ENERGY LABORATORY

MASSACHUSETTS INSTITUTE
OF TECHNOLOGY

TITAN CODE DEVELOPMENT FOR APPLICATION TO A PWR STEAM LINE BREAK ACCIDENT FINAL REPORT: 1983 - 1984

by

C. K. Tsai, M. S. Kazimi and A. F. Henry

Energy Laboratory Report No. MIT-EL 84-014

July 1984

Energy Laboratory

and

Department of Nuclear Engneering

Massachusetts Institute of Technology
Cambridge, Massachusetts 02139

TITAN CODE DEVELOPMENT FOR APPLICATION TO A PWR STEAM LINE BREAK ACCIDENT

by

C. K. Tsai, M. S. Kazimi and A. F. Henry

July 1984

Final Report: 1983 - 1984 for a Project
sponsored by

Long Island Lighting Company
Northeast Utilities Service Company
Yankee Atomic Electric Company

under

MIT Energy Laboratory Electric Utility Program

Energy Laboratory Report No. MIT-EL 84-014

ABSTRACT

Modification of the TITAN computer code which enables it to be applied to a PWR steam line break accident has been accomplished. The code now has the capability of simulating an asymmetric inlet coolant temperature transient by employing different temperature transient forcing functions for different core inlet regions. Up to ten regions of the core can be considered and each region can have at most 50 channels. A total inlet coolant mass flow rate boundary condition option has been added to the code. Flow/coolant temperature transient and control rod transient can be simulated simultaneously by the code as necessary for a steam line break accident simulation. Also, the transient restart capability has been fixed which allows users to change core conditions during a transient calculation for various purposes. All these modifications have been tested by a ten-channel test calculation.

Three steam line break accident simulations (YA-1, YA-2, and YA-3) with different pressure forcing functions have been performed. Each simulation included both closed and open-channel calculations. The steady-state results show that a 1-D thermalhydraulic analysis gives accurate results.

Case YA-1 employed a pressure forcing function taken from a Yankee Atomic report. No boiling during the whole calculation was observed. Also, no significant difference between closed and open-channel calculations was found.

Case YA-2 employed a reduced pressure forcing function with constant pressure after 45 seconds (because of the limitation of W-3 correlation data base). Boiling was observed around 42 seconds after the beginning of the transient. The MCHFR dropped to a value below 6 after boiling. The MCHFR went back to a high value (~30) at 50 seconds for the open-channel calculation while the MCHFR for the closed-channel case still remained below 6. The open-channel model provided a better condition of flow mixing among channels.

Case YW-3 had the same pressure forcing function as that of case YA-2 except the pressure kept decreasing after 45 seconds. The MCHFR was about equal for open-and closed-channels. It is concluded that the closed-channel calculations may produce conservative core power values, but the effect on MCHFR is not always conservative.

Publications of the TITAN Project

Papers

1. D. P. Griggs, M. S. Kazimi and A. F. Henry, "TITAN: An Advanced Three-Dimensional Neutronics/Thermal-Hydraulics Code for LWR Safety Analysis," Proc. ANS Conf. on Advances in Reactor Physics and Core Thermal Hydraulics, Sept. 21-24, 1982, Kiamesha Lake, NY.

2. D. Griggs, C. Tsai, A. Henry and M. Kazimi, "TITAN: An Advanced Three Dimensional Coupled Code," Trans. Am. Nucl. Soc. 46, pp. 1984.

Technical Reports

1. D. P. Griggs, A. F. Henry and M. S. Kazimi, "Development of a Three-Dimensional Two-Fluid Code with Transient Neutronic Feedback for LWR Applications," MIT-EL 81-013, Energy Laboratory, M.I.T., April 1981 (NTIS #PB-82-180 217).

2. D. P. Griggs, M. S. Kazimi, and A. F. Henry, "Advanced Methods Development of LWR Transient Analysis, Final Report: 1981 - 1982," E-Lab Report No. MIT-EL 82-021, May 1982.

3. C. K. Tsai, D. P. Griggs, M. S. Kazimi and A. F. Henry, "Development and Quarter Core PWR Rod Ejection Accident Application of the TITAN Code, Final Report: 1982 - 1983," Energy Laboratory Report No. MIT-EL 83-007, June 1983.

4. C. K. Tsai, M. S. Kazimi and A. F. Henry, "TITAN Code Development for Application to a PWR Steam Line Break Accident," MIT-EL 84-014, July 1984.

5. D. P. Griggs, M. S. Kazimi and A. F. Henry, " TITAN: An Advanced Three Dimensional Coupled Neutronic/Thermal-Hydraulics Code for Light Water Nuclear Reactor Core Analysis," MIT-EL 84-011, June 1984.

TITAN Code Development for Application to A PWR Steam Line Break Accident

List of Figures

List of Figures (continued)

List of Table

I. Introduction

I.1 Background

The concern in a steam line break accident is the possibility of a return to reactor power. The sequence of events of a steam line break can be described as follows. Once the break occurs, the pressure at the break point drops to the environment pressure level. The large difference between the steam generator (secondary loop) system pressure and the pressure at the break point accelerates the secondary loop fluid and a blow down occurs. As a result of the blow down, the secondary side coolant can remove much more heat from the primary loop than in the normal operating condition. The consequence of the excess heat removal is that the coolant temperature at the inlet of the core will decrease with time, for a substantial period of time.

The effects of the primary loop coolant temperature drop are: 1) the primary coolant volume starts shrinking because the coolant density is increasing; 2) the primary system pressure keeps dropping. In addition, the reactor will trip because of the low pressure level of the secondary loop. Reactor scram reduces the power generation and enhances the primary coolant inventory shrinkage. Also, the fuel temperature drops after the scram.

Several competing factors affect the net reactor power generation. Factors for positive reactivity addition are: 1) coolant temperature drop; 2) coolant density increase; and 3) fuel temperature drop. Factors for negative reactivity addition are: 1) control rod insertion (scram); 2) possible void formation because of the system depressurization. This may occur once the saturation temperature of the coolant becomes lower than the coolant temperature.

A recent consideration, in response to post-TMI concerns, is to automatically start the auxiliary feed water pumps once a low steam generator pressure signal is received. The result of this action is that the secondary coolant mass flow rate at the steam generator inlet will increase at the early stage of the accident. The introduction of auxiliary feed water will enhance the positive reactivity addition (because more heat is removed from the primary loop).

Under nominal steam line break accident conditions, the negative reactivity insertion by scram should be sufficient to compensate for all positive reactivity addition and keep the system in a subcritical condition. However, based on the instructions of NRC, in accident analysis reports the highest worth control element assembly should be assumed as failing to fall into the coldest part of the

core. Even with this restriction, the core should still be in a subcritical condition. Otherwise, a return to power may occur and high power peaking may damage the fuel rods.

To simulate this more complicated accident, a complete core computer code is required. That is, a computer code should be a coupled code (thermal-hydraulics and neutronics) with coolant temperature, coolant density and fuel temperature feedback models. Also needed is the capability of simulating the system pressure drop as a function of time, and the inlet coolant mass flow rate change as a function of time. Furthermore, in order to investigate 3-D effects of the asymmetric probelm (the inlet coolant temperature varies from channel to channel), a 3-D code is necessary. Also, the 3-D effects of any local boiling in the later stage of a PWR accident require a two-phase flow model to simulate them.

The TITAN computer code [1,2,3] developed at M. I. T. is a complete coupled core code. The thermal-hydraulics part of the code is the THERMIT-2 code [4], which is a 3-D, two-phase, two-fluid, ten-equation code with the most advanced constitutive models. The neutronics part of the TITAN code is the QUANDRY code [5], which is a 3-D, 2 group, neutron diffusion nodal code. The advantages of the

nodal code are its efficiency and accuracy with large mesh sizes.

Basically, TITAN satisfied most of the requirements described above to simulate a steam line break accident. However, some modifications were needed to complete the requirements.

I.2 Organization of this Report

The work presented here demonstrates the capability of the TITAN code for steam line break accident simulations with a ten-channel PWR model. In addition, preliminary investigation of the 3-D effects, and hence the adequacy of 1-D modeling is included.

In section II the necessary code developments for steam line break accident simulations are described. In section III application to a quarter core model is described. In section VI the major conclusions are presented and some future work is proposed.

II. Code Development and Testing

Several modifications were made and tests were performed before TITAN was applied to simulate a steam line break accident in a PWR. These are described in the following sections.

II.1 Inlet Coolant Temperature Transient Forcing Function

The original TITAN had the capability of simulating the core inlet coolant temperature as a function of time. However, only one forcing function could be employed for all channels. As mentioned before, the steam line break accident is an asymmetric problem, and the inlet coolant temperature would not necessarily be the same for all channels. A modification has been made to extend the capability of the code so that a more flexible forcing function can be employed for different channels at the core inlet.

The code now has the capability to simulate up to 10 regions with 10 different inlet temperature forcing functions for each region. Each region can have at most 50 channels. This is quite enough even for a whole core analysis. Usually, it is acceptable to consider three regions for a steam line break accident. That is, one region for the cold part (broken side), one region for the hot part (intact side), and one region for the mixing part (between the broken side and the intact side).

A ten-channel test problem was studied to test this new modified capability. The cross section geometry is shown in Figure 1. Channels 2, 3, 4, and 7 are in the hot region. Channels 5, 8, 9, and 10 are in the cold region. Channels 1 and 6 are in the mixing region. Each region has its own temperature forcing function, as shown in Figure 2.

Two items ought to be checked in the code output. The first check is whether the inlet coolant temperature is changing as described by the given forcing function for each region. From the output of a sample calculation, this has been ensured. The second thing to check is the core power history. Since nothing is changing except the inlet coolant temperature which keeps dropping, positive reactivity is added by the coolant temperature, coolant density coefficients and the fuel Doppler feedback effect. Therefore, a power excursion is expected.

Four temperature transients were studied. All the calculations restarted from steady-state results of the 10-channel model, with open channel (for thermal-hydraulics, see section III.2, Model B). The four calculations are:

Case (1) Open-channel uniform inlet coolant temperature distribution case,

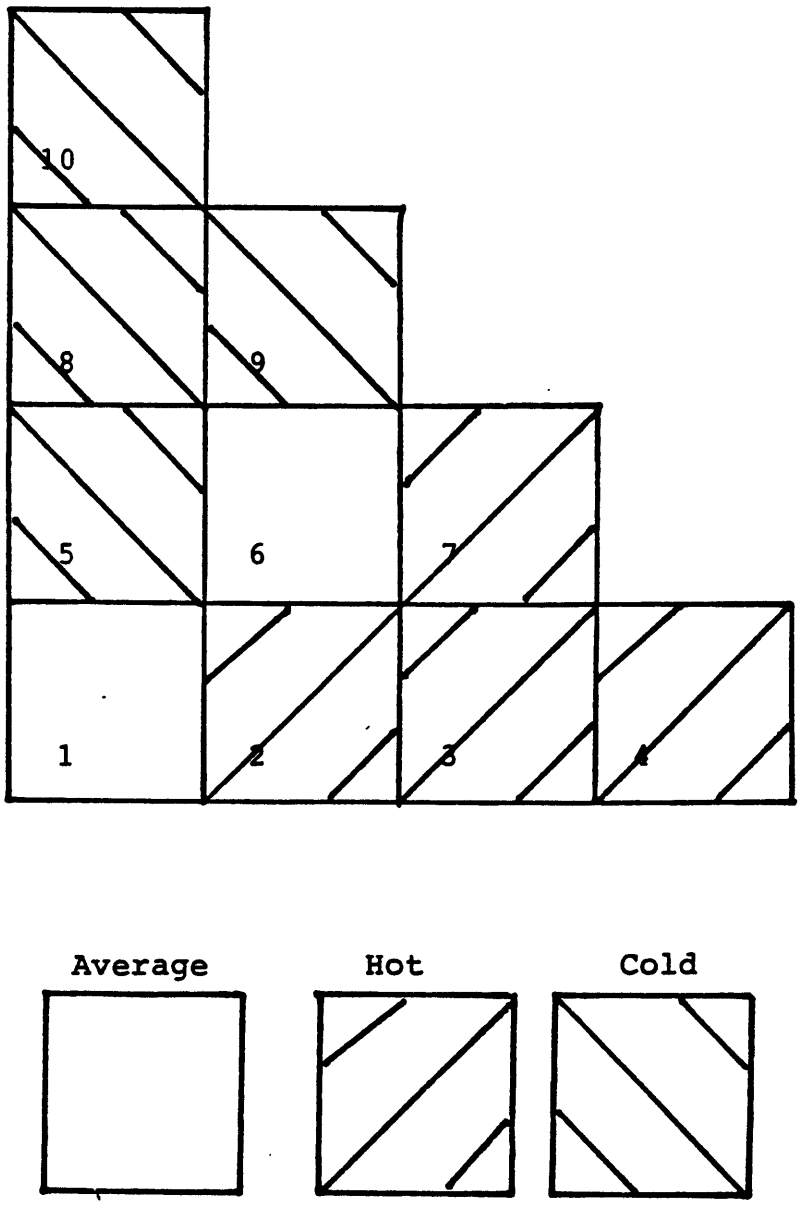


Figure 1 Three inlet coolant temperature zones steam line break transient simulation.

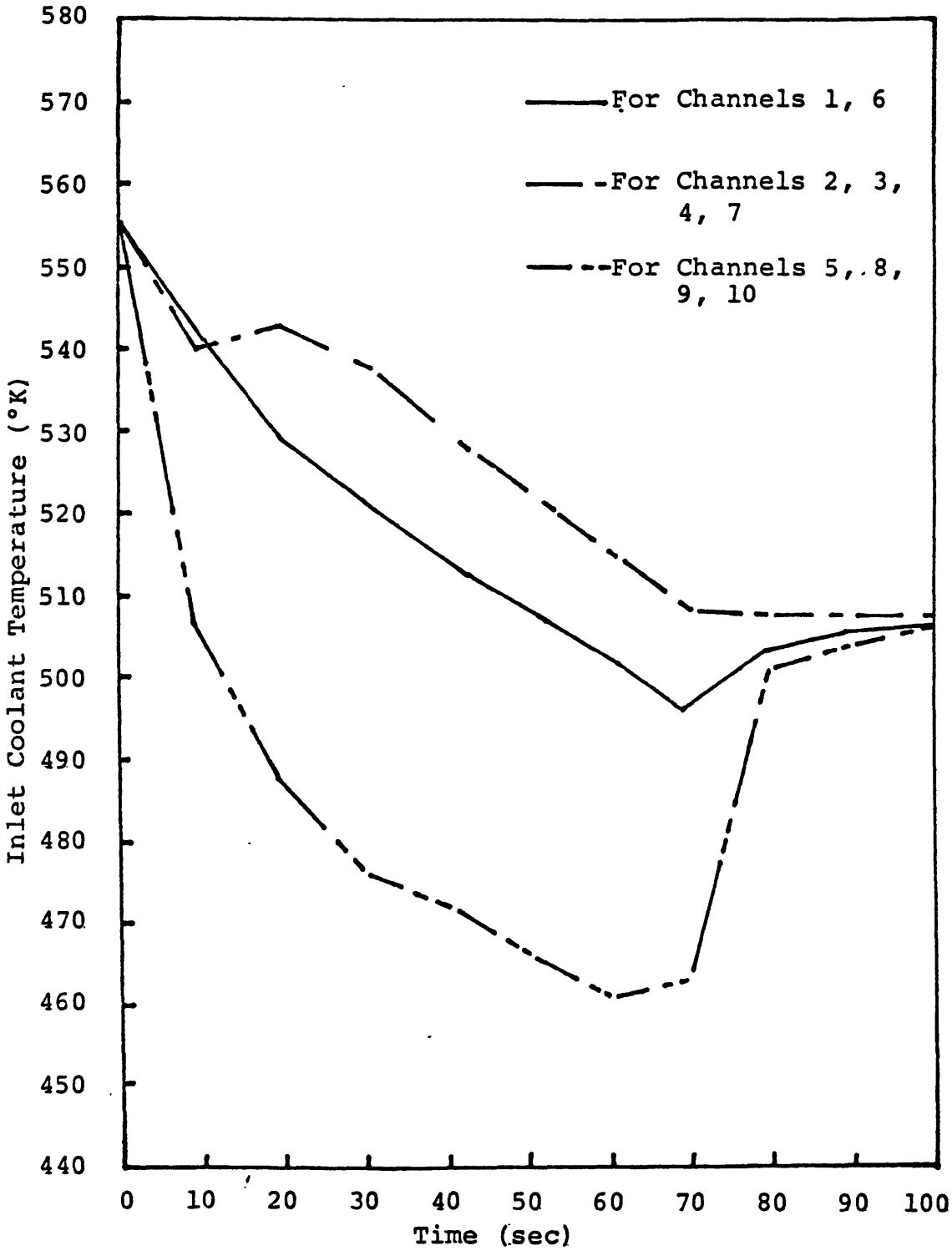


Figure 2 Inlet coolant temperature v.s. time, non-uniformly distributed case.

Case (2) Closed-channel uniform inlet coolant temperature distribution case,

Case (3) Open-channel non-uniform inlet coolant temperature distribution case,

Case (4) Closed-channel non-uniform inlet coolant temperature distribution case.

The non-uniform inlet coolant temperature cases included three inlet coolant temperature zones as described before. The forcing function for the mixing temperature zone is also used as the forcing function for cases (1) and (2).

No significant differences between cases (1) and (2), or cases (3) and (4) were observed. The maximum differences of the total power for cases (1) and (2), and cases (3) and (4) are about 0.003% and 0.006%, respectively. The power histories for cases (1) and (3) are shown in Figure 3. Both cases have power excursions. as expected. The non uniform case has a higher power history than the uniform case. This is because of the large temperature drop in the cold region.

The power histories in Figure 3 also show the correlation between the inlet coolant temperature forcing function and the results. As can be seen in Figure 2, at about 70 seconds, the coolant temperature starts to rise again. The reactivity feedback, therefore, should be negative.

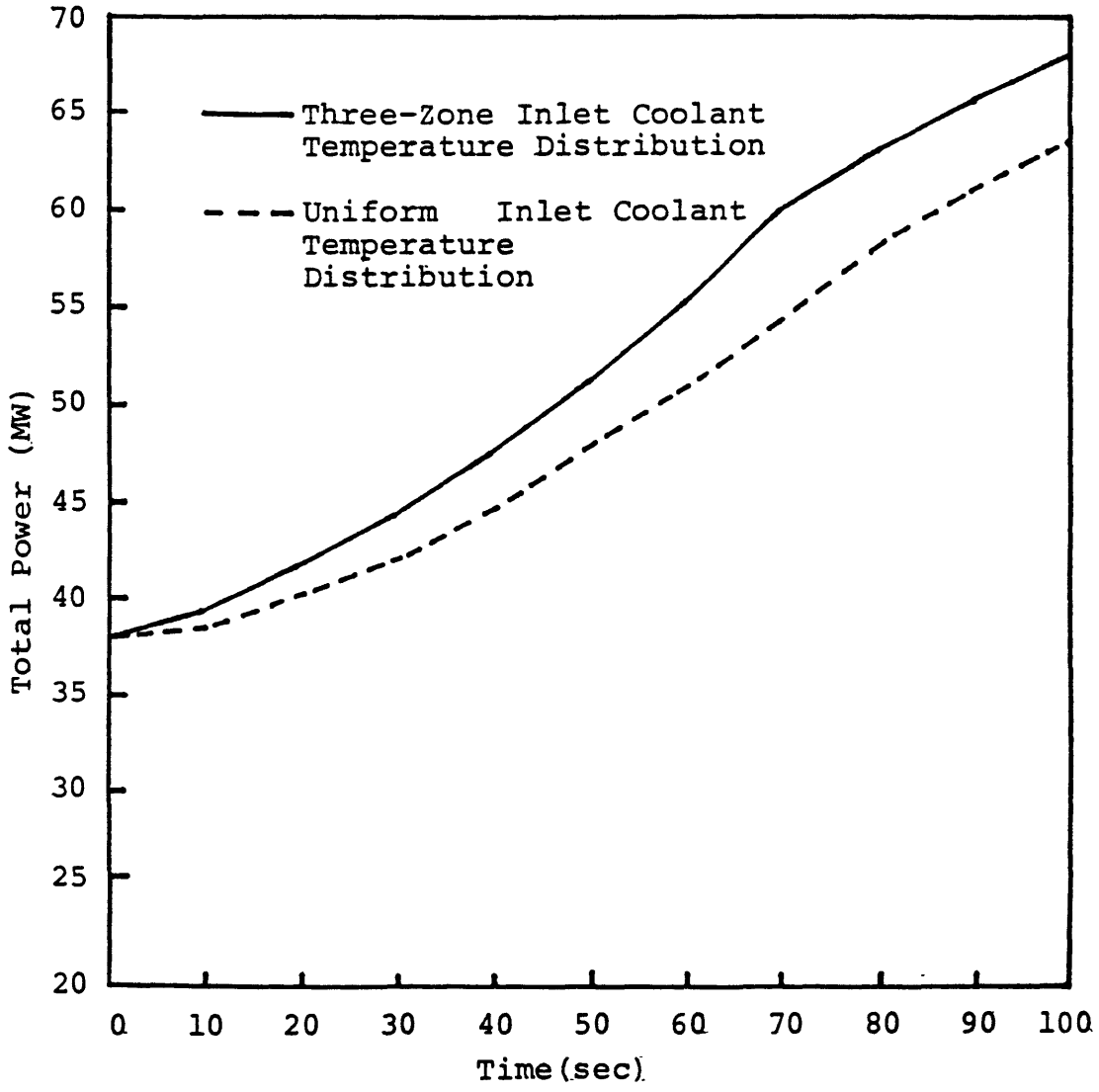


Figure 3 Power Histories of 10-Channel PWR Steam Line Break Transient Simulations.

This is reflected in Figure 3 where the slopes for both cases started to decrease at about 70 seconds.

The computation statistics are summarized in Table 1. From this table, we can see that about 14% cpu time was saved for case (4) compared to case (3). However, only about 4% cpu time was saved for case (2) compared to case (1).

In addition, the cold region is expected to have a higher power than the other two regions. Figure 4 shows the correct trends of our testing calculation.

In general, the test calculations of the new inlet coolant temperature transient forcing function capability have demonstrated two points: 1) reasonable trends were predicted by TITAN for this kind of transients; 2) the transient resutls do not depend on whether the channels were closed or open. However, one should be aware that no boiling was predicted for all the four cases.

II.2 Scram Simulation

One of the major simulation needs during a steam line break accident is the reactor scram as mentioned in section I.1. Before performing a steam line break accident simulation, we would like to make sure that the code predicts correct scram results. With the same testing problem described in section II.1, we performed a scram only tran-

Table 1

Coolant Temperature Transient-Only Test Case

Computation Statistics

Case	Time Steps	CPU Time (sec)	CPU Time/ Step/ Node
Uniform Open-Channel	2000	8150.03	0.041
Uniform Closed-Channel	2000	7828.58	0.039
Non-Uniform Open-Channel	2000	9364.40	0.047
Non-Uniform Closed-Channel	2000	8034.10	0.040

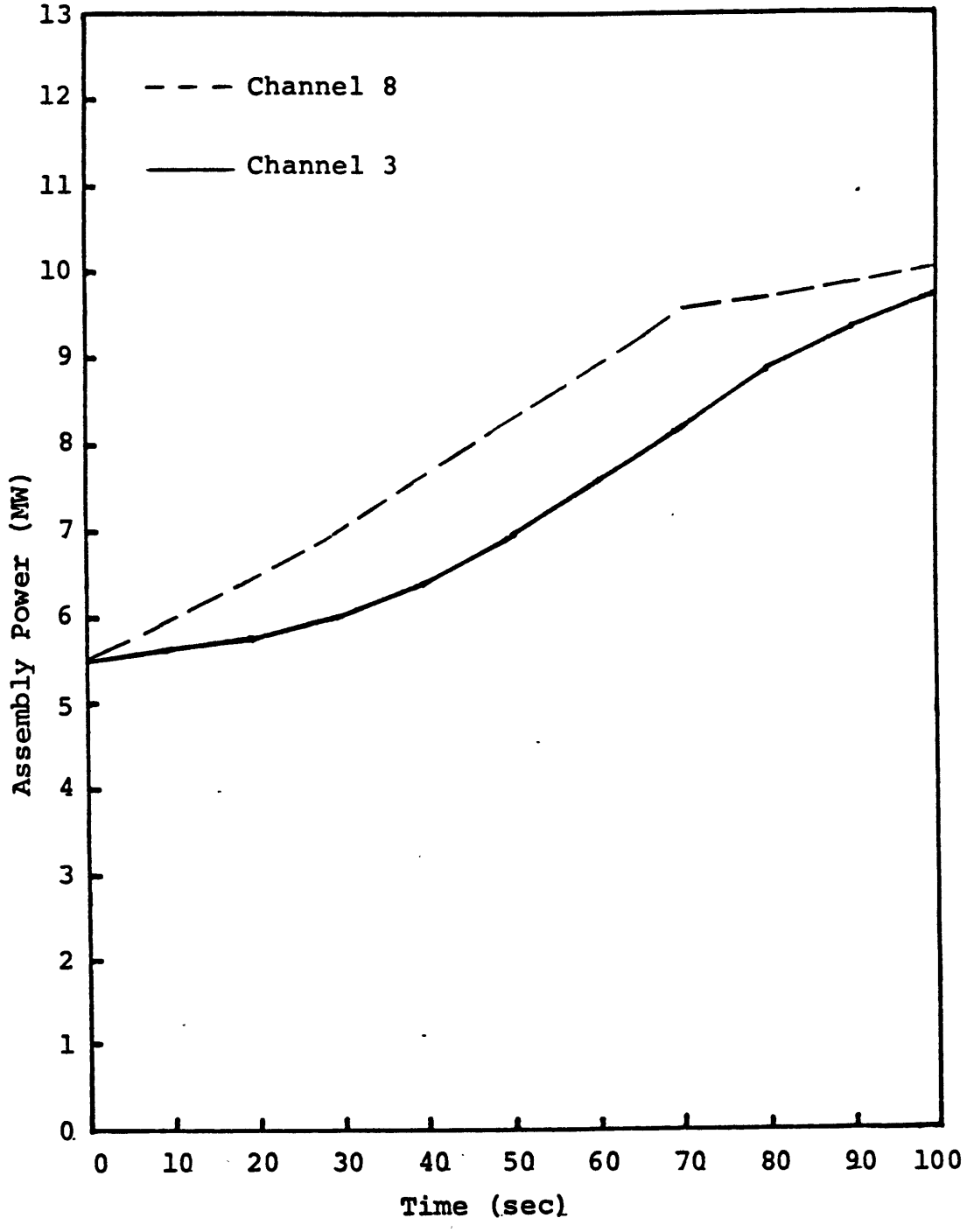


Figure 4 Assembly power histories of steam line break transient for ch. 3,8 non-uniform inlet coolant temperature distribution case.

sient calculation. The two partially inserted control rods fall into the core starting at t=2.5 seconds and ending at t=5 seconds after the transient has begun.

