Argonne NATIONAL LABORATORY

ANL/NE-16/19

The SAS4A/SASSYS-1 Safety Analysis Code System

Nuclear Engineering Division

About Argonne National Laboratory

Argonne is a U.S. Department of Energy laboratory managed by UChicago Argonne, LLC under contract DE-AC02-06CH11357. The Laboratory's main facility is outside Chicago, at 9700 South Cass Avenue, Argonne, Illinois 60439. For information about Argonne and its pioneering science and technology programs, see www.anl.gov.

Document Availability

Disclaimer

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor UChicago Argonne, LLC, nor any of their employees or officers, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of document authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof, Argonne National Laboratory, or UChicago Argonne, LLC.

Argonne NATIONAL LABORATORY

ANL/NE-16/19

The SAS4A/SASSYS-1 Safety Analysis Code System

Chapter 3:

Steady-State and Transient Thermal Hydraulics in Core Assemblies

F. E. Dunn
Nuclear Engineering Division
Argonne National Laboratory

March 31, 2017

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

NOMENCLATURE

Subscript	Description
c	Coolant
e	Cladding
f	Fuel
g	Plenum gas
kz	Reflector zone
p	Gas plenum region
si	Structure inner node
so	Structure outer node
1	Beginning of the time step
2	End of the time step

Symbol	Description	Units
A_c	Coolant flow area	m^2
A_{ep}	Cross sectional area of the clad in the gas plenum region	m^2
A_g	Cross sectional area of the plenum gas	m^2
A_{fr}, b_{fr}	Coefficients in the friction factor correlation: $f = A_{fr}(Re)^{b_{fr}}$	--
c	Specific heat	J/kg-K
$\bar{c}_c$	Coolant specific heat	J/kg-K
c_e	Cladding specific heat	J/kg-K
$\bar{c}_f$	Average fuel specific heat, averaged over a time step	J/kg-K
c_m	Modified specific heat in the melting range	J/kg-K
c_{mix}	Coolant specific heat in the mixing zone used for re-entry temperature calculation	J/kg-K
c_1,c_2,c_3	Correlation constants used in coolant heat-transfer coefficients	--
D	Right-hand-side terms in the matrix equations for radial temperature profiles	J/m
D_h	Hydraulic diameter	m
d_{sti}	Structure inner node thickness	m
d_{sto}	Structure outer node thickness	m
E_{ec}	Heat flux from clad to coolant, integrated over a time step	J/m^2
E_{sc}	Heat flux form structure to coolant, integrated over a time	J/m^2

Symbol	Description	Units
	step	
f	Friction factor	--
f_i	Fraction of the structure thickness represented by the inner node	--
f_o	Fraction of the structure thickness represented by the outer node	--
g	Acceleration of gravity	m/s^2
h_b	Bond gap conductance	W/m^2-K
h_c	Coolant-film heat-transfer coefficient	W/m^2-K
h_{cond}	Condensation heat-transfer coefficient for sodium vapor	W/m^2-K
H_{eg}	Heat-transfer coefficient form the gas in the gas plenum to the cladding	W/m^2-K
H_{erc}	Heat-transfer coefficient from the cladding or reflector outer node to the coolant	W/m^2-K
h_r	Equivalent radiation heat-transfer coefficient	W/m^2-K
H_{rio}	Heat-transfer coefficient from the structure inner node to the reflector outer node	W/m^2-K
H_{sic}	Heat-transfer coefficient from the structure inner node to the coolant	W/m^2-K
H_{stio}	Heat-transfer coefficient from the structure inner node to the structure outer node	W/m^2-K
i	Radial node number	--
ic	Core channel number	--
I_1	Inertial integral in the momentum equation	m^{-1}
I_2	Acceleration integral in the momentum equation	m/kg
I_3	Friction integral in the momentum equation	m/kg
I_4	Orifice term in the momentum equation	m/kg
I_5	Density integral in the momentum equation	kg/m^2
j	Fuel axial node number	--
jc	Coolant axial node number	--
JC	Axial node number	--
k	Thermal conductivity	W/m-K
k_{ep}	Cladding thermal conductivity in the gas plenum region	W/m-K
$\bar{k}_{i,j}$	Weighted average thermal conductivity for heat flow from node i to j	W/m-K
K_{or}	Orifice coefficient	--

Symbol	Description	Units
L	1 for subassembly inlet, 2 for outlet	--
MZC	Total number of coolant axial nodes	--
m_e	Cladding mass	kg
m_f	Fuel mass	kg
M_{mix}	Mass of sodium in the mixing volume	kg
NC	Radial node number of the coolant node	--
NE	Radial node number for the cladding mid-point	--
NE′	Radial node number for the cladding outer surface node	--
NE″	Radial node number for the cladding inner surface node	--
NN	NT-1	--
N_{ps}	Number of fuel pins represented by a channel	--
NR	Radial node number for the fuel outer surface radius. (note NR=NE″)	--
NSI	Radial node number for the inner structure node	--
NSO	Radial node number for the outer structure node	--
NT	Radial node number for the fuel outer surface temperature node	--
p	Pressure	Pa
p_b	Pressure at the bottom of the subassembly	Pa
p_{b1},p_{b2}	p_b at beginning and end of a time step	Pa
p_{in}	Pressure in the coolant inlet plenum	Pa
$\overline{P}_j$	Heat production rate in axial node j	W
P_r	Radial power shape, per unit mass	--
p_t	Pressure at the top of the subassembly	Pa
p_{t1},p_{t2}	p_t at the beginning and end of a time step	Pa
p_x	Pressure in the coolant outlet plenum	Pa
$\left(\frac{\partial p}{\partial z}\right)_{fr}$	Friction pressure drop	Pa/m
$\left(\frac{\partial p}{\partial z}\right)_{k}$	Orifice pressure drop	Pa/m
Q_c	Coolant heat source due to direct heating by neutrons and gamma rays	W/m^3
Q_{ct}	Total steady-state heat source per unit of coolant volume	W/m^3
Q_{ec}	Heat flow from clad to coolant	W/m^3
q_{fe}	Fuel-to-cladding heat flux	W/m^2

Symbol	Description	Units
Q_{sc}	Heat flow from structure to coolant	W/m^3
$Q_{sm}(i)$	Sum of the heat sources for all radial nodes inside and including node i	W
Q_{st}	Structure heat source due to direct heating by neutrons and gamma rays	W/m^2
Q_v	Heat source per unit volume	W/m^3
r	Radius	m
r_{brp}	Clad inner radius in the gas plenum region	m
Re	Reynolds number	--
R_{ec}	Thermal resistance between clad and coolant	m^2-K/W
R_{ehf}	Thermal resistance of the outer fourth of the cladding	m^2-K/W
r_{erp}	Cladding outer radius in the gas plenum region	m
R_g	Thermal resistance of the gas in the plenum	m^2-K/W
$r_o(i)$	Steady-state radial mesh	m
S_{er}	Perimeter of the cladding or reflector in contact with the coolant	m
S_{st}	Structure perimeter, heat-transfer area per unit height	m
T	Temperature	K
t	Time	s
T_{cin}	Coolant inlet temperature	K
T_{cout}	Coolant outlet temperature	K
T_{eex}	Extrapolated clad temperature	K
T_{eq}	Equilibrium temperature in the mixing volume	K
$\overline{T}_{exp}$	Temperature of the sodium expelled from the subassembly into a mixing volume, averaged over a time step	K
$\overline{T}_f$	Average fuel temperature at an axial node, mass-weighted average	K
T_g	Plenum gas temperature	K
T_{liq}	Liquidus temperature	K
T_{out}	Bulk temperature in the coolant outlet plenum	K
t_{p1}, t_{p2}	Times at the beginning and end of a PRIMAR time step	s
T_{ri}	Reflector inner node temperature	K
T_{ro}	Reflector outer node temperature	K
T_{sol}	Solidus temperature	K
T_o	Temperature at the beginning of a time step	K

Symbol	Description	Units
T_1	Temperature at the beginning of a time step	K
T'_1	Temperature of the coolant entering an axial node at the end of a time step	K
T_2	Temperature at the end of a time step	K
T'_2	Temperature of the coolant entering an axial node at the end of a time step	K
U_{melt}	Heat of fusion	J/kg
v	Velocity	m/s
w	Coolant mass flow rate	kg/s
w_e	Estimated mass flow rate	kg/s
w_{fe}	Thickness of the liquid-sodium film left on the cladding after voiding occurs	m
w_{fr}	Thickness of the liquid-sodium film left on the reflector after voiding occurs	m
w_{fst}	Thickness of the liquid-sodium film left on the structure after voiding occurs	m
w_1,w_2	w at beginning and end of a time step	kg/s
Δw	Change in w during a time step	kg/s
x	Distance	m
x_{I1}(JC)	Nodal contribution to I_1	m^{-1}
x_{I2}(JC)	Nodal contribution to I_2	m/kg
x_{I3}(JC)	Nodal contribution to I_3	
x_{I5}(JC)	Nodal contribution to I_5	kg/m^2
z	Axial position	m
z	Elevation	m
Δz	Node height	m
z_{pb}	Elevation at the bottom of the gas plenum	m
z_{pt}	Elevation at the top of the gas plenum	m
$z_{p\ell\ell}$	Reference elevation of the coolant inlet plenum	m
$z_{p\ell u}$	Reference elevation of the coolant outlet plenum	m
α	Heat capacity terms in the matrix equations for radial temperature profiles	J/m-K
α_e	Cladding thermal expansion coefficient	K^{-1}
α_f	Fuel thermal expansion coefficient	K^{-1}
β	Thermal conductivity terms in the matrix equations for	J/m-K

Symbol	Description	Units
	radial temperature profiles	
γ_c	Fraction of the total heat production that goes directly into the coolant	--
γ_e	Fraction of the total heat production that goes directly into the cladding	--
γ_s	Fraction of the total heat production that goes directly into the structure	--
γ_2	Ratio of the structure perimeter to the cladding perimeter	--
Δr	Radial node size	m
$\Delta r_{i,j}$	Effective radial distance for heat flow from node I to node j	m
Δt	Time-step size	s
Δz	Axial node height	m
ε	Thermal emissivity	--
θ_1,θ_2	Degree of explicitness or implicitness in the solution	--
ρ	Density	kg/m^3
ρ_c	Coolant density	kg/m^3
ρ_{cin}	Coolant density in the inlet plenum	kg/m^3
ρ_{cout}	Coolant density in the outlet plenum	kg/m^3
$(\rho c)_g$	Density times specific heat of the plenum gas	J/m^3-K
$(\rho c)_r$	Density times specific heat for the reflector	J/m^3-K
ρ_e	Cladding density	kg/m^3
σ	Stefan-Bolzmann constant	W/m^2-K^4
τ	Time constant for flow rate changes	s
τ_c	Condensation heat-transfer time constant	s
τ_{ro}	Time constant for temperature changes in the outer reflector node	s
τ_{sti}	Time constant for temperature changes in the inner structure node	s
μ	Coolant viscosity	Pa-s
$\bar{\mu}(JC)$	Average value of μ for node JC	Pa-s
ψ	Source terms in the matrix equations for radial temperature profiles	--

STEADY-STATE AND TRANSIENT THERMAL HYDRAULICS IN CORE ASSEMBLIES

3.1 Introduction

The core assembly thermal hydraulics treatment in SAS4A/SASSYS-1 includes the calculation of fuel, cladding, coolant, and structure temperatures, as well as coolant flow rates and pressure distributions. This treatment includes melting of the fuel and cladding. Boiling of the coolant is also handled, as described in Chapter 12. The relocation of fuel and cladding after pin disruption is described in Chapters 13, 14, and 16; and relocation of molten fuel before pin disruption is described in Chapter 15.

Prior to version 3.0, all of the core subassembly models in SAS4A/SASSYS-1 were single pin models: a single fuel pin and its associated coolant were used to represent a subassembly; and pin-to-pin variations within a subassembly were ignored. A multiple pin option has been added to the code in version 3.0. A number of pins and their associated coolant can now be used to represent a subassembly, so variations within a subassembly can be accounted for. Currently the multiple pin option is only available for single-phase thermal hydraulics; it does not handle coolant boiling, in-pin fuel relocation, or pin disruption. Therefore, typical SAS4A/SASSYS-1 cases that do not get into coolant boiling can be handled with the multiple-pin model, but typical core disruption cases can only be handled with single pin models.

Although SAS4A/SASSYS-1 is mainly a transient code, both steady-state and transient temperatures and coolant pressures are calculated. The steady-state solutions are obtained from the transient equations after dropping all time derivatives. In general, the steady-state solutions in the single pin per subassembly model are not obtained by running a transient calculation at constant power and flow until the results approach a steady-state solution. Instead, the steady-state temperatures are obtained rapidly from a direct solution based on conservation of energy and the use of the same spatial finite differencing as used in the transient. On the other hand, a direct steady-state solution for the multiple pin option would be much more complicated, especially if subassembly-to-subassembly heat transfer is included. Therefore, a null transient with powers and flows held constant is used to obtain steady-state conditions for the multiple pin option.

The thermal hydraulics calculations are carried out in a number of separate modules, and each module is designed for a specific type of calculation. A steady-state thermal hydraulics module provides the initial conditions for the transient. The transient temperatures are calculated in a pre-voiding module (TSHTRN) until the onset of boiling. After the onset of boiling, the fuel-pin temperatures are calculated in a separate module (TSHTRV) that couples with the boiling module.

