore, a pressure relief. The rate at which the bypass valves open is provided by a table in the specifications (position versus time). A TRAC-M control variable is necessary to precisely model the valve operation. However, in the current model, the bypass valves are opened instantaneously when the turbine stop valve is closed. Therefore, in the current model, the pressure wave is mitigated, and the total reactivity and power are somewhat delayed and smaller than the actual test data. For the final TRAC-M model of the PBTT2 transient, the actual bypass valve operation will be modeled, and the reactivity and power should agree well with the plant measurements.

Even though the results shown here are from preliminary modeling of the PBTT2 transient with TRAC-M, the agreement with measured plant data is reasonably good. The PBTT2 results shown here demonstrate that the spatial kinetics capability in TRAC-M is functional for a BWR.

REFERENCES

- 7.2.1 Moberg, L., "RAMONA Analysis of the Peach Bottom-2 Turbine Trip Transient," EPRI-NP-1869, Electric Power Research Institute, 1981.
- 7.2.2 "Core Design and Operating Data for Cycles 1 and 2 of Peach Bottom 2," EPRI-NP-563, Electric Power Research Institute, June 1978.

Table 7.2.1 PB2 Fuel Assembly Data

	Initial Load			Reload (Cycle 1)	Reload (Cycle 2)	LTA Special
Assembly Type	1	2	3	4	5	6
No. of Assemblies, Initial Core	168	263	333	0	0	0
No. of Assemblies, C2	0	261	315	68	116	4
Geometry	7x7	7x7	7x7	8x8	8x8	8x8
Assembly Pitch, in	6.0	6.0	6.0	6.0	6.0	6.0
Fuel Rod Pitch	0.738	0.738	0.738	0.640	0.640	0.640
Fuel Rods per Assembly	49	49	49	63	63	62
Water Rods per Assembly	0	0	0	0	1	2
Burnable Poison Positions	0	4	5	5	5	5
No. of Spacer Grids	7	7	7	7	7	7
Inconel per Grid, lb	0.102	0.102	0.102	0.102	0.102	0.102
Zr-4 per Grid, lb	0.537	0.537	0.537	0.614	0.614	0.614
Spacer Width, in	1.625	1.625	1.625	1.625	1.625	1.625
Assembly Average Fuel Composition:						
Gd ₂ O ₃ , g	0	441	547	490	328	313
UO ₂ , kg	222.44	212.21	212.06	207.78	208.0	207.14
Total Fuel, kg	222.44	212.65	212.61	208.27	208.33	207.45

Table 7.2.2 PB2 Reference Design Information

Parameter	Value
Rated core thermal power, MWt	3293
Rated core total flow rate, (Mlb/hr)/(kg/s)	102.5/12915
Fraction of core thermal power passing through fuel cladding	.96
Approximate bypass coolant total power fraction	.02
Approximate active coolant total power fraction	.02
Approximate channel wall direct heating fraction	.0075
Design minimum critical power ratio for 7x7 assemblies (Cycle 2)	≥1.28
Design minimum critical power ratio for 8x8 assemblies (Cycle 2)	≥1.31
Design overpower for turbine-generator system	105% rated steam
Turbine inlet pressure, psia/Pa	965/6.653E06
Rated reactor dome pressure, psia/Pa	1020/7.033E06
Rated steam flow rate, (Mlb/hr)/(kg/s)	13.381/1685.98
Steam moisture content, fraction	.001
Rate steam dryer and separator pressure drop, psia/Pa	15/103421
Rated core pressure, psia/Pa	1035/7.1361E06
Core pressure drop at rated conditions, psia/Pa	22/151685
Approximate core inlet pressure, psia/Pa	1050/7.2395E06
Core inlet enthalpy, (Btu/lb)/(J/kg)	521.3/1.2125E06
Enthalpy rise across core, (Btu/lb)/(J/kg) (average)	109.6/2.5491E05
Core support plate pressure drop, psi/Pa	18/1.24105E05
Reactor average exit quality at rated conditions	.129
Design hot channel active coolant exit quality	.25
Design bypass coolant exit quality	.0
Total feedwater flow rate, (Mlb/hr)/(kg/s)	13.331/1679.7
Feedwater temperature, °F/°K	376.1/464.32
Control rod drive flow rate, (lb/hr)/(kg/s)	50000/6.2999
Control rod drive flow temperature, °F/°K	80/299.82
Cleanup demineralizer flow rate, (lb/hr)/(kg/s)	133300/16.7958
Cleanup demineralizer inlet temperature, °F/°K	528/548.7
Cleanup demineralizer outlet temperature, °F/°K	431/494.82
Location of demineralized water return	Feedwater line
Jet pump design M ratio	1.96
Jet pump design N ratio	.16
Number of recirculation pumps	2
Recirculation pump type	Centrifugal
Recirculation pump rated flow, (Mlb/hr)/(kg/s)	17.1/2154.56
Total developed pump head, ft/m	710/216.41
Recirculation pump efficiency, percent	87
Head loss from vessel recirculation outlet to vessel inlet, ft/m	59/17.98
Head loss from vessel recirculation inlet to jet pump 180° bend entrance, ft/m	11/3.353