The expected result of a scram is a reactor power drop because of the large negative reactivity addition to the core. As seen from Figure 5, the total core power history decreases after 2.5 seconds. There is no power change during the first 2.5 seconds because there is no scram during that period. It should be pointed that in TITAN, no fission product decay heat is included. Only fission heating is calculated.

This calculation gives confidence in the code's scram simulation capability.

II.3 Flow/Temperature Transient plus Control Rod Transient Option and the Transient Restart Capability

A logic modification was done to provide the user an option to simulate events which include both control rod movement and flow/coolant temperature transient. This is necessary for a steam line break accident simulation. The original code could handle either a control rod transient or a flow transient, but not both transients simultaneously.

Also, the transient restart capability has been fixed and tested. With this capability, it is easier to perform a

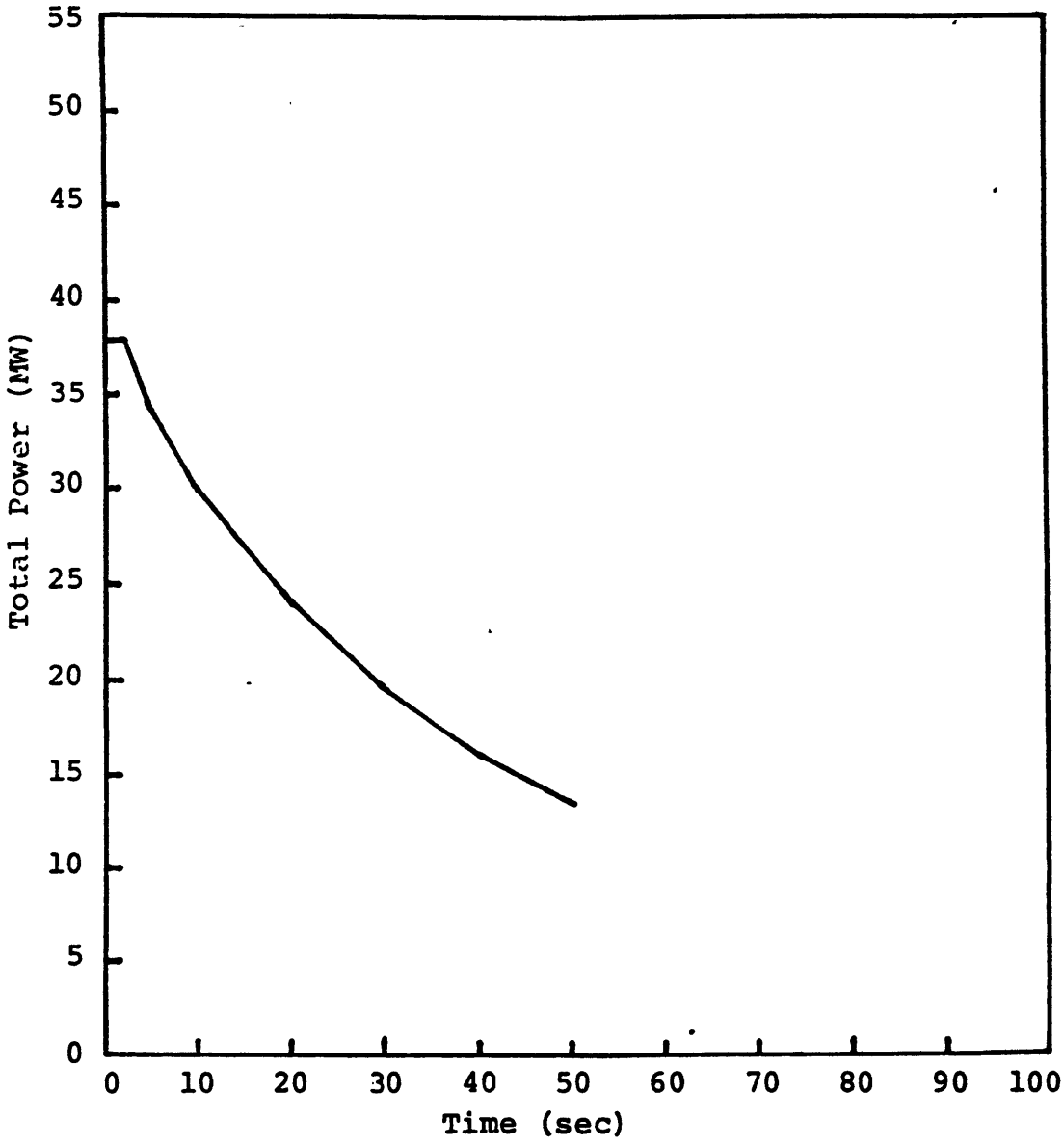


Figure 5 Power history of 10-channel PWR scram transient without temperature transient.

lengthy transient calculation, such as the problems involved here. The simulation conditions can be changed during a transient calculation for various purposes.

II.4 Total Inlet Coolant Mass Flow Rate Boundary Condition

A total inlet mass flow rate boundary condition is required for transients that only their total inlet flow rates are known.

The approach of previous THERMIT work was adopted [6]. The idea of the total inlet flow rate boundary condition is that, for a given total inlet flow rate, the code will calculate the lower plenum pressure, P_{ℓ}, based on the current guess for pressures inside the domain. Then, an additional equation is solved as part of the usual pressure solution. The resulting correction, δP, is used to update the pressure in the plenum, and the other pressure corrections are used to update the pressures inside the domain.

Once the new lower plenum pressure and the new pressures of entrance nodes of the core are obtained, the corresponding inlet coolant velocities of all channels are calculated. This is very important for steam line break accident simulations, when the only available information is the total inlet flow rate, instead of the inlet coolant velocities of various channels. If channel velocity bounda-

ry conditions at the core inlet are used, inaccuracy will be introduced.

The next modification is the capability of the code to simulate total inlet coolant flow rate transients. This is necessary since the inlet flow rate is a function of time for steam line break accidents.

By checking a sample calculation output, it has been proven that the total inlet flow rate follows the given flow transient forcing function.

Detailed user guidance is provided in the updated TITAN User's Guide which is attached to this report as an Appendix.

III. Application to a Quarter Core Model

III.1 10-Channel PWR Quarter Core Models

Two similar 10-channel PWR quarter core models were employed for our investigation. Figure 6 shows the x-y plane cross section view of these models. Channels 3 and 8 have partially inserted control rods (~45%) (see Figure 7). There is no control rod in channel 1 for Model A, but a fully inserted control rod for Model B. Model A is used to simulate the case of stuck control rod outside the core during a scram. Model B is used to test the general behavior of the TITAN code in dealing with steam line break accident simulations.

The nuclear composition distribution is shown in Figures 6 and 7. The nuclear data were taken from a BNL report [7]. The neutronic boundary conditions along the core center lines (x-y plane) are zero neutron current because of the symmetric geometry. Both top and bottom of the core have albedo boundary conditions. The outer side of the core (x-y plane) has an albedo boundary condition also.

Axially, there are 10 nodes for each channel, in addition to the fictitious boundary nodes. The first node (the bottom one) has no fuel rod. This is the node used to simulate part of the lower plenum. The flow distribution is calculated in this node when the total inlet flow rate

I. 10-Channel Artificial PWR 1/4th Core Model

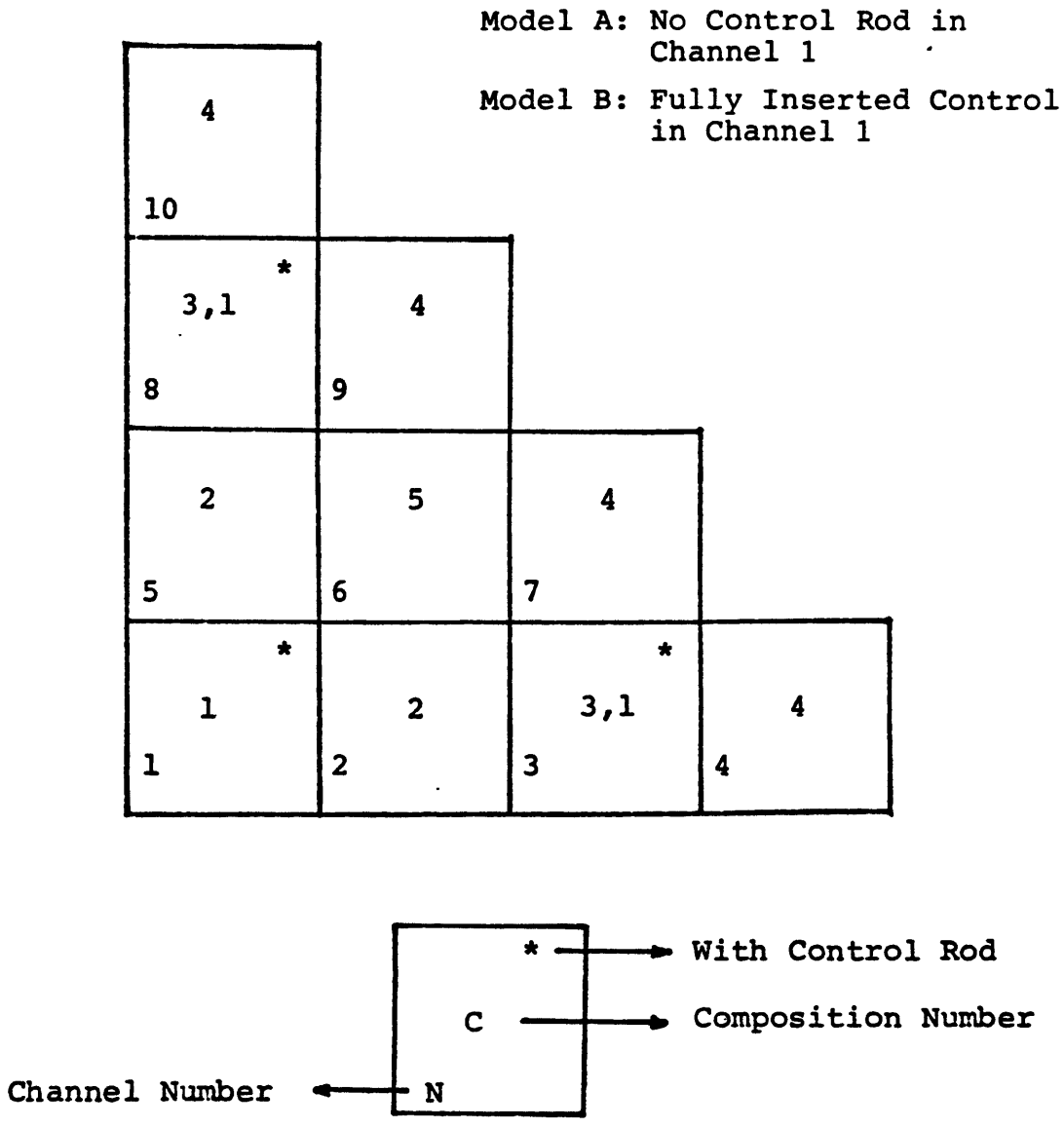


Figure 6 Composition layout for 10-channel steam line break transient.

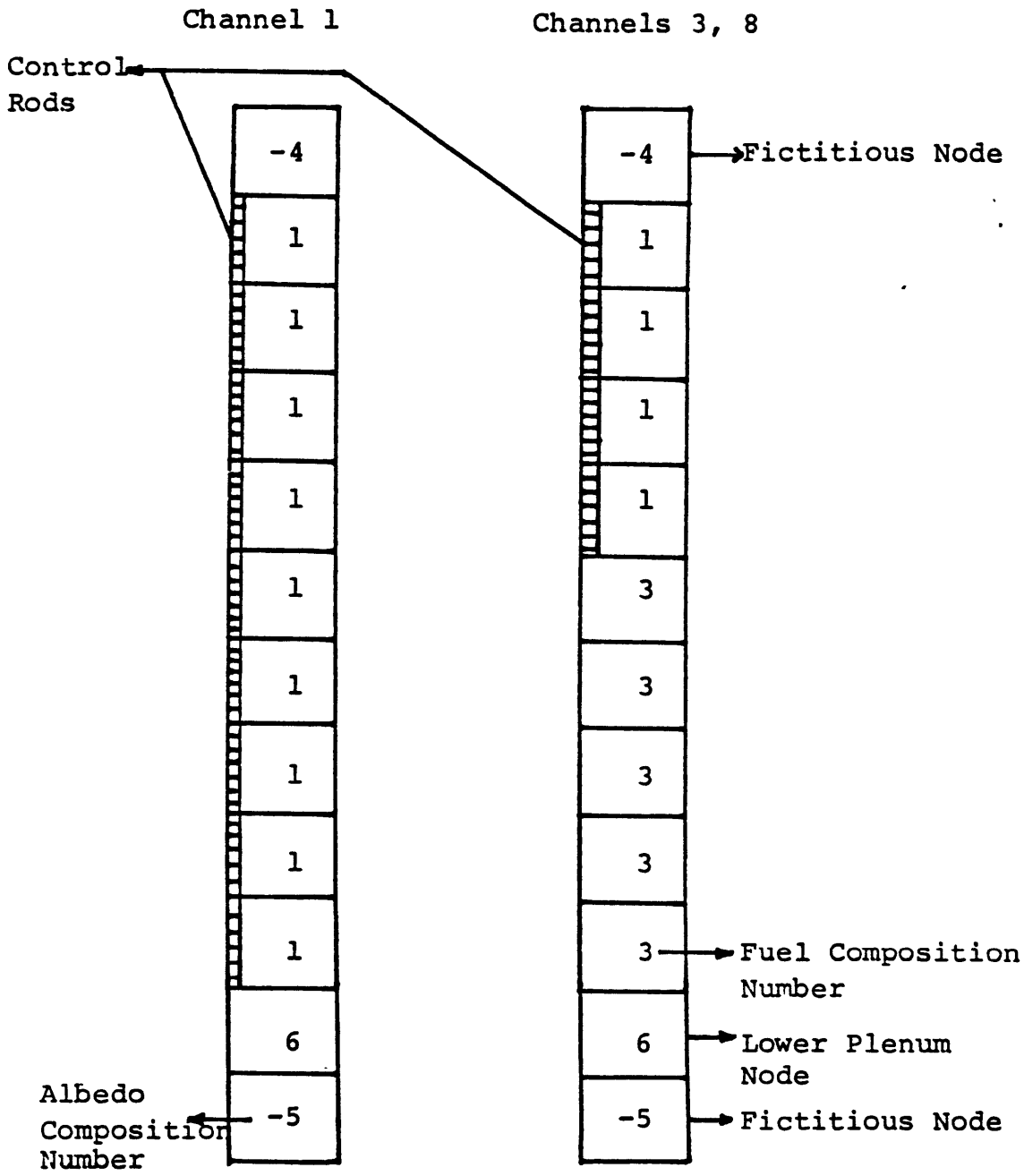


Figure 7 Assemblies with control rod inserted.

boundary condition is chosen. Channel 1 has only quarter size of a normal channel. Channels 2, 3, 4, 5, 8, and 10 have only half the size of a normal channel. With this arrangement, an exact quarter core is simulated.

III.2 Steady-State Simulation

The steady-state core power (quarter core) was 37.95 MW. The total inlet coolant flow rate was 651.2 Kg/sec (quarter core). The system (exit) pressure was 15.65 MPa. The inlet coolant temperature was 555°K. For the thermal hydraulics part calculation, exit pressure and total inlet coolant flow rate boundary conditions were chosen for top and bottom of the core, respectively. About 5% core power was assumed to be the direct heating power from coolant.

For both Models A and B, the convergence criteria are the same:

pressure iteration convergence crit. = 1.0×10^{-7}

newton iteration convergence crit. = 1.0×10^{-7}

eigen value convergence crit. = 1.0×10^{-6}

power convergence crit. = 1.0×10^{-6}

The procedure of performing steady-state calculations is described as follows. At the beginning, all channels were closed and no cross flow among channels was allowed. After several time steps calculation, the nuclear cross section average option (see Appendix for parameter "ixavg")

was used to accelerate the convergence of the results. Once converged closed-channel results were obtained, the calculation was restarted with channels opened. Finally, converged open-channel results were obtained.

To judge if the result is converged or not, five parameters must be checked. The definitions of the five parameters are:

flow error = (exit mass flow rate / inlet mass flow rate) ^{n} - 1.0

energy error = (total core enthalpy rise / core power) ^{n} - 1.0

Win = (calculated inlet mass flow rate / input inlet mass flow rate) ^{n} - 1.0

wchk = [(total core exit mass flow rate) $^{n-1}$ / (total core exit mass flow rate) ^{n}]- 1.0

qchk = [(total core enthalpy rise) $^{n-1}$ / (total coreenthalpy rise) ^{n}]- 1.0

where n means at time step n, and n-1 means at time step n-1.

The "Win" parameter is to be checked when the total inlet flow rate option is chosen. The parameters "wchk" and "qchk" are used to make sure that the results are not only converged within a time step (checked by flow and energy

errors), but also converged timewise (i.e., real steady state results).

Table 2 gives the results of the above five parameters for both Model A and Model B. Closed-channel results and open-channel results are shown in the Table for each model. Convergence is achieved with a flow error of less than 0.1% and energy error of about 2%.

Figure 8 shows bottom peaking of the axial power distributions for both models. This is expected because of the top inserted control rods. Also, from Figure 8, we can see that the power drops in the top part of channel 3 because of the partially inserted control rods. The average power and peak power of channel 3 with Model A is lower than that of channel 3 with Model B. Recall that in Model A, there is no control rod in channel 1. With equal total core powers of the two models, the power of channel 1 with Model A is much higher than that with Model B. Therefore, the power of channel 3 with Model A is lower than that of channel 3 with Model B.

The cpu usages of the two models are summarized in Table 3. The cpu time per time step per node is much higher than the values shown in Table 1 because steady-state calculations require more iterations within each time step. An interesting investigation was made to determine the cpu

Table 2

Steady-State Convergence Criteria

Model A:

	Closed-Channel	Open-Channel
Energy Error	-1.9183×10^{-3}	-1.9734×10^{-2}
Flow Rate Error	1.1878×10^{-4}	-7.9465×10^{-4}
Win	-1.1470×10^{-5}	2.3080×10^{-6}
qchk	-4.4010×10^{-4}	2.1927×10^{-4}
wchk	4.1833×10^{-5}	2.3080×10^{-6}

Model B:

Energy Error	-2.1088×10^{-3}	-1.9016×10^{-2}
Flow Rate Error	1.1268×10^{-4}	-7.6004×10^{-4}
Win	4.1357×10^{-6}	-4.9676×10^{-6}
qchk	3.8178×10^{-5}	-9.9310×10^{-6}
wchk	-4.1239×10^{-4}	-3.1878×10^{-4}

Table 3

Steady-State Computational Usage

Model A:

	Time Step	CPU Time (sec)	CPU Time/ Time Step/ Node
Closed-Channel	20	647.86	0.324
Open-Channel	10	321.78	0.32178

Model B:

Closed-Channel	20	527.23	0.2636
Open-Channel	10	246.94	0.24694

time used for the neutronic calculations in each time step, with the help of the parameter "nfeed". The significance of "nfeed" can be explained with an example. If nfeed=2, that means there will be 2 thermal-hydraulic calculations per one neutronic calculation. More explicitly, with nfeed=2, time step 1 includes both thermal-hydraulic and neutronic calculations. In time step 2, only thermal-hydraulic calculation is performed. In time step 3, both calculations are performed again. For steady-state calculations, our conclusion is that, with nfeed=1, about 78% of the cpu usage is used for the neutronic calculation in each time step. Whenever converged and accurate steady-state results can be obtained with "nfeed" larger than 1, significant cpu time will be saved.

As to the comparison of the closed-channel results and open-channel results, no significant difference was found for both models (see Figure 8). The conclusion of this observation is that a 1-D numerical scheme should be sufficient for steady-state calculations. A more efficient and faster scheme should be added to the code as an option for steady-state calculations.

III.3 Transient Results and Discussions

One test calculation was made with Model B and three calculations were made with Model A. All of them included

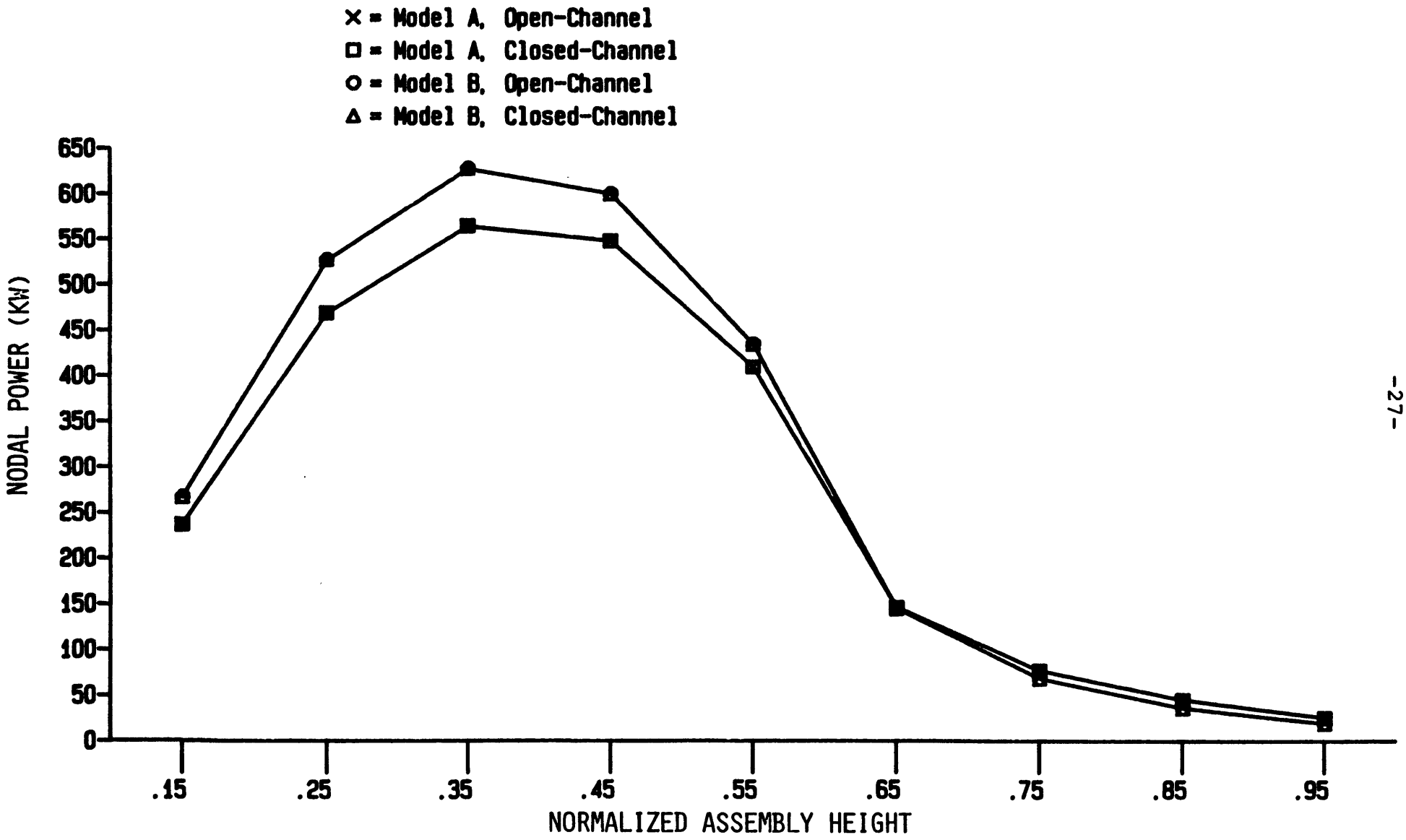


FIGURE 8 STEADY-STATE AIXAL POWER DISTRIBUTION OF CHANNEL 3, FOR MODELS A AND B.

both open and closed-channel cases and non-uniform inlet temperature distribution was employed in both cases. The inlet coolant temperature forcing function of the test calculation made with Model B was based on the Final Safety Analysis Report of Maine Yankee Nuclear Power Plant [8]. The three calculations made with Model A were based on functions used in Ref. [9].

III.3.1 Test Calculations

The test calculation actually is a continuation of the two test calculations described in Section II.1 and II.2. In Section II.1, an inlet coolant temperature transient was presented. In Section II.2, a scram- only simulation was described. These are the two major parts of a steam line break accident simulation. A combined calculation was made to give confidence in TITAN's steam line break accident simulations. The test case used a combination of non-uniform inlet coolant temperature transient and scram transient with an open-channel model.