The core thermal-hydraulic routines interact with a number of other modules, as shown in Figure 3.1–1. Before the onset of voiding, TSHTRN calculates the coolant temperatures used in the hydraulic calculations, whereas the hydraulics routines calculate the coolant flow rates used in TSHTRN. After the onset of voiding, coolant temperatures are calculated in TSBOIL, and this module supplies the heat flux at the cladding outer surface or the fuel outer surface to TSHTRV. TSBOIL uses the cladding temperatures from TSHTRV in its coolant temperature calculations. The point kinetics module supplies the power level used in the heat-transfer routines, and the heat-transfer routines supply the Doppler feedback reactivity as well as other temperature-dependent reactivity feedback. TSBOIL supplies the voiding reactivity. The inlet plenum temperature computed by PRIMAR-4 is used in calculating the inlet temperature for TSHTRN or TSBOIL, and TSHTRN or TSBOIL provides the subassembly outlet temperatures used by PRIMAR-4 to compute the outlet plenum temperature. If flow reversal occurs in a subassembly, then the outlet plenum temperature computed by PRIMAR-4 is used in calculating the coolant temperature at the top of the subassembly, and the temperature computed by TSHTRN or TSBOIL for the coolant leaving the bottom of the subassembly is used by PRIMAR-4 to calculate the inlet plenum temperature. PRIMAR-4 supplies the inlet and outlet plenum pressures that drive the coolant hydraulics calculations, and the core channel flows are provided to PRIMAR-4 by TSBOIL and the pre-voiding hydraulics. The initial coolant flow rate and pressure distribution are supplied to TSBOIL by the pre-voiding hydraulics routines.

The transient calculations in the code used a multi-level time step approach, with separate time steps for each module. For the heat-transfer routines, all temperatures are known at the beginning of a heat-transfer step, and the routines calculate the new temperatures at the end of the step. The heat-transfer time step can be longer than the coolant time step or the PRIMAR time step, but the heat-transfer time step can be no longer than the main time step that is used for reactivity feedback and main printouts.

Figure 3.1–2 shows the flow through the pre-voiding core channel thermal hydraulics driver, TSCL0. This routine is entered once for each channel during each coolant time step. The coolant flow rates are calculated before the heat-transfer module (TSHTRN) is called. TSHTRN is only called if the current coolant time step completes a heat-transfer time step. The voiding model, TSHTRV, is described in Chapter 12.

In this chapter, Section 3.2 describes the mesh structure used for heat-transfer calculations. Then, Section 3.3 describes the pre-boiling transient heat-transfer calculations, followed by the steady-state thermal hydraulics calculations in Section 3.4. The pre-voiding transient heat transfer is discussed before the steady-state thermal hydraulics for two reasons. First, the code is primarily a transient code, so the transient calculations are more important. Second, the finite difference approximations were made with the transient calculations in mind, and the steady-state solution was formulated to be consistent with the approximations used in the transient. Section 3.5 describes TSHTRV, the fuel-pin heat-transfer calculations in the boiling module. Section 3.6 describes the treatment of the bond-gap conductance between the fuel and the cladding. Section 3.7 describes modifications to the fuel pin heat transfer calculations for PLUTO2 and LEVITATE. The heat transfer time step control is described in Section 3.8. Section 3.9 describes steady-state and pre-voiding transient hydraulics. Section 3.10 describes the multiple pin option. Subassembly-to-subassembly heat transfers described in Section 3.11. Section 3.12 describes interaction with other modules. It is followed by sections providing subroutine descriptions and flowcharts, subchannel model treatment details, thermal properties of fuel and cladding, and a description of the input to, and output from, the thermal hydraulic routines.

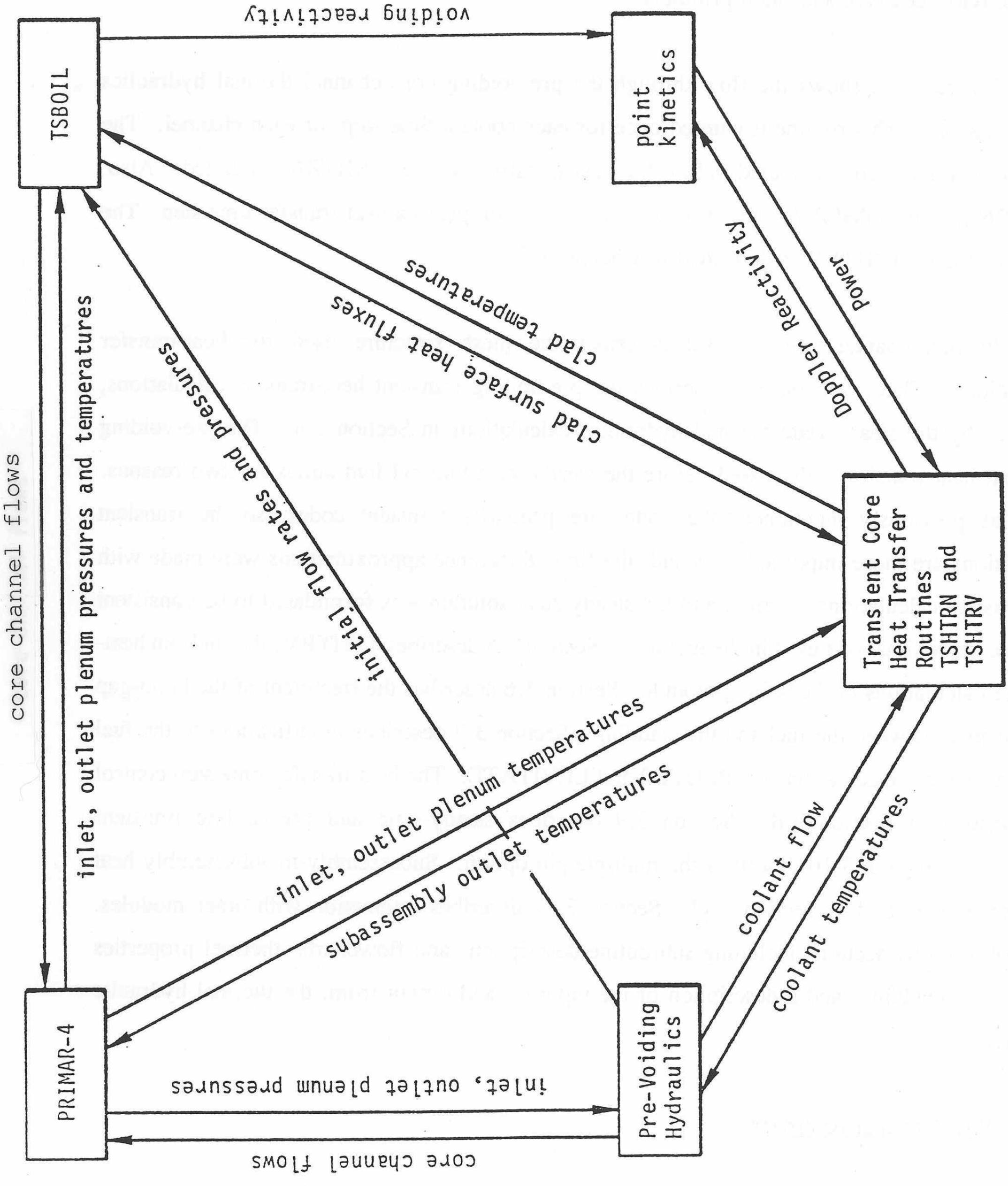

TSBOIL
voiding reactivity
point kinetics
power
Doppler Reactivity
clad temperatures
clad surface heat fluxes
initial flow rates and pressures
core channel flows
inlet, outlet plenum pressures and temperatures
Transient Core Heat Transfer Routines TSHTRN and TSHTRV
inlet, outlet plenum temperatures
subassembly outlet temperatures
coolant flow
coolant temperatures
PRIMAR-4
inlet, outlet plenum pressures
core channel flows
Pre-Voiding Hydraulics

Figure 3.1–1: Interactions of Thermal-Hydraulic Routines with Other Modules

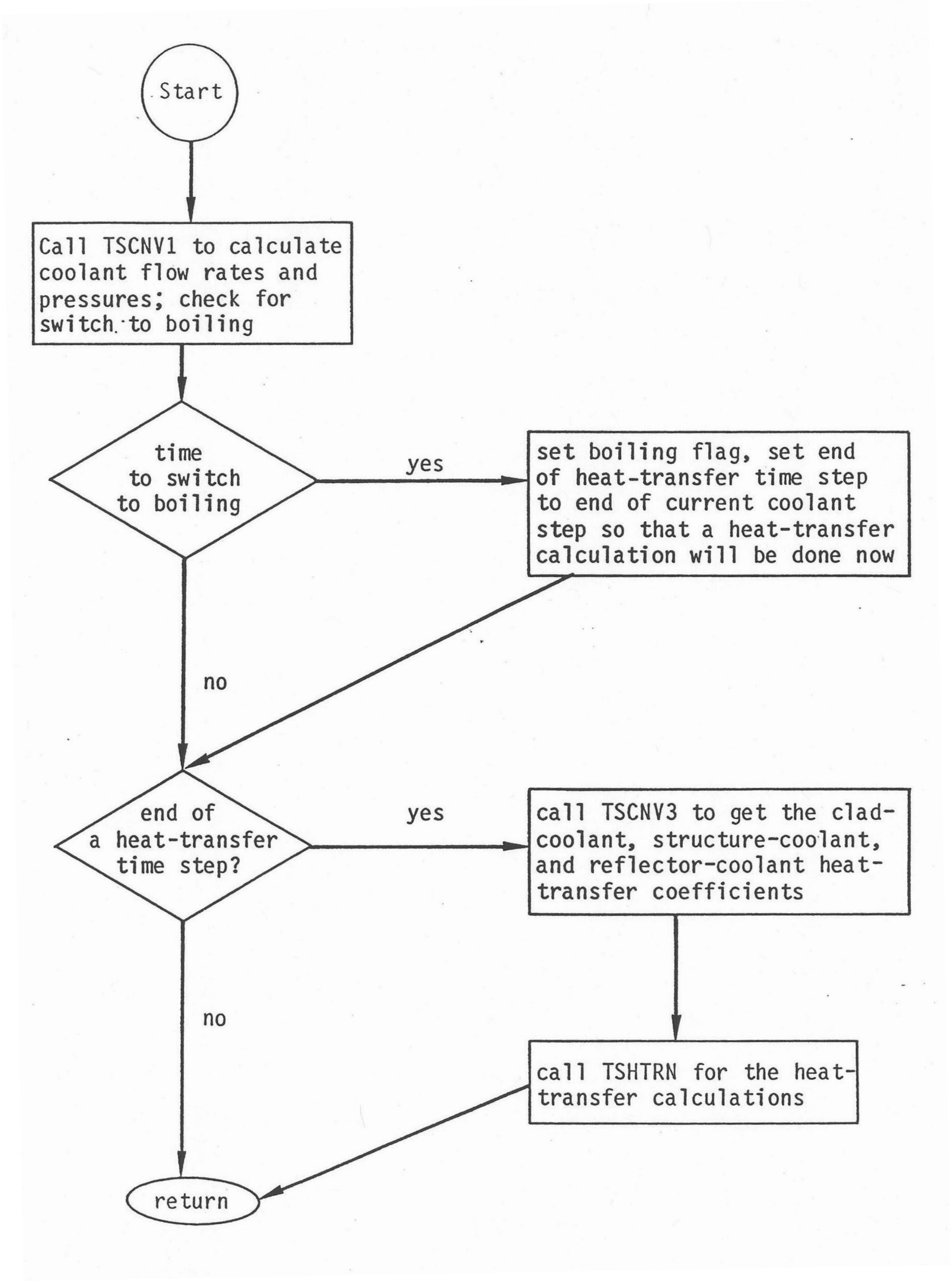

Start
Call TSCNV1 to calculate coolant flow rates and pressures; check for switch to boiling
time to switch to boiling
yes
set boiling flag, set end of heat-transfer time step to end of current coolant step so that a heat-transfer calculation will be done now
no
end of a heat-transfer time step?
yes
call TSCNV3 to get the clad-coolant, structure-coolant, and reflector-coolant heat-transfer coefficients
no
call TSHTRN for the heat-transfer calculations
return

Figure 3.1–2: Flowchart for the Pre-Voiding Core Channel Thermal Hydraulics Driver (Subroutine TSCL0)

3.2 SAS4A/SASSYS-1 Channel Approach

SAS4A/SASSYS-1 is capable of using a multi-channel treatment. Each channel represents a fuel pin, its associated coolant, and a fraction of the subassembly duct wall, as indicated in Figure 3.2–1. Usually, a channel is used to represent an average pin in a fuel subassembly or a group of similar subassemblies. A channel can also be used to represent a blanket assembly or a control-rod channel, and the hottest pin in a subassembly can be represented instead of the average pin. Different channels can be used to account for radial and azimuthal power variations within the core, as well as variations in coolant flow orificing and fuel burn-up. In the multiple pin option, more than one channel can be used to represent a subassembly.

3.2.1 Axial Mesh Structure

A channel usually represents the whole length of the subassembly, from coolant inlet to coolant outlet. A number of axial zones are used, as indicated in Figure 3.2–2. One zone represents the fuel-pin section, including the core, axial blankets, and gas plenum. Other zones represent reflector regions above and below the pin section. A maximum of 7 zones can be used in a channel. In general, radial dimensions and thermal properties are constant with a reflector zone. The pin section zone is treated separately in considerably more detail than the reflector zones. The gas plenum can be either above or below the core.

Figure 3.2–3 shows the axial mesh structure used for a channel. The coolant and structure nodes run the whole length of the channel. The coolant nodes are staggered with respect to the fuel, cladding, reflector, and structure nodes. Using coolant temperatures defined at the axial boundaries between cladding and structure nodes makes it easier to calculate accurate coolant temperatures. If non-uniform axial mesh sizes are used, a simple finite differencing of the coolant temperature equation gives accurate coolant temperatures with a staggered mesh, whereas if the coolant nodes were at the middle of the cladding nodes, then obtaining accurate coolant temperatures would require extra terms in the finite differencing of the coolant temperature equation.