Table 7.2.3 Reactor Core Layout

	Type No. (*)	Array Size	Orif Dia. in (cm)	No. of Bundles
CHAN 50	6	8x8	2.211 (5.616)	4
CHAN 51	5	8x8	2.211 (5.616)	116
CHAN 52	4	8x8	2.211 (5.616)	68
CHAN 53	3	7x7	2.211 (5.616)	297
CHAN 54	2	7x7	2.211 (5.616)	187
CHAN 55	3	7x7	1.469 (3.731)	18
CHAN 56	2	7x7	1.469 (3.731)	74
CHAN 57	Bypass			

(*) See Table 7.2.1

Table 7.2.4 Steady-State Parameters

Parameter	Plant	TRAC	Error (%)
Power (MWt) ¹	3293.00	3293.00	0.00
Core flow rate (kg/s)	12915.00	12915.00	0.00
Bypass flow rate (kg/s)	1097.78	1103.00	-0.48
Turbine inlet pressure (psia) ¹	965.00	965.00	0.00
Steam dome pressure (psia)	1020.00	1020.00	0.00
Steam flow rate (kg/s)	1686.00	1681.00	0.30
Core inlet pressure (psia)	1050.00 ²	1050.00	0.00
Core inlet subcooling (K)	12.50 ²	11.20	10.40
Feedwater temperature (K) ¹	464.30	464.30	0.00
Separator carryunder quality	0.0010	0.0011	-10.00
Downcomer collapsed liquid level (m)	11.24	11.24	0.03
Jet pump m-ratio	1.96	2.05	-4.59
Jet pump n-ratio	0.16	0.16	0.00
Recirculation pump torque (N-m)	30533.00	30200.00	1.09
Recirculation pump speed (rad/s)	174.67	175.00	-0.19

1. Imposed as boundary conditions

2. Inferred from the measured data

Table 7.2.5 Comparison of T/H Steady-State Parameters for PBTT2

PARAMETER	P1-EDIT	RAMONA (Ref. 7.2.1)	TRAC-M
Core Thermal Power, MWth	2028	2028	2028
Core Average Void Fraction	30.1	33.3	31.3
Subcooling, K	8.9	-	8.3
K _{eff}	1.00000	1.01170	1.00397