The transient was followed for 100 seconds. No boiling was observed throughout the entire calculation. The total power (see Figure 9) rose a bit for the first 2.5 seconds because of the inlet coolant temperature drop. Then, once the scram started, the power kept decreasing. The rate of power reduction is not very fast. The reasons may be: 1)

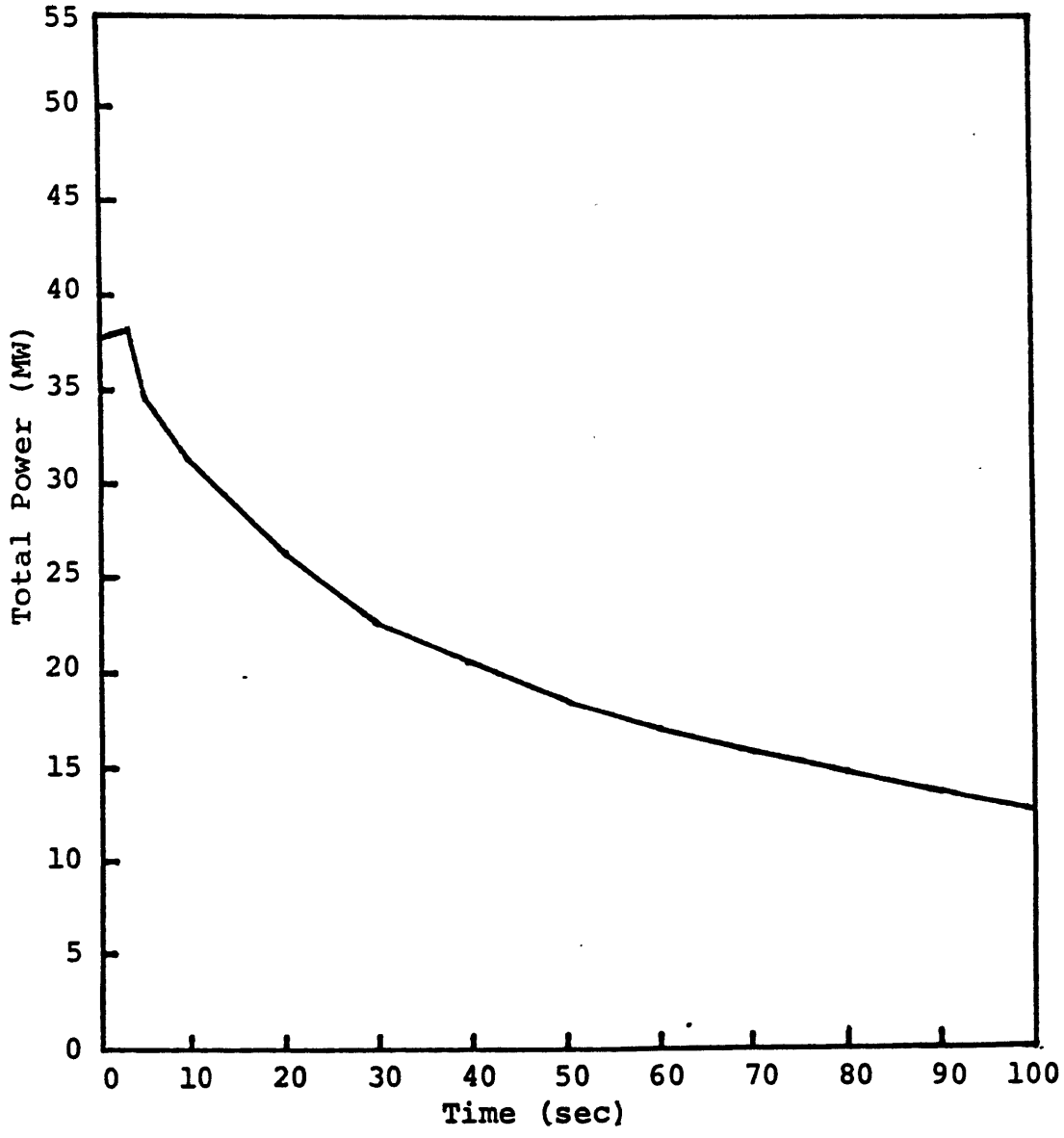


Figure 9 Power history of 10-channel PWR steam line break transient with scram.

the strong effect of the large coolant temperature drop in the cold region; 2) the control rods worth is not large enough to speed the power drop.

Figures 10 and 11 show the radial power distributions. As expected, two valleies occur at channels 3 and 8 where the control rods exist.

From these results, we see that TITAN predicted what was expected to happen based on the physical behavior. This gave confidence in TITAN's capability for simulating this kind of accident.

III.3.2 Original Yankee Atomic Steam Line Break Pressure Forcing Function Case, YA-1

Three calculations were performed with Model A of the 10-channel geometry. All of them are based on a Yankee Atomic report [9]. The first one, designated case YA-1, used the transient forcing functions as stated in the report.

Figure 12 shows the three inlet coolant temperature forcing functions. The main difference between these functions and Maine Yankee's (Figure 2) is that there is no inlet temperature increase after 70 seconds. Figure 13 shows the system pressure forcing function which was used at the top of the core. Figure 14 shows the total inlet coolant mass flow rate forcing function. Before 60 seconds,

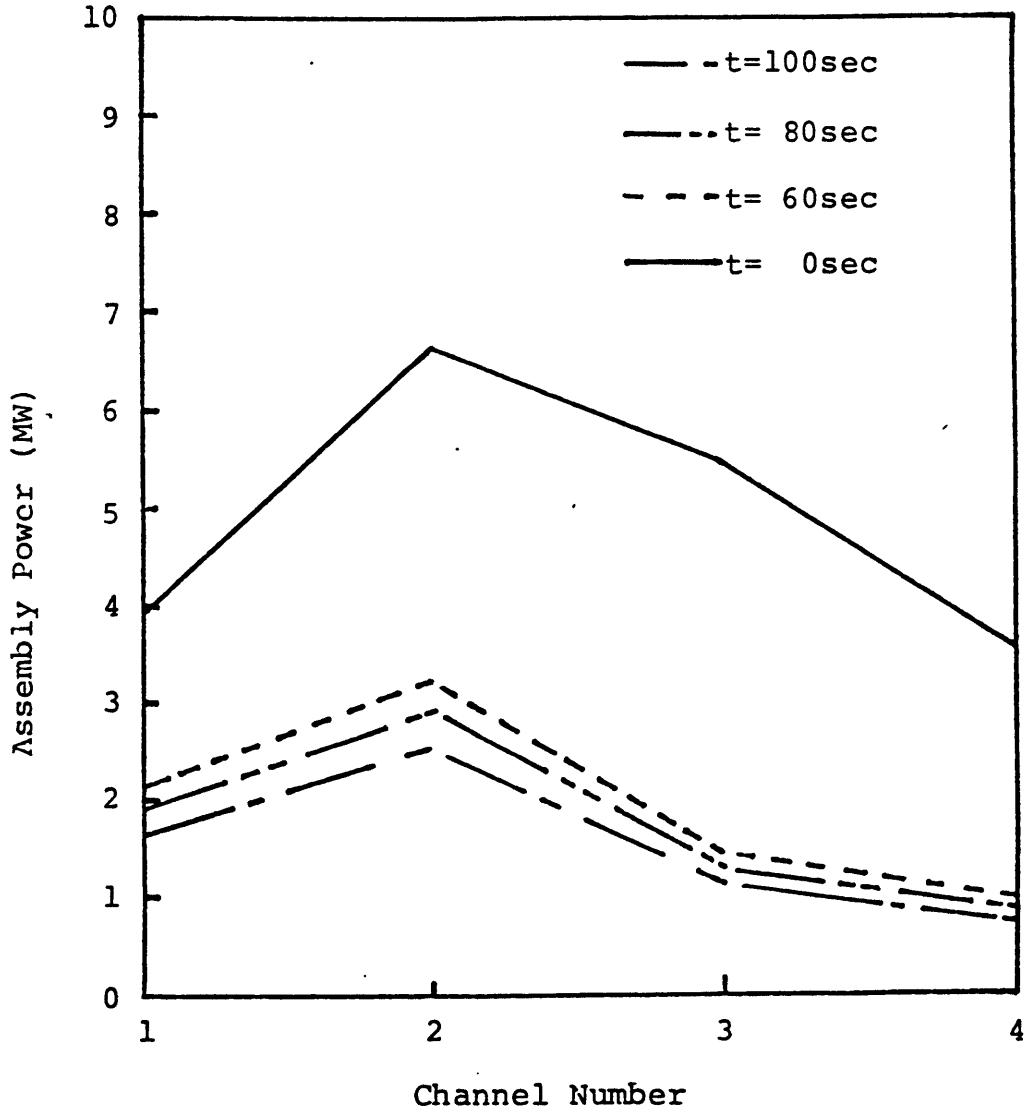


Figure 10 Radial assembly power distribution v.s. time for channels 1, 2, 3, and 4. Scram and temperature feedback.

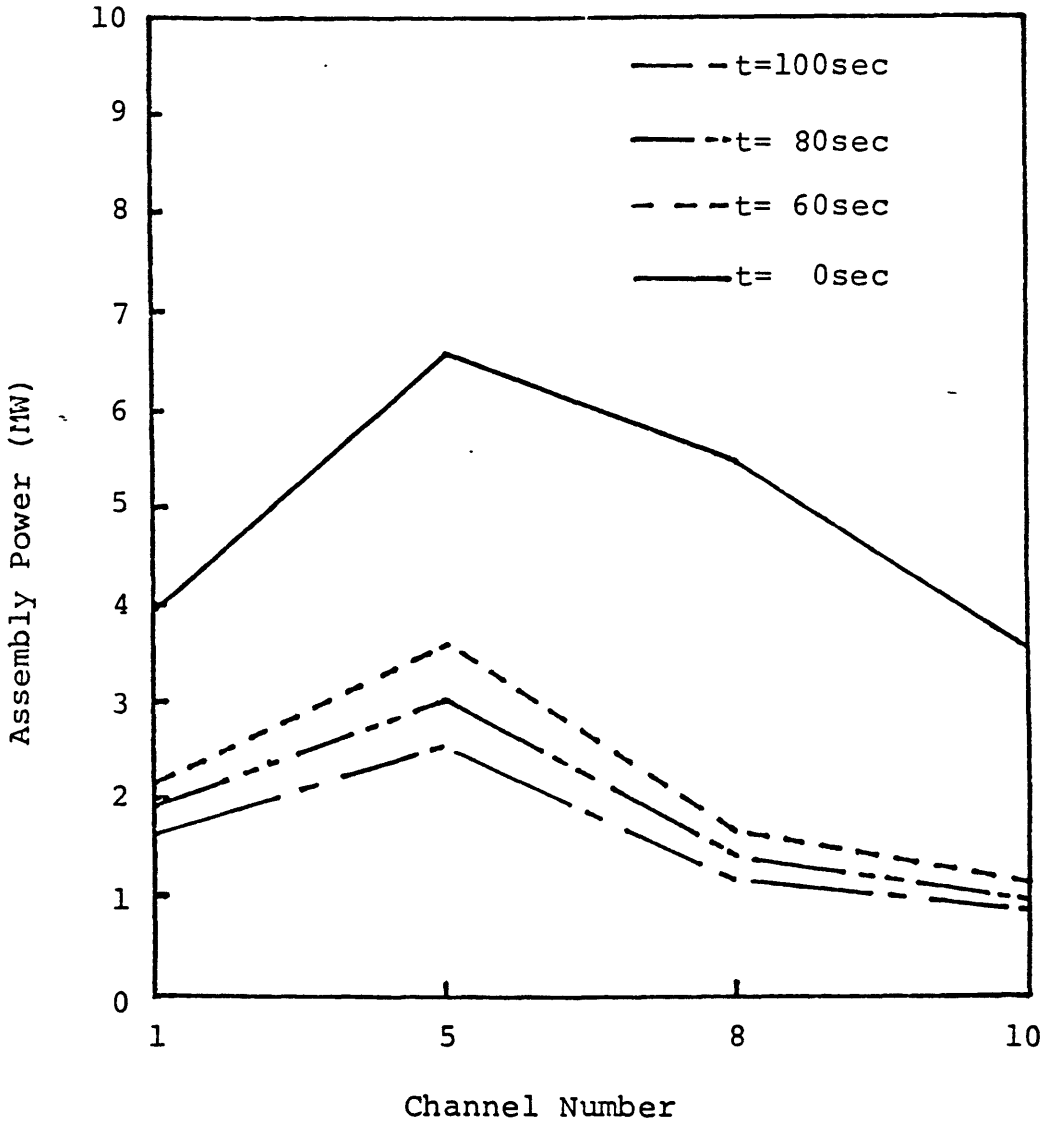


Figure 11 Radial assembly power distribution v.s. time for channels 1, 5, 8, and 10. Scram and temperature feedback.

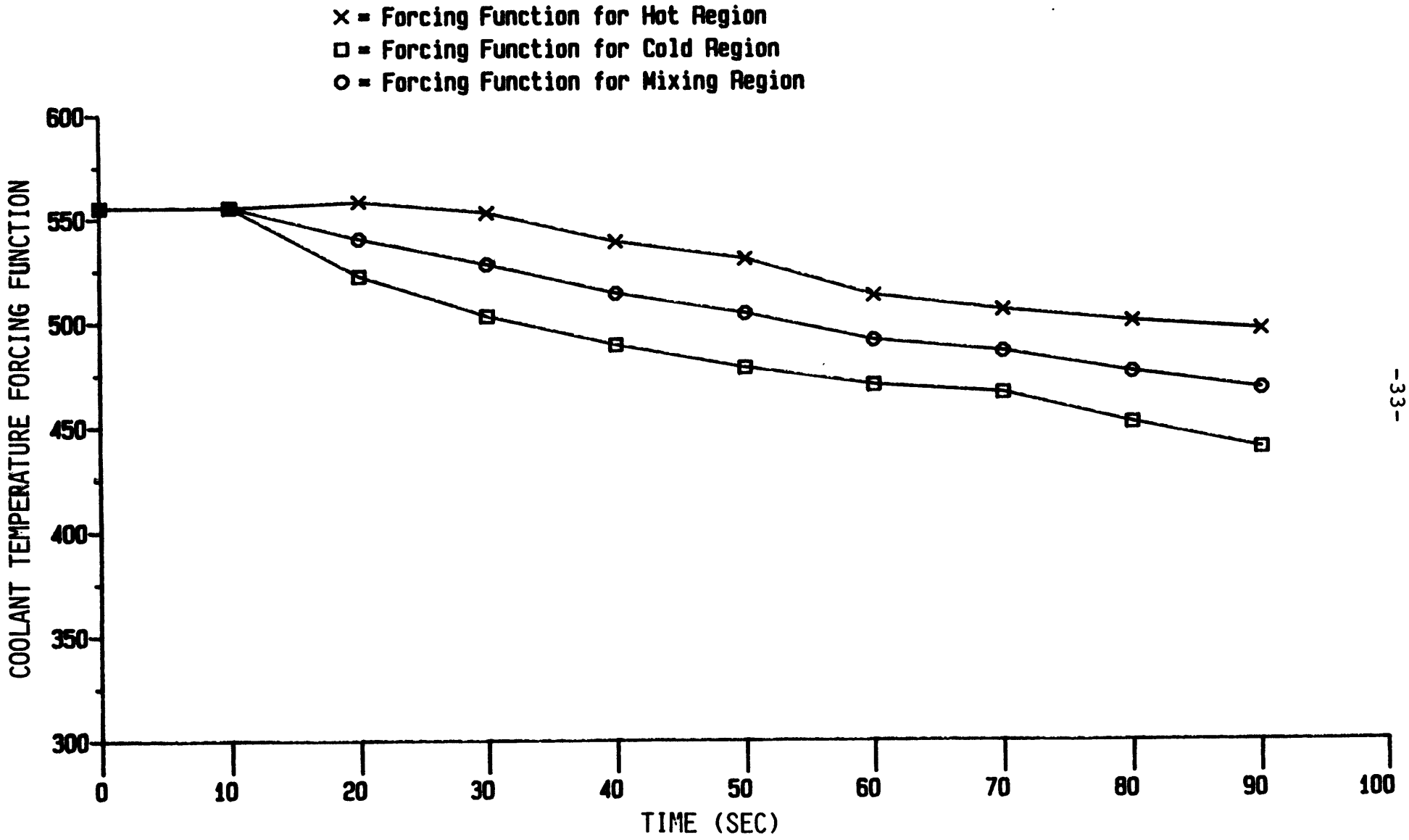


FIGURE 12 COOLANT TEMPERATURE FORCING FUNCTION FOR CASES YA-1, YA-2, AND YA-3.

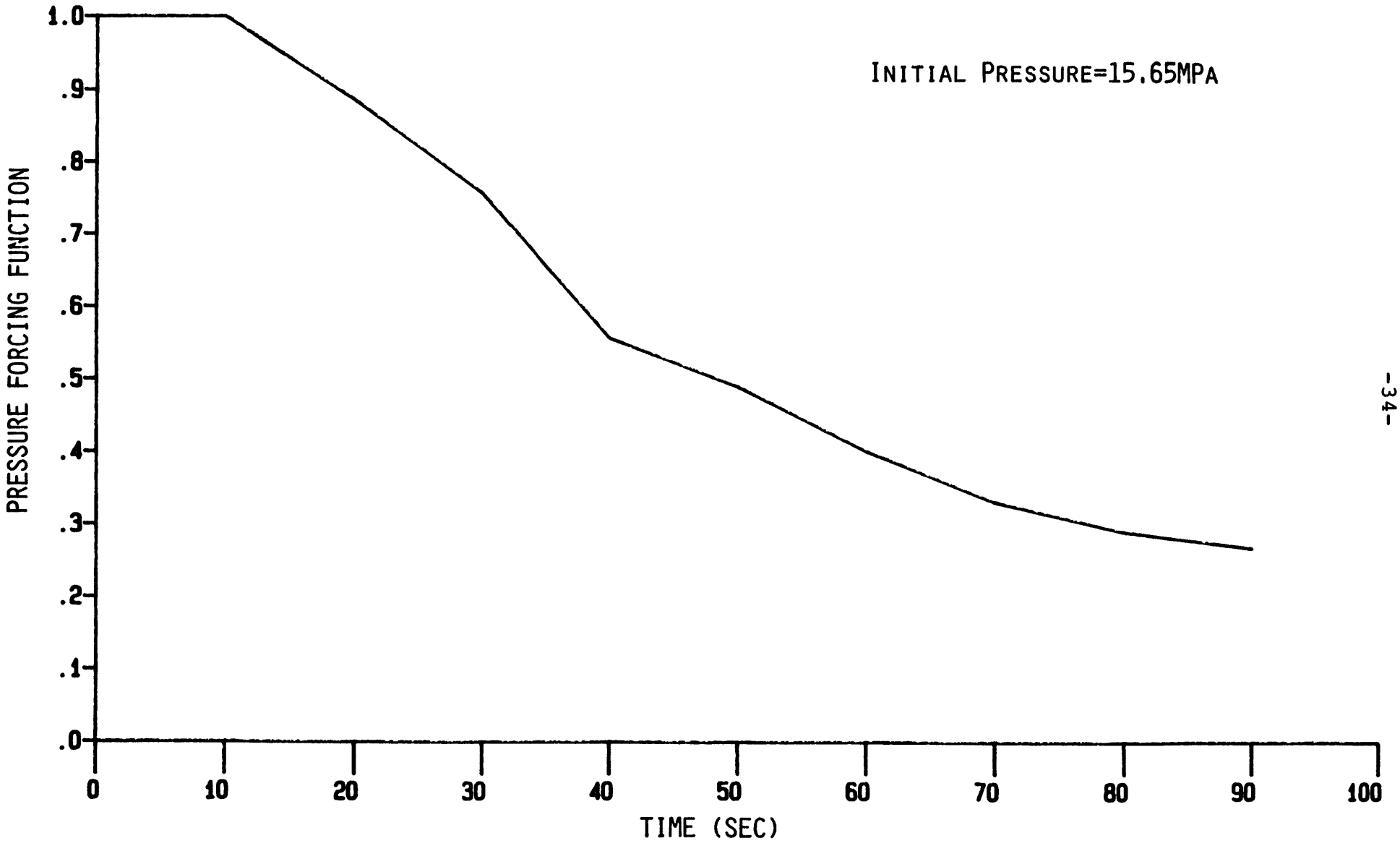


FIGURE 13 PRESSURE FORCING FUNCTION FOR CASE YA-1

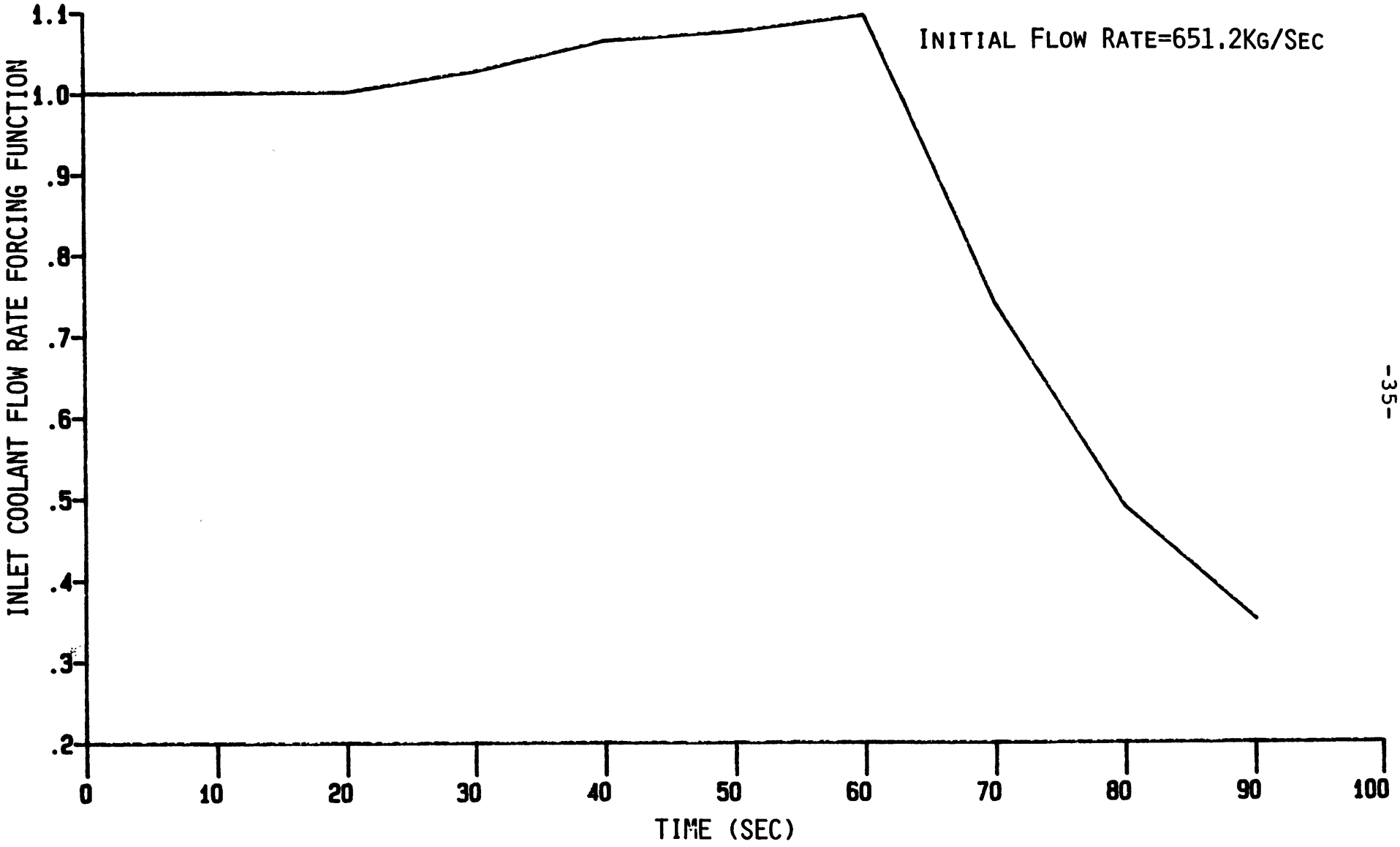


FIGURE 14 INLET COOLANT FLOW RATE FORCING FUNCTION FOR CASE YA-1

the total inlet flow rate increases because of the increase of the coolant density. The reactor coolant pumps are tripped at 60 seconds, therefore, the inlet flow rate drops sharply after 60 seconds. Scram started at 10.9 seconds after the transient had begun, and ended at 15.9 seconds. All other initial conditions are the same as the test case described before.

The transient was followed up to 90 seconds. Again, no boiling was observed in this calculation. Figure 15 shows the power history of this transient. At the period of 11 to 20 seconds, the power drops sharply because of the scram action. Later on, the slope of the curve gradually decreases. This corresponds to the inlet coolant temperature forcing functions.

Figure 16 shows the coolant saturation temperature history, which corresponds to the system pressure history, and the fuel wall temperature history of the hottest node (channel 2 node 6). As we can see, these two curves are approaching each other as time goes by. At 90 seconds, they meet together.

Up to now, no boiling was observed and no significant cross flow was observed, either. Neutronically, TITAN did predict a higher power profile in the cold region than that in the hot region.