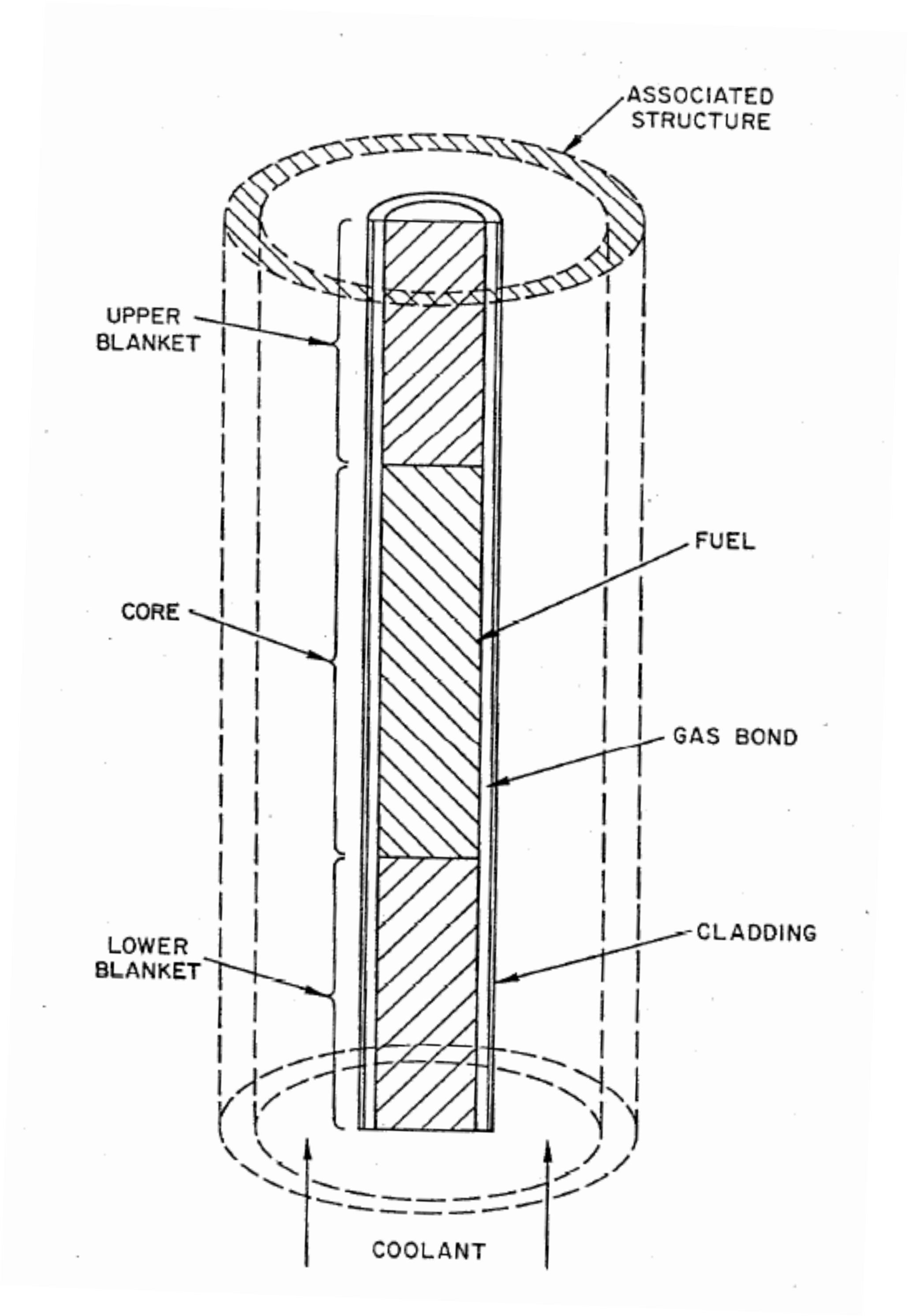


Figure 3.2–1: SAS4A/SASSYS-1 Channel Treatment

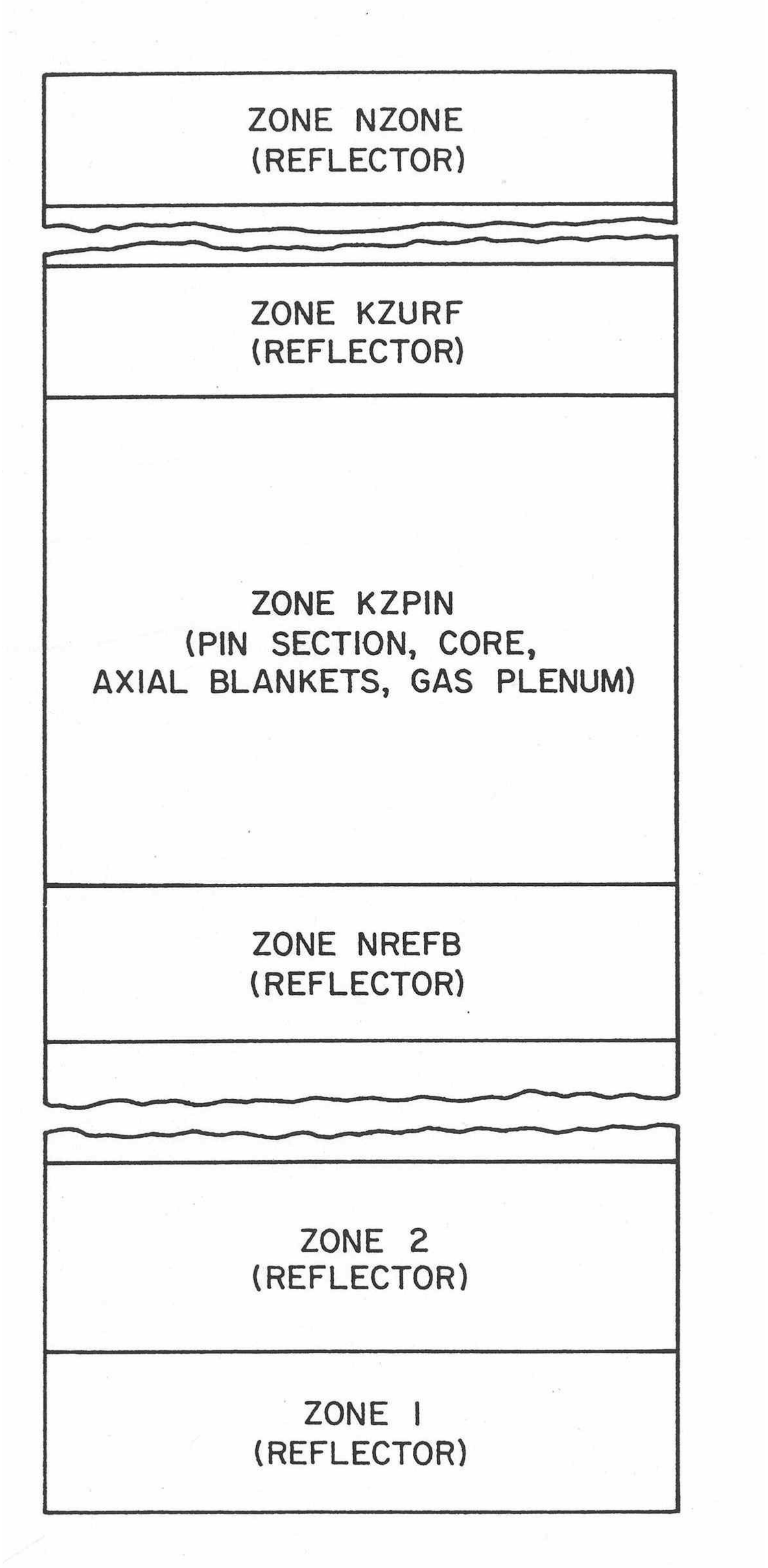


Figure 3.2–2: Axial Zones in a SAS4A/SASSYS-1 Channel

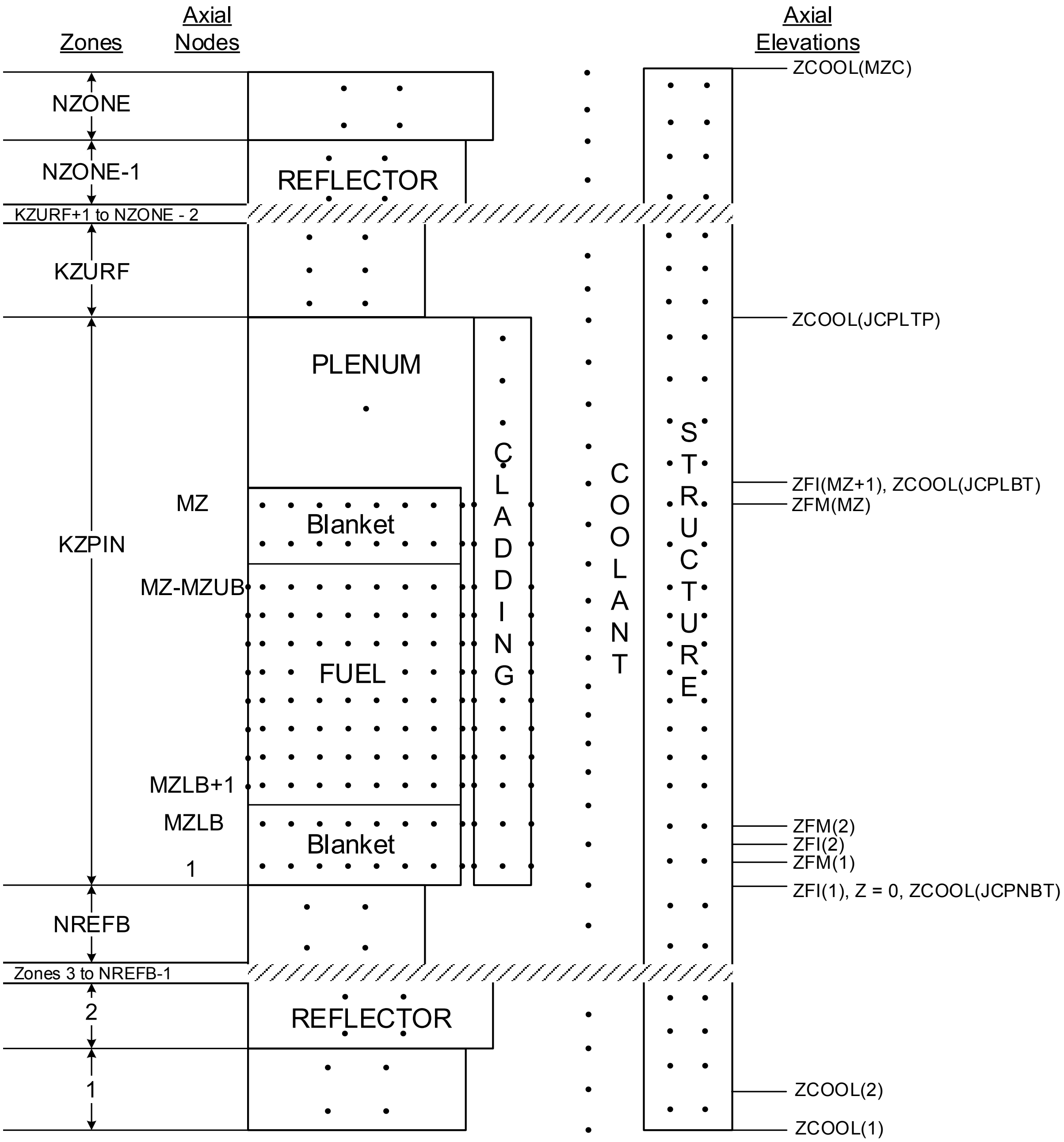


Figure 3.2–3: Schematic of SAS4A/SASSYS-1 Channel Discretization

3.2.2 Radial Mesh Structure

3.2.2.1 Core and Blanket Region

Figure 3.2–4 shows the radial mesh structure used for temperature calculations in the core and blanket regions. This figure represents one axial node. Between four and eleven radial nodes are used in the fuel, three in the cladding, one in the coolant, and two in the structure. In the fuel, the nodes can be set up on either an equal radial difference basis or an equal mass basis. In either case, the first and last nodes are half-size. For a given number of nodes, an equal radial difference mesh will usually give more accurate center-line temperatures, but equal mass nodes are sometimes used to get more nodes in the outer part of the fuel, where temperature gradients are steeper. Steady-state fuel restructuring can change the node sizes. Also, during the transient calculation, the radii will move with the fuel as it expands or contracts due to temperature changes. After the steady-state initialization, the mass of fuel associated with a radial node is constant, at least until fuel-pin disruption and coupling is made to PLUTO2, PINACLE, or LEVITATE.

The inner fuel node is at $r = 0$ if there is no central void. Otherwise, it is at the fuel inner surface. The outer fuel node is at the fuel outer surface.

The “structure” represents each pin’s share of the duct wall. A wrapper wire can be lumped in with either the cladding or the structure.

3.2.2.2 Gas Plenum Region

The radial mesh structure used in the gas plenum region is shown in Figure 3.2–5. The plenum gas is represented by a single axial and radial node. This gas is in contact with a number of axial cladding nodes. At each axial node, there is one radial node in the cladding, one in the coolant, and two in the structure.

3.2.2.3 Reflector Regions

The radial mesh structure in an axial node in a reflector region is shown in Figure 3.2–6. Two nodes are used in the reflector, one in the coolant, and two in the structure.