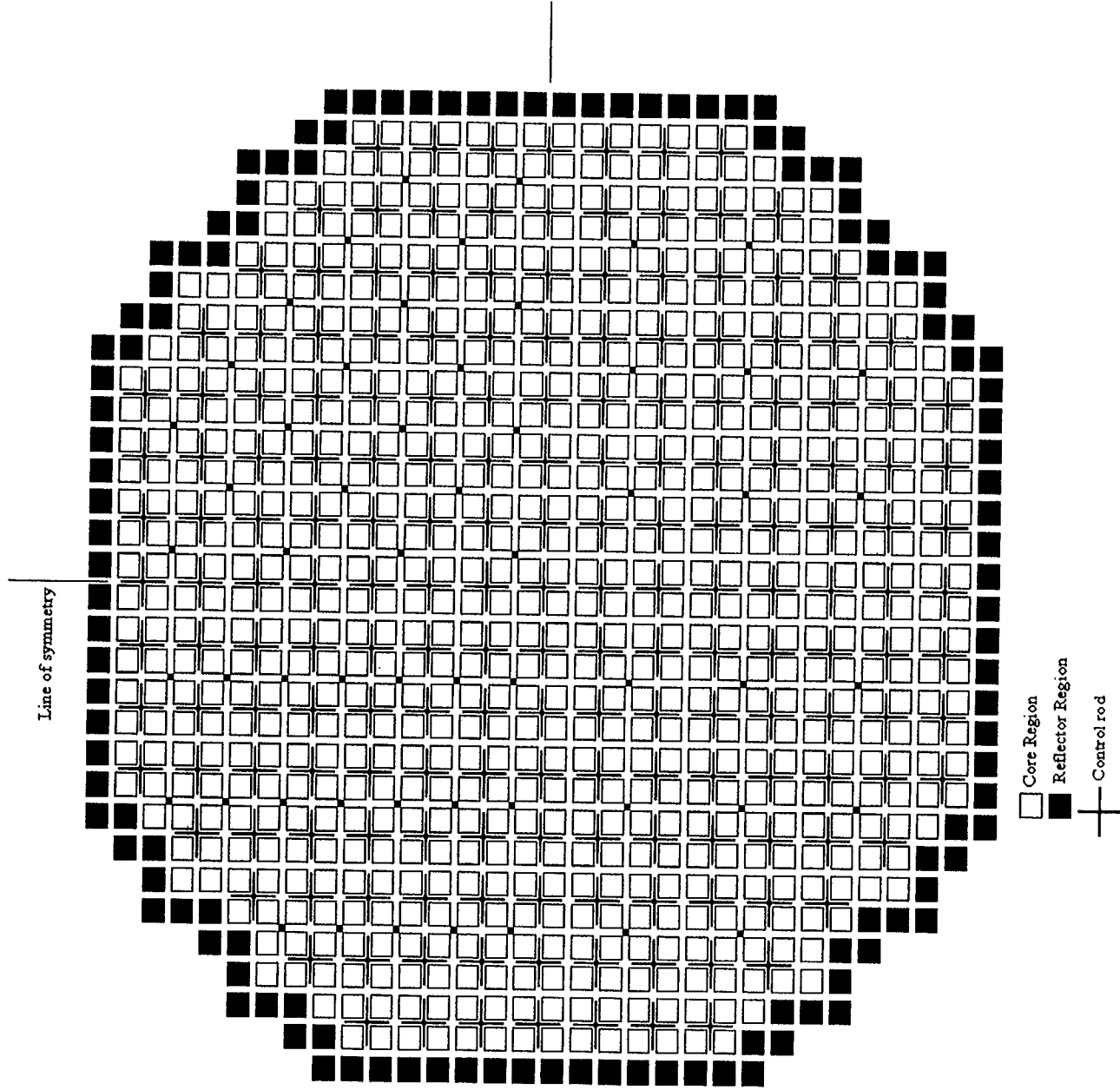


Figure 7.2.1 Reactor Core Cross-Sectional View

```

19 19 19 19 19 19 19 19
19 19 11 5 6 5 6 6 6
19 19 19 6 6 1 5 1 6 1 1
19 7 13 13 1 10 1 10 1 10 12
19 19 6 7 1 14 12 14 15 16 15 15
19 19 19 13 10 1 16 1 15 3 15 1 15 2
19 8 6 9 1 15 10 15 14 15 14 17 14 17
19 19 12 7 1 15 4 15 3 17 3 15 2 15 15
19 19 6 13 1 16 10 17 15 17 17 15 15 17 14 17
19 11 6 1 16 1 15 3 17 2 15 2 15 2 15 2
19 5 1 10 10 15 12 17 16 15 14 17 15 15 14 17
19 6 6 1 16 3 17 3 15 2 15 2 17 2 17 17
19 6 1 9 16 17 14 16 15 15 14 16 15 17 17 17
19 7 5 1 17 3 15 2 15 2 15 2 17 2 17 2
19 6 1 10 15 16 15 15 15 17 15 17 17 15 15
19 6 1 12 16 2 16 13 16 2 17 15 17 2 15 15
19 7 1 12 15 2 17 15 15 2 15 15 15 2 16 17
19 6 1 10 16 17 15 15 15 17 14 17 15 17 15 16
19 6 6 1 15 1 17 2 15 2 15 2 17 2 17 2
19 6 1 9 16 17 12 16 15 15 14 17 14 17 14 17
19 6 6 1 16 3 17 3 15 2 15 2 17 2 17 16
19 5 1 10 10 17 14 15 14 17 15 15 15 17 14 15
19 11 5 1 16 1 17 3 15 2 17 2 15 2 17 2
19 19 7 13 1 16 10 15 14 15 15 15 15 17 16 17
19 19 13 7 1 15 4 15 3 15 3 15 2 17 14
19 8 6 10 1 17 10 17 12 15 14 17 13 17
19 19 19 13 10 1 16 1 17 3 17 1 17 2
19 19 6 7 1 16 10 16 16 18 18 15
19 7 11 15 1 10 1 9 1 10 10
19 19 19 6 6 1 6 1 6 1 1
19 19 11 5 6 6 7 6 6
19 19 19 19 19 19 19 19

```

Figure 7.2.2 Two-Dimensional Assembly Type Map (Half Core)