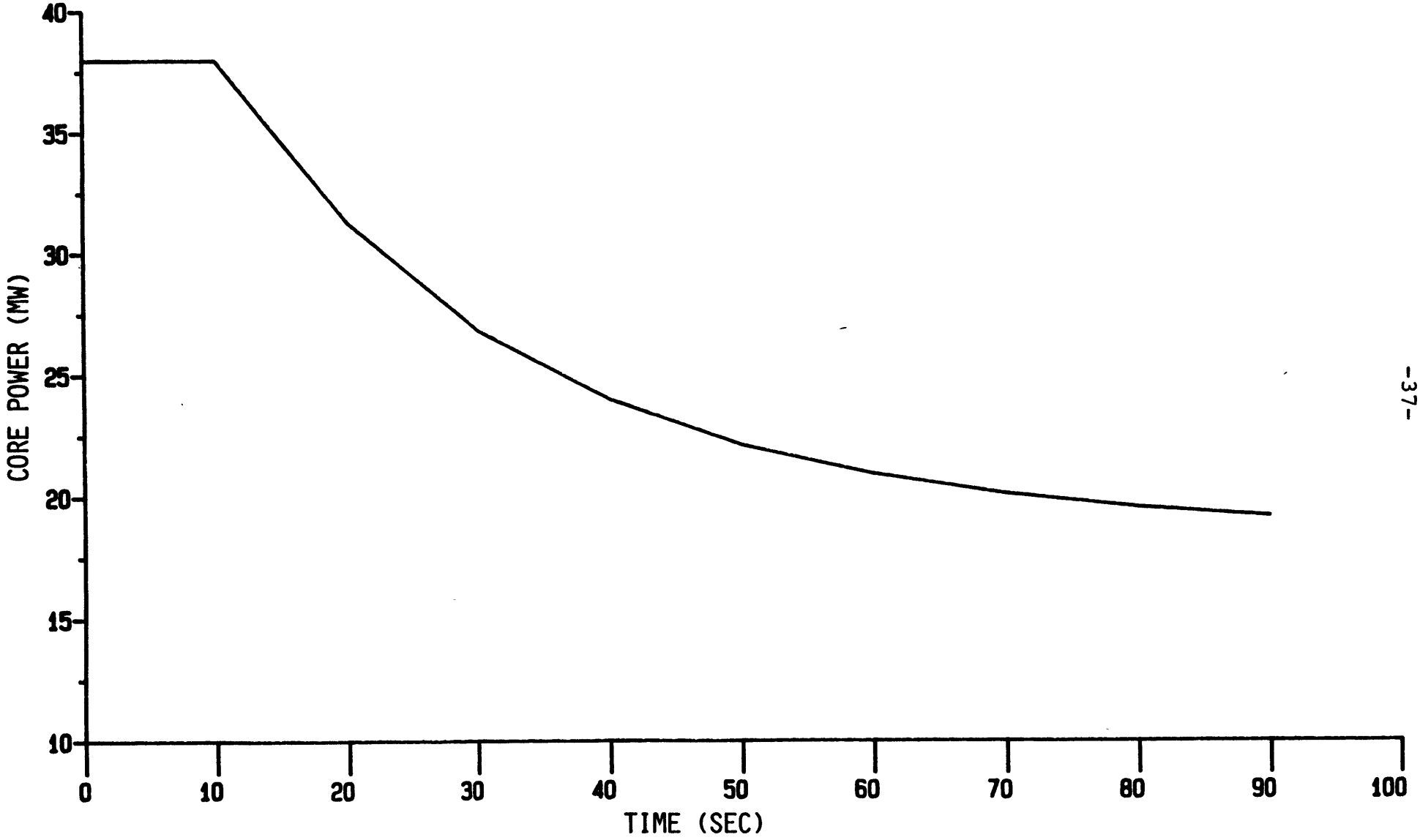


FIGURE 15 TOTAL POWER HISTORY OF CASE YA-1

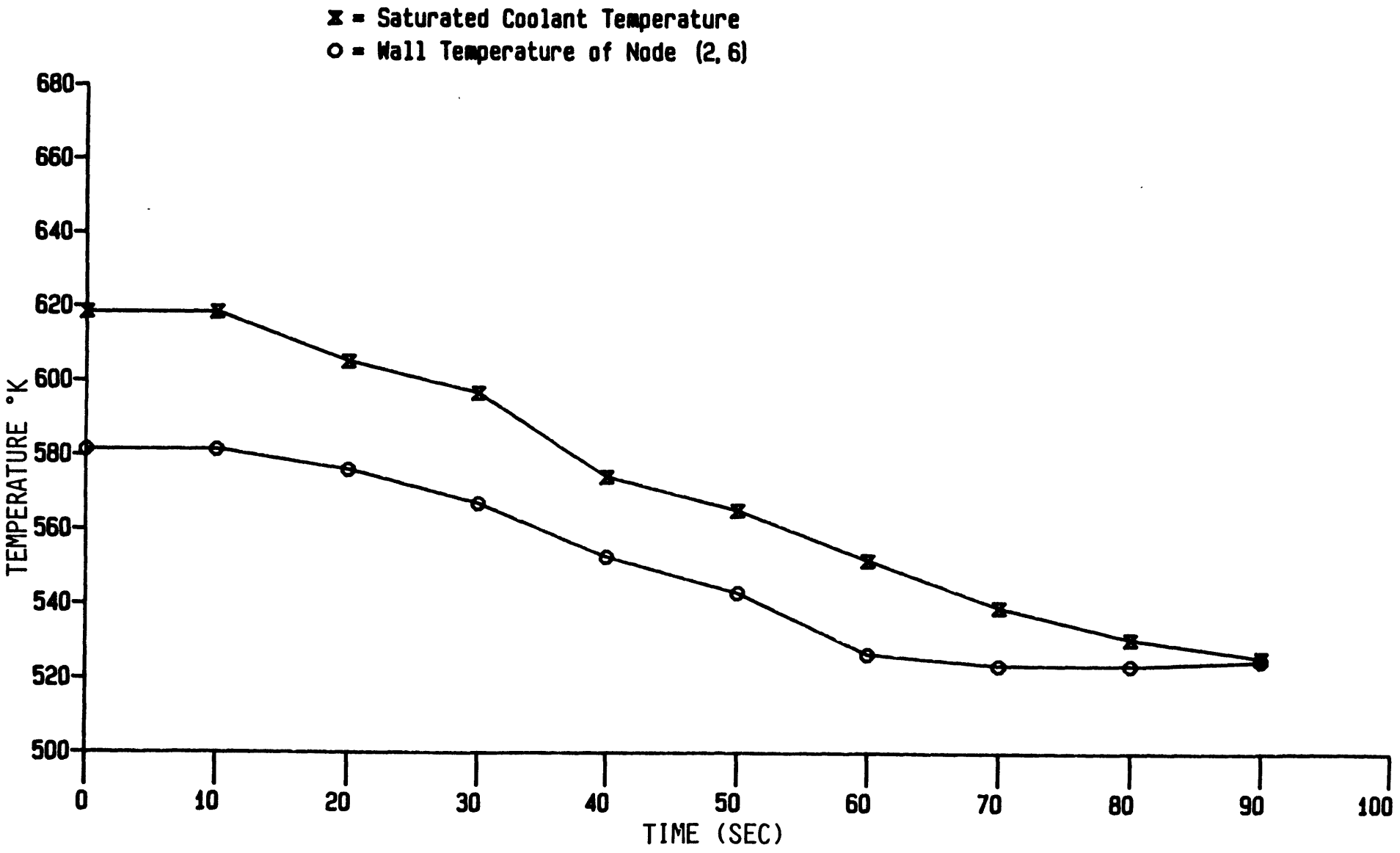


FIGURE 16 COOLANT SATURATION AND FUEL WALL TEMPERATURE HISTORIES OF CASE YA-1

Since 3-D hydraulic effects are of interest in this transient, particularly once boiling starts, a reduced pressure forcing function case was performed to accelerate the boiling phenomena. This is to be described in the next section, as case YA-2.

The total computation time used for this case YA-1 was 18,314.25 cpu seconds on Honeywell machine with Multics operating system. Altogether, there were 1610 time steps with time step size around 0.05 seconds.

III.3.3 Reduced Pressure Forcing Function Case, YA-2

As explained in the previous section, in order to induce earlier boiling, we reduced the pressure forcing function as shown in Figure 17. After 45 seconds, the pressure is kept constant because of the limitation of the data base of the W-3 CHF correlation (around 800 psia) used in our calculations. The total inlet flow rate forcing function is shown in Figure 18. All other conditions are the same as case YA-1.

Two calculations were performed. They were closed channel and open-channel cases. The open-channel case was performed throughout the 50 seconds transient period, and was restarted form the open-channel steady-state results. The closed-channel case was restarted from a dump file which had the results of the open-channel transient calcu-

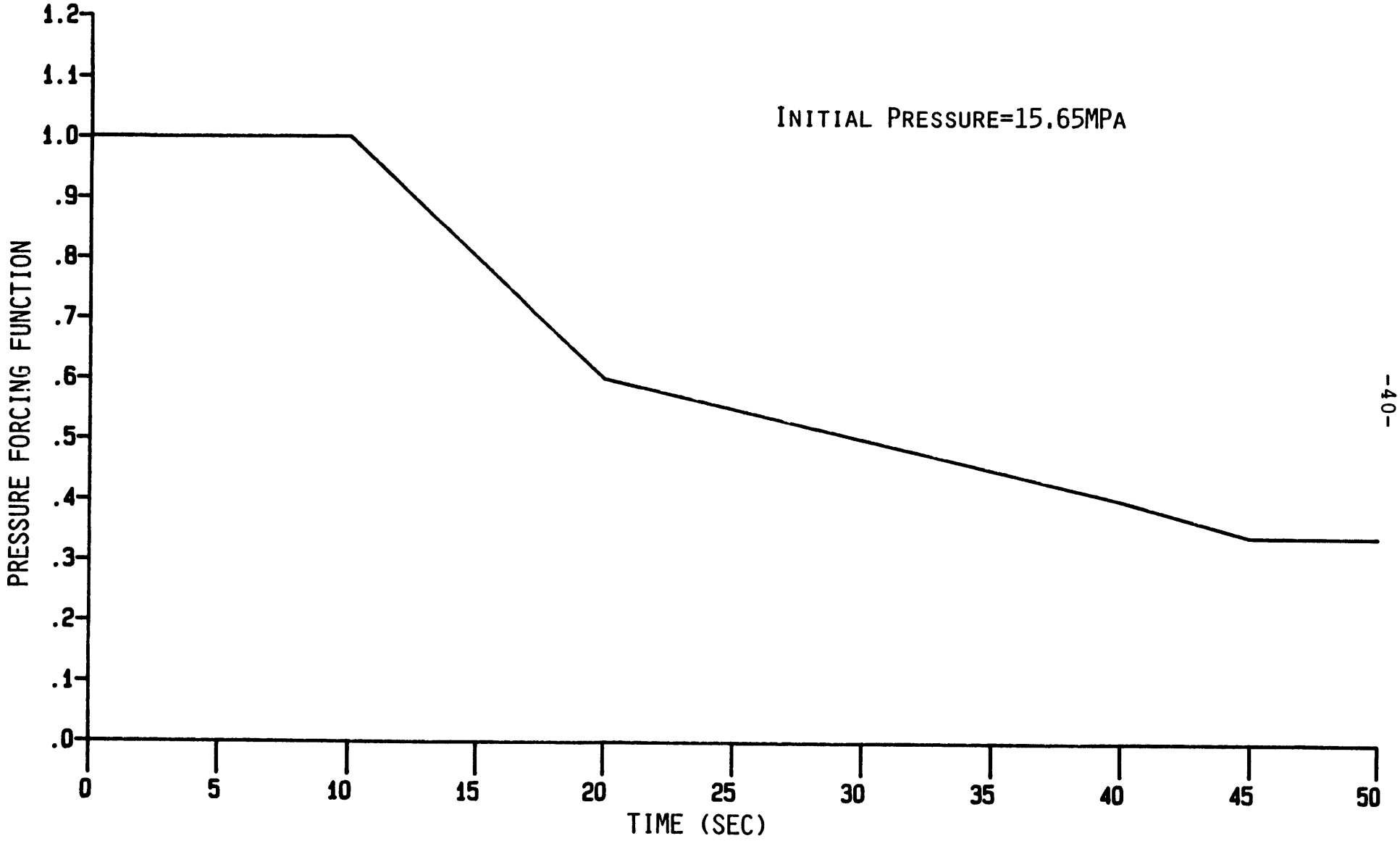


FIGURE 17 PRESSURE FORCING FUNCTION FOR CASE YA-2

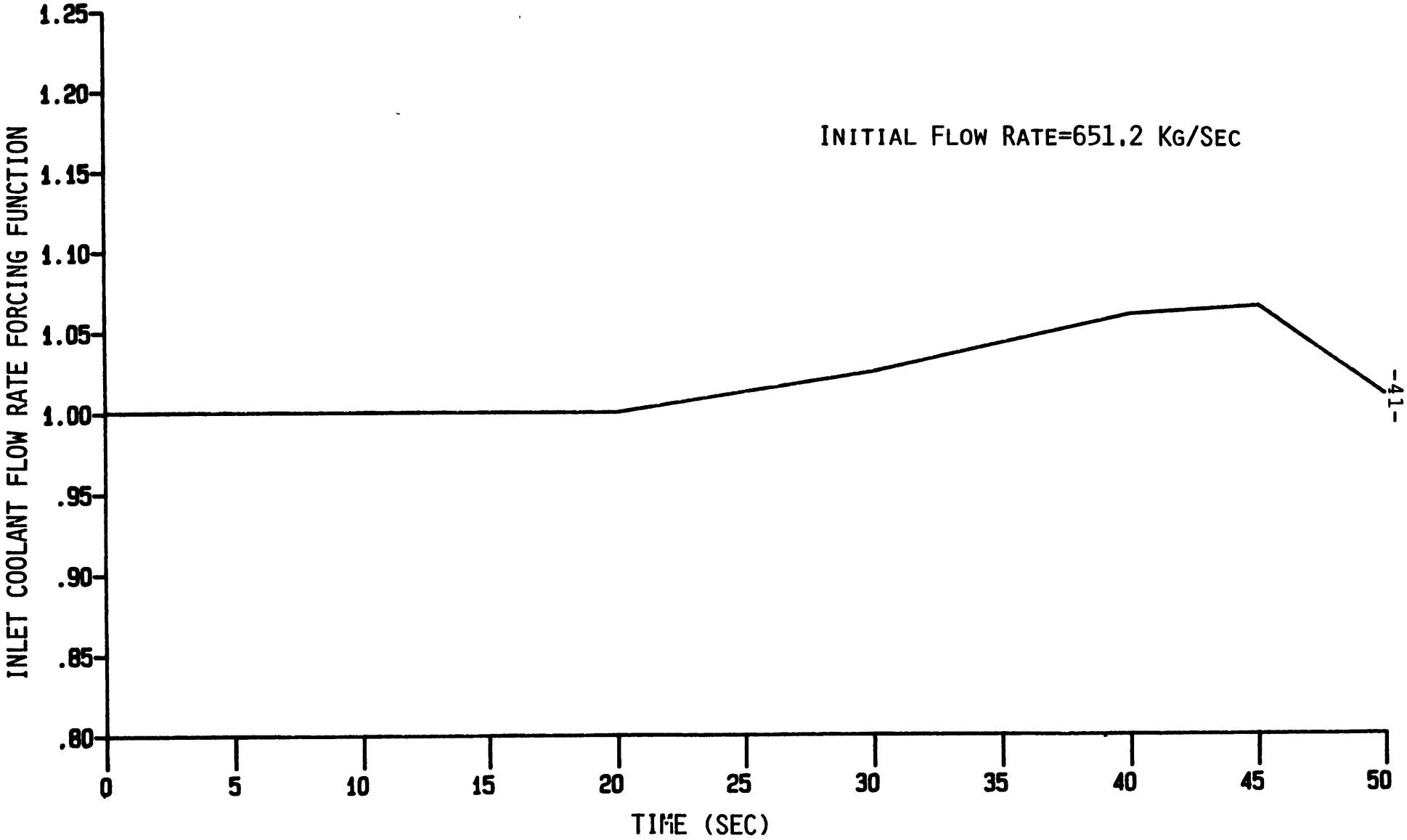


FIGURE 18 INLET COOLANT FLOW RATE FORCING FUNCTION FOR CASE YA-2

lation at the time when boiling was observed approximately. We did not restart the closed-channel transient calculation from the steady-state results because of the experience of no significant differences existing before boiling is observed.

Around 42 seconds after the beginning of the transient boiling started. All the voids were found in the hot and mixing regions. From the results of the total core powr histories after boiling started (see Figure 19), no significant difference between open-channel and closed-channel calculations is observed. However, if we check the minimum CHFR histories predicted by the code after boiling started (see Figure 20), we find the minmum CHFR of the open-channel result returns to 32.24 at 50 seconds and the closed-channel result still remains below 6 at 50 seconds.

To explain these results, let us examine what happened after 45 seconds. Recall that after 45 seconds, the pressure forcing function was kept constant. But the inlet coolant temperatures of the three zones are still dropping. Therefore, the voids start being condensed. Figure 20 shows this phenomenon. From Figure 21, we see that after 45 seconds, both open-channel and closed-channel MCHFR start rising. This is correct based on the discussion above. Now, since the open-channel model provides a better condition of

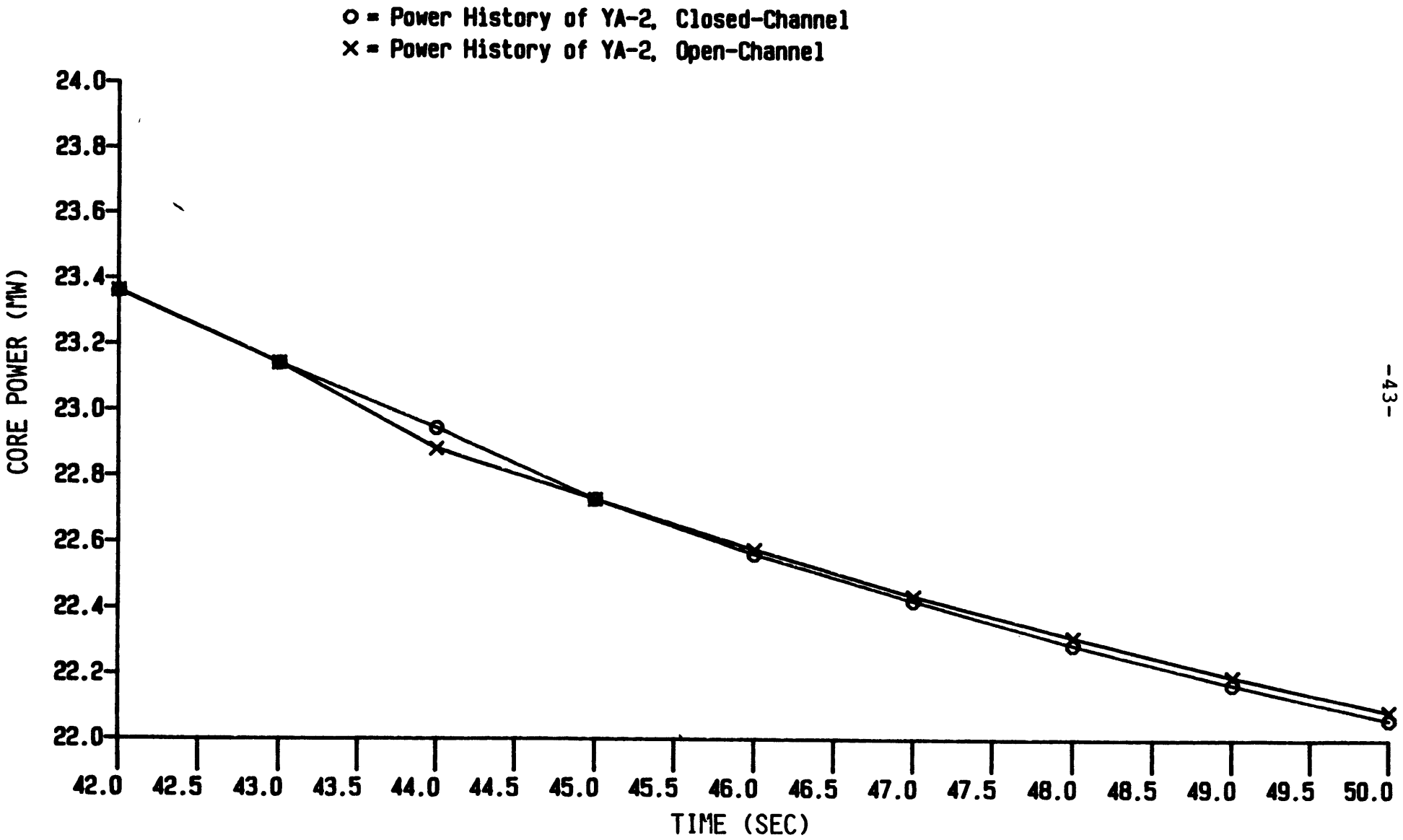


FIGURE 19 TOTAL POWER HISTORIES OF CASE YA-2

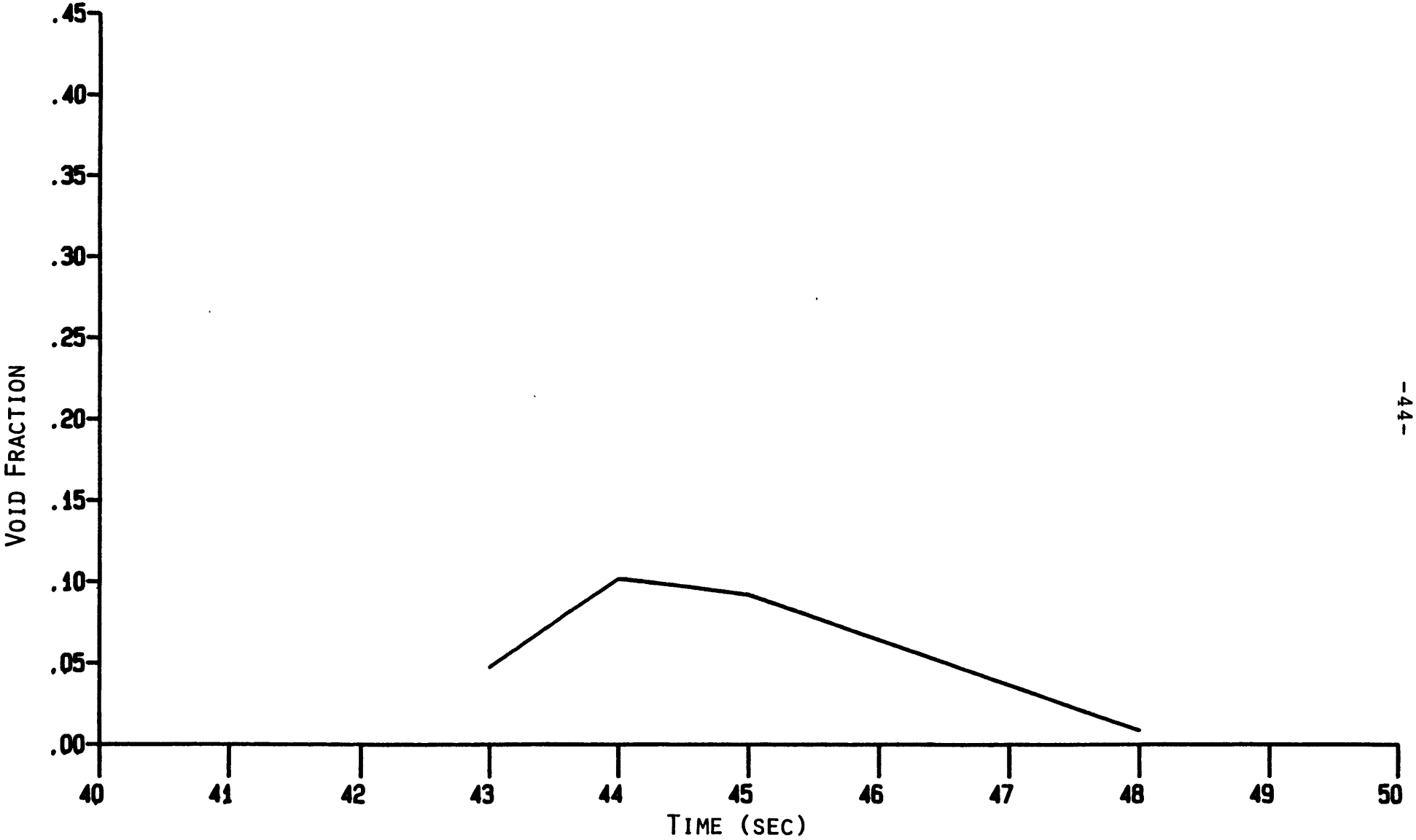


FIGURE 20 VOID FRACTION V.S. TIME FOR NODE (2,6), CASE YA-2.

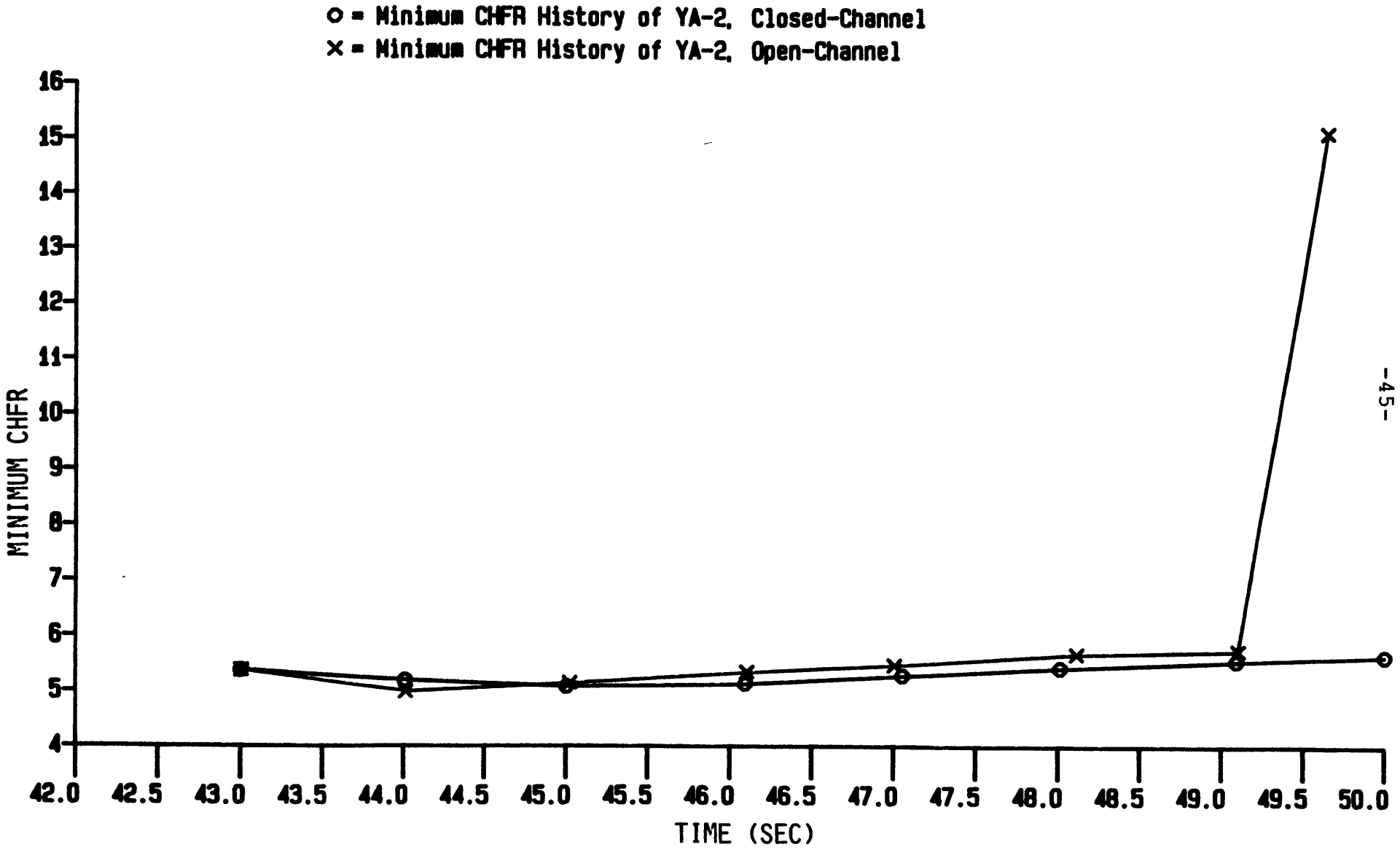


FIGURE 21 MINIMUM CHFR HISTORIES OF CASE YA-2

flow mixing among channels, it is easier for voids to be condensed. Therefore, the minimum CHFR of the open-channel case gose back to a higher value earlier than the closed channel case.

Table 4 gives the cross flow rate at 40 seconds and 44 seconds (before and after boiling started). The cross flow shown here is the cross flow between channels 1 and 2. Minus sign indicates the flow direction from channel 2 to channle 1. Obviously, before boiling started, there was no vapor cross flow. After boiling started, there was some vapor cross flow. The liquid cross flow was higher than that before boiling started (at 42 seconds). Also, it seems that coolant was driven out of channel 2. This is why the minimum CHFR occurred in channel 2.

The computation cpu time for this calculation was about 12,620 cpu seconds on Honeywell machine for open channel calculation. The closed-channle calculation spent 1936 cpu seconds after boiling was observed, while the open-channel calculation spent 2427 cpu seconds. About 20% cpu time was saved.