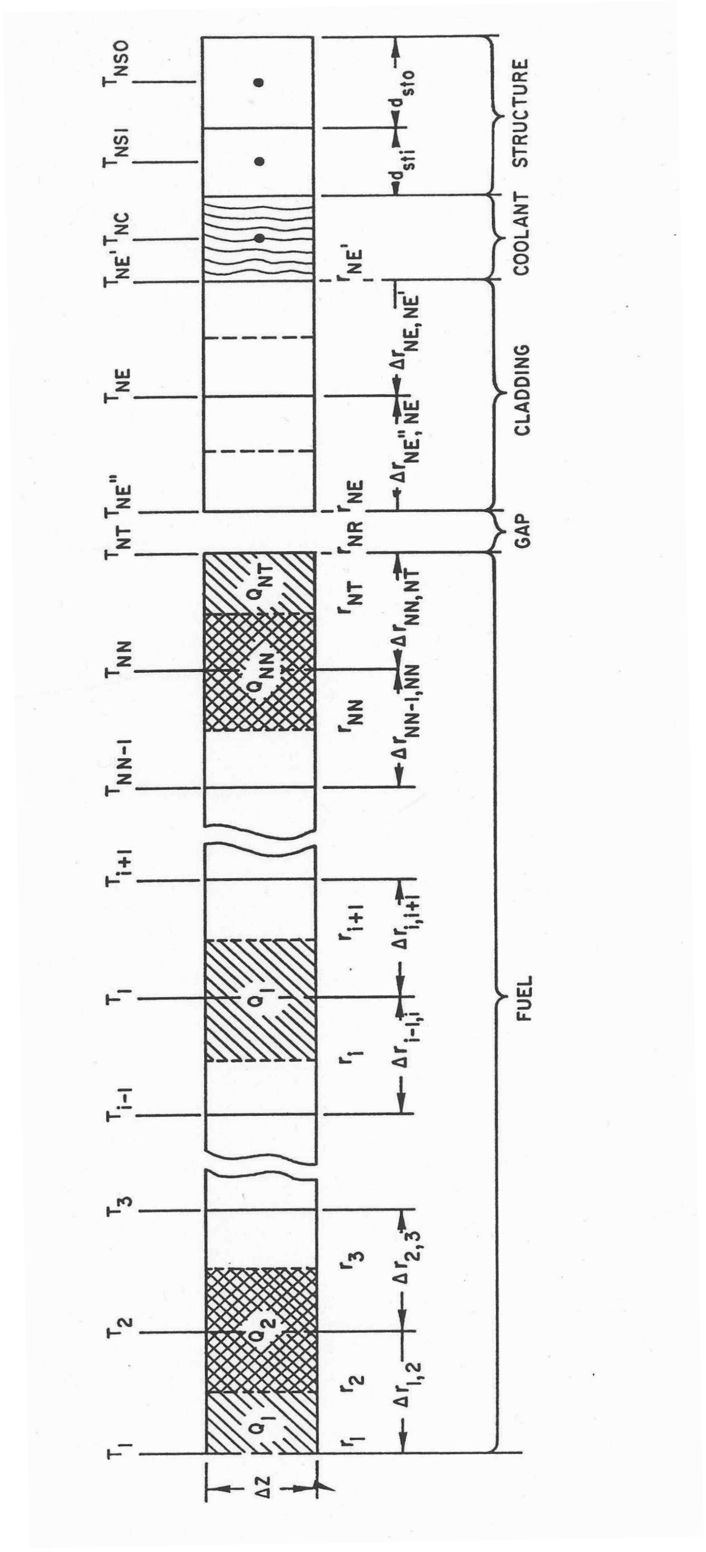


Figure 3.2–4: Radial Temperature Nodes, Core and Axial Blanket Regions

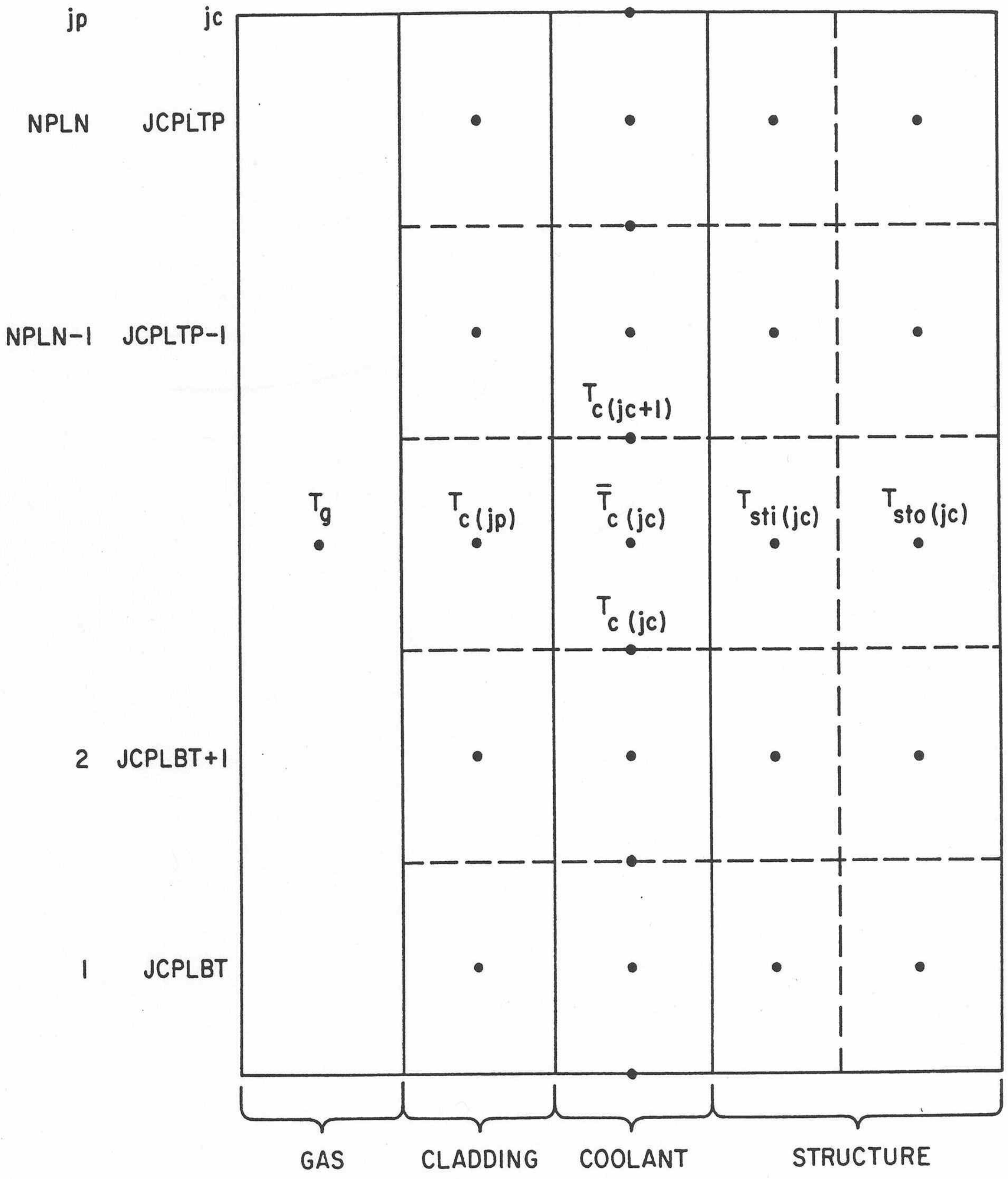


Figure 3.2–5: Radial Temperature Nodes, Gas Plenum Region

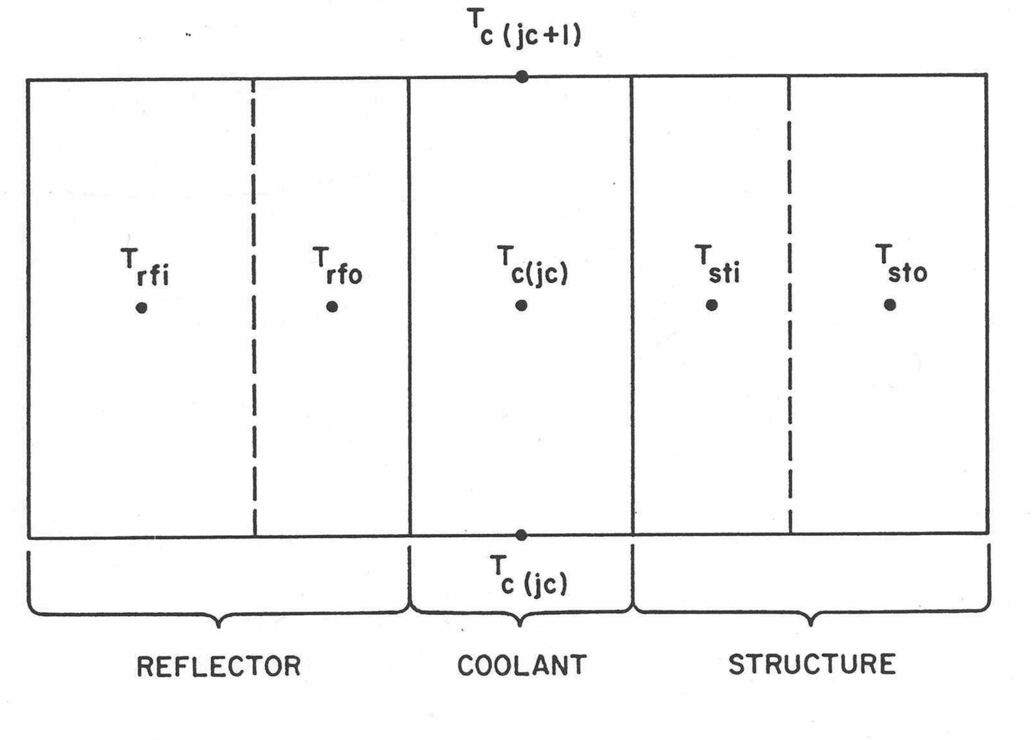


Figure 3.2–6: Radial Temperature Nodes, Reflector Region

3.3 Pre-boiling Transient Heat Transfer, Single Pin Model

The transient fuel-pin temperature calculations in SAS4A/SASSYS-1 are similar to the Crank-Nickolson scheme [3-1] used in SAS2A [3-2] and SAS3D [3-3], but there are a number of significant differences. One reason for these differences is to allow the use of larger heat-transfer time steps with less computing time per step. Another reason is to obtain greater accuracy and more precise energy balance.

In order to use very long heat-transfer time steps on one second or more in the pre-voiding phase of a very slow transient, the fuel, cladding, coolant, and structure temperatures at an axial node are computed simultaneously in non-voiding cases. Therefore, coolant temperatures are computed in the fuel-pin heat-transfer routine in the non-voiding situation.

The coupling between TSHTRN and the pre-voiding coolant dynamics routines is different from the coupling between TSHTRV and the voiding routines, although in both cases the coolant dynamics or voiding calculations are done before the fuel-pin heat-transfer calculations. In the non-voiding case, extrapolated coolant temperatures are used to obtain the temperature-dependent sodium properties used in the calculation of the coolant flow rates and heat-transfer coefficients. Then, TSHTRN uses these values to compute the coolant temperatures, and these new coolant temperatures are used in the next extrapolation. In the voiding routines, extrapolated cladding temperatures are used in the implicit calculation of the heat flux from cladding to coolant to obtain new coolant temperatures. The voiding routines then sum the integrated heat flux from cladding to coolant at each axial node, and this integrated heat flux at the cladding surface is used as a boundary condition in TSHTRV.

TSHTRN and TSHTRV are somewhat simpler than the corresponding TSHTR in SAS3A and SAS3D. In SAS3D, TSHTR contains extraneous material related to other modules, such as initiating cladding and fuel motion; this material is not included in the SAS4A/SASSYS-1 heat-transfer routines. Also, in SAS3D, the fuel mass for each radial node of each axial node is recomputed every time step from the temperature-dependent fuel density and the node radii. In SAS4A/SASSYS-1 the node mass is computed in the steady-state module and then stored for used in the transient. This node mass is held constant until fuel relocation starts.

Another difference between the SAS3D and SASSYS-1 heat-transfer routines is in the amount of vectorization of the algorithms and the coding. Some of the calculations in the SAS3D routines happen to vectorize, but a special effort was made to vectorize many more of the calculations in the SAS4A/SASSYS-1 heat-transfer routines. In general, vectorizing involves setting up arrays so that all elements in the array are processed in the same way and so that the results of a given calculation for one element in the array do not depend on the results for any other element in the array. Vectorization is partly just a coding matter, but it also involves choosing vectorizable algorithms or adapting algorithms for vectorization.

To some extent, the descriptions in the following sections reflect the emphasis on vectorization. Intermediate quantities used in the solution are usually defined as elements in arrays, and, where possible, the array elements are defined so that all of the elements in an array can be treated in the same manner in the calculations. In order to emphasize the array nature of the solution, when a new array is introduced, all of the elements in the array are defined in the same section, even though many of the elements are often not used until later sections.

As indicated above in Section 3.3.2, three different radial mesh structures are used in the heat-transfer calculations: one for the core and blanket region, one for the gas plenum region, and one for the reflector regions. Different heat-transfer calculations are done for each of these three regions.

3.3.1 Core and Axial Blankets

3.3.1.1 Basic Equations

The basic transient heat-transfer equation within the fuel and within the cladding is

$$\rho c \frac{\partial T}{\partial t} = \frac{1}{r}\frac{\partial}{\partial r}\left(kr\,\frac{\partial T}{\partial r}\right) + Q \qquad (3.3\text{-}1)$$

where

T = temperature

ρ = density

c = specific heat

r = radius

k = thermal conductivity

t = time

and

Q = heat source per unit volume

Melting of fuel and cladding is treated using a melting range, bounded by solidus and liquidus temperatures, T_{sol} and T_{liq}, respectively, and a heat of fusion, U_{melt}. Separate values of T_{sol}, T_{liq}, and U_{melt} are used for the fuel and cladding. In the melting range, Eq. 3.3-1 is modified and becomes

$$\rho c_m \frac{\partial T}{\partial t} = \frac{1}{r}\frac{\partial}{\partial r}\left(kr\frac{\partial T}{\partial r}\right) + Q \qquad (3.3\text{-}2)$$

where

$$c_m = \frac{U_{melt}}{T_{liq} - T_{sol}} \qquad (3.3\text{-}3)$$

In the actual solution of Eq. 3.3-2 for a time step, c is used instead of c_m in the calculation of the temperature change for the step. Then, if the temperatures are in the melting range, the computed temperature change is modified to account for the heat of fusion, as described in Section 3.3.5.