57 57 57 57 57 57 57 57 57 57 57 57 57 57 57 57
 57 57 57 52 50 50 50 50 50 50 50 50 50 50 52 57 57 57
 57 57 57 50 51 55 51 55 51 55 55 55 55 51 55 51 55 51 50 57 57 57
 57 57 50 52 53 55 51 55 51 55 51 51 51 51 55 51 55 51 55 53 52 50 57 57
 57 57 57 50 51 55 51 51 51 53 51 53 53 51 53 53 51 53 51 53 55 51 50 57 57 57
 57 57 57 52 51 55 51 55 53 55 53 55 53 54 53 55 53 55 53 55 53 55 51 52 57 57 57
 57 57 50 50 51 55 53 51 53 51 53 51 53 51 53 51 53 51 53 51 53 55 51 50 50 57 57
 57 57 57 50 51 55 53 56 53 55 53 55 53 54 53 53 51 53 54 53 55 53 55 51 56 53 55 51 52 57 57 57
 57 57 50 53 55 51 51 53 53 53 53 53 53 51 53 53 53 53 51 53 53 53 51 51 51 53 55 53 50 57 57
 57 52 51 55 51 55 53 55 53 54 53 54 53 54 53 54 53 54 53 54 53 54 53 55 53 55 53 55 51 52 57
 57 50 55 51 51 53 51 53 51 53 51 53 53 53 51 53 53 53 53 53 53 53 51 53 51 53 51 51 55 50 57
 57 50 51 55 51 55 53 55 53 54 53 54 53 54 53 54 53 54 53 54 53 54 53 55 53 55 53 55 51 50 57
 57 50 55 51 51 53 51 51 53 53 51 51 53 53 53 53 53 53 53 53 53 53 53 53 53 53 51 53 51 51 51 55 50 57
 57 50 51 55 51 55 53 54 53 54 53 54 53 54 53 54 53 54 53 54 53 54 53 55 53 55 53 55 51 50 57
 57 50 55 51 53 51 53 53 53 53 53 53 53 53 53 53 53 53 53 53 53 53 53 53 53 53 51 53 51 51 55 50 57
 57 50 55 51 51 54 51 53 51 54 53 53 53 54 53 53 53 53 54 53 53 53 53 54 53 53 51 54 53 51 55 50 57
 57 50 55 51 53 53 53 53 53 51 53 53 53 53 51 53 53 53 53 53 53 53 53 53 53 53 51 53 53 51 55 50 57
 57 50 51 55 53 55 53 54 53 54 53 54 53 54 53 54 53 54 53 54 53 54 53 54 53 54 53 55 53 55 51 50 57
 57 50 55 51 51 53 51 51 53 53 51 53 51 53 53 53 51 53 51 53 53 53 51 51 53 51 55 50 57
 57 50 51 55 51 55 53 55 53 54 53 54 53 54 53 51 53 53 54 53 54 53 54 53 55 53 55 53 55 51 50 57
 57 50 55 51 51 53 51 53 51 53 53 53 53 51 53 53 53 53 51 53 51 53 51 53 51 53 51 51 55 50 57
 57 52 51 55 51 55 53 55 53 54 53 54 53 54 53 54 53 54 53 54 53 54 53 55 53 55 51 55 51 50 57
 57 57 50 53 55 51 51 53 51 53 53 53 53 51 53 51 51 53 53 53 53 51 53 51 51 55 51 50 57 57
 57 57 57 52 51 55 53 56 53 55 53 55 53 54 53 51 51 53 54 53 55 53 55 53 56 53 55 51 52 57 57 57
 57 57 50 50 51 55 53 51 53 51 53 51 53 53 53 53 51 51 53 51 53 51 53 55 51 50 50 57 57
 57 57 57 52 51 55 51 55 53 55 53 55 53 54 53 55 53 55 51 55 51 55 51 52 57 57 57
 57 57 57 50 51 55 51 51 51 51 51 53 51 53 51 51 51 51 55 51 50 57 57 57
 57 57 50 52 53 55 51 55 51 55 51 51 51 51 55 51 55 51 55 53 52 50 57 57
 57 57 57 50 51 55 51 55 51 55 55 55 51 55 51 55 51 50 57 57 57
 57 57 57 52 50 50 50 50 50 50 50 50 50 50 50 50 52 57 57 57
 57 57 57 57 57 57 57 57 57 57 57 57 57 57 57 57 57 57