The results presented in this section provide two conclusions: 1) cross flow helps the minimum CHFR stay higher than the limitation point during a condensing process; 2) TITAN shows its capability of simulating compli-

Table 4

Cross Flow of Case YA-2

(Channel #, Node #)	After Boiling At 44 sec		Before Boiling At 40 sec
	$\dot{m}_\ell$	$\dot{m}_v$	$\dot{m}_\ell$
2,5	-0.2344	-0.0008	0.546
2,6	-0.7889	-0.0047	0.820
2,7	-0.7184	-0.0022	-0.080
2,8	-0.8552	-0.0072	0.030
2,9	-0.8164	-0.0105	0.180
2,10	-1.5133	-0.0315	0.480

Unit: (Kg/Sec)

cated steam line break accidents, such as the cases described in this section, which include depressurization, inlet coolant temperature transient, scram, and inlet coolant flow transient.

III.3.4 Reduced Pressure Forcing Function Case, YA-3

This is the same case as the one described in the previous section except that the system pressure forcing function and the total inlet flow rate forcing function were changed. In this case, YA-3, the system pressure was allowed to go down below the data base limitation of W-3 correlation (800psi). The pressure forcing function is shown in Figure 22. The total inlet flow rate forcing function is shown in Figure 23 which describes the transient period up to 60 seconds. Again, the closed-channel case was restarted after boiling started, and the open channel case was restarted from the open-channel steady state results.

Figure 24 shows the total core power histories of the two cases after boiling started. The difference between the two results gets to be significant after 50 seconds. Also, the closed-channel case has a somewhat higher power history. Increased mixing of the colder coolant with hotter coolant in the core helps reduce the power level for the open channel case. Form Figure 25 we see that the void

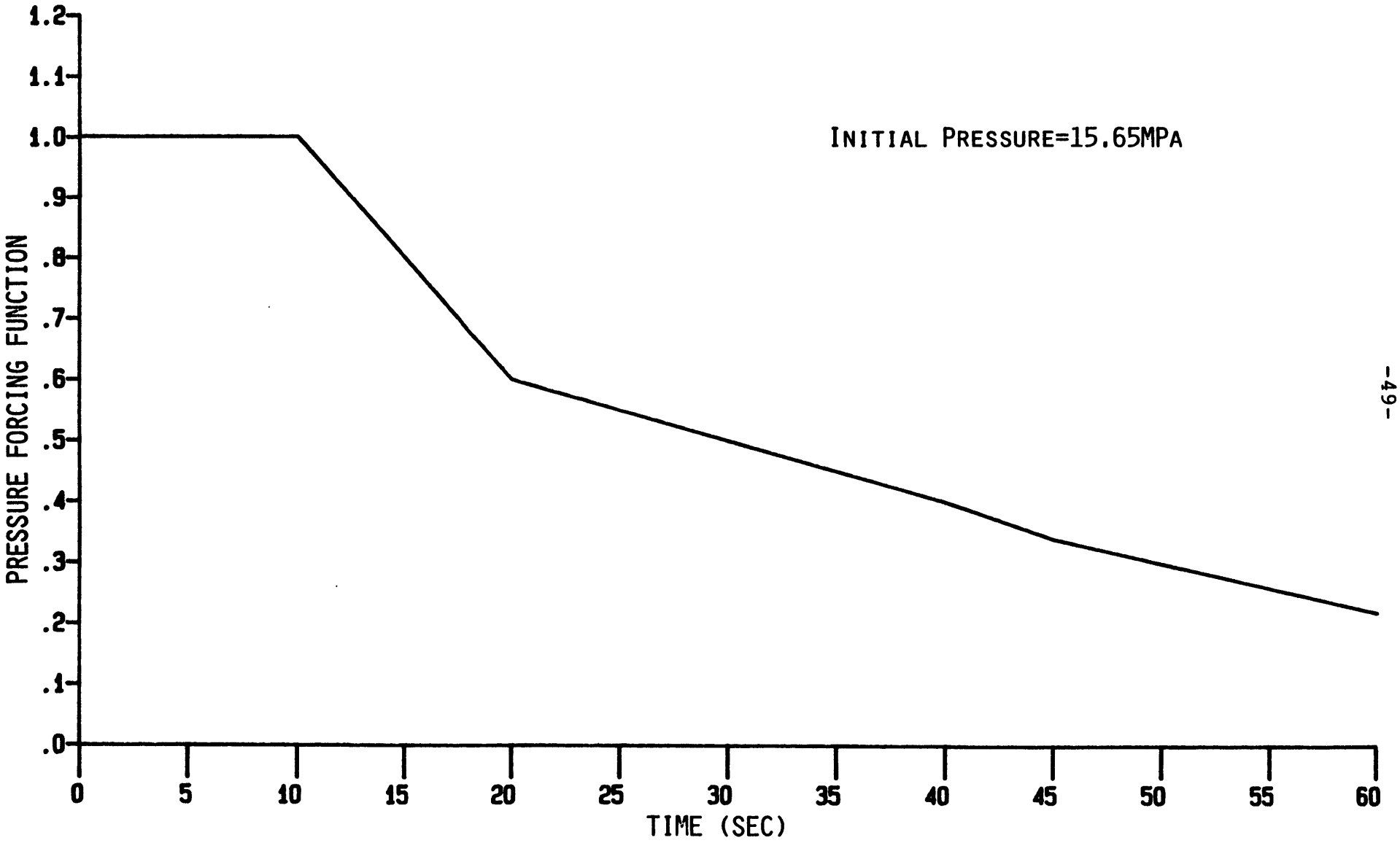


FIGURE 22 PRESSURE FORCING FUNCTION FOR CASE YA-3

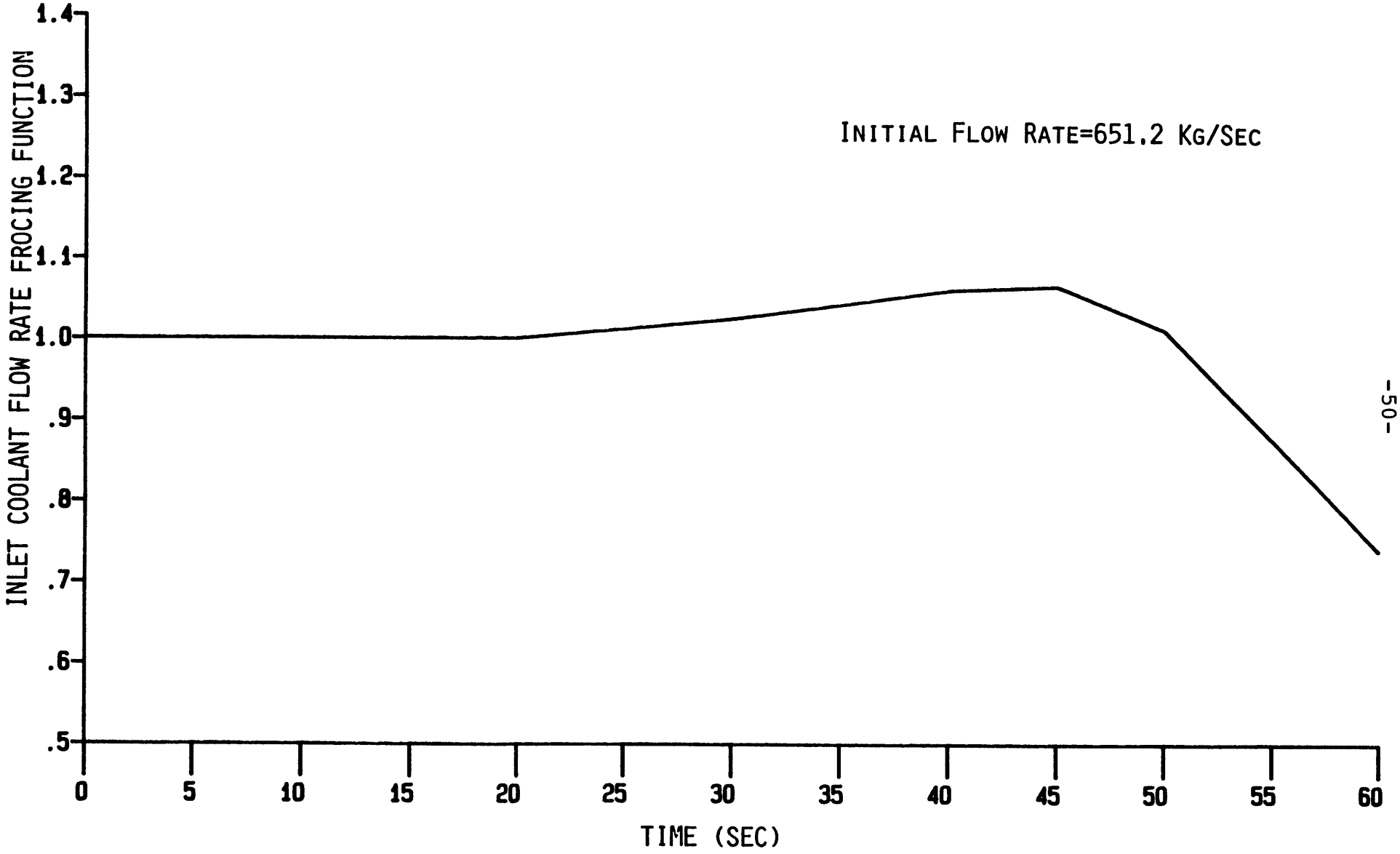


FIGURE 23 INLET COOLANT FLOW RATE FORCING FUNCTION FOR CASE YA-3

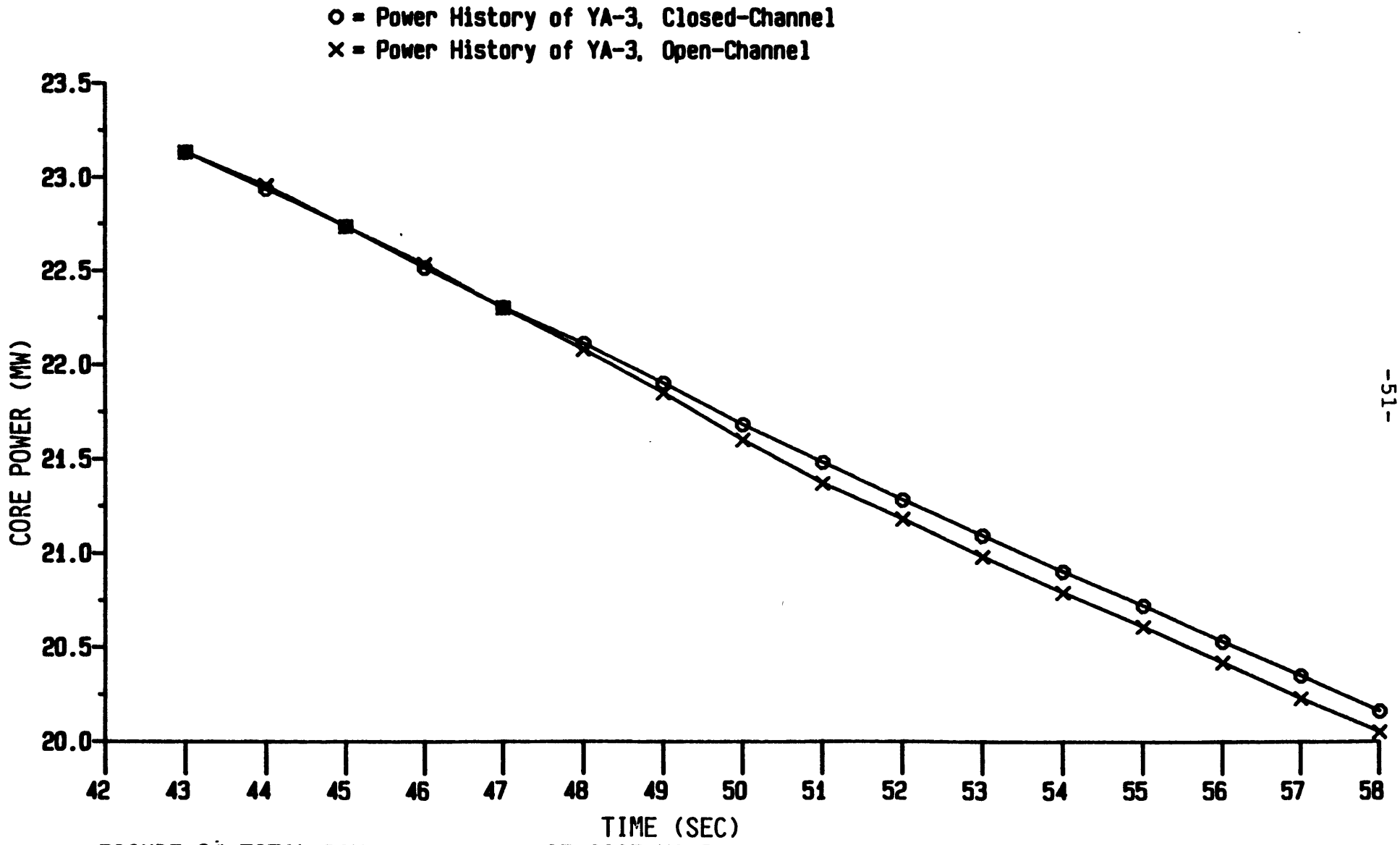


FIGURE 24 TOTAL POWER HISTORIES OF CASE YA-3

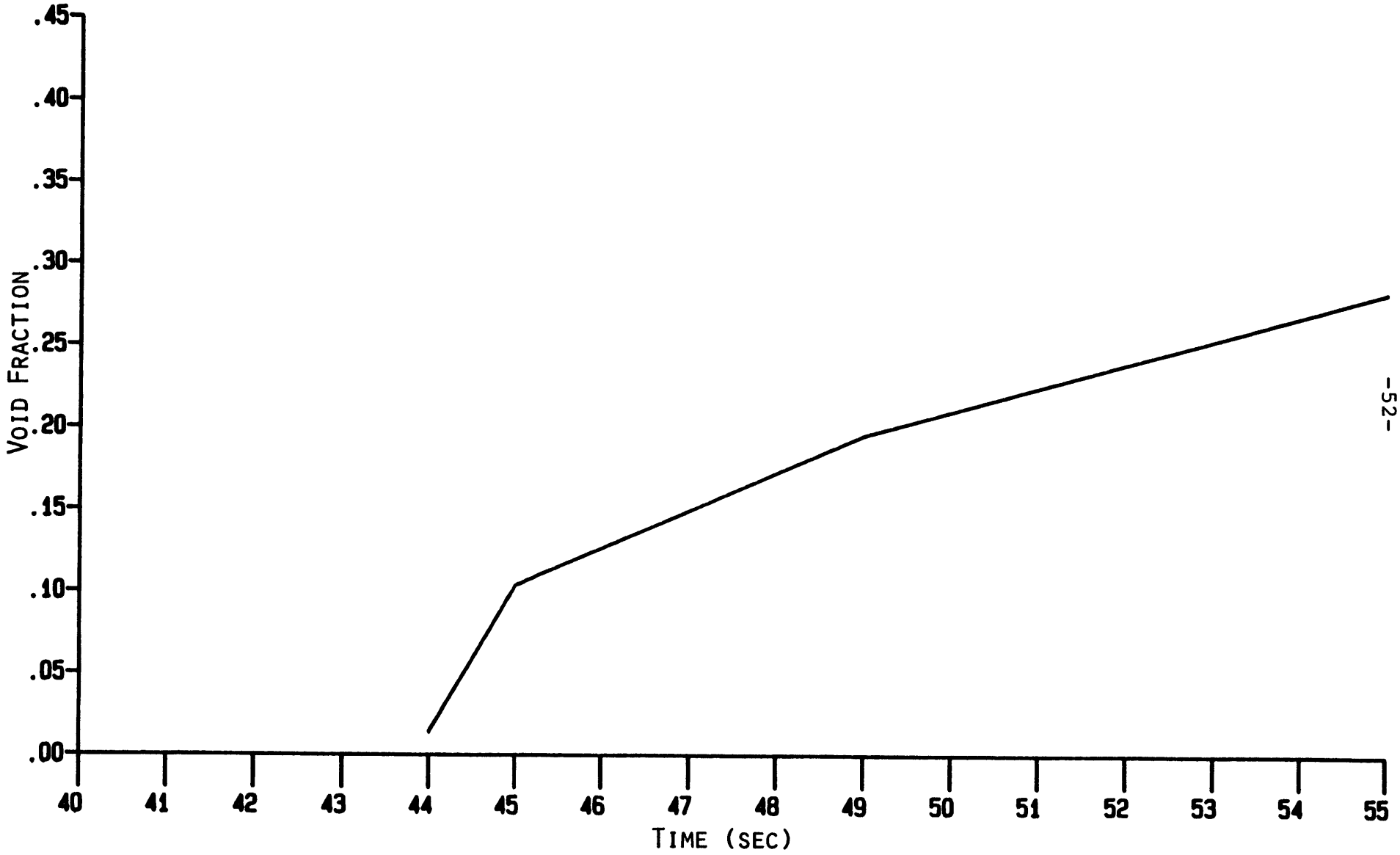


FIGURE 25 VOID FRACTION V.S. TIME FOR NODE (2,6), CASE YA-3.

fraction keeps increasing as time goes on, i.e., the coolant keeps boiling which correlates the pressure forcing function.

An interesting difference between the minimum CHFR of case YA-3 (see Figure 26) and that of case YA-2 is observed. In Figure 20, we see that after 45 seconds, the closed-channel results are always below those of the open channel. However, in Figure 26, we see that most of the time the results of open-channel are below those of the closed channel. Remember that in case YA-2, the system pressure was kept constant after 45 seconds. In case YA-3, the system pressure was decreasing during the whole calculation. Therefore, in case YA-3, more voids were being generated instead of being condensed as in case YA-2 after 45 seconds. These factors may affect the flow condition and hence the CHFR. However, one should notice that the CHFR results presented here are just for reference since the system pressure after 50 seconds is below the data range of W-3 correlation.

Table 5 gives the cross flow rate between channels 1 and 2 at 43 seconds and 57 seconds. The cross flows at 57 seconds are much higher than the cross flows at 43 seconds. This means that 3-D effects will be more important as the boiling keeps going on.

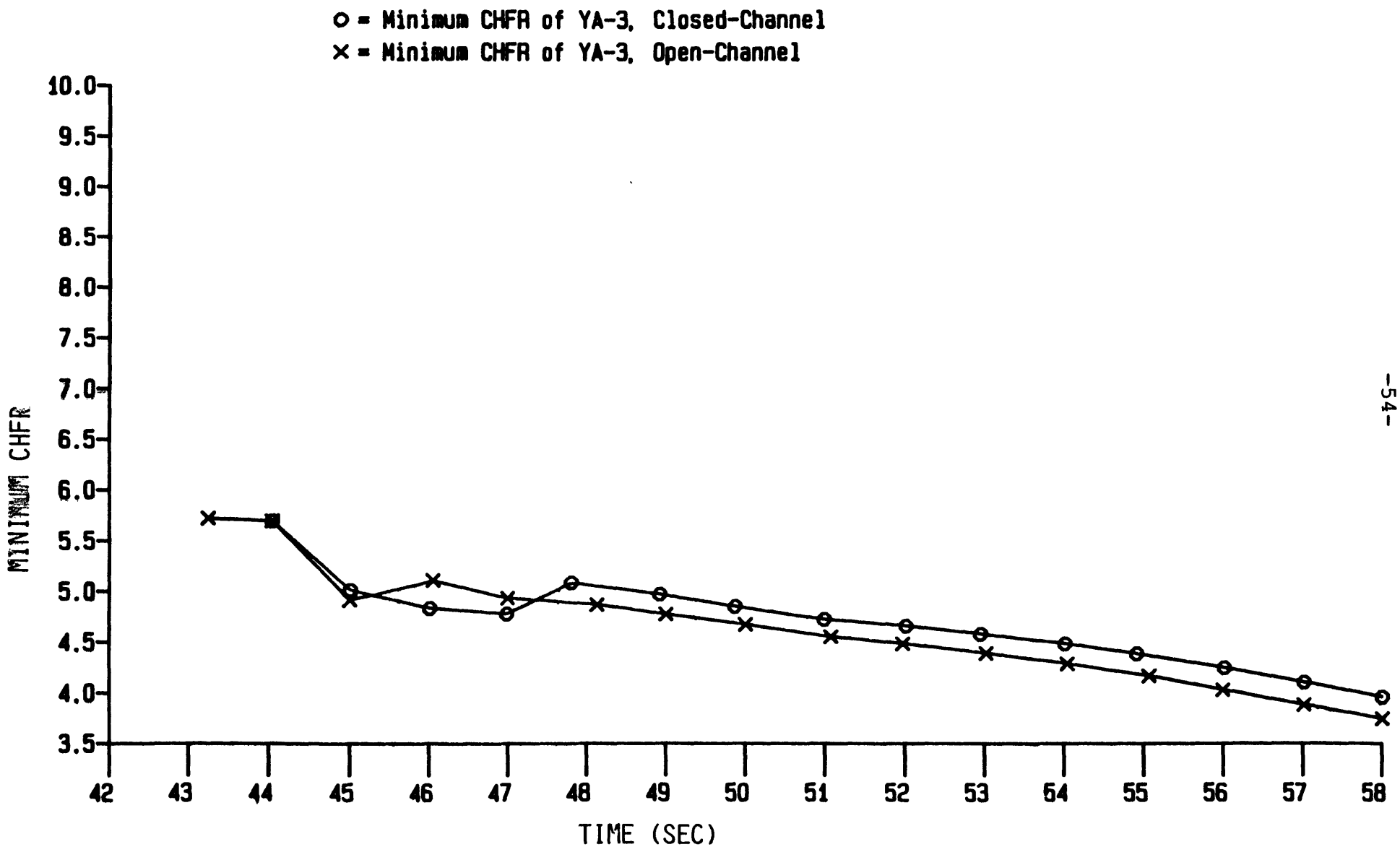


FIGURE 26 MINIMUM CHFR HISTORIES OF CASE YA-3

Table 5

Cross Flow of Case YA-3

(Channel #, Node #)	After Boiling At 57 sec		Before Boiling At 43 sec
	$\dot{m}_\ell$	$\dot{m}_v$	$\dot{m}_\ell$
2,5	-2.21	-2.08×10^{-2}	0.5176
2,6	-2.67	-5.82×10^{-2}	0.7856
2,7	-2.06	1.09×10^{-2}	-0.0935
2,8	-1.81	-4.27×10^{-2}	0.0083
2,9	-1.60	-5.35×10^{-2}	0.1610
2,10	-2.15	-1.40×10^{-1}	0.4670

Unit: (Kg/sec)

For the open-channel case, the total cpu usage was about 20,400 cpu seconds for 48 seconds out of 58 seconds transient period (the first 10 seconds were essentially for null transient, and no calculation was performed). For the closed-channel case, 10,500 cpu seconds was used after boiling started.

The conclusions of these calculations are: 1) a closed-channel calculation in the core may produce conservative results with regards to the total power; 2) a closed-channel calculation may not be conservative with regards to MCHFR calculation compared with the open-channel results which has increased cross flow out from the hot node.

VI. Conclusions

Based on the work presented in this report, the major conclusions are summarized here:

(1) TITAN has proved its capability of simulating complicated steam line break accidents.

(2) A 3-D analysis seems to be necessary for accident analysis of such transients once boiling is observed.

(3) Closed-channel analysis is conservative with regard to total core power prediction. The effect on MCHFR calculation is not always conservative.

(4) A CHF correlation with a wider data base than W-3 may be needed in some cases.

Some future work is proposed here:

(1) A faster 1-D numerical scheme is a good option for steady-state calculations.

(2) More investigations should be done to get optimum values of the unmber of thermal-hydraulic calculations needed between neutronic calculations.

(3) A true 1-D calculation should be done (no inlet temperature distribution) and compared with the fully 3-D results of the sample case.

(4) A boron concentration model should be included since in the latter stage of a steam line break

accident borated water will be pumped into the core which may affect the total nuclear absorption cross section.

(5) The time step size should be decoupled for the thermal-hydraulics part and the neutronics part calculations. This gives the code flexibility for various purposes.

References

1. D. P. Griggs, M. S. Kazimi, and A. F. Henry, "TITAN: An Advanced Three-Dimensional Neutronics/Thermal Hydraulics Code for Light Water Reactor Safety Analysis." NUREG/CP-0034, Vol. 2, Proc. of Topical Mtg. on Advances in Reactor Physics and Core Thermal Hydraulics, Kiamesha Lake, NY, pp. 766-780 (Sept. 1982).

2. D. P. Griggs, M. S. Kazimi and A. F. Henry, "Advanced Methods Development for LWR Transient Analysis, Final Report: 1981-1982," MIT-EL-82-021 (1982).

3. C. K. Tsai, D. P. Griggs, M. S. Kazimi, and A. F. Henry, "Development and Quarter Core PWR Rod Ejection Accident Application of the TITAN Code, Final Report: 1982-1983," MIT-EL-83-007 (1983).

4. J. E. Kelly, S. P. Kao, and M. S. Kazimi, "THERMIT-2: A Two-Fluid Model for Light Water Reactor Subchannel Transient Analysis," MIT-EL-81-014 (1981).

5. K. S. Smith, "An Analytic Nodal Method for Solving the Two-Group, Multidimensional, Static and Transient Neutron Diffusion Equations," Neclear Engineer Thesis, MIT (1979).

6. A. T. Schor and N. E. Todreas, "A Four-Equation Two Phase Flow Model for Sodium Boiling Simulation of LMFBR Fuel Assemblies," MIT-EL-82-038 (1982).

7. H. S. Cheng, A. L. Aronson, J. W. Herczeg, and D. J. Diamond, "The Use of MEKIN-B for Light Water Reactor Transient Calculations," BNL-NUREG-23785.

8. FSAR of Maine Yankee Nuclear Power Plant.

9. Yankee Atomic Report, Attachment A, "Cycle 6 MSLB Analysis."

Appendix

TITAN User's Guide

1 Introduction

In this part of the report, the TITAN code input information is given. The basic structure of the input format is based on the THERMIT code[4] and the QUANDRY code[5].

The description of the required input variables is presented in the following sections.

2 Detailed Input Description

2.1 Introduction

Three types of input formats are used in the code. The first is that associated with the standard FORTRAN READ statement. Both format-free and fixed format type variables are used. The format-free input is referred to as *-format consistent with IBM FORTRAN. All integer and real non-array variables are input via the format-free option. Only the title card is input in fixed character format.