The heat flux, q_{fe}, from the fuel outer surface to the cladding inner surface contains both a bond gap conductance term, h_b, and a radiation term:

$$q_{fe} = h_b[T(\mathrm{NT}) - T(\mathrm{NT''})] + \varepsilon\sigma[T(\mathrm{NT})^4 - T(\mathrm{NT''})^4] \qquad (3.3\text{-}4)$$

where

ϵ = thermal emissivity of the fuel

σ = Stefan=Boltzman constant

$T(\mathrm{NT})$ = fuel outer surface temperature

and

$T(\mathrm{NE''})$ = cladding inner surface temperature

For the coolant, the basic heat-transfer equation is

$$\rho c \; A_c \frac{\partial T}{\partial r} + \frac{\partial}{\partial r}(wcT) = (Q_c + Q_{ec} + Q_{sc})A_c \tag{3.3-5}$$

where A_c = coolant flow area. The heat, Q_c, produced directly in the coolant is computed from the fraction, γ_c, of the total energy production that goes into neutron and gamma heating of the coolant:

$$Q_c = \frac{\gamma_c \bar{P}(j)}{A_c \Delta z(j)} \tag{3.3-6}$$

where

$\bar{P}(j) =$ total heat production rate in axial node j

and

$\Delta z(j) =$ axial node height

The heat flow from the cladding to the coolant, Q_{ec} is calculated as

$$Q_{ec} = h_c[T(\mathrm{NE}') - T(\mathrm{NC})]\frac{2\pi r(\mathrm{NE}')}{A_c} \tag{3.3-7}$$

and the heat flow from structure to coolant, Q_{sc}, is calculated as

$$Q_{sc} = h_c[T(\mathrm{NSI}) - T(\mathrm{NC})]\frac{S_{st}}{A_c} \tag{3.3-8}$$

where S_{st} is the perimeter of the structure. The coolant heat-transfer coefficient, h_c, is calculated using

$$\frac{h_c D_h}{k} = c_1 \left[\frac{D_h \; w \; c}{k \; A_c}\right]^{c_2} + c_3 \tag{3.3-9}$$

which is a form used in correlations for convective heat-transfer coefficients for low Prandtl number fluids, such as liquid metal [3-4]. The user supplied constants c_1, c_2, and c_3 depend on the particular correlation used.

The structure is treated with a one-dimensional heat conduction equation:

$$\rho c \frac{\partial T}{\partial t} = \frac{\partial}{\partial x}\left[k \frac{\partial T}{\partial x}\right] + Q \tag{3.3-10}$$

The treatment of the heat source, Q, in the structure is discussed in Section 3.3.1.2.8.

3.3.1.2 Finite Difference Equations

Finite differencing in both space and time is used for the transient heat-transfer calculations. The radial mesh structure is described in Section 3.2.2 above. In the equations below, the time t represents the beginning of the temp step, and Δt is the step size. The parameters θ_1 and θ_2 determine the degree of implications. For an explicit scheme, θ_1 = 1.0 and θ_2 = 0.0. For a fully implicit scheme, θ_1 = 0.0 and θ_2 = 1.0. For a semi-implicit scheme θ_1 – 0.5 and θ_2 = 0.5. The degree of implicitness is calculated by the code, based on the ratio of the time-step size, Δt, to a user-supplied time constant for fuel-pin heat transfer, τ_{ht}. As explained in Appendix 3.1, the expression used for θ_1 and θ_2 in TSHTRN are

$$\theta_2 = \frac{1.65 + x}{3.3 + x} \tag{3.3-11a}$$

where

$$x = \Delta t / \tau_{ht} \tag{3.3-11b}$$

and

$$\theta_1 = 1.0 - \theta_2 \tag{3.3-12}$$

Also, large relative changes in coolant flow rate, w, during a long heat transfer time step can lead to anomalous coolant temperature changes if the value of θ_1 is too large; therefore, θ_2 is calculated as

$$\theta_2 = \frac{|w_1|}{|w_1| + |w_2|} \tag{3.3-13}$$

if Eq. 3.3-13 gives a larger value than Eq. 3.3-11. In Eq. 3.3.13, w_1 and w_2 are the coolant flow rates at the beginning and end of the heat-transfer time step, respectively. In any case, Eq. 3.3-12 is used for θ_1.

The heat transfer time step size is limited to a relatively small value (0.02 s or less) after the onset of boiling, and so TSHTRV used $\theta_1 = \theta_2 = 0.5$.

The finite difference equations used for the fuel and for the two inner cladding nodes are the same in TSHTRN and TSHTRV. The differences between these routines start at the outer cladding node.

In general, the time derivative of a variable y is approximated as

$$\frac{\partial y}{\partial t} = \frac{y(t+\Delta t) - y(t)}{\Delta t} \tag{3.3-14}$$

and the spatial derivative is approximated as

$$\frac{\partial y}{\partial t} \simeq \theta_1 \left[\frac{y(t,z+\Delta z) - y(t,z)}{\Delta z} \right] + \theta_2 \left[\frac{y(t+\Delta t, z+\Delta z) - y(t+\Delta t, z)}{\Delta z} \right] \tag{3.3-15}$$

In the following sections, it will be useful to refer to Figure 3.2–4 for the radial node structure and the definitions of the radial node indexes.

3.3.1.2.1 Fuel Inner Surface, Node 1

There is an adiabatic boundary at the fuel inner surface; so the node 1, Eq. 3.3-1 becomes

$$\begin{aligned} & m_f(1)\, \bar{c_f}(1) \left[\frac{T_2(1) - T_1(1)}{\Delta t} \right] \\ & = \frac{2\pi r(2) \Delta z(j) \bar{k}_{1,2}}{\Delta r_{1,2}} \{ \theta_1 [T_1(2) - T_1(1)] + \theta_2 [T_2(2) - T_2(1)] \} + Q(1) \end{aligned} \tag{3.3-16}$$

where

$m_f(i)$ = fuel mass at node i

$\bar{c}(i)$ = fuel heat capacity at radial node i at $t = t_1 + \theta_2 \Delta y$

$T_2(i)$ = temperature at $t + \Delta t$

$T_1(i)$ = temperature at time t

$\Delta z(j)$ = axial mesh height

$$\Delta r(1) = 2\left[r(2) - r(1) \right] \tag{3.3-17}$$

$$\Delta r(i) = r(i+1) - r(i) \quad 2 \le i \le \mathrm{NN} \tag{3.3-18}$$

$$\Delta r(\mathrm{NT}) = 2[r(\mathrm{NR}) - r(\mathrm{NT})] \tag{3.3-19}$$

$$\Delta r(i) = \frac{[r(NE') - r(NE)]}{2} \quad i = NE", NE, NE' \tag{3.3-20}$$

(note: NE" = NR)

$$\Delta \mathrm{r}_{\mathrm{i,i+1}} = \frac{\Delta \mathrm{r}(\mathrm{i}) + \Delta r(i+1)}{2} \; 1 \le \mathrm{i} \le \mathrm{NE} \tag{3.3-21}$$

$$Q(i) = \frac{\overline{P}(j)\,(1-\gamma_e-\gamma_c-\gamma_s)P_r(i)m_f(i)}{\sum_{ii=1}^{NT} P_r(ii)m_f(ii)} \; 1 \le i \le NT \tag{3.3-22}$$

$$Q(\mathrm{NE}'') = \frac{\overline{P}(j)\gamma_e}{4} \tag{3.3-23}$$

$$Q(\mathrm{NE}) = \frac{\overline{P}(j)\gamma_e}{2} \tag{3.3-24}$$

$$Q(\mathrm{NE}') = \frac{\overline{P}(j)\gamma_e}{4} = Q(\mathrm{NE}'') \tag{3.3-25}$$

where

$\overline{P}(j) =$ total power (watts) in axial node j

$P_r(i)$ = radial power shape per unit mass

and

$\gamma_e, \gamma_c, \gamma_s$ = fraction of power in direct heating of clad, coolant, and structure, respectively.

The thermal conductivity, $\overline{k}_{1,2}$ used in Eq. 3.3-16 is a weighted average of the values for the two adjacent nodes. It is calculated as

$$\overline{k}_{i,i+1} = \frac{k(i)k(i+1)\left[\Delta r(i) + \Delta r(i+1)\right]}{k(i)\Delta r(i+1) + k(i+1)\Delta r(i)} \tag{3.3-26}$$

where

k(i) = thermal conductivity for radial node *i*, evaluated using the fuel temperature extrapolated to $t + \theta_2 \Delta t$

Equation 3.3-26 is carried over from SAS3D. The fuel restructuring algorithm used in SAS3D uses up to three different fuel types (columnar, equiaxed, and unrestructured) for the fuel at each axial node. Sharp boundaries between fuel types are used. The radial mesh is adjusted, if necessary, so that fuel-type boundaries fall on radial node boundaries. One consequence of the sharp fuel-type boundaries in SAS3D is that the fuel thermal conductivity can change significantly from one node to the next at a fuel-type boundary. The average thermal conductivity of Eq. 3.3-26 will give accurate steady-state fuel temperatures even if the thermal conductivity has large jumps at node boundaries. The fuel restructuring provided by DEFORM-IV in SAS4A is somewhat smoother than that used in SAS3D, and the node-to-node changes in thermal conductivity in SAS4A are usually smaller than the corresponding changes at fuel-type boundaries in SAS3D, so the need for special weighting of the fuel thermal conductivities in SAS4A Lis less than in SAS3D, but Eq. 3.3-26 still provides more accurate fuel temperatures than simpler weighting schemes.

Note that even though most of the variables in Eqs. 3.3-16 to 3.3-26 vary with axial node *j*, the subscript *j* has only been included for some of these variables.

3.3.1.2.2 Inner Fuel Nodes, Nodes 2 to NN

For fuel radial node I, Eq. 3.3-1 becomes

$$
\begin{aligned}
& m_f(i)\bar{c}_f(i)\,\frac{\left[T_2(i)-T_1(i)\right]}{\Delta t} \\
&\quad = \frac{2\pi r(i+1)\Delta z(j)\bar{k}_{i,i=1}}{\Delta r_{i,i+1}}\left\{\,\theta_1\,\left[T_1(i+1)-T_1(i)\right]+\theta_2\,\left[T_2(i+1)-T_2(i)\right]\,\right\} \\
&\quad - \frac{2\pi r(i)\Delta z(j)\bar{k}_{i-1,i}}{\Delta r_{i-1,i}}\left\{\,\theta_1\,\left[T_1(i)-T_1(i-1)\right]+\theta_2\,\left[T_2(i)-T_2(i-1)\right]\,\right\}+Q(i)
\end{aligned}
\tag{3.3-27}
$$

Note that the left-hand side of Eq. 3.3-27 represents the change in internal energy as node *i*, whereas the terms on the right-hand side represent heat conduction into node *i* from nodes *i*+1 and *i*-1, as well as the heat source in node *i*.

3.3.1.2.3 Fuel Outer Surface Node, Node NT

The heat flux q_{fc}, from the fuel outer surface to the cladding inner surface contains both a bond gap conductance and a radiation term.

$$
q_{fc} = h_b\left[\;T(\mathrm{NT})-T(\mathrm{NE}'')\;\right]+\varepsilon\sigma\left[\;T(\mathrm{NT})^4-T(\mathrm{NE}'')^4\;\right]
\tag{3.3-28}
$$

where

h_b = bond conductance

ε = thermal emissivity of the fuel

and

σ = Stefan-Boltzmann constant.

The T^4 terms are rewritten as

$$T(\mathrm{NT})^4 - T(\mathrm{NE}'')^4 = h_r \left[T(\mathrm{NT}) - T(\mathrm{NE}'')\right] \tag{3.3-29}$$

where

$$h_r = \frac{T_1(\mathrm{NT})^4 - T_1(\mathrm{NE"})^4}{T_1(\mathrm{NT}) - T_1(\mathrm{NE"})} = [T_1(\mathrm{NT}) + T_1(\mathrm{NE"})][T_1(\mathrm{NT})^2 + T_1(\mathrm{NE"})^2] \tag{3.3-30}$$

The approximation is then made that h_r is a constant for a time step, and the equation for node NT becomes

$$\begin{aligned} m_f(\mathrm{NT})\bar{c}_f(\mathrm{NT})\ \frac{T_2(\mathrm{NT}) - T_1(\mathrm{NT})}{\Delta t} \\ &= \frac{2\pi r(\mathrm{NT})\Delta z(j)\bar{k}_{\mathrm{NN,NT}}}{\Delta r_{\mathrm{NN,NT}}}\{\theta_1[T_1(\mathrm{NN}) - T_1(\mathrm{NT})] \\ &+ \theta_2[T_2(\mathrm{NN}) - T_2(\mathrm{NT})]\}\ 2\pi r(\mathrm{NR})\Delta z(j)[h_b \\ &+ \epsilon\sigma h_r]\ \{\theta_1[T_1(\mathrm{NE"})\ -\ T_1(\mathrm{NT})] + \theta_2[T_2(\mathrm{NE"}) \\ &- T_2(\mathrm{NT})]\} + Q(\mathrm{NT}) \end{aligned} \tag{3.3-31}$$

3.3.1.2.4 Cladding Inner Node, Node NE”

For the cladding inner node, Eq. 3.3-1 becomes

$$\begin{aligned}
\frac{m_e c_e}{4}\,\frac{T_2(\mathrm{NE}'')-T_1(\mathrm{NE}'')}{\Delta t}
&= \frac{2\pi\Delta z(j)\,\bar{r}_{NE}\,k_{NE'',NE}}{\Delta r_{NE'',NE}}\left\{\theta_1\left[T_1(\mathrm{NE})-T_1(\mathrm{NE}'')\right]+\theta_2\left[T_2(\mathrm{NE})-T_2(\mathrm{NE}'')\right]\right\} \\
&\quad -2\pi r(\mathrm{NR})\Delta z(j)\left(h_b+\varepsilon\sigma h_r\right)\left\{\theta_1\left[T_1(\mathrm{NE}'')-T_1(\mathrm{NT})\right]\right. \\
&\quad \left. +\theta_2\left[T_2(\mathrm{NE}'')-T_2(\mathrm{NT})\right]\right\}+Q(\mathrm{NE}'')
\end{aligned} \tag{3.3-32}$$

where

c_e = cladding heat capacity

m$_e$ = cladding mass $= 2\pi\rho_e[r(\mathrm{NE}')^2 - r(\mathrm{NE})^2]\Delta z(j)$ (3.3-33)

ρ_e = cladding density

and

$$\bar{r}_{NE} = r(\mathrm{NE}) + \tfrac{1}{4}\left[r(\mathrm{NE}')-r(\mathrm{NE})\right] \tag{3.3-34}$$

Note that the factors of 4 in Eqs. 3.3-32 and 3.3-34 come about because the inner cladding node represents one fourth of the thickness of the cladding.