Figure 7.2.3 Mapping of Fuel Assemblies to CHAN Components

VOID FRACTION AT CORE EXIT

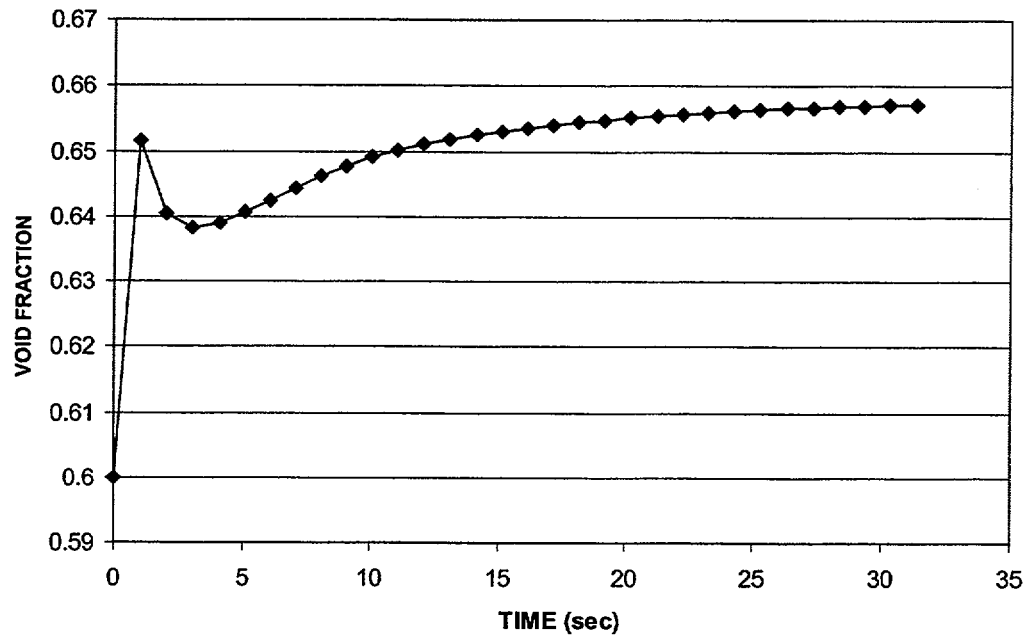


Figure 7.2.4 Steady-State Convergence During Initialization of TRAC-M Model

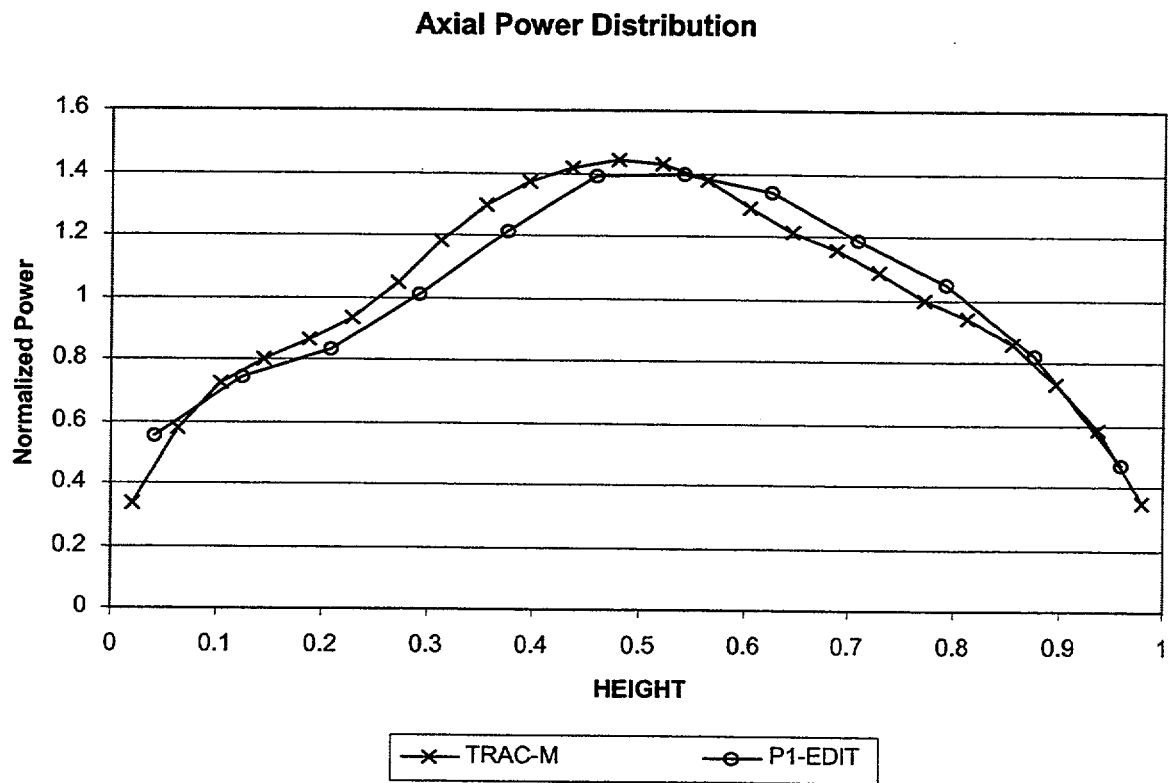


Figure 7.2.5 Axial Power Distribution for the PBTT2 Initial Conditions

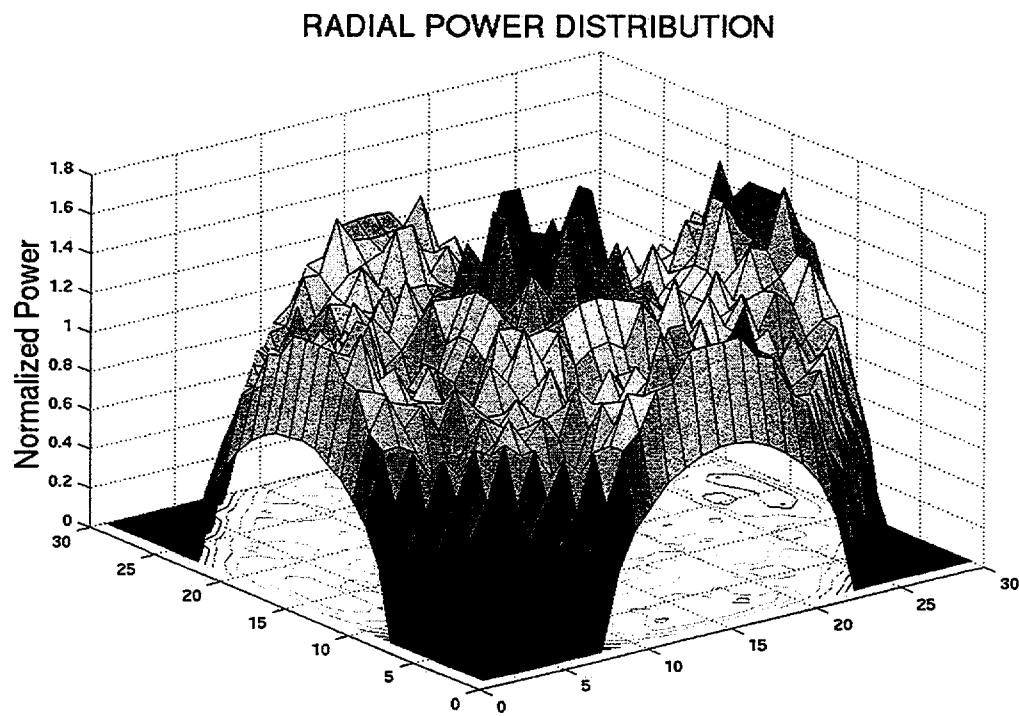


Figure 7.2.6 Steady-State Radial Power Distribution of the PBTT2

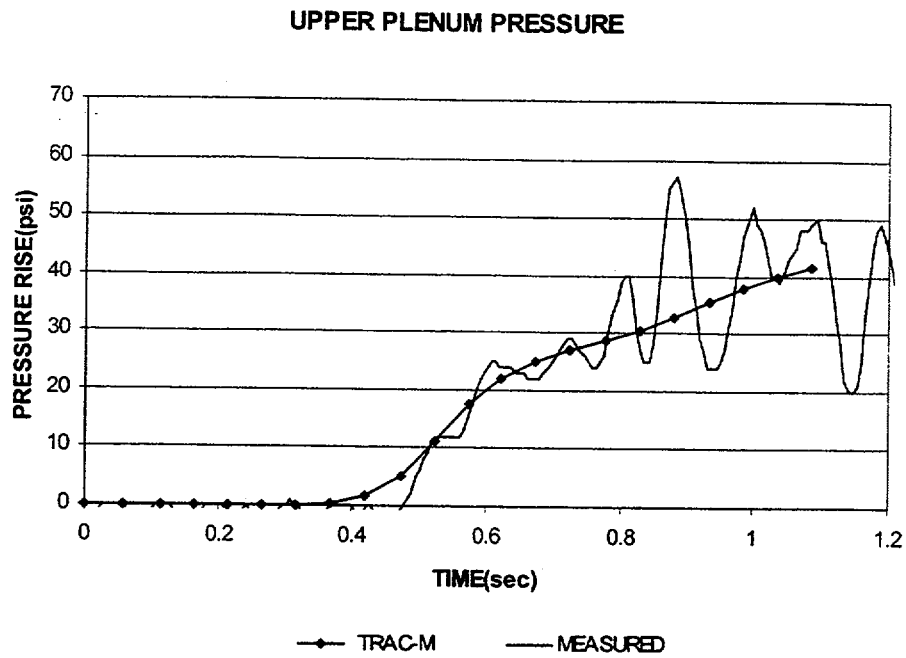


Figure 7.2.7 Upper Plenum Pressure During PBTT2

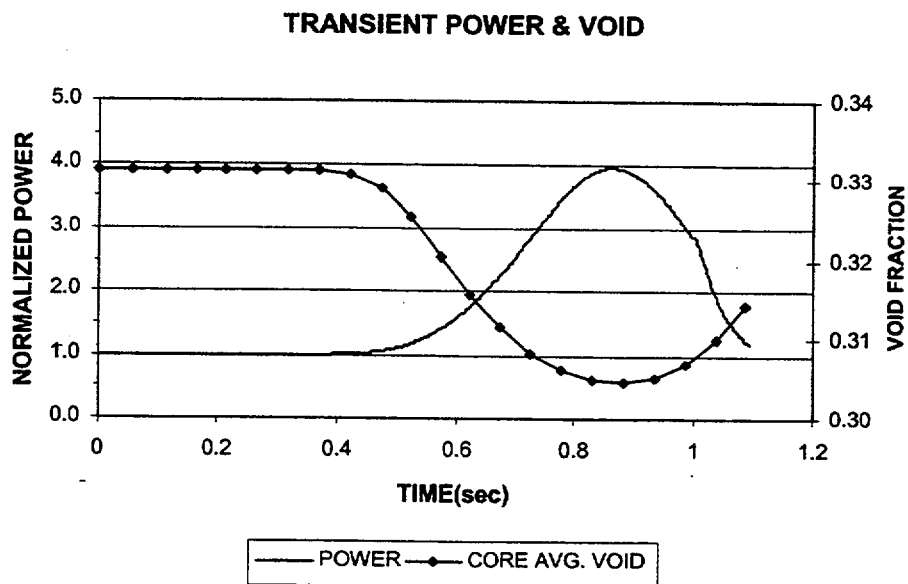


Figure 7.2.8 PBTT2 Core Average Power and Void Fraction Predicted by TRAC-M

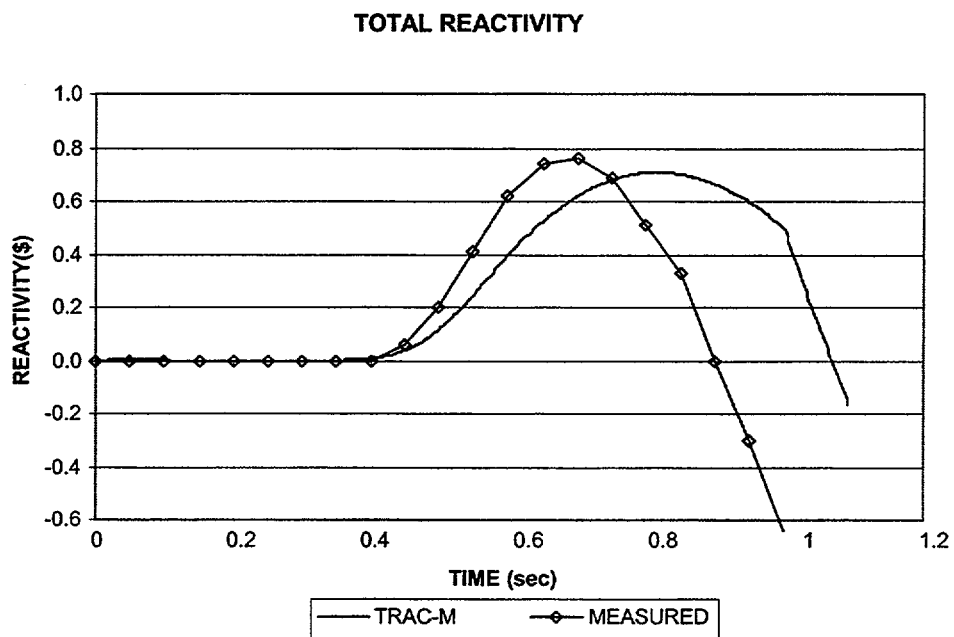


Figure 7.2.9 Comparison of Measured and TRAC-M Predicted Core Reactivity

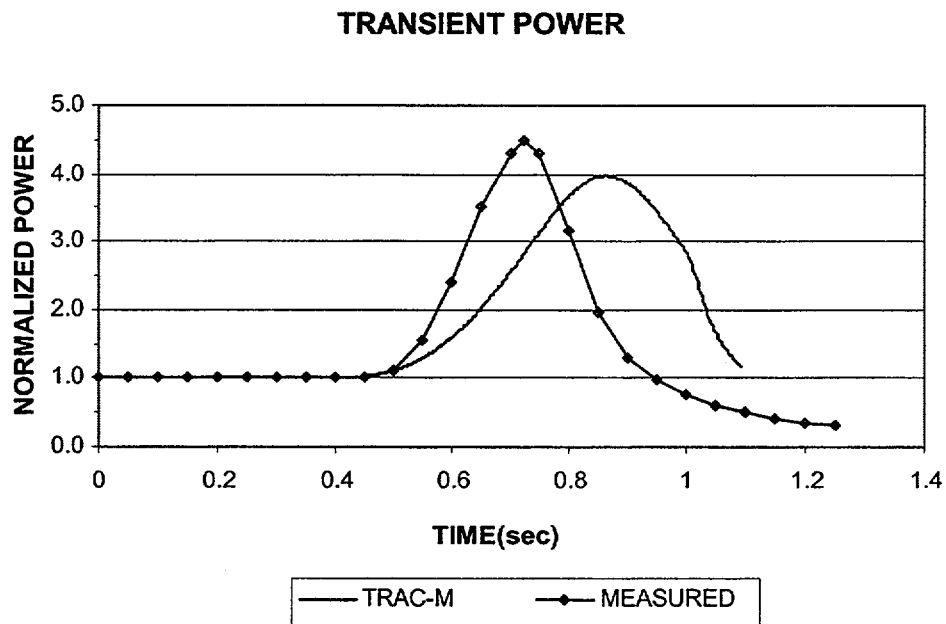


Figure 7.2.10 Comparison of Measured and TRAC-M Predicted Core Average Power

8. Conclusions