The second type of input format is that associated with the standard FORTRAN Namelist option. This option is part of the restart feature and allows the user to change selectively the value of any of a variety of variables. The details of the Namelist option can be found in FORTRAN reference manuals

and only an example will be given here. If the variable 'iflash' is to be changed from 1 to 2 during a restart, then the input statement would be

$restart iflash=2$

This statement would set iflash equal to 2 while not affecting any other variable. Of course, if other variables are to be changed, they also can be included in the Namelist statement. As indicated above, this type of input format is only used for the restart option.

The third type of input format is that associated with the input processor found in subroutine 'nips'. This subroutine is used to read the array data. The input processor permits relatively easy input of the values for the arrays. The key to this processor is that blocks of data may be repeatedly read. To achieve this result, a special type of format is used. Input fields are separated by blanks (no commas are allowed) with repeated fields inside parentheses proceeded by an integer multiplier. The end of a card group is marked with a dollar sign ($). An example serves to illustrate the use of this format. Suppose the array P(6,4) (6 levels, 4 channels) must be read in. There are 24 total values which are required. If these values are all the same (e.g., 6.9MPa), then the input would be

24 (6.9e6) $P

(Everything after the $-sign is ignored so that comments can be placed here). If the four channels all have the same distribution, but not axially uniform, then the input would

be

4(6.9e6 6.85e6 6.8e6 6.75e6 6.7e6 6.65e6)$P

It should be noted that the values for the variables can be given in any format, but will be interpreted according to the variable type. Up to 10 levels of parentheses nesting are permitted. Also no blank may appear between a left parentheses and the integer proceeding it. With this type of format the array data can be specified with a minimum amount of input.

2.2 General Problem Information, Real and Integer Constants

The first group of input contains the general information of the case we are dealing with. There are thirteen cards in this group. The variables in each card are described in the following sections. Note here, all inputs for TITAN are in the free format except for the title information card (card two), which is in the A format.

2.2.1 Card One

(i) Variable: ntc

(ii) Meaning: Two meanings

- The number of title cards.
- An input flag indicating whether the job is a restart or a new problem.

(iii) Description:

- ntc>0, a new problem is started and ntc is the actual number of title cards to be read in card 2.

- ntc=0, the execution is terminated.
- ntc=-2, the job is restarted from a previously created dump file.

This is a steady-state restart. (See 3.2.1)

- ntc=-3, the job is restarted from a previously created dump file. This is a transient restart. (See 3.2.2)

2.2.2 Card Two

(i) Variable: Title information

(ii) Description:

The number of cards which are read in is equal to ntc. On each card 80 characters of information may be given.

2.2.3 Card Three

(i) Variables: nc,nr,nz,ncf,ncc,nopt,noppt

(ii) Description:

nc=Number of channels.

nr=Number of rows.

nz=Number of axial nodes.

ncf=Number of nodes in the fuel.

ncc=Number of nodes in the clad.

nopt=Number of channels to be printed out.

noppt=Indicator of print out(0/1) (print all channels/optional print out)

(iii) Example: Refer to Figures A.1 and A.2.

nc=6

nr=3

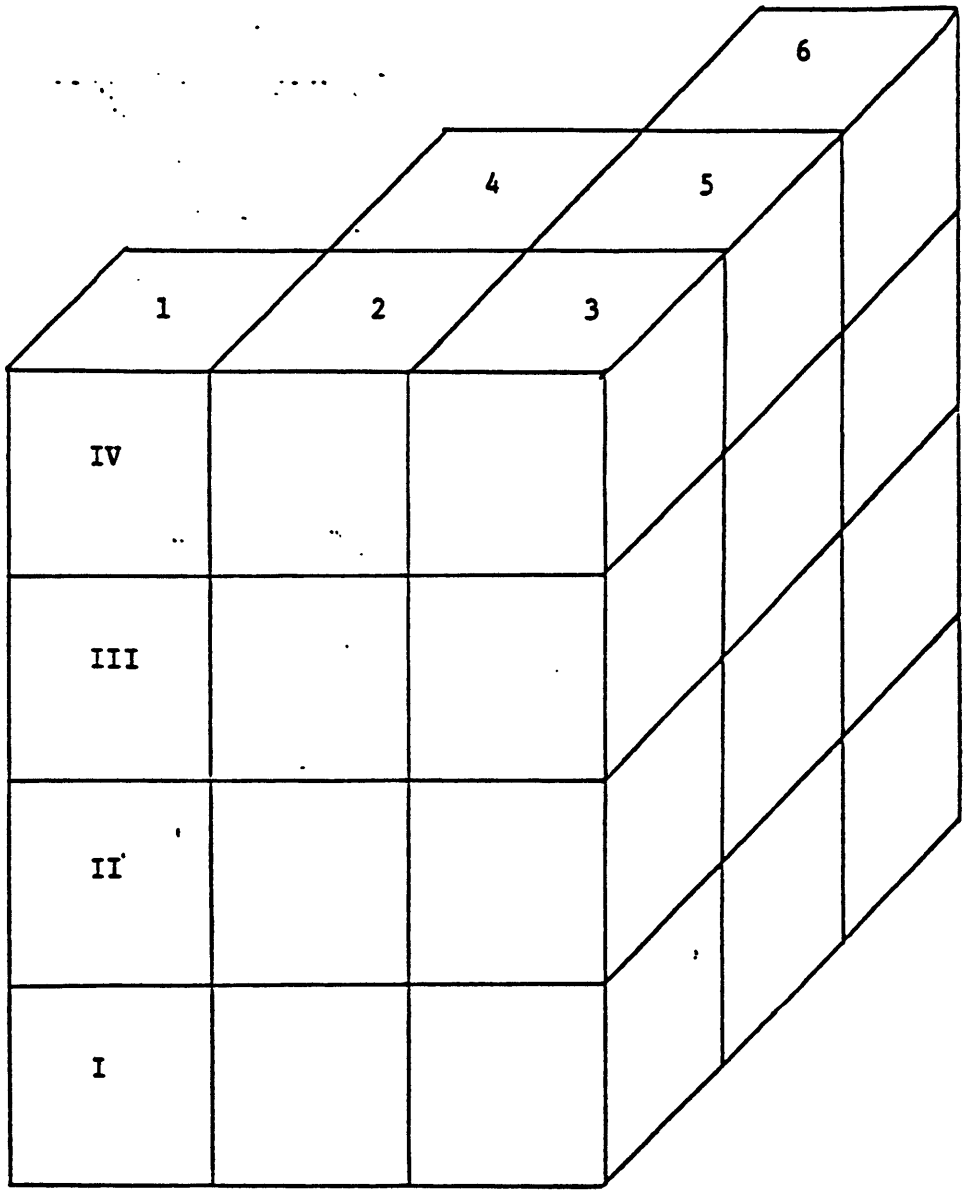


Figure A.1 Six-channel example

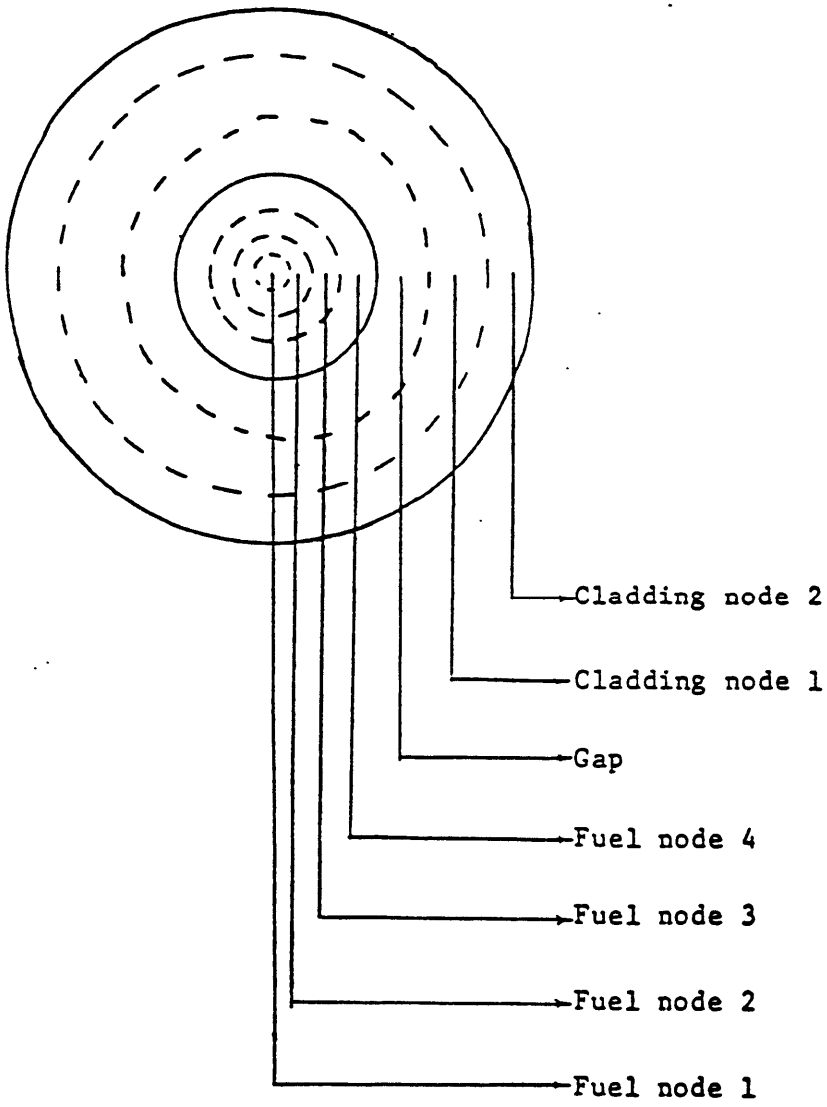


Figure A.2 Fuel and cladding nodes example

nz=4

ncf=4

ncc=2

noppt=0, print all six channels.

nopt=6

If noppt=1 and nopt=2, two of the six channels will be printed out. The two channels will be determined by the array input "ncopt(nopt)" (Card 14).

2.2.4 Card Four

(i) Variable: itb,ibb,iflash,ifintr,iht,iss,
iqss,ichf,iwft,ivec,itam,imixm,imixe,
iafm,itfm,igfm,grav,hdt,velx

(ii) Description:

These variables are related to the thermohydraulics model. The interger variables act as indicators for the options which the user may select.

itb=Top boundary condition indicator (0/1) (pressure/velocity)

ibb=Bottom boundary condition indicator (0/1/2) (pressure/velocity/total inlet flow rate)

iflash=Interfacial mass exchange model (0/1/2) (Nigmatulin Model/Suppressed, i.e., Γ=0/Nonequilibrium Boiling Model)

ifinter=Interfacial momentum exchange model (0/1) (MIT/LASL)

iht=Heat transfer indicator (0/1/2/3) (No heat transfer/Constant gap conductance, temperature independent fuel, cladding conductance/ Constant gap conductance, temperature dependent fuel, cladding conductance/ Temperature dependent gap, fuel, cladding conductance)

iss=Heat transfer calculation type (0/1/2) (Transient/Steady-state/Steady-state with critical heat flux check suppressed)

iqss=Steady-state heat flux indicator (0/1) (Heat flux is held constant and no fuel temperatures are calculated/Heat flux is not held constant and the fuel temperature is calculated)

ichf=Critical heat flux indicator (1/2/3/4/5/6) (Biasi and CHF-Void correlations/W-3/ CISE/Barnett/Bowring/Hench-Levy)

iwft=Transverse friction model indicator (0/1) (No friction/Gunter-Shaw correlation)

ivec=Tranverse velocity indicator (0/1) (Actual transverse velocity is used/ The magnitude of the velocity vector is used)

itam=Fluid dynamics indicator (0/1) (No transverse flow allowed, i.e., closed-channel calculation/Normal, i.e., open-channel calculation)

imixm=Momentum turbulent mixing indicator (0) (No mixing is allowed in this version of TITAN)

imixe=Energy turbulent mixing indicator (0) (No mixing is allowed in this version of TITAN)

iafm=Axial friction model indicator (0/1) (Default/User supplied, see card six)

itfm=Transverse friction model indicator (0/1) (Default/User supplied, see card six)

igfm=Grid friction model indicator (0/1) (Default/User supplied, see card six)

grav=Gravitational constant (Usually: -9.81 M/s^2)

hdt=Hydraulic diameter in transverse direction (m) =4*Free volume/Rod surface area

velx=Velocity multiplier for transverse friction, normally should be set equal to the ratio of the maximum to average transverse flow area

2.2.5 Card Five

This card is required only if ibb=2.

(i) Variable: winlet

(ii) Description:

winlet=The total inlet flow rate [Kg/s].

2.2.6 Card Six

(i) Variables:

If iafm=1, need a0,rex,a,b

If itfm=1, need a0,ret,a,b

If igfm=1, need a,b

(ii) Description:

The friction coefficient is defined as

(a) Laminar flow

$$f = ao/re;\ re < rex/ret \tag{A.1}$$

(b) Turbulent flow:

$$f = a \cdot re^{**}b;\ \ re > rex/ret \tag{A.2}$$

Therefore, we need a0,rex,ret,a,b as input variables. The default values are in Tabel A.1.

2.2.7 Card Seven

(i) Variables: idump,nitmax,iitmax,epsn,epsi

(ii) Description:

This card contains the thermohydraulic iteration control and dump indicator variables.

idump=Dump file request indicator (0/1) (No/Yes)

nitmax=Maximum number of Newton iterations

iitmax=Maximum number of inner iterations

epsn=Newton iteration convergence criterion

epsi=Inner iteration convergence criterion, i.e., the pressure iteration convergence criterion

A relative error check on the pressure is used in this code, so that the iteration proceeds until the condition

$$\max \left| \frac{p^{m} - p^{m-1}}{p^{m}} \right| < \text{eps} \tag{A.3}$$

is met, where m designates either the Newton or inner iteration and where eps is either epsn or epsi. The maximum is taken over all mesh cells, but only the pressure is checked for convergence during the Newton iteration. In no case, how-

Table A.1 Friction Model Default Constants

	a0	rex/ret	a	b
Axial Friction Model	64	1502.11	0.184	-0.2
Transverse Friction Model	180	202.5	1.92	-0.145
Grid Friction Model	--	--	3.0	-0.1

ever, will the total iteration count be allowed to exceed the limits specified by nitmax and iitmax. If these limits are reached and nitmax> 0, iteration ceases, and code operation continues as if the iteration had converged. If nitmax<0, then the calculation stops when the limits are reached.

It is important to remember that all variables in the calculation of thermohydraulic part of this code are derived from the pressure solution, therefore, if the pressure solution is not converged tightly enough, errors in pressure solution may be amplified as other variables are computed from it. The user is therefore cautioned to be certain that the convergence criteria are sufficiently small by repeating the calculation with smaller values for those quantities, if possible. One must also remember, however, that on a finite precision machine there is a lower limit to these quantities, below which roundoff errors will prevent convergence.

2.2.8 Card Eight

(i) Variables: q0,qcf,to,omg,hrdr,thc,thg,
hgap,ftd,fpuo2,fpress,cpr,expr,
grgh,pgas,(gmix(k),k=1,4),burn

(ii) Description:

This card is required only if iht is not equal to 0. (See card four).

q0=Initial total power (W)

If q0<0, then q0 is set to be the current power.

qcf=Fraction of power which is generated from the

coolant.

t0=Delay time (s)=0.0 for this version of TITAN.

omg=Inverse reactor period (1/s)=0.0 for this version of TITAN.

radr=Outer fuel rod radius (m)

thc=Clad thickness (m)

thg=Gap width (m)

hgap=Gap heat transfer coefficient (W/m*m*DEG.K)
Suggested value = 5.678e3

ftd=Fraction of theoretical density of fuel

fpuo2=Fuel pressure on clad for gap conductance model (Pa=N/m**2)

cpr=Coefficient for the above pressure

expr=Exponent for the above pressure

grgh=Gap roughness(m). If zeor is given as input, a default value of 4.4e-6 m is assumed.

pgas=Gap gas pressure (Pa)

gmix(1)=Helium fraction in gap gas

gmix(2)=Argon fraction in gap gas

gmix(3)=Krypton fraction in gap gas

gmix(4)=Xenon fraction in gap gas

burn=Fuel average burnup (MWD/MTU). This variable is used in the cracked-pellet model, which accounts for partial contact of fuel against clad.

The variables fpress,cpr,expr are used if the gap conductance is to be supplemented by a term CP_f^m to represent the effect

of a closed gap (grgh> thg) with fuel pressing against clad with pressure P_f . The term may be suppressed by giving cpr=0, fpress=0 and expr=1.

2.2.9 Card Nine

(i) Variables:

idiag,irstrt,jprinta,jprintb,jprintc,jprintd,
jprinte,jprint,ibpont,
icore,itran,ndpg,ithfbk,ixenon,ecf

(ii) Description

idiag=Diagonal symmetry, lower right to upper left in the reactor plane (0/1) (No/Yes)

irstrt=Leakage Approximation/Point Kinetics extrapolation

irstrt	Leakage Approximation	Point Kinetics Extrapolation
0	Quadratic	Yes
1	Flat	Yes
2	Quadratic	No
3	Flat	No

Suggested value=0 for most problems.

jprinta=Print flag for total power(0/1) (No/Yes)

jprintb=Print flag for normalized nodal power(0/1) (No/Yes)

jprintc=Print flag for nertron fluxes(0/1) (No/Yes)

jprintd=Print flag for nertron leakages(0/1) (No/Yes)

jprinte=Print flag for normalized assembly power(0/1) (No/Yes)

jprint=Print flag (<3/3/4/5/6) (No print for expansion coefficient, steady state matrices, albedo oriented map and reactor oriented map/expansion coefficient/steady state matrices/albedo oriented map/albedo oriented and reactor oriented maps)

ibpont=BPOINTER print flag=0 (Not used in this version of TITAN)

icore=container array size in CDC words=0 (Not used in this version of TITAN)

itran=Transient problem (0/1/2/3/4) (Static/Initiate transient with control rod, Cusping applied/Initiate Transient with flow rate/Initiate transient with inlet temperature/Same as 1 but without Cusping effect)

ndpg=Number of delayed neutron groups ($\leq$6)

ithfbk=Type of thermohydraulic feedback (0/1/2/3) (None/Cross sections are linear functions of fuel, moderator temperatures and moderator density/Quadratic feedback model is used/In addition to option 2, feedback coefficients of perturbed portion are considered)

ixenon=Equilibrium xenon model indicator (0/1) (No/Yes)

ecf=Energy conversion factor in W-sec per fission, default=3.204e-11

2.2.10 Card Ten

(i) Variables: minout,maxout,minflx,maxflx,ninner,noutpu

This card describes the neutronic iteration specifications.

(ii) Description:

minout=Minimum number of outer iterations

maxout=Maximum number of outer iterations, if maxout=0, default number, 100, is used.

minflx=Minimum number of flux iterations

maxflx=Maximum number of flux iterations, if maxflx=0, default number, 3, is used.

ninner=Number of inner iterations, if ninner=0, default number, 1, is used. Default number is recommended.

noutpu=Number of outer iterations per matrix update, if noutpu=0, default number, 5, is used.

2.2.11 Card Eleven

(i) Variables: guessk,guark,shiftk

(ii) Description: This card describes the input eigenvalue and shift factors.

guessk=Initial eigenvalue guess, if guessk=0, default number, 1.0, is used.

guark=Shift factor

shiftk=Initial eigenvalue shift, if shiftk=0, default number, 1.5, is used. Default number is

recommended.

2.2.12 Card Twelve

(i) Variables: epsk,epsphi,errorr

(ii) Description: This card describes the neutronic convergence criteria.

epsk=Eigenvalue convergence criteria, default value is 1.e-6.

epsphi=Node power convergence criteria, default value is 1.e-4.

errorr=Error reduction on flux iterations, default value is 0.3.

2.2.13 Card Thirteen

(i) Variables: nx,nunqpl,nalb,ncomp,nedtx,nedty,nedtz

(ii) Description: This card describes the problem size.

nx=Number nodes in x-direction, for example, in Figure A.1, nx is 3.

nunqpl=Number of unique planes, i.e., the number of different composition mapped planes. This variable counts the bottom and top fictitious cells too.

nalb=Number of albedo sets.

ncomp=Number of compositions, i.e., unique cross-section sets.

nedtx=Number of edit bounds, x-direction.

nedty=Number of edit bounds, y-direction.

nedtz=Number of edit bounds, z-direction.

(iii) Example: See Figure A.3.

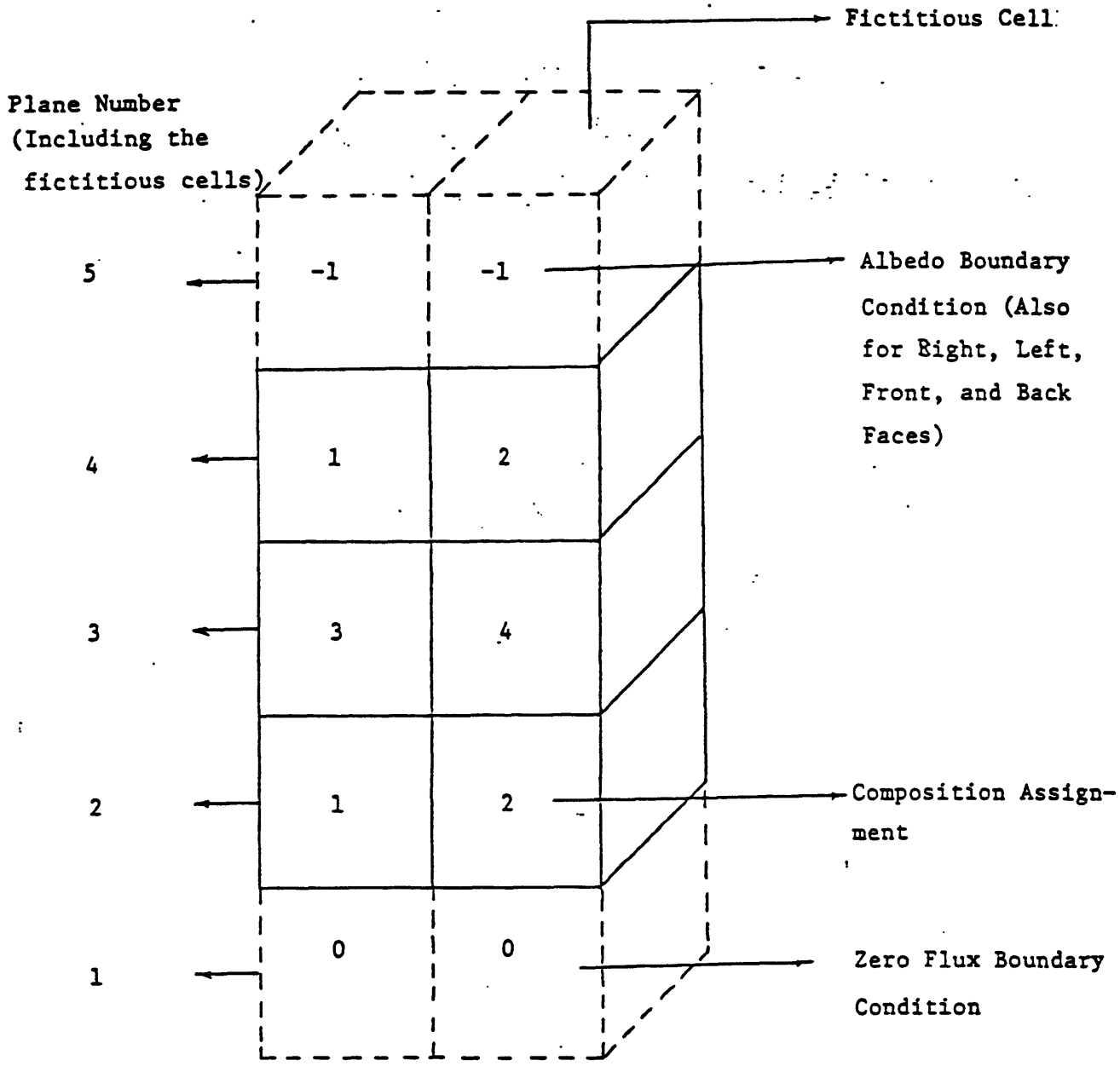


Figure A.3 Card twelve example

nx=2

nunqpl=4

nalb=2, see Figure A.6 too.

ncomp=4

nedtx=2

nedty=1

nedtz=3

Note here, nedtx, nedty,nedtz are not necessary to be 2, 1, 3 respectively. This will be explained more clearly in Section 2.3.6.9.

2.3 Array Data

The second group of input is that related to the array data. These data are read in via the 'NIPS' input subroutine. The array data is divided into six general sections: geometrical data, friction model, initial and boundary conditions, heat transfer model, transient forcing function, and the neutronic data. Each of these is discussed below.

2.3.1 Geometrical Data

The mesh is basically a regular, orthogonal, x-y-z grid, but boundaries in the x-y plane may be irregular and mesh spacings in all three dimensions can vary with location. The node numbering schemes of THERMIT and QUANDRY are different. This was discussed in the report.

The numbering scheme assigns the index 1 to the left-most cell in the bottom row of cells, incrementing the

index from left to right and bottom to top as indicated in Figure A.4. The convention for the positive direction for the x and y axes is also indicated in this figure. This convention must be remembered in interpreting the signs of the velocities printed out by the code.

The geometrical data required is described in the following sections.

2.3.1.1 Card Fourteen

(i) Variables: ncopt(nopt)

(ii) Description:

This card is required only if nopt>0 and noppt=1.

ncopt=The channel numbers whose information will be printed out.

(iii) Example: See Figure A.4.

Case I: Print out all 12 channels--

nopt=12

noppt=0

No ncopt(nopt) is required.

Case II: No thermal-hydraulic print out--

nopt=0

noppt=1

No ncopt(nopt) is required.

Case III: Channels 2 and 3 are printed out--

nopt=2

noppt=1

ncopt(1)=2

ncopt(2)=3

2.3.1.2 Card Fifteen

(i) Variables: ncr(nr)

(ii) Description:

ncr=The number of cells in each row. nr is the dimension of ncr. Recall that nr is the number of row. No gaps are allowed in a row of cells.

(iii) Example: See Figure A.4.

ncr(1)=1

ncr(2)=2

ncr(3)=3

ncr(4)=5

ncr(5)=1

2.3.1.3 Card Sixteen

(i) Variables: indent(nr)

(ii) Description:

indent=Identation for each row. The identation is specified from a left boundary which is a fixed position defined by the leftmost node (channel). This can be explained in the following example.