3.3.1.2.5 Cladding Mid-point, Node NE

For the cladding mid-point node, Eq. 3.3-1 becomes

$$\begin{aligned}
\frac{m_e c_e}{2}\,\frac{T_2(\mathrm{NE})-T_1(\mathrm{NE})}{\Delta t}
&= \frac{2\pi\bar{r}_{\mathrm{NE}'}\Delta z(j)\bar{k}_{\mathrm{NE,NE}'}}{\Delta r_{\mathrm{NE,NE}'}}\{\theta_1[T_1(\mathrm{NE}')-T_1(\mathrm{NE})]+\theta_2[T_2(\mathrm{NE}') \\
&\quad - T_2(\mathrm{NE})]\} - \frac{2\pi\bar{r}_{\mathrm{NE}}\Delta z(j)\bar{k}_{\mathrm{NE'',NE}}}{\Delta r_{\mathrm{NE'',NE}}}\{\theta_1[T_1(\mathrm{NE})-T_1(\mathrm{NE''})] \\
&\quad + \theta_2[T_2(\mathrm{NE})-T_2(NE'')]\Big\} + Q(\mathrm{NE})
\end{aligned} \tag{3.3-35}$$

where

$$\bar{r}_{NE'} = r(\text{NE}) + 3\!/\!4\left[r(\text{NE}') - r(\text{NE})\right] \qquad 3.3\text{-}36)$$

3.3.1.2.6 Cladding Outer Node, Node NE′

The out cladding node transfers heat to both the cladding mid-point node and the coolant node, so the equation for the outer cladding node temperature is

$$\begin{aligned}\frac{m_e c_e}{4}\frac{T_2(\text{NE}') - T_1(\text{NE}')}{\Delta t}& \\ &= -\frac{2\pi\bar{r}_{\text{NE}'}\Delta z(j)\bar{k}_{\text{NE,NE}'}}{\bar{r}_{\text{NE,NE}'}}\{\theta_1[T_1(\text{NE}') - T_1(\text{NE})] \\ &+ \theta_2[T_2(\text{NE}') - T_2(\text{NE})]\} \\ &+ 2\pi r(\text{NE}')\Delta z(j)\,\{\theta_1 h_{c1}(j)[T_1(\text{NC}) - T_1(\text{NE}')] \\ &+ \theta_2 h_{c2}(j)[T_2(\text{NC}) - T_2(\text{NE}')]\Big\} + Q(NE')\end{aligned} \qquad (3.3\text{-}37)$$

Note that $T(\text{NC}) = \bar{T}_c(jc)$. Also, *hc1* and *hc2* are the coolant heat-transfer coefficients at t and $t\ +\Delta t$ as calculated from Eq. 3.3-9,

3.3.1.2.7 Coolant, Node NC

Coolant temperatures are calculated for the whole length of the subassembly, whereas fuel temperatures are computed in the core and blankets only. Therefore, the axial coolant node mesh extends beyond the fuel mesh; and the coolant axial node index, *jc*, is related to the fuel axial node index, *j*, by

$$jc = j + j_{cblbt} - 1 \qquad (3.3\text{-}38)$$

where j_{cblbt} is the coolant node at the bottom of the lower blanket. In the *T1(i,j)* and *T2(i,j)* arrays, the coolant node corresponds to

$$T_2(\text{NC}, j) = \bar{T}_c(jc) \qquad (3.3\text{-}39)$$

In the non-voiding case, the coolant flow is usually upward. In such a situation, the transient calculation for a time step starts at the subassembly inlet and works upward through the lower reflector zones, through the pin section, and finally through the upper reflector zones. The code can also handle downward coolant flow during the transient, although the initial steady-state coolant flows must all be upward. In the downward situation, the calculation starts at the top of the subassembly and works down to the inlet.

In the core and blanket regions, Eq. 3.3-5 becomes

$$\begin{aligned}
&\bar{\rho}_c(jc)\bar{c}_c(jc)\frac{T_2(\mathrm{NC})-T_1(\mathrm{NC})}{\Delta t}+\frac{2\bar{c}_c(jc)}{\Delta z(j)A_c(jc)}\left\{\theta_1\left|w_1\right|\left[T_1(\mathrm{NC})-T_1'\right]\right.\\
&\quad\left.+\theta_2\left|w_2\right|\left[T_2(\mathrm{NC})-T_2'\right]\right\}=\frac{Q(\mathrm{NC})}{A_c(jc)\Delta z(j)}+\frac{2\pi r(\mathrm{NE'})}{A_c(jc)}\\
&\quad\left\{\theta_1 h_{c1}(j)\left[T_1(\mathrm{NE'})-T_1(\mathrm{NC})\right]+\theta_2 h_{c2}(j)\ \left[T_2(\mathrm{NE'})-T_2(\mathrm{NE})\right]\right\}\\
&\quad+\frac{S_{pr}}{A_c(jc)}\left\{\theta_1 H_{sic1}(jc)\ \left[T_1(\mathrm{NSI})-T_1(\mathrm{NC})\right]\right.\\
&\quad\left.+\theta_2 H_{sic2}(jc)\left[T_2(\mathrm{NSI})-T_2(\mathrm{NC})\right]\right\}
\end{aligned}\tag{3.3-40}$$

where

S_{pr} = structure perimeter

w_1 and w_2 = the coolant mass flow rates (kg/s) at t and $t + \Delta t$

NSI = the inner structure node

$$T_1'=\begin{cases}T_{c1}(jc) & \text{if } w_1\geq 0\\ T_c(jc+1) & \text{if } \mathrm{w}_1<0\end{cases}\tag{3.3-41a-b}$$

$$T_2'=\begin{cases}T_{c2}(jc) & \text{if } w_2\geq 0\\ T_{c2}(jc+1) & \text{if } \mathrm{w}_2<0\end{cases}\tag{3.3-42a-b}$$

and

$\mathrm{H_{sic}}$ = the heat-transfer coefficient from the structure inner node to the coolant

$$H_{sic}=\frac{2h_c\ k_{si}}{2k_{si}+h_c d_{sti}}\tag{3.3-43}$$

Note that Eq. 3.3-43 is obtained by adding thermal resistance in series:

$$\frac{1}{H_{sic}} = \frac{1}{h_c} + \frac{d_{sti}}{2k_{si}} \tag{3.3-44}$$

3.3.1.2.8 Structure Inner Node, Node NSI

In the core and blanket regions,

$$\begin{aligned}(\rho c)_{sti}\, d_{sti} \frac{T_2(\mathrm{NSI}) - T_1(\mathrm{NSI})}{\Delta t} &= \theta_1 H_{sic1}(jc)\left[T_1(\mathrm{NC}) - T_1(\mathrm{NSI})\right] \\ &+ \theta_2 H_{sic2}(jc)\left[T_2(\mathrm{NC}) - T_2(\mathrm{NSI})\right] \\ &+ H_{stio}(jc)\{\, \theta_1 \left[T_1(\mathrm{NSO}) - T_1(\mathrm{NSI})\right] \\ &+ \theta_2 \left[T_2(\mathrm{NSO}) - T_2(\mathrm{NSI})\right] \} + Q_{st} \frac{d_{sti}}{d_{sti} + d_{sto}}\end{aligned} \tag{3.3-45}$$

where

NSO = outer structure node

d_{sti} = thickness of inner structure node

d_{sto} = thickness of outer structure node

$$H_{stio} = \frac{2k_{si}k_{so}}{d_{sti}k_{so} + d_{sto}k_{si}} \tag{3.3-46}$$

k_{si} = thermal conductivity of the inner structure node

k_{so} = thermal conductivity of the outer structure node

and

Q_{st} = direct heating source in the structure.

The left-hand side of Eq. 3.3-45 represents the change in internal energy in the node. The terms on the right-hand side represent heat flow from the coolant and the outer structure node, as well as direct heating of the structure by neutrons and gamma rays. It is assumed that the direct heating source is divided between the inner and outer nodes in proportion to their thicknesses.

3.3.1.2.9 Structure Outer Node, Node NSO

In the core and blankets,

$$
\begin{aligned}
(\rho c)_{sto} d_{sto} \frac{T_2(\mathrm{NSO}) - T_1(\mathrm{NSO})}{\Delta t} \\
&= H_{stio}(jc)\{\theta_1[T_1(\mathrm{NSI}) - T_1(\mathrm{NSO})] \\
&+ \theta_2[T_2(\mathrm{NSI}) - T_2(\mathrm{NSO})]\} + Q_{st}(jc)\frac{d_{sto}}{d_{sti} + d_{sto}} \\
&+ Q_{chch}(jc)
\end{aligned}
\qquad (3.3\text{-}47)
$$

where

$Q_{chch} =$ subassembly-to-subassembly heat transfer heat flux

The values used for Q_{chch} are discussed in Section 3.11.

3.3.1.2.10 Solution of Finite Difference Equations

Equations 3.3-16, 3.3-27, 3.3-31, 3.3-32, 3.3-35, 3.3-37, 3.3-40, 3.3-45, and 3.3-47 can be written in matrix form, yielding a tri-diagonal matrix of the form

$$
\begin{bmatrix}
\alpha_1 + \beta_1 & -\beta_1 & 0 & 0 & \cdots \\
-\beta_1 & \alpha_2 + \beta_1 + \beta_2 & -\beta_2 & 0 & \cdots \\
0 & -\beta_2 & \alpha_3 + \beta_2 + \beta_3 & -\beta_2 & \cdots \\
\cdot & & & & \\
\cdot & & & & \\
\cdot & & & & \\
\cdot & & & & \\
\cdot & & & & \\
\cdot & & & & \\
0 & \ldots 0 & -\beta_{\mathrm{NC}} & \alpha_{\mathrm{NSI}} + \beta_{\mathrm{NC}} + \beta_{\mathrm{NSI}} & -\beta_{\mathrm{NSI}} \\
0 & \ldots. & 0 & -\beta_{\mathrm{NSI}} & \alpha_{\mathrm{NSO}} + \beta_{\mathrm{NSI}} + \beta_{\mathrm{NSO}}
\end{bmatrix}
\begin{bmatrix}
T_2(1) \\ T_2(2) \\ T_2(3) \\ \\ \\ \\ \\ \\ \\ T_2(\mathrm{NSI}) \\ T_2(\mathrm{NSO})
\end{bmatrix}
=
\begin{bmatrix}
D_1 \\ D_2 \\ D_3 \\ \\ \\ \\ \\ \\ \\ D_{\mathrm{NSI}} \\ D_{\mathrm{NSO}}
\end{bmatrix}
\qquad (3.3\text{-}48)
$$

where

$$
\alpha_1 = \frac{\mathrm{M_f}(i)\bar{c}_f(i)}{2\pi\Delta z(j)} \quad \text{for } i = 1,\ldots\ NT \qquad (3.3\text{-}49a)
$$

$$
\alpha_{NE''} = \frac{\mathrm{M_e} c_e}{8\pi\Delta z(j)} \qquad (3.3\text{-}49b)
$$

$$\alpha_{NE} = \frac{\mathrm{M_e} c_e}{4\pi \Delta z(j)} = 2\alpha_{NE''} \tag{3.3-49c}$$

$$\alpha_{NE'} = \alpha_{NE''} \tag{3.3-49d}$$

$$\alpha_{NC} = \frac{\bar{\rho}_c(\mathrm{jc})\bar{\mathrm{c}}_\mathrm{c}(\mathrm{jc})\mathrm{A_c}(\mathrm{jc})}{2\pi} + \frac{\bar{c}_c(jc)}{\pi \Delta z(jc)}\theta_2 \, |w_2| \Delta t \tag{3.3-49e}$$

$$\alpha_{NSI} = \frac{(\rho \mathrm{c})_\mathrm{sti} \mathrm{d_{sti}} \mathrm{S_{pr}}}{2\pi} \tag{3.3-49f}$$

$$\alpha_{NSO} = \frac{(\rho \mathrm{c})_\mathrm{sto} \mathrm{d_{sto}} \mathrm{S_{pr}}}{2\pi} \tag{3.3-49g}$$

$$\beta_i = \frac{r(i+1)\bar{k}_{i,i+1}}{\Delta r_{i,i+1}} \theta_2 \, \Delta t \quad \text{for } i = 1,\dots.NN \tag{3.3-50a}$$

$$\beta_{NT} = r(NR)\,\theta_2 \left[h_b + \varepsilon\sigma h_r\right]\Delta t \tag{3.3-50b}$$

$$\beta_{NE''} = \frac{\bar{r}_{\mathrm{NE}} \, \bar{k}_{\mathrm{NE'',NE}}}{\Delta r_{\mathrm{NE'',NE}}} \theta_2 \Delta t \tag{3.3-50c}$$

$$\beta_{NE} = \frac{\bar{r}_{\mathrm{NE'}} \, \bar{k}_{\mathrm{NE,NE'}}}{\Delta r_{\mathrm{NE,NE'}}} \theta_2 \Delta t \tag{3.3-50d}$$

$$\beta_{\mathrm{NE'}} = r_{\mathrm{NE'}} \, h_{c2}(j)\theta_2 \Delta t \tag{3.3-50e}$$

$$\beta_{\mathrm{NC}} = \frac{S_{pr}}{2\,\pi} H_{\mathrm{sic2}}(jc)\;\theta_2 \Delta t \tag{3.3-50f}$$

$$\beta_{\mathrm{NSI}} = \frac{S_{pr}}{2\,\pi} H_{\mathrm{stio}}(jc)\;\theta_2 \Delta t \tag{3.3-50g}$$

$$\beta_{\mathrm{NSO}} = 0 \tag{3.3-50h}$$

$$D_1 = T_1(1)\left[\alpha_1 - \frac{\theta_1}{\theta_2}\beta_1\right] + \frac{\theta_1}{\theta_2}\beta_1 T_1(2) + \psi_1 \tag{3.3-51a}$$

$$\begin{aligned} D_i = {} & \frac{\theta_1}{\theta_2}\beta_{i-1}T_1(i-1) + T_1(i)\left[\alpha_1 - \frac{\theta_1}{\theta_2}(\beta_{i-1}+\beta_i)\right] \\ & + \frac{\theta_1}{\theta_2}\beta_i T_1(i+1) + \psi_1 \quad \text{for } i = 2,\dots,NSI \end{aligned} \tag{3.3-51b}$$

$$D_{\mathrm{NSO}} = \frac{\theta_1}{\theta_2}\beta_{NSI}T_1(\mathrm{NSI}) + T_1(\mathrm{NSO})\left[\alpha_{\mathrm{NSO}} - \frac{\theta_1}{\theta_2}(\beta_{\mathrm{NSI}} + \beta_{\mathrm{NSO}})\right] + \psi_{\mathrm{NSO}} \tag{3.3-51c}$$

$$\psi_i = \frac{\mathrm{Q(i)\Delta t}}{2\pi\Delta z(j)} \text{ for } i = 1,\dots, \mathrm{NE'} \tag{3.3-52a}$$

$$\psi_{NC} = \frac{\mathrm{Q(NC)}\Delta t}{2\pi\Delta z(j)} + \frac{\bar{c}_c(jc)\Delta t}{\pi\Delta z(j)}\left[\theta_1|w_1|T_1' + \theta_2|w_2|T_2'\right] \tag{3.3-52b}$$

$$\psi_{NSI} = \frac{Q(\mathrm{NSI})\Delta t}{2\pi\Delta z(j)} \tag{3.3-52c}$$

and

$$\psi_{NSO} = \frac{\mathrm{Q(NSO)}\Delta t}{2\pi\Delta z(j)} + \frac{S_{pr}}{2\pi}\mathrm{Q_{chch}\Delta T} \tag{3.3-52d}$$

In these equations the α array is related to heat capacity, the β array is related to heat transfer between adjacent nodes, and the ψ array is related to the heat source.