Modernization of TRAC-P is successfully accomplished. This is demonstrated by comparing the results of predictions by TRAC-P (Version 5.50) designated as TRAC-M(F77), and TRAC-M (Version 3690) designated as TRAC-M(F90), for test cases presented in Chapter 3, "Analytical Test Problems," Chapter 4, "PWR Developmental Assessment," and Chapter 7, "Plant Transient Calculations."

The test matrix for "Analytical Test Problems" contains 42 cases, involving relatively simple and small problems designed to exercise various parts of the code, with an estimated coverage of about 80%. These problems are exercised with each version of the code as each version is created, to ensure that new errors have not been introduced. Comparisons between the two code predictions show either "Excellent" or "Reasonable" agreements. Chapter 3 contains a sample of five test cases; the results of the remaining test cases are available for review by interested parties.

The test matrix for "PWR Developmental Assessment" contains 20 test cases exercising various parts of the code. The test cases are more complicated than those in the "Analytical Test Problems," in that they represent cases in which coding and modeling of two-phase flow phenomena in PWR plant system transients are tested. Comparisons between the two code predictions show either "Excellent" or "Reasonable" agreements. Most of these test cases also provide comparison with the test data. Since modernization did not change physical models in the code, agreements of the code predictions with the test data remain the same as shown in Ref. 1.2. It should be noted that the test data obtained in the 2-D/3-D program are not presented in this report, since the data are restricted to the use of 2-D/3-D program participants. When the participants agree to release these test data, this report will be revised accordingly.

Finally, a limited number of plant transients are calculated using both codes. These are long-running transients; therefore, many transients could not be included in this report, since the report was prepared in a relatively short time. Comparisons between the two code predictions show "Excellent" or "Reasonable" agreements.