(iii) Example: See Figure A.4.

ident(1)=2

ident(2)=1

ident(3)=1

ident(4)=0

ident(5)=1

2.3.1.4 Card Seventeen

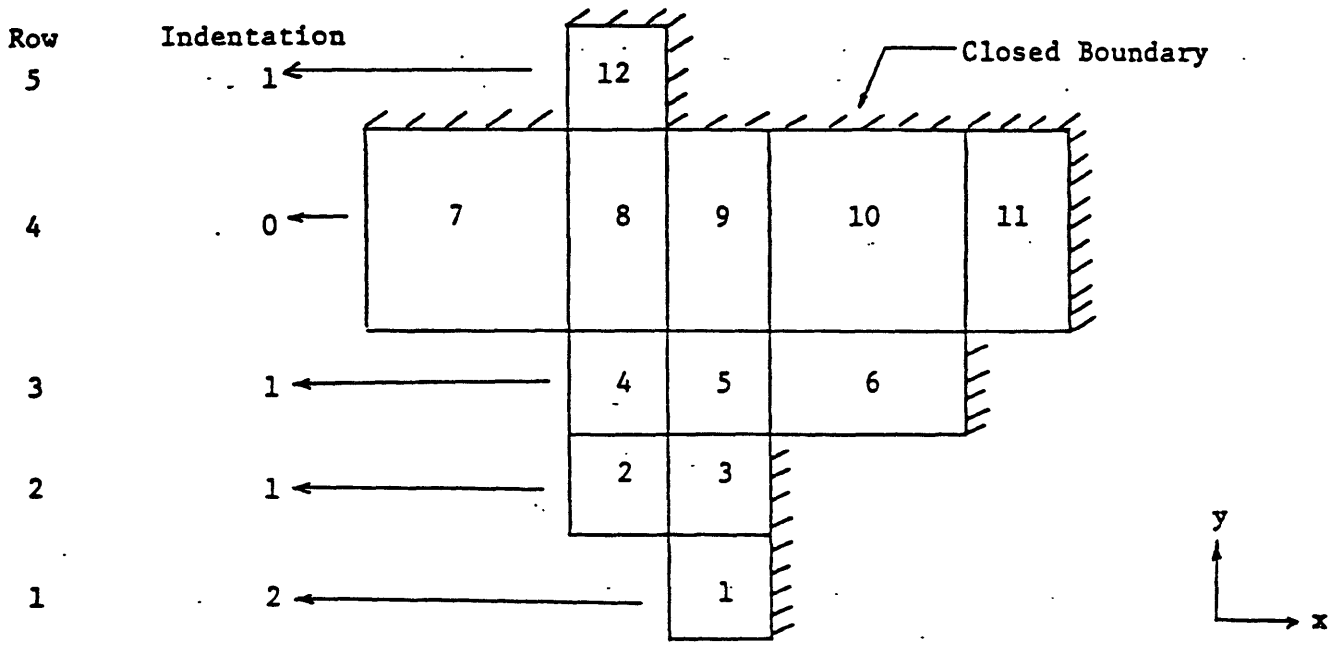


Figure A.4 A possible arrangement of channels in the horizontal plane, showing the index conventions

(i) Variables: arx(nz,nc)

(ii) Description:

arx=Mesh cell areas in the x-direction (m**2). arx for each mesh cell is the area on the left side of the cell. If we set arx at one mesh cell to be equal to zero, that means no flow is allowed to cross that boundary. Also note here, the right most cells have been automatically set to be zero, see Figure A.4. The dimension (nz,nc) means that we have to specify the areas from bottom to top for each channel. The flow area should be the volume average areas between the two appropriate cells.

(iii) Example: See Figure A.1.

The order of arx is

arx(1,1) arx(2,1) arx(3,1) arx(4,1)

arx(1,2) arx(2,2) arx(3,2) arx(4,2)

arx(1,3) arx(2,3) arx(3,3) arx(4,3)

. . .

. . .

. . .

arx(1,6) ..... arx(4,6)

The value of arx(3,5), for example, should be

$$arx(3,5) = \frac{Vol(3,4) + Vol(3,5)}{hx(3,4) + hx(3,5)} \qquad (A.4)$$

where vol and hx are defined in cards 18 and 19 respectively.

2.3.1.5 Card Eighteen

(i) Variables: ary(nz,nc)

(ii) Description:

ary=Mesh cell areas in the y-direction (m**2). ary for each mesh cell is the area on the lower side of the cell in the x-y plane. All other things about ary are the same as those of arx.

(iii) Example: See Figure A.1.

The value of ary(3,5), for example, should be

$$ary = \frac{vol(3,3) + vol(3,5)}{hy(3,3) + hy(3,5)} \qquad (A.5)$$

where hy is defined in card 20.

2.3.1.6 Card Ninteen

(i) Variable: arz(nz+1,nc)

(ii) Description:

arz=Mesh cell flow area in the z-direction (m**2). Here, for each channel i, the quantities arz(j,i) for j=1,2,3... nz+1 represent the areas beginning with the bottom face of the first nonfictitious cell in the channel and ending with the top face of the final non-fictitious cell.

(iii) Example: See Figure A.5.

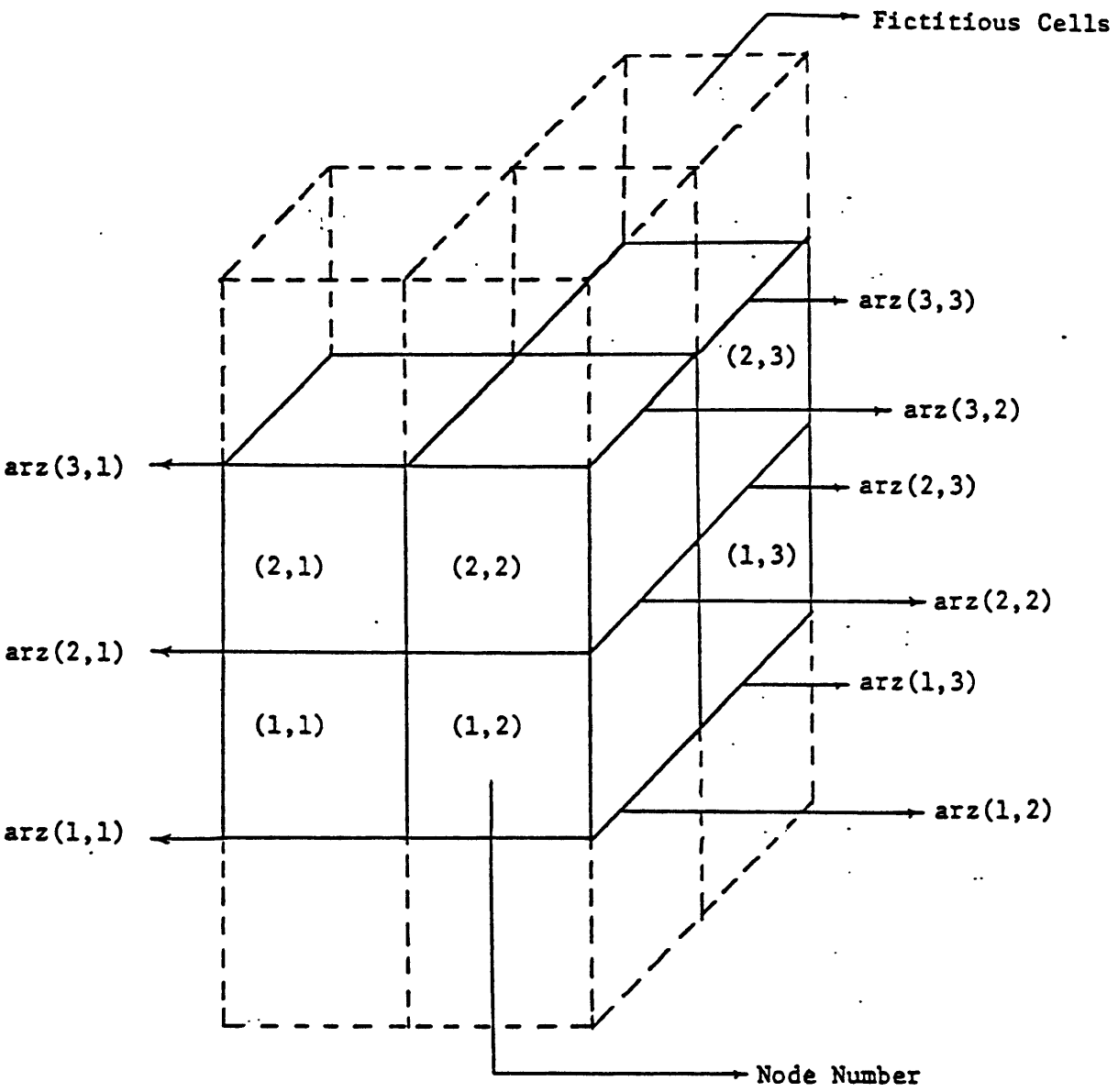


Figure A.5 Example of arz and fictitious cells

In Figure A.5, we can see the positions of arz. Note, the definition of arz is the flow area in the z-direction, therefore, we have to exclude the area occupied by the fuel rods and all other structure within that cell when we calculate arz.

2.3.1.7 Card Twenty

(i) Variables: vol(nz,nc)

(ii) Description:

vol=Mesh cell free volumes (m**3). The meaning of free volume is that the mesh cell volume excluding the volume of fuel rods and all other structure volume within that cell.

2.3.1.8 Card Twenty-One

(i) Variables: hx(nx)

(ii) Description:

hx=Mesh spacing in the x-direction (m). Since all hx must be the same for a given column, therefore, we need to specify only nx values of hx.

2.3.1.9 Card Twenty-Two

(i) Variables: hy(nr)

(ii) Description:

hy=Mesh spacing in the y-direction (m). Since all hy must be the same for a given row, therefore, we need to specify only nr values of hy.

2.3.1.10 Card Twenty-Three

(i) Variables: dz(nz+2)

(ii) Description:

dz=Mesh spacing in the z-direction (m). nz+2 means we have to specify the z-direction mesh spacing for both bottom and top fictitious cells. For fictitious cells, see Figure A.5.

2.3.1.11 Card Twenty-Four

(i) Variables: hdz(nz+2,nc)

(ii) Description:

hdz=Axial hydraulic diameter for each channel (m). This is defined as

$$hdz = 4 \cdot arz / P_{wet}$$

where P_{wet} =Wetted perimeter within the cell.

2.3.1.12 Card Twenty-Five

(i) Variables: sij(4,nc)

(ii) Description:

sij=Gap interconnections for each channel (m)=0.0 for this version of TITAN.

2.3.2 Friction Model Data, Card Twenty-Six

(i) Variables: iwfz(nz+1)

(ii) Description:

iwfz=Indicator for axial friction Model. Axial friction and form loss are specified by the array iwfz(nz+1), with one value associated with each axial velocity level. Form loss is attributed to a given axial velocity level if the spacer grid lies anywhere between the two

neighboring pressure points. The indicator iwfz is made up of a tens digit and a unit digit, whose meanings are as follows:

-- tens digit=0 axial friction only
1 axial friction+form loss
2 as 1 + funnel effect

-- unit digit=0 no friction
1 Martinelli multiplier
2 Martinelli-Nelson multiplier with mass flow effect
3 Levy multiplier
4 Rough tube correlation with Levy multiplier

If iwfz=10, that means that we chose form loss without axial friction.

2.3.3 Initial and Boundary Conditions

The initial conditions are required for both transient and steady-state calculations. For steady-state calculations, the initial condition is simply a guess, the final solution is independent of this guess. But, because of the characteristics of the boiling curve, if the initial rod temperatures are in the stable film boiling regime, the final steady-state solution may yield rod temperatures in this regime, whereas a starting guess of a lower rod temperature may yield a final steady-state solution with rod temperatures in the nucleate boiling heat transfer regime.

2.3.3.1 Card Twenty-Seven

(i) Variables: p(nz+2,nc)

(ii) Description:

p=Initial pressures (Pa). nz+2 means we have to give the initial guess to the bottom and top fictitious mesh cells too. If pressure boundary condition is chosen, the boundary velocities are determined by solving momentum equations at the boundary.

2.3.3.2 Card Twenty-Eight

(i) Variabels: alp(nz+2,nc)

(ii) Description:

alp=Initial vapor volume fraction. Same as p, we have to give the initial guess to the bottom and top fictitious mesh cells too. Note, if there is only single vapor phase exists, use alp=0.9999 instead of 1.

2.3.3.3 Card Twenty-Nine

(i) Variables: tv(nz+2,nc)

(ii) Description:

tv=Initial vapor temperature (°K). The initial liquid temperature is set equal to tv.

2.3.3.4 Card Thirty

(i) Variables: vvz(nz+1,nc)

(ii) Description:

vvz=Initial vapor axial velocity (m/s). The initial liquid axial velocity is set equal to vvz. All the initial transverse vapor and liquid

velocities are set to be zero. The (nz+1,nc) array was explained in section 2.3.1.5. If velocity boundary condition is chosen, there is no momentum equation is solved at the boundary, and the boundary velocities are set to be the input values.

2.3.4 Heat Transfer Model Input Data

This part of data is required if iht>0.

2.3.4.1 Card Thirty-One

(i) Variables: icr(nc)

(ii) Description:

icr=Adjacent channel number for a given rod. For this version of TITAN, icr(nc)=nc.

(iii) Example: For the channel arrangement in Figure A.5,

icr(1)=1

icr(2)=2

icr(3)=3

2.3.4.2 Card Thirty-Two

(i) Variables: hdh(nz+2,nc)

(ii) Description:

hdh=Equivalent heated diameter for given channel

$$= 4 \cdot \text{Flow Area/Heated Perimeter} \qquad \text{(A.7)}$$

2.3.4.3 Card Thirty-Three

(i) Variables: tw(nz,nc)

(ii) Description:

tw=Initial wall surface temperature (°K).

2.3.4.4 Card Thirty-Four

(i) Variables: qr(ncf+1+ncc)

(ii) Description:

qr=Fuel pin radial power shape.

2.3.4.5 Card Thirty-Five

(i) Variables: rn(nc)

(ii) Description:

rn=Number of fuel rods in each channel.

2.3.4.6 Card Thirty-Six

(i) Variables: fracp(nc)

(ii) Description:

fracp=Fraction of heated perimeter facing adjacent channel. In this version of TITAN, we cannot use it for subchannel analysis, therefore, we set fracp=1 for all channels.

2.3.5 Transient Forcing Functions

The transient forcing functions are used to change the boundary conditions as a function of time so that reactor transients may be simulated. The code linearly interpolates between given multipliers. If at any time which is less than the first entry then a multiplier of 1.0 is used. If at any time which is greater than the last entry, the last factor in the table is used. All of these forcing function tables can be changed or updated in the tffdata restart namelist(See 3.2.1.3).

2.3.5.1 Card Thirty-Seven

This card includes four transient forcing function indicators.

(i) Variables: nb,nt,ntemp,nq

(ii) Description:

nb=Number of entries in bottom boundary condition forcing function table ($\leq$30).

nt=Number of entries in top boundary condition forcing function table ($\leq$30).

ntemp=Number of entries in inlet temperature forcing function table ($\leq$30).

nq=Number of entries in reactor power forcing function table. Since the reactor power is generated by the code itself, nq=0 for this version of TITAN.

nitr=Number of inlet temperature regions (<10)

2.3.5.2 Card Thirty-Eight

This card is required only if nb>0.

(i) Variables: botfac(i),yb(i); i=1,nb

(ii) Description:

botfac=Bottom boundary condition multiplier. The multipliers are for pressure, velocity and total inlet flow rate according to ibb=0, 1, 2, respectively.

yb=Time corresponding to multiplier

These variables should be read in as pairs, i.e.,botfac(1),yb(1),botgad(2),

yb(2),....botfac(nb),yb(nb).

(iii) Example: If the bottom boundary condition option is velocity, and we have a flow decay transient, the multiplier is 1,0.5,0.2 corresponding to time 0 sec, 1 sec and 2 sec, respectively, then, the input should be;

1.,0.,0.5,1.,0.2,2.

2.3.5.3 Card Thirty-Nine

This card is required only if nt>0.

(i) Variables: topfac(i),yt(i); i=1,nt

(ii) Description:

topfac=Top boundary condition multiplier. The multipliers are for pressure, velocity according to ibb=0,1, respectively.

yt=Time corresponding to multiplier

These variables should be read in as pairs i.e., topfac(1),yt(1),topfac(2),yt(2),.....,

topfac(nt),yt(nt)

2.3.5.4 Card Forty

This card is required only if ntemp>0.

(i) Variables: tinfac(i,j),ytemp(i,j); i=1,ntemp, j=1,nitr

(ii) Description:

tinfac=Inlet temperature multiplier

ytemp=Time corresponding to multiplier

These variables should be read in as pairs, i.e., tinfac(1,1),ytemp(1,1),tinfac(2,1),

ytemp(2,1),.....,tinfac(ntemp,1),ytemp(ntemp,1),
tinfac(1,2),ytemp(1,2),.....,tinfac(ntemp,2),
ytemp(ntemp,2),.....,tinfac(ntemp,nitr),
ytemp(ntemp,nitr)

2.3.5.5 Card Forty-One

This card is required only if ntemp>0.

(i) Variables: nctr(j); j=1,nitr

(ii) Description:

nctr(j)=Number of channels in region j (<50)

2.3.5.6 Card Forty-Two

This card is required only if ntemp>0.

(i) Variables: ncit(i,j); i=1,nctr, j=1,nitr

(ii) Description:

ncit(i,j)=The ith channel number in region j.

2.3.6 The Nuclear Data

In this part of input data, necessary nuclear information is read.

2.3.6.1 Card Forty-Three

(i) Variables: nplane(nunqpl)

(ii) Description:

nplane=Number of planes in this unique plane.

(iii) Example: See Figure A.3.

nplane(1)=1 -- the 1st bottom fictitious plane
nplane(2)=2 -- the 2nd and 4th planes
nplane(3)=1 -- the 3rd plane
nplane(4)=1 -- the 5th top fictitious plane

2.3.6.2 Card Forty-Four

(i) Variables: iasn(nz+2)

(ii) Description:

iasn=Plane numbers belong to a certain unique plane

(iii) Example: See Figure A.3.

iasn(1)=1 -- nplane(1)

iasn(2)=2 -- nplane(2)

iasn(3)=4 -- nplane(2)

iasn(4)=3 -- nplane(3)

iasn(5)=5 -- nplane(4)

2.3.6.3 Card Forty-Five

(i) Variables: irow(nx+2,nr+2,nunqpl)

(ii) Description:

irow=Composition assignment for each unique plane. This input data must be input from left to right, bottom to top.

(iii) Example: See Figures A.3 and A.6.

irow(1,1,1)=0
irow(2,1,1)=0
irow(4,1,1)=0
irow(1,2,1)=0
.
.
. 1st plane
.
irow(4,2,1)=0
irow(1,3,1)=0
.
.

				Plane Number
0	0	0	0	
0	0	0	0	1
0	0	0	0	

-2	-1	-1	-1
-2	1	2	-1
-2	-1	-1	-1

2

-2	-1	-1	-1
-2	3	4	-1
-2	-1	-1	-1

3

-2	-1	-1	-1
-2	1	2	-1
-2	-1	-1	-1

4

-1	-1	-1	-1
-1	-1	-1	-1
-1	-1	-1	-1

5

Figure A.6 Cross-section view of Figure A.3, showing the nuclear composition assignment

```
irow(4,3,1)=0

irow(1,1,2)=-2
irow(2,1,2)=-1
irow(3,1,2)=-1
irow(4,1,2)=-1
irow(1,2,2)=-2
irow(2,2,2)=1      2nd and 4th planes
irow(3,2,2)=2
irow(4,2,2)=-1
irow(1,3,2)=-2
irow(2,3,2)=-1
irow(3,3,2)=-1
irow(4,3,2)=-1

irow(1,1,3)=-2
irow(2,1,3)=-1
irow(3,1,3)=-1
irow(4,1,3)=-1
irow(1,2,3)=-2
irow(2,2,3)=3        3rd plane
irow(3,2,3)=4
irow(4,2,3)=-1
irow(1,3,3)=-2
irow(2,3,3)=-1
irow(3,3,3)=-1
```

irow(4,3,3)=-1

irow(1,1,4)=-1
.
.
.
irow(4,2,3)=-1

5th plane

2.3.6.4 Card Forty-Six

(i) Variables: xsrf(14,ncomp)

(ii) Description:

xsrf=The reference cross section of each composition material. For each composition, we need 14 different cross sections corresponding to two-group approximation. They are described as follows.

xsrf(1,j)=Group 1 diffusion coefficient in x-direction, D_{1x}

xsrf(2,j)=Group 1 total cross section, $\Sigma_1 = \Sigma_a + \Sigma_{21}$

xsrf(3,j)=Group 1 scattering cross section Σ_{21}

xsrf(4,j)=Group 1 ν *fission cross section $\nu\Sigma_{f1}$

xsrf(5,j)=Group 1 fission cross section Σ_{f1}

xsrf(6,j)=Group 2 diffusion coefficient in x-direction, D_{2x}

xsrf(7,j)=Group 2 total cross section $\Sigma_2 = \Sigma_a$

xsrf(8,j)=Group 2 scattering cross section $\Sigma_{12}=0$

xsrf(9,j)=Group 2 ν *fission cross section $\nu\Sigma_{f2}$

xsrf(10,j)=Group 2 fission cross section Σ_{f2}

xsrf(11,j)=Group 1 diffusion coefficient in y-direction, D_{1y}

xsrf(12,j)=Group 2 diffusion coefficient in y-directionn, D_{2y}

xsrf(13,j)=Group 1 diffusion coefficient in z-direction, D_{1z}

xsrf(14,j)=Group 2 diffusion coefficient in z-direction, D_{2z}

where j is the jth composition; j=1,ncomp. All the above variables have the same dimension [cm^{-1}].

2.3.6.5 Card Forty-Seven

This card is required only if ixenon=1.

(i) Variables: xesig(ncomp)

(ii) Description:

xesig=Xenon cross section of each composition [b].

2.3.6.6 Card Forty-Eight

This card is required only if ixenon=1.

(i) Variables: yield(ncomp)

(ii) Description:

yield=Xenon yield. This yield includes γ_{Xe} and γ_I , i.e.,

$$\gamma_{sum} = \gamma_{Xe} + \gamma_I \tag{A.8}$$

This is due to the fact that xenon is created by the decay of I and the direct fission process.

2.3.6.7 Card Forty-Nine

This card is required only if ixenon=1.

(i) Variables: xelam(ncomp)

(ii) Description:

xelam=Xenon decay constant of each composition.

2.3.6.8 Card Fifty

This card is required only if nalb>0.

(i) Variables: alb(15,nalb)

(ii) Description:

alb=albedoes and expansion factors. Altogether we need 15 variables, five for each direction (x,y,z). The five basic variables are defined as follows.

$$\begin{bmatrix} \phi_1 \\ \phi_2 \end{bmatrix} = \begin{bmatrix} a_1 & a_2 \\ a_3 & a_4 \end{bmatrix} \begin{bmatrix} J_1 \\ J_2 \end{bmatrix} \qquad (A.9)$$

where ϕ_1, ϕ_2 are the group 1 and 2 neutron fluxes, and J_1, J_2 are the group 1 and 2 neutron currents. a_1, a_2, a_3, and a_4 are the first four variables. The fifth one is so called ALRATIO, which is defined as the ratio of the tranverse leakage in the last node of the reactor to the next nonexistant node. A value of -1.0 is usually a good value.

In Summary:

alb(1,j)=x-directed a_1

alb(2,j)=x-directed a_2

alb(3,j)=x-directed a_3

alb(4,j)=x-directed a_4

alb(5,j)=x-directed ALRATIO

alb(6,j)=y-directed a_1

alb(7,j)=y-directed a_2

alb(8,j)=y-directed a_3

alb(9,j)=y-directed a_4

alb(10,j)=y-directed ALRATIO

alb(11,j)=z-directed a_1

alb(12,j)=z-directed a_2

alb(13,j)=z-directed a_3

alb(14,j)=z-directed a_4

alb(15,j)=z-directed ALRATIO

where j=1,nalb.

2.3.6.9 Card Fifty-One

Variables: iedx(nedtx)

(ii) Description:

iedx=Last node number in each edit segment; x-direction.

(iii) Example: See 2.3.6.11.

2.3.6.10 Card Fifty-Two

(i) Variables; iedy(nedty)

(ii) Description:

iedy=Last node number in each edit segment, y-direction.

(iii) Example: See 2.3.6.11.

2.3.6.11 Card Fifty-Three

(i) Variables: iedz(nedtz)

(ii) Description:

iedz=Last node number in each edit segment, z-direction.

(iii) Example: See Figure A.3 and refer to 2.2.12.

The example given in 2.2.12 set nedtx=2, nedty=1 and nedtz=3. This means that we want to edit the power for each node. Therefore, iedx, iedy, iedz should be:

idex(1)=1

iedx(2)=2

iedy(1)=1

iedz(1)=1

iedz(2)=2

iedz(3)=3

Now, if we want to edit the power for the two channels together, then

nedtx=1

nedty=1

nedtz=3

and

iedx(1)=2

iedy(1)=1

iedz(1)=1

iedz(2)=2

iedz(3)=3

2.3.6.12 Card Fifty-Four

This card reads the cross section feedback coefficients with repect to the moderator density, ρ_m .

(i) Variables: aa(10,ncomp) [cm^{-1} /(gm/cm^3)]

(ii) Description:

aa=Partial of cross sections w.r.t. .

For each composition we need 10 aa's.

aa(1,j)= $\partial D_1 / \partial \rho_m$; if ithfbk=1, then

aa(1,j)= $\partial (D_1^{-1}) / \partial \rho_m$.

aa(2,j)= $\partial \Sigma_{c1} / \partial \rho_m$ = $\partial \Sigma_1 / \partial \rho_m - \partial \Sigma_{f1} / \Sigma \rho_m$

aa(3,j)= $\partial \Sigma_{21} / \partial \rho_m$

aa(4,j)= $\partial (\nu \Sigma_{f1}) / \partial \rho_m$

aa(5,j)= $\partial \Sigma_{f1} / \partial \rho_m$

aa(6,j)= $\partial D_2 / \partial \rho_m$; if ithfbk=1, then

aa(6,j)= $\partial (D_2^{-1}) / \partial \rho_m$.

aa(7,j)= $\partial \Sigma_{c2} / \partial \rho_m$ = $\partial \Sigma_2 / \partial \rho_m - \partial \Sigma_{f2} / \partial \rho_m$

aa(8,j)= $\partial \Sigma_{12} / \partial \rho_m$ =0

aa(9,j)= $\partial (\nu \Sigma_{f2}) / \partial \rho_m$

aa(10,j)= $\partial \Sigma_{f2} / \partial \rho_m$

where j=1,ncomp.

2.3.6.13 Card Fifty-Four

This card reads the cross section feedback coefficients with respect to the coolant temperature.