The matrix equations 3.3-49 are solved by Gaussian elimination. First arrays A_i and S_i are defined:

$$A_1 = \alpha_1 + \beta_1 \tag{3.3-53a}$$

$$A_i = \alpha_i + \beta_i + \beta_{i=1} - \frac{\beta_{i-1}^2}{A_{i-1}} \quad \text{for } i = 2,\ldots,\text{ NSO} \tag{3.3-53b}$$

$$S_1 = D_1 \tag{3.3-54a}$$

and

$$S_i = D_i + \frac{\beta_{i-1} S_{i-1}}{A_{i-1}} \quad \text{for } i = 2,\ldots,\text{ NSO} \tag{3.3-54b}$$

Then,

$$T_2(\text{NSO}) = \frac{S_{\text{NSO}}}{A_{\text{NSO}}} \tag{3.3-55a}$$

and

$$T_2(i) = S_i + \frac{\beta_i}{A_i}\ T_2(i+1) \quad \text{for } i = \text{NSI, NC},\ldots,\ 1 \tag{3.3-55b}$$

3.3.2 Reflector Zones

In reflector zones, a two-node slab geometry treatment is used at each axial node for heat transfer to the “reflector”. The reflector represents any material in the subassembly outside the pin section.

Typically, this material includes shield orifice blocks near the subassembly inlet, and instrumentation in the upper part of the subassembly. Usually, this material does not come to the form of either pure slabs or pure cylinders, and so any simple geometrical treatment of it will be only an approximation. The best that one is likely to do with a simple heat-transfer calculation is to use a slab calculation with parameters chosen to match total heat capacity, total heat-transfer surface area, and the approximate effective thickness of the material.

3.3.2.1 Basic Equations

The basic equations used for coolant and structure temperatures in reflector zones are the same as Eqs. 3.3-5 and 3.3-10 used in the core and blanket regions, except that no heat generation is considered in the reflector zones. Therefore, the Q_c, term of Eq. 3.3-5 and the Q term of Eq. 3.3-10 are eliminated in the reflector zones. The reflector is treated with a one-dimensional heat conduction equation that is the same as the one used for the structure, except that the thermal properties ρ, c, and k used for the reflector can be different from those used for the structure.

3.3.2.2 Finite Difference Equations

Figure 3.2–6 shows the radial mesh structures used for an axial node in a reflector region.

3.3.2.2.1 Reflector Inner Node

Equation 3.3-10 becomes

$$\begin{aligned}(\rho c)_r d_{ri} \frac{T_{ri2}(jc)-T_{ri1}(jc)}{\Delta t} = H_{rio}(jc)\{&\theta_1 [T_{ro1}(jc)-T_{ri1}(jc)\ ] \\ &+\theta_2[T_{ro2}(jc)-T_{ri2}(jc)]\}\end{aligned} \tag{3.3-56}$$

where

$(\rho c)_r$ = density times specific heat of the reflector

d_{ri} = thickness of inner node

T_{ri1}, T_{ri2} = reflector inner node temperature at the beginning and end of the time step

T_{ro1}, T_{ro2} = reflector outer node temperature at the beginning and end of the time step

d_{ro} = thickness of outer reflector node

and

$$H_{rio} = \frac{2k_r}{d_{ri}+d_{ro}} \tag{3.3-57}$$

Equation 3.3-57 is obtained by adding thermal resistances in series:

$$\frac{1}{H_{rio}} = \frac{d_{ri}}{2k_r} + \frac{d_{ro}}{2k_r} \tag{3.3-58}$$

Equation 3.3-56 is similar to Eq. 3.3-47 for the outer structure node, except there is no direct heating source in the reflectors.

3.3.2.2.2 Reflector Outer Node

The equation for the reflector outer node temperature is similar to Eq. 3.3-45 for the structure inner node:

$$
\begin{aligned}
&(\rho c)_r d_{ro} \frac{T_{ro2}(jc)-T_{ro1}(jc)}{\Delta t} \\
&\quad = \theta_1 H_{erc1}(jc)\left[\overline{T}_{c1}(jc)-T_{r01}(jc)\ \right]+\theta_2 H_{erc2}(jc)\left[\overline{T}_{c2}(jc)-T_{r02}(jc)\ \right] \\
&\quad\quad + H_{rio}(jc)\left\{\theta_1\left[T_{ri1}(jc)-T_{ro1}(jc)\right]+\theta_2\left[T_{ri2}(jc)-T_{ro2}(jc)\right]\right\}
\end{aligned}
\tag{3.3-59}
$$

where

$$
H_{erc} = \frac{h_c k_r}{k_r + \dfrac{h_c d_{ro}}{2}}
\tag{3.3-60}
$$

Note that

$$
\overline{T}_c(jc) = \left[T_c(jc)+T_c(jc+1)\right]/2
\tag{3.3-61}
$$

3.3.2.2.3 Coolant Node

In the reflector zones and in the gas plenum, Eq. 3.3-5 becomes

$$
\begin{aligned}
&\overline{\rho}_c(jc)\overline{c}_c(jc)\frac{\overline{T}_{c2}(jc)-T_{c1}(jc)}{\Delta t}+\frac{2\overline{c}_c(jc)}{\Delta z(jc)A_c(jc)}\left\{\theta_1|w_1|\left[\overline{T}_c(jc)-T_1'\right]\right. \\
&\quad \left. +\theta_2|w_2|\left[\overline{T}_{c2}(jc)-T_2'\ \right]\right\} \\
&\quad =\frac{S_{er}(KZ)}{A_c(jc)}\left\{\theta_1 H_{erc1}\left[T_{er1}(jc)-T_{c1}(jc)\right]\right. \\
&\quad \left. +\theta_2 H_{erc2}\left[T_{er2}(jc)-\overline{T}_{c2}(jc)\ \right]\right\}+\frac{S_{pr}}{A_c(jc)}\left\{\theta_1 H_{sic1}\left[T_{ri1}(jc)\right.\right. \\
&\quad \left.\left. -T_{c1}(jc)\right]+\theta_2\ H_{sci2}(jc)\left[T_{sti2}(jc)-\overline{T}_{c2}(jc)\right]\right\}
\end{aligned}
\tag{3.3-62}
$$

where, in reflector zones,

T_{er1} = T_{rol}

T_{er2} = T_{ro2}

S_{er} = Sr = reflector perimeter

S_{pr} = structure perimeter

and

T_{sti} = structure inner node temperature

3.3.2.2.4 Structure Nodes

The finite difference approximation to Eq. 3.3-10 for the structure nodes is the same as in the core and blanket regions. Equations 3.3-45 and 3.3-47 are used, except that Q_{st} is zero outside the core and blankets.

3.3.2.3 Solution of Finite Difference Equations

$$\begin{bmatrix} \alpha_1+\beta_1 & -\beta_1 & 0 & 0 & 0 \\ -\beta_1 & \alpha_2+\beta_1+\beta_2 & -\beta_2 & 0 & 0 \\ 0 & -\beta_2 & \alpha_3+\beta_2+\beta_3 & -\beta_3 & 0 \\ 0 & 0 & -\beta_3 & \alpha_4+\beta_3+\beta_4 & -\beta_4 \\ 0 & 0 & 0 & -\beta_4 & \alpha_5+\beta_4+\beta_5 \end{bmatrix} \begin{bmatrix} T_{ri2} \\ T_{ro2} \\ \overline{T}_{c2} \\ T_{Sti2} \\ T_{St02} \end{bmatrix} = \begin{bmatrix} D_1 \\ D_2 \\ D_3 \\ D_4 \\ D_5 \end{bmatrix} \tag{3.3-63}$$

where

$$\alpha_1 = (\rho c)_r \, d_{ri} S_{er} (\mathrm{KZ}) \tag{3.3-64a}$$

$$\alpha_2 = (\rho c)_r \, d_{ro} S_{er} (\mathrm{KZ}) \tag{3.3-64b}$$

$$\alpha_3 = \overline{\rho}_c \, \overline{c}_e \, A_c + 2 \, \overline{c}_c \, \theta_2 |w_2| \Delta t \tag{3.3-64c}$$

$$\alpha_4 = (\rho c)_{sti} \, d_{sti} S_{pr} \tag{3.3-64d}$$

$$\alpha_5 = (\rho c)_{sto} \, d_{sto} S_{pr} \tag{3.3-64e}$$

$$\beta_1 = \theta_2 H_{rio} (jc) S_{er} \Delta t \tag{3.3-65a}$$

$$\beta_2 = \theta_2 H_{erc2} (jc) S_{er} \Delta t \tag{3.3-65b}$$

$$\beta_3 = \theta_2 H_{sic2} (jc) S_{pr} \Delta t \tag{3.3-65c}$$

$$\beta_4 = \theta_2 H_{stio} (jc) S_{pr} \Delta t \tag{3.3-65d}$$

$$\beta_5 = \theta \tag{3.3-65e}$$

$$D_1 = T_{ri1}\left[\alpha_1 - \frac{\theta_1}{\theta_2}\beta_1\right] + \frac{\theta_1}{\theta_2}\beta_1\, T_{ro1} \tag{3.3-66a}$$

$$\begin{aligned} D_i = {} & \frac{\theta_1}{\theta_2}\beta_{i-1}\, T_1''(i-1) + T_1''(i)\left[\alpha_1 - \frac{\theta_1}{\theta_2}(\beta_{i-1} + \beta_1)\right] \\ & + \frac{\theta_1}{\theta_2}\beta_i\, T_1''(i+1) \quad \text{for } i = 2, 4 \end{aligned} \tag{3.3-66b}$$

$$\begin{aligned} D_3 = {} & \frac{\theta_1}{\theta_2}\beta_2\, T_1''(2) + T_1''(3)\left[\alpha_3 - \frac{\theta_1}{\theta_2}(\beta_2 + \beta_3)\right] + \frac{\theta_1}{\theta_2}\beta_3\, T_1''(4) \\ & + 2\frac{\bar{c}_c \Delta t}{\Delta z}\left[\theta_1 |w_1|\, T_1' + \theta_2 |w_2| T_2'\right] \end{aligned} \tag{3.3-66c}$$

$$D_5 = \frac{\theta_1}{\theta_2}\beta_4\, T_1''(4) + T_1''(5)\left[\alpha_5 - \frac{\theta_1}{\theta_2}(\beta_4 + \beta_5)\right] + S_{pr}\, Q_{chch}\Delta t \tag{3.3-66d}$$

$$T_1''(1) = T_{ri1}(jc) \tag{3.3-67a}$$

$$T_1''(2) = T_{ro1}(jc) \tag{3.3-67b}$$

$$T_1''(3) = \overline{T}_{c1}(jc) \tag{3.3-67c}$$

$$T_1''(4) = T_{sti1}(jc) \tag{3.3-67d}$$

and

$$T_1''(5) = T_{sto1}(jc) \tag{3.3-67e}$$

This tri-diagonal matrix is solved in the same manner as in Section 3.3.1.2 above.

3.3.3 Gas Plenum Region

In the gas plenum region, a single gas node is in contact with all axial cladding nodes. A single radial node is used in the cladding, as indicated in Figure 3.2–5.

3.3.3.1 Basic Equations

The basic equations used in the cladding, coolant, and structure in the gas plenum region are the same as those used in the core and blankets, except that there is no heat source term outside of the core and blankets. The gas is assumed to transfer heat only to the cladding, and the basic equation used for the gas temperature is

$$\left(z_{pt}-z_{pb}\right)\left(\rho c\right)_g A_g \frac{dT_g}{dt} = 2\pi r_{brp} H_{eg} \int_{z_{pb}}^{z_{pt}} \left(T_e - T_g\right) dz \tag{3.3-68}$$

where

z_{pb} = elevation at the bottom of the gas plenum

z_{pt} = elevation at the top of the gas plenum

$(\rho c)_g$ = density times specific heat for the plenum gas

A_g = cross sectional area of the gas plenum

$$A_g = \pi\, r^2_{brp} \tag{3.3-69}$$

r_{brp} = cladding inner radius in the gas plenum region

T_g = gas temperature

H_{eg} = heat-transfer coefficient from the plenum gas to the cladding node

T_e = cladding temperature

3.3.3.2 Finite Difference Equations

3.3.3.2.1 Gas Plenum

Equation 3.3-69 becomes

$$\left(\rho c\right)_g A_g \frac{\left(T_{g2}-T_{g1}\right)}{\Delta t} = 2\pi r_{brp} H_{eg} \frac{\sum_{jp} \left\{\theta_1\left[T_{e1}(jp)-T_{g1}\right]+\theta_2\left[T_{e2}(jp)-T_{g2}\right]\right\}\Delta z(jp)}{\sum_{jp}\Delta z(jp)} \tag{3.3-70}$$

where

jp = plenum node

T_{g1} = plenum gas temperature at the beginning of the time step

T_{g2} = plenum gas temperature at the end of the time step

$$H_{eg} = \frac{1}{R_g + \frac{r_{erp} - r_{brp}}{2k_{ep}}} = \frac{2k_{ep}}{r_{erp} - r_{brp} + 2k_{ep} R_g} \tag{3.3-71}$$

k_{ep} = cladding thermal conductivity in the gas plenum

and

R_g = thermal resistance of the gas.