Many BWR components are integrated to the consolidated TRAC-M(F90) code, as is one component to provide RELAP5-type capability and BWR specific models. These components, models, and modules are as follows:

Components:

- Channel (CHAN) component
- Containment (CONTAN) component
- Feedwater heater (HEATR) component
- Jet pump (JETP) component
- Separator (SEPT) component
- Single-junction (SJACS) component
- Turbine (TRBN) component

Models:

- BWR control system (CONSYS) model
- Level tracking (LTRCK) model

Modules:

- Power module
- Spatial kinetics module

The assessment of the integration of the CHAN, HEATR, JETP, SJACS, and TRBN components, as well as the CONSYS and LTRCK models and the 3-D Spatial Kinetics module is reported in Chapters 5 and 6 of this report. Verification tests regarding the integration show that these five components, two models, and one module are correctly integrated. The only exception is that the tests revealed an error in the TRBN component, which prevents verification of the control system operation for the turbine. This error will be corrected in a future version of the code, and the assessment will be repeated using the corrected version. Documentation of the SEPT and CONTAN components is not complete; therefore, their assessments are not discussed in this report. The work on the power module is completed. Its assessment will be included in a future report. Verification tests show that the first eight components and models including 3-D kinetics module are integrated correctly.

Verification tests indicate that each of those models and components have been integrated to the consolidated code correctly; however, they do not show how all of these components and models with the control system would function together. A BWR transient (Peach Bottom Turbine Trip transient) exercising BWR components, spatial kinetics and the BWR control system is being calculated and will be compared against the test data. Preliminary calculations performed for this transient indicate that the spatial kinetics capability has been integrated correctly.

BIBLIOGRAPHIC DATA SHEET

(See instructions on the reverse)

1. REPORT NUMBER
(Assigned by NRC, Add Vol., Supp., Rev.,
and Addendum Numbers, if any.)

NUREG-1752

2. TITLE AND SUBTITLE

Assessment of Modernization and Integration of BWR Components and Spatial Kinetics in the
TRAC-M, Version 3690, Code

3. DATE REPORT PUBLISHED

MONTH

YEAR

December

2001

4. FIN OR GRANT NUMBER

5. AUTHOR(S)

Principal Author: F. Odar (NRC)*
Contributing Authors: J. Uhle, S. Lu (NRC)*
J. F. Dearing (LANL)**
B. Aktas, R. Shumway, S. Lucas (ISL)***
T. Downar (Purdue University)****

6. TYPE OF REPORT

Technical

7. PERIOD COVERED (Inclusive Dates)

8. PERFORMING ORGANIZATION - NAME AND ADDRESS (If NRC, provide Division, Office or Region, U.S. Nuclear Regulatory Commission, and mailing address; if contractor, provide name and mailing address.)

*Division of Systems Analysis and Regulatory Effectiveness ** Los Alamos National Laboratory, Los Alamos, NM 87545
Office of Nuclear Regulatory Research ***ISL, Inc., 11140 Rockville Pike, Rockville, MD 20852
U.S. Nuclear Regulatory Commission ****Purdue Univ., 1290 Nuclear Engr. Bldg., W. Lafayette, IN 47907
Washington, D.C. 20555-0001

9. SPONSORING ORGANIZATION - NAME AND ADDRESS (If NRC, type "Same as above"; if contractor, provide NRC Division, Office or Region, U.S. Nuclear Regulatory Commission, and mailing address.)

Same as above

10. SUPPLEMENTARY NOTES

11. ABSTRACT (200 words or less)

This report presents an assessment of three different activities which led to creation of TRAC-M, Version 3690. The first is the modernization of the old Transient Reactor Analysis Code for pressurized water reactors (TRAC-P), the second is the integration of boiling water reactor (BWR) components from the TRAC-B code and the third is the integration of a spatial kinetics capability from the PARCS code. The assessment shows that all three aspects of the modernization and integration have been successful.

12. KEY WORDS/DESCRIPTORS (List words or phrases that will assist researchers in locating the report.)

TRAC-M
THERMAL HYDRAULICS
COMPUTER CODES

13. AVAILABILITY STATEMENT

unlimited

14. SECURITY CLASSIFICATION

(This Page)

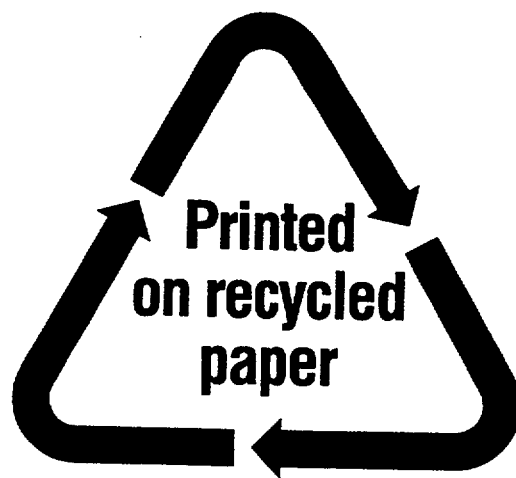
unclassified

(This Report)

unclassified

15. NUMBER OF PAGES

16. PRICE



Federal Recycling Program