(i) Variables: bb(10,ncomp) [$cm^{-1}/°K$]

(ii) Description:

bb=Partial of cross sections w.r.t. T_c . The definitions of bb's are the same as aa's ex-

cept ρ_m now is T_c .

2.3.6.14 Card Fifty-Six

This card reads the cross section feedback coefficients with respect to the square root of the fuel temperature. If ithfbk=1, the coefficients are dependent of the fuel temperature with square root.

(i) Variables: cc(10,ncomp) [$cm^{-1}/\sqrt{°K}$] or [$cm^{-1}/°K$]

(ii) Description:

cc=Partial of cross sections w.r.t $\sqrt{T_f}$ of T_f .

The definitions of cc's are the same as aa's except ρ_m now is $\sqrt{T_f}$ if ithfbk $\neq$ 1. Otherwise, ρ_m now is T_f .

2.3.6.15 Card Fifty-Seven

This card is required only if ithfbk $\neq$ 1.

(i) Variables: dd(10,ncomp)

(ii) Description:

dd=Partial of unperturbed cross sections w.r.t. ρ_m^2. The definitions of dd's are the same as aa's except $\partial/\partial\rho_m$ now is $\partial^2/\partial\rho_m^2$.

In summary:

Cross Section	ithfbk	aa	bb	cc	dd
D	$\neq 1$	$\frac{\partial D}{\partial \rho_m}$	$\frac{\partial D}{\partial T_c}$	$\frac{\partial D}{\partial \sqrt{T_f}}$	$\frac{\partial^2 D}{\partial \rho_m^2}$
D	$= 1$	$\frac{\partial (1/D)}{\partial \rho_m}$	$\frac{\partial (1/D)}{\partial T_c}$	$\frac{\partial (1/D)}{\partial T_f}$	___

Σ	$\neq 1$	$\frac{\partial \Sigma}{\partial \rho_m}$	$\frac{\partial \Sigma}{\partial T_c}$	$\frac{\partial \Sigma}{\partial \sqrt{T_f}}$	$\frac{\partial^2 \Sigma}{\partial \rho_m^2}$
Σ	$= 1$	$\frac{\partial \Sigma}{\partial \rho_m}$	$\frac{\partial \Sigma}{\partial T_c}$	$\frac{\partial \Sigma}{\partial T_f}$	——

2.4 Neutronic Boundary Conditions, Card Fifty-Eight

This card reads the neutronic boundary conditions. Define the notations as follows.

0=Zero Flux

1=Zero Current

2=Albedo

(i) Variables: ibcxl,ibcxu,ibcyl,ibcyu,ibczl,ibczu

(ii) Description:

ibcxl=Boundary condition at x-direction lower side

ibcxu=Boundary condition at x-direction upper side

ibcyl=Boundary condition at y-direction lower side

ibcyu=Boundary condition at y-direction upper side

ibczl=Boundary condition at z-direction lower side

ibczu=Boundary condition at z-direction upper side

(iii) Example: See Figure A.7.

ibcxl=1 (left side)

ibcxu=2 (right side)

ibcyl=0 (front side)

ibcyu=2 (back side)

ibczl=1 (bottom side)

ibczu=0 (top side)

2.5 Thermal-Hydraulic Reference Data

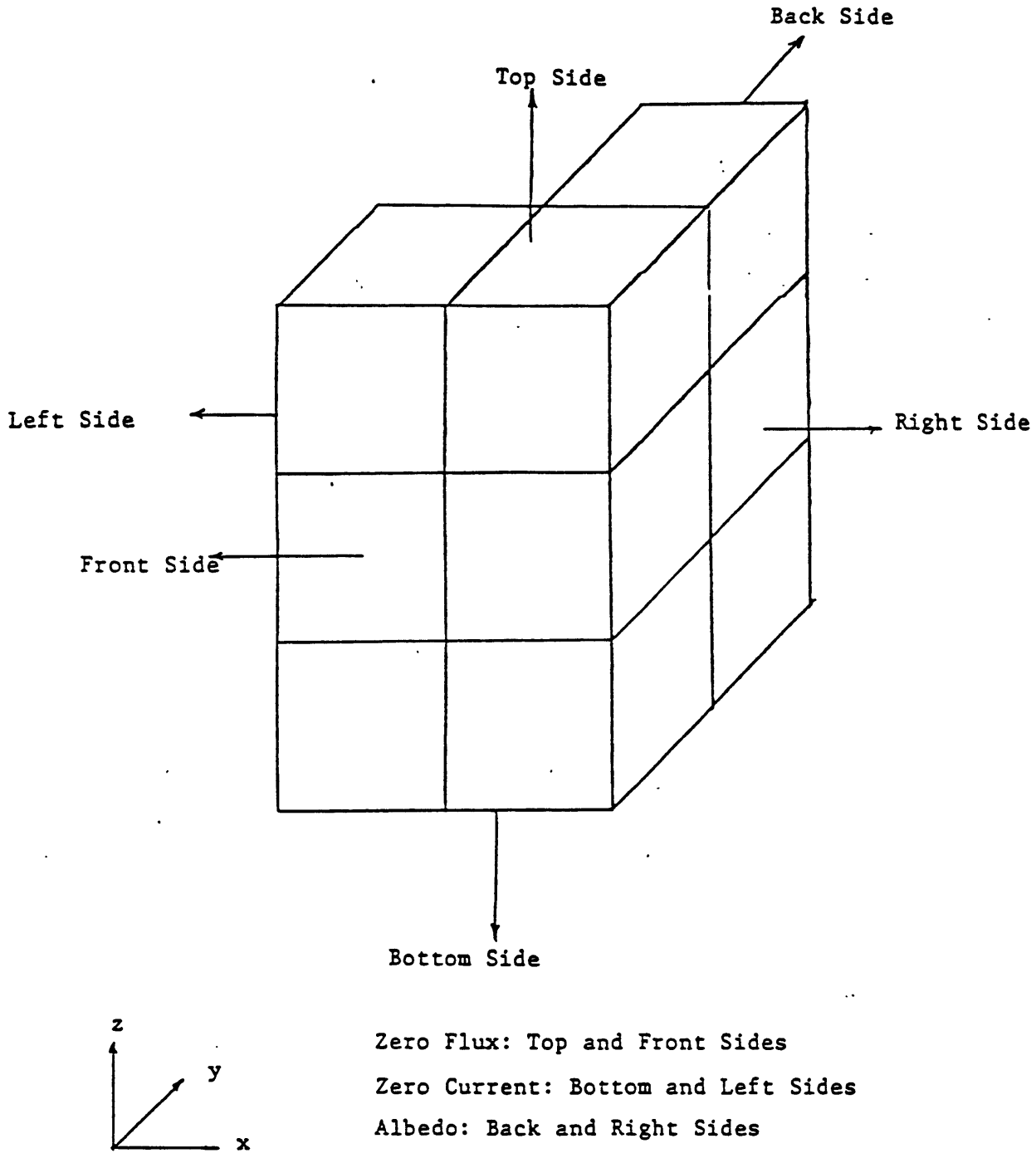


Figure A.7 Example of neutronic boundary condition

2.5.1 Card Fifty-Nine

(i) Variables: cfuel,cmod,rofuel,flozro,hzero,u,
ah,vfracm,tempin,ratiom,pressr,drhdtm

(ii) Description:

The variables required in this card are needed as the initial guess for the simple heat transfer model used in the code.

cfuel=Specific heat of fuel [Erg/(gm-°K)]

cmod=Specific heat of coolant [Erg/(gm-°K)]

rofuel=Density of fuel [gm/cm^3]

flozro=Initial mass flow rate through the core [gm/sec]

hzero=Film coefficient at initial flow rate [Erg/(cm^2 -°K-sec)], obtained from $hD/k=0.023Re^{0.8}Pr^{0.4}$

u=Conductivity/conduction length of fuel clad [Erg/(cm^2 -°K-sec)]

ah=Surface area of clad/Volume of coolant [cm^{-1}]

vfracm=Volume fraction of coolant, i.e.,

$$V_{coolant}/(V_{coolant} + V_{fuel})$$

tempin=Inlet temperature of the coolant [°K]

ratiom=Fraction of fission energy released into the coolant

pressr=Coolant pressure [Pa]

drhdtm=Patial of density*enthalpy w.r.t. coolant temperature [Erg/(cm^3 -°C)], i.e.,

$\partial(\rho h)_c / \partial T_c$

2.5.2 Card Sixty

(i) Variables: tfref,tmref,denref,ixavg

(ii) Description:

tfref=Reference fuel temperature [°K]

tmref=Reference coolant temperature [°K]

denref=Reference coolant density [gm/cm^3]

ixavg=Cross section average option (0/1) (No/Yes)

(See Section 4.2 of the report)

2.6 Time Step Control, Card Sixty-One

For each time zone, eight variables are required.

(i) Variables:

tend,dtmin,dtmax,dtsp,dtlp,clm,iredmx,nfeed

(ii) Description:

tend=End of time zone

dtmin=Minimum time step size allowed in time zone

dtmax=Maximum time step size allowed in time zone

dtsp=Time interval for short prints

dtlp=Time interval for long prints

clm=Multiplier for convective time step limit

iredmx=Maximum allowed number of time strp reduction

nfeed=Number of the thermohydraulic calculations per neutronic calculation

The time step sizes for both neutronic part and thermohydraulic part are the same. To determine the time step

sizes in a time zone, at the beginning of a time step, the z-direction vapor velocities and axial mesh spacings are used to compute the convective time step limit; the z-direction liquid velocities and tranverse velocities for vapor and liquid are ignored, under the assumption that the true convective limit will normally be determined by axial vapor velocities. There are, of course, situations in which this is not the case. We next multiply the convective limit by the parameter clm and call the result $\overline{\Delta t}$. The time step size actually used by the code is then set to the following value:

$$\Delta t = \min(\text{dtmax}, \overline{\Delta t}) \qquad \text{(A.10)}$$

When the user sets dtmax=dtmin the code bypasses the calculations of the convective limit and sets Δt=dtmin.

Printing occurs at selected time steps as determined by the parameters dtsp and dtlp. These parameters are used to determine the times at which a print is desired. If t_o represents the time at the beginning of the time zone, then prints should occur at the time t_o +k*dtlp for k=1,2,... In fact these times may not correspond to time step boundaries, so the code attemps to print at the time steps nearest the above times. Computation continues in the above manner until the time exceeds "tend" or until the time equals "tend" within a tolerance of 1.e-7 sec. At this point, new values of the above eight quantities are input, defining a new time zone. As many time zones as desired can be used in any one problem. The code will continue the computation as long as a positive

value is input for tend. The value tend=0 is always taken to signify the end of the problem, and at this point the code attempts to read data for another problem from the input data file. If tend<0, it means that the code requests a restart. The code will then prompt the user to enter data and will request a new time zone card. For more detailed discussion, see the next section, problem restart.

If at any time step the pressure problem diverges (e.g. negative void fraction), then the code automatically reduces the time step size by a factor of 10 and tries to converge using this smaller time step size. If with this smaller size the code still does not converge, the time step is again reduced. This procedure continues until $\Delta t<dtmin$ or until the number of reductions is greater than iredmx at which point execution is terminated. Of course, if with the smaller time step the code converges, then the calculations continue and the time step size is gradually increased.

For steady-state calculation, the user can request the code to perform "nfeed" times neutronic calculations per one time thermohydraulic calculation. This will save computation time since the variation of the thermohydraulic behavior is not as sensitive as that of the neutronic behavior. Here we say one time neutronic calculation means that with the thermohydraulic boundary conditions from the previous time step, the neutronic part of the calculation will iterate until the critical condition obtained at this time step. While one time thermohydraulic calculation means that the pressure

solution meets the convergerce criteria but not necessary obtain the steady-state situation.

For transient calculation, nfeed can be set any number because the code bypasses it, and calculate both neutronoc and thermohydraulic parts one time at each time step.

3 Problem Restart

Several restart options are described in this section. The restart option makes use of external devices to dump and read-in the common blocks. The dump file is automatically created at the end of a run when idump=1.

3.1 Continue Running A Problem

This option is not a real restart option. As described in 2.6, every time the code finishing on time zone, it reads the next time control card. If tend=0, the job is finished, the common blocks are then dumped into dump file. If tend<0, this means you want the code to continue the same problem by supplying the following two cards through the terminal directly.

3.1.1 Card One

This card is required only if the problem is a restart transient problem, i.e., itrans $\neq$ 0 (see 3.2.2.8).

(i) Variables: itd,nupdat,nedit,
nprint,theta1,theta2,error

(ii) Description:

All these variables will be described in detail in

3.2.2.20. Since this card is required only if the problem is a restart transient problem, all the above variables have been defined in the previous time zone, we want to keep this information except itd, we have to set it to be 2. This is an indicator tells the code that now we want to continue the restart transient problem. Therefore, the input for itd should be 2.

3.1.2 Card Two

This card is required whenever you use this option.

(i) Variables:

tend,dtmin,dtmax,dtsp,dtlp,clm,iredmx,nfeed

(ii) Description:

All the above variables have been described in 2.6.

3.2 Restart A Problem

If we have a dump file obtained from the previous calculation, we wnat to restart this problem, two types of restart options are available depend on the value of ntc.

3.2.1 Steady-State restart

This is the case that ntc=-2 and ithfbk=2. No neutronic transient data for perturbed cross sections are read in. With ntc=-2, the job restarts from a previously created dump file.

A number of variables may be changed at a restart option. This is accomplished through use of the FORTRAN namelist input feature. Two namelists are available, the 'restart' and the 'tffdata'.

3.2.1.1 Card One

(i) Variable: ntc

(ii) Description:

ntc=-2 for this option.

3.2.1.2 Card Two

In this card, ten flags and two variables are read in again, which means we are able to restart a problem with different options for these ten flags and two variables.

(i) Variables:

ithfbk,ixavg,idiag,jprinta,jprintb,jprintc,jprintd, jprinte,jprint,irstrt,nitr,ntemp

(ii) Description:

All the variables are defined. Please refer to the following sections:

ithfbk: 2.2.9

ixavg: 2.5.2

idiag: 2.2.9

jprinta: 2.2.9

jprintb: 2.2.9

jprintc: 2.2.9

jprintd: 2.2.9

jprinte: 2.2.9

jprint: 2.2.9

irstrt: 2.2.9

nitr: 2.3.5.1

ntemp: 2.3.5.1

3.2.1.3 Card Three

This card contains the 1st namelist, 'restart', variables.

(i) Variables: nitmax,iitmax,epsn,epsi,iflash,
itb,ibb,hdt,grav,iht,iss,
iwft,ivec,idump,itam,ichf,iqss,imixm,imixe

(ii) Description:

All the above variables have been defined. We can choose the variables we want to change during the restart calculation, let the input look like

$ restart f1=x1,f2=x2,...$

where fi is the name of the variable we want to change and xi is the new assigned value.

(iii) Example: $restart iht=2$

3.2.1.4 Card Four

This card contains the 2nd namelist, 'tffdata', variables.

(i) Variables: nb,nt,nq,botfac(i),topfac(i),
gfac(i),yb(i),yt(i),yq(i)

(ii) Description:

All the above variables have been defined. We can choose the variables we want to change during the restart calculation, let the input look like

$ tffdata f1=x1,f2=x2,...$

where fi is the name of the variable we want to

change and xi is a single new assinged value or a set of new assigned values.

(iii) Example:

$tffdata nb=2,botfac(1)=1,0.7,yb(1)=0,0.2$

3.2.1.5 Card Five

This card is required only if ntemp>0.

(i) Variables: tinfac,ytemp

(ii) Description: See 2.3.5.4.

3.2.1.6 Card Six

This card is required only if ntemp>0.

(i) Variables: nctr

(ii) Description: See 2.3.5.5.

3.2.1.7 Card Seven

This card is required only if ntemp>0.

(i) Variable: ncit

(ii) Description: See 2.3.5.6.

3.2.1.8 Card Eight

This is the time control card. Please refer to 2.6.

3.2.2 Transient Restart

This is the case that ntc=-3. A set of neutronic transient data should be read in, which takes care of the perturbed cross sections feedback coefficients. If you set ntc=-3 and ithfbk=2, then the perturbed cross sections feedback coefficients are neglected.

3.2.2.1 Card One

(i) Variable: ntc

(ii) Description:

ntc=-3 for this option.

3.2.2.2 Card Two

This card is the same as card two described in 3.2.1.2.

3.2.2.3 Card Three

This card contains the 1st namelist, 'restart', variables. It is exactly the same as described in 3.2.1.2.

3.2.2.4 Card Four

This card contains the 2nd namelist,'tffdata', variables. See 3.2.1.3.

3.2.2.5 Card Five

This card is the same as the card described in 3.2.1.5.

3.2.2.6 Card Six

This card is the same as the card described in 3.2.1.6.

3.2.2.7 Card Seven

This card is the same as the card described in 3.2.1.7.

3.2.2.8 Card Eight

This card contains transient indicators.

(i) Variables: itrans,ndpg,nodalt,epsk,epsphi,errorr

(ii) Description:

itrans=Transient type indicator

=0: Null transient

=1: Control rod transient with Cusping correction

=2: Flow transient or temperature transient

=3: Only thermohydraulic part calcula-

tion is performed

=4: Control rod transient without Cusping correction

=5: Combined condition 2 and 1

=6: Combined condition 2 and 4

ndpg=Number of delayed neutron groups

nodalt=Number of nodes in which rod will move

epsk=See 2.2.12

epsphi=See 2.2.12

errorr=See 2.2.12

3.2.2.9 Card Nine

(i) Variables: v(2,ncomp)

(ii) Description:

v=Group neutron velocity

v(1,j)=Group one neutron velocity [cm/sec]

v(2,j)=Group two neutron velocity [cm/sec]

3.2.2.10 Card Ten

(i) Variables: beta(ndpg)

(ii) Description:

beta(j)=delayed neutron fraction for group j; j=1,ndpg.

3.2.2.11 Card Eleven

(i) Variables: lmda(ndpg)

(ii) Description:

lmda(j)=Precursor delay constant for group j [1/sec]; j=1,ndpg.

3.2.2.12 Card Twelve

This card is required only if nodalt ≠ 0.

(i) Variables: ni(nodalt)

(ii) Description:

ni=The x-direction index of node to be perturbed.

(iii) Example: See Figure A.8.

From this figure, we can see that we have 5 nodes that are going to be perturbed. Therefore,

nodalt=5

ni(1)=1

ni(2)=1

ni(3)=1

ni(4)=1

ni(5)=1

3.2.2.13 Card Thirteen

This card is required only if nodalt ≠ 0

(i) Variables: nj(nodalt)

(ii) Description:

nj=The y-direction index of node to be perturbed.

(iii) Example: See Figure A.8.

nj(1)=1

nj(2)=1

nj(3)=1

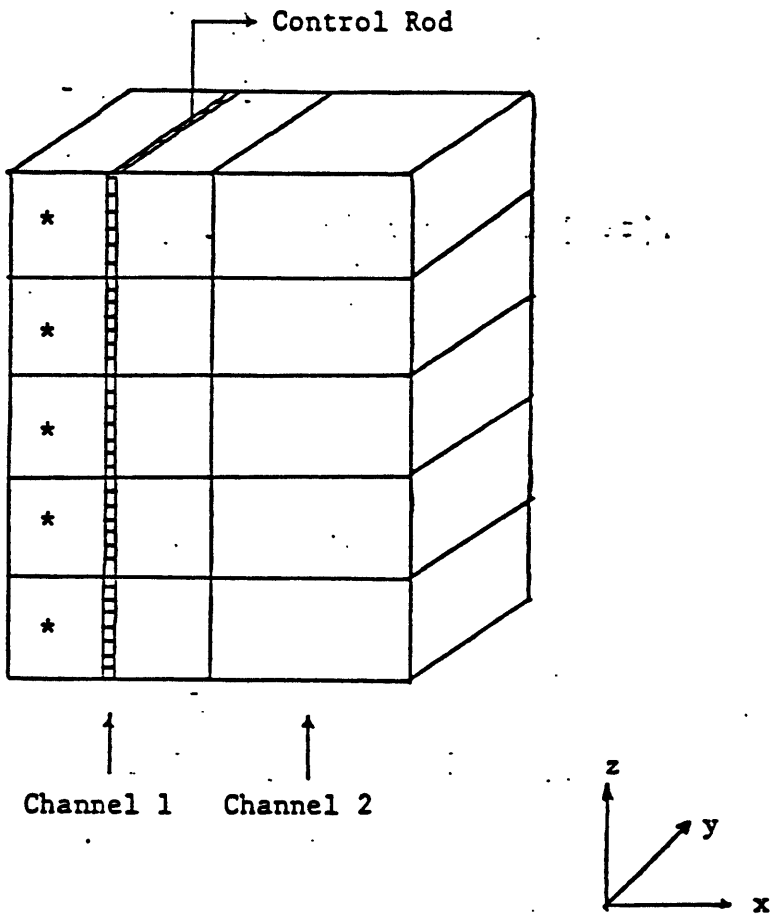
nj(4)=1

nj(5)=1

3.2.2.14 Card Fourteen

This card is required only if nodalt ≠ 0.

(i) Variables: nk(nodalt)



* is the node perturbed

Figure A.8 Example of ni, nj, nk

(ii) Description:

nk=The z-direction index of node to be perturbed.

(iii) Example: See Figure A.8.

nk(1)=1

nk(2)=2

nk(3)=3

nk(4)=4

nk(5)=5

(iv) Composition Perturbation:

There is another way to use ni, nj, nk, and nodalt to model the perturbed portion of the core. This is so called the "composition perturbation", i.e., once the perturbation is initiated, the nodes which have the same composition (identified by "nk") are perturbed.

If we set:

ni=0

nj=0, the code will automatically go to the "composition perturbation" option. Now,

nk=The compositions to be perturbed

ndoalt=Number of the compositions to be perturbed

For example, let

ndoalt=3

nk(1)=2

nk(2)=3

nk(3)=5, then, the code will know that the nodes with composition 2 are going to be perturbed at the time tstart(1) (See 3.2.2.15). The perturbation ceases at the time tend(1)

(See 3.2.2.16). The same procedure is applied to compositions 3 and 5.

3.2.2.15 Card Fifteen

This card is required only if nodalt ≠ 0.

(i) Variables: tstart(nodalt)

(ii) Description:

tstart=Time that node perturbation commences [sec].

(iii) Example: See Figure A.8.

Now, suppose we have the control rod ejection accident, the control rod in channel will be ejected from the node (1,1,1) all the way out of that channel within 0.1 sec. Therefore, the time that node perturbation commences should be

tstart(1)=0.

tstart(2)=0.02

tstart(3)=0.04

tstart(4)=0.06

tstart(5)=0.08

Note, the value 0.02 is an averaged value obtained from 0.1 sec. divided by 5 nodes.

3.2.2.16 Card Sixteen

This card is required only if nodalt ≠ 0.

(i) Variables: tend(nodalt)

(ii) Description:

tend=Time that node perturbation ceases [sec].

(iii) Example: See Figure A.8.

tend(1)=0.02

tend(2)=0.04

tend(3)=0.06

tend(4)=0.08

tend(5)=1.0

3.2.2.17 Card Seventeen

This card is required only if nodalt ≠ 0.

(i) Variables: dlxs(14,nodalt)

(ii) Description:

dlxs=Actual perturbed cross sections

dlxs(1,j)=Perturbed x-direction group 1 diffusion constant

= ΔD_{1x}

dlxs(2,j)=Perturbed group 1 total cross section

= $\Delta \Sigma_1$

dlxs(3,j)=Perturbed group 1 scattering cross section

= $\Delta \Sigma_{21}$

dlxs(4,j)=Perturbed group 1 ν*fission cross section

= $\Delta(\nu \Sigma_{f1})$

dlxs(5,j)=Perturbed group 1 fission cross section

= $\Delta \Sigma_{f1}$

dlxs(6,j)=Perturbed x-direction group 2 diffusion constant

= ΔD_{2x}

dlxs(7,j)=Perturbed group 2 total cross section

= $\Delta \Sigma_2$

dlxs(8,j)=Perturbed group 2 scattering cross section

= $\Delta \Sigma_{12} = 0$

dlxs(9,j)=Perturbed group 2 ν*fission cross section

= $\Delta(\nu \Sigma_{f2})$

dlxs(10,j)=Perturbed group 2 fission cross section

$= \Delta\Sigma_{f2}$

dlxs(11,j)=Perturbed y-direction group 1 diffusion constant

$= \Delta D_{1y}$

dlxs(12,j)=Perturbed y-direction group 2 diffusion constant

$= \Delta D_{2y}$

dlxs(13,j)=Perturbed z-direction group 1 diffusion constant

$= \Delta D_{1z}$

dlxs(14,j)=Perturbed z-direction group 2 diffusion constant

$= \Delta D_{2z}$

where Δ =(Cross section after perturbed)-(Cross section before perturbed)

3.2.2.18 Card Eighteen

This card is required only if nodalt $\neq$ 0 and ithfbk=3.

(i) Variables: ee(10,ncomp)

(ii) Description:

ee=Partial of the perturbed cross sections w.r.t. ρ_m in the perturbed nodes. The definitions of ee's are the same as aa's described in 2.3.6.12 except that ee's are for the perturbed part of the nodes only.

3.2.2.19 Card Ninteen

This card is required only if nodalt $\neq$ 0 and ithfbk=3.

(i) Variables: ff(10,ncomp)

(ii) Description:

ff=Partial of the perturbed cross sections w.r.t. ρ_m^2 in the perturbed nodes. The definition of ff's are the same as dd's described in 2.3.6.15, except that ff's are for the perturbed part of the nodes only.

3.2.2.20 Card Twenty

(i) Variables: itd,nupdat,nedit, nprint,theta1,theta2,error

(ii) Description:

itd=Number of time domains. Suppose we have a steady-state dump file which has

been created by a steady-state calculation. We want to perform a 2 seconds transient calculation by performing two one second calculations. Two steps should be followed.

1: restart the steady-state dump file with itd=1 for the 1st second calculation. A new dump file is created after this calculation.

2: restart the dump file created in step 1 with itd=2 for the 2nd second calculation.

nupdat=Steps for updating the matrix

nedit=Times for editing the neutronic data.

For example, if nedit=2, the code will print the neutronic data once every 2 time steps.

nprint=Print flag for point kinetics omegas(0/1) (No/Yes)

theta1=Flux theta (0.0< to $\leq$1.0) (Recommend 1.0)

theta2=Delayed theta (0.0< to $\leq$1.0) (Recommend 1.0)

error=Convergence criteria (Recommend 1.e-3,1.e-4)

3.2.2.21 Card Twenty-One

This is the time control card. See 2.6.