3.3.3.2.2 Cladding Node

The equation for the cladding node is

$$\begin{aligned} \rho_e c_e A_{ep} & \frac{T_{e2}(jp) - T_{e1}(jp)}{\Delta t} \\ & = 2\pi r_{erp} \left\{ \theta_1 H_{erc1} \left[\overline{T}_{c1}(jc) - T_{e1}(jp) \right] \right. \\ & \quad + \theta_2 H_{erc2} \left[\overline{T}_{c2}(jc) - T_{e2}(jp) \right] \left. \right\} + 2\pi r_{brp} H_{eg} \left\{ \theta_1 \left[T_{g1} - T_{e1}(jp) \right] \right. \\ & \quad \left. + \theta_2 \left[T_{g2} - T_{e2}(jp) \right] \right\} \end{aligned} \tag{3.3-72}$$

where

ρ_e = cladding density

c_e = cladding specific heat

$$A_{ep} = \pi \left[r_{erp}^2 - r_{brp}^2 \right] \tag{3.3-73}$$

and

r_{erp} = cladding outer radius in the gas plenum region

3.3.3.2.3 Coolant Node

The equation for the coolant node in the gas plenum region is the same as Eq. 3.3-62 for reflector zones, except that in the gas plenum

$$H_{erc} = \frac{h_e k_e}{k_e + \frac{h_c\left(r_{erp} - r_{brp}\right)}{2}} \tag{3.3-74}$$

$$S_{cr} = 2\pi r_{erp} \tag{3.3-75}$$

and

k_e = cladding thermal conductivity.

3.3.3.2.4 Structure Nodes

The finite difference approximations used for the structure in the gas plenum region are the same as those used in the core and blankets. Equations 3.3-45 and 3.3-47 are used, except that Q_{st} is zero.

3.3.3.3 Solution of Finite Difference Equations

A direct solution of the finite difference equations would be complicated, since Eq. 3.3-70 connects a number of axial nodes, each containing a number of radial nodes. Instead, an approximate solution method is used. This method uses the assumption that the total heat capacity of the gas is much less than the total heat capacity of the cladding in the gas plenum region, or

$$(\rho c)_g A_g << \rho_e c_e A_{ep} \tag{3.3-76}$$

which should always be the case.

The solution method contains five steps:

1. Set $T_{g2} = T_{g1}$.
2. Solve Eqs. 3.3-72, 3.3-62, 3.3-45, and 3.3-47 for all axial nodes in the gas plenum to get new cladding, coolant, and structure temperatures.
3. Use Eq. 3.3-70 to obtain a new computed value for T_{g2}.
4. For each axial node, calculate the heat flow error, ΔE, due to the assumption in step 1:

$$\Delta E = (\rho c)_g A_g \left(T_{g2} - T_{g1}\right) \tag{3.3-77}$$

5. Add this heat flow to the cladding, changing the cladding temperature:

$$T_{e2} = T_{c2} + \frac{\Delta E}{\rho_e c_e A_{ep}} \tag{3.3-78}$$

In step 2, the equations for each axial node give a matrix equation of the form

$$\begin{bmatrix} \alpha_1+\beta_1 & -\beta_1 & 0 & 0 \\ -\beta_1 & \alpha_1+\beta_1+\beta_2 & -\beta_2 & 0 \\ 0 & -\beta_2 & \alpha_3+\beta_2+\beta_3 & -\beta_3 \\ 0 & 0 & -\beta_3 & \alpha_4+\beta_3+\beta_4 \end{bmatrix} \begin{bmatrix} T_2''(1) \\ T_2''(2) \\ T_2''(3) \\ T_2''(4) \end{bmatrix} = \begin{bmatrix} D_1 \\ D_2 \\ D_3 \\ D_4 \end{bmatrix} \tag{3.3-79}$$

where

$$T_2''(1) = T_{e2}(jp) \tag{3.3-80a}$$

$$T_2''(2) = \bar{T}_{c2}(jc) \tag{3.3-80b}$$

$$T_2''(3) = T_{sti2}(jc) \tag{3.3-80c}$$

$$T_2''(4) = T_{st02}(jc) \tag{3.3-80d}$$

$$\alpha_1 = \rho_e c_e A_{ep} + 2\pi\, r_{brp}\, \Delta t\ \theta_2\, H_{eg} \tag{3.3-81a}$$

$$\alpha_2 = \bar{\rho}_e \bar{c}_e A_e + 2\bar{c}_c\ \theta_2 \left|w_2\right| \Delta t \tag{3.3-81b}$$

$$\alpha_3 = \left(\rho c\right)_{sti} d_{sti}\, S_{pr} \tag{3.3-81c}$$

$$\alpha_4 = \left(\rho c\right)_{sto} d_{sto}\, S_{pr} \tag{3.3-81d}$$

$$\beta_1 = 2\pi\, r_{erp}\ \theta_2\, H_{erc2}\left(jc\right)\Delta t \tag{3.3-82a}$$

$$\beta_2 = \theta_2\, H_{sic2}\left(jc\right) S_{pr}\, \Delta t \tag{3.3-82b}$$

$$\beta_3 = \theta_2\, H_{stio2}\left(jc\right) S_{pr}\, \Delta t \tag{3.3-82c}$$

$$\beta_4 = 0 \tag{3.3-82d}$$

$$D_1 = T_1''(1)\left[\alpha_1 - \frac{\theta_1}{\theta_2}\beta_1\right] + \frac{\theta_1}{\theta_2}\beta_1 T_1''(2) + 2\pi r_{brp}\Delta t H_{eg}\left[T_{g1} - T_{e1}(jp)\right] \tag{3.3-83a}$$

$$D_2 = \frac{\theta_1}{\theta_2}\beta_1 T_1''(1) + T_1''(2)\left[\alpha_2 - \frac{\theta_1}{\theta_2}(\beta_1+\beta_2)\right] + \frac{\theta_1}{\theta_2}\beta_2 T_1''(3) + 2\frac{\overline{c}_c \Delta t}{\Delta z}\left[\theta_1 |w_1| T_1' + \theta_2 |w_2| T_2'\right] \tag{3.3-83b}$$

$$D_3 = \frac{\theta_1}{\theta_2}\beta_2 T_1''(2) + T_1''(3)\left[\alpha_1 - \frac{\theta_1}{\theta_2}(\beta_2+\beta_3)\right] + \frac{\theta_1}{\theta_2}\beta_3 T_1''(4) \tag{3.3-83c}$$

$$D_4 = \frac{\theta_1}{\theta_2}\beta_3 T_1''(3) + T_1''(4)\left[\alpha_4 - \frac{\theta_1}{\theta_2}(\beta_3+\beta_4)\right] + S_{pr} Q_{chch}\Delta t \tag{3.3-83d}$$

$$T_1''(1) = T_{e1}(jp) \tag{3.3-84a}$$

$$T_1''(2) = \overline{T}_{c1}(jc) \tag{3.3-84b}$$

$$T_1''(3) = T_{sti1}(jc) \tag{3.3-84c}$$

and

$$T_1''(4) = T_{sto1}(jc) \tag{3.3-84d}$$

This tri-diagonal matrix equation is solved in the same manner as in Section 3.3.1.2 above. For step 3, Eq. 3.3-70 is rewritten as

$$T_{g2} = \frac{T_{g1} + d' s'}{1 + d'\theta_2} \tag{3.3-85}$$

where

$$d' = \frac{2\pi r_{brp} H_{eg} \Delta t}{(\rho c)_g A_g} \tag{3.3-86}$$

and

$$s' = \frac{\sum_{jp} \left\{ \theta_1 \left(T_{c1}(jp) - T_{g1} \right) + \theta_2 \, T_{e2}(jp) \right\} \Delta z(jp)}{\sum_{jp} \Delta z(jp)} \tag{3.3-87}$$

3.3.4 Order of Solution

For a time step, the coolant flow rates are calculated before any temperatures are calculated. Extrapolated coolant temperatures are used to obtain temperature-dependent coolant properties for the flow-rate calculations. Section 3.9 describes the pre-voiding coolant flow-rate calculations.

The order in which the temperature calculations are carried out depends on whether the coolant flow is up or down. In either case, the calculation goes in the direction of the flow. For upward flow, the calculation starts at the subassembly inlet. The coolant temperature at the first coolant node is set equal to the inlet temperatures, T_{in}:

$$T_{c2}(1) = T_{in} \tag{3.3-88}$$

The inlet temperature is supplied by the primary loop calculation, as discussed in Chapter 5. For each axial coolant node, *jc*, the coolant temperature, $T_{c2}(jc)$, at the bottom of the node is used as input for the simultaneous calculation of temperatures at all radial nodes. The solutions in Sections 3.3.1, 3.3.2, and 3.3.3 provide $\overline{T}_{c2}(jc)$, the average coolant temperature for the axial node. The T_{c2} (*jc* + 1) is obtained from

$$T_{c2}(jc+1) = 2\overline{T}_{c2}(jc) - T_{c2}(jc) \tag{3.3-89}$$

This coolant temperature at the top of the node *jc* is the coolant inlet temperature used as input for the calculation at node *jc* + 1.

If the flow is downward, the process is the same, except that the calculation starts at the top of the subassembly and works down. The coolant reentry temperatures, $T_{up\ell}$, described in Section 3.3.6 is used as the starting point for the last coolant node MZC:

$$T_{c2}(\mathrm{MZC}) = T_{up\ell} \tag{3.3-90}$$

For each coolant node, *jc*, T_{c2} (*jc* + 1) is used as input to the calculation of all radial nodes. The solutions in Sections 3.3.1, 3.3.2, and 3.3.3 again provide $\overline{T}_{c2}(jc)$. Then T_{c2} (*jc*) is obtained from

$$T_{c2}(jc) = 2\overline{T}_{c2}(jc) - T_{c2}(jc+1) \tag{3.3-91}$$

3.3.5 Melting of Fuel or Cladding

As discussed in Section 3.3.1.1, melting of both fuel and cladding is treated with a melting range form a solidus temperature, T_{sol}, to a liquidus temperature, T_{liq}, rather than using a sharp melting temperature. Between the solidus and the liquidus, an effective specific heat, c_m, based on the heat of fusion is used, as in Eq. 3.3-3. In the actual temperature calculations for a time step, the heat of fusion is neglected. Then, at the end of the step the temperatures are modified to account for the heat of fusion at any radial node that is going through the melting range. A number of different cases are considered, depending on the relationships between T_1, the temperature at the beginning of the step, and T_2, the temperature calculated for the end of the step, ignoring melting, T_{sol} and T_{liq}. In all cases, T_2'' is the final adjusted temperature at the end of the step, and T_2' would be the final adjusted temperature if it did not go beyond the melting range.

Case 1: $T_1 < T_{\text{sol}}, T_2 > T_{\text{sol}}$

In this case, an adjusted temperature, T_2', is calculated as

$$T_2' = T_{sol} + \left(T_2 - T_{sol}\right)\frac{c}{c_m} \tag{3.3-92}$$

where c is the normal specific heat that was used in the calculation of T_2. Then,

$$T_2' = \begin{cases} T_{liq} + \left(T_2' - T_{liq}\right)\frac{c_m}{c} \text{ if } T_2' > T_{liq} \\ T_2' \quad \text{otherwise} \end{cases} \tag{3.3-93a-b}$$

In essence, Eq. 3.3-92 divides the energy above the solidus by c_m to get the temperature above the solidus; and Eq. 3.3-93 divides any energy above the liquidus by c to get the temperature above the liquidus.

Case 2: $T_1 > T_{\text{liq}}, T_2 < T_{\text{liq}}$

In this case,

$$T_2' = T_{liq} + \left(T_2 - T_{liq}\right)\frac{c}{c_m} \tag{3.3-94}$$

$$T_2' = \begin{cases} T_{sol} + \left(T_2' - T_{sol}\right)\frac{c_m}{c} \text{ if } \quad T_2' > T_{sol} \\ T_2' \quad \text{otherwise} \end{cases} \tag{3.3-95a-b}$$

Case 3: $T_{\text{liq}} \geq T_1 \geq T_{\text{sol}}$

In this case

$$T_2' = T_1 + \left(T_2 - T_1\right)\frac{c}{c_m} \tag{3.3-96}$$

$$T_2' = \begin{cases} T_{liq} + \left(T_2' - T_{liq}\right)\dfrac{c_m}{c} \text{ if } T_2' > T_{liq} \\ T_{sol} + \left(T_2' - T_{sol}\right)\dfrac{c_m}{c} \text{ if } T_2' < T_{sol} \\ T_2' \quad \text{otherwise} \end{cases} \tag{3.3-97a-b-c}$$

3.3.6 Coolant Inlet and Re-entry Temperature

The coolant inlet temperature at the subassembly inlet and the re-entry temperature at the subassembly outlet are determined mainly by the coolant plenum temperatures calculated in PRIMAR-4; but, in addition, mixing zones are modeled at the inlet and outlet of each channel. Therefore, the temperature of coolant entering a subassembly is based on the temperature in the part of the plenum in the immediate vicinity of the end of the subassembly, and the reentry temperature soon after an expulsion is based on both the bulk plenum temperature and the temperature of the coolant that was recently expelled.

The mixing mode uses a mass, M_{mix}, of the coolant in the mixing volume, a specific heat, C_{mix}, of the coolant, and a time constant, τ_{mix}, for heat transfer or mixing between the mixing volume and the bulk plenum coolant. If all temperatures and flows are constant, then the mixing volume temperature, T_{mix}, is assumed to approach an equilibrium value asymptotically with an exponential decay. Figure 3.3–1 shows the model for the outlet mixing volume.

If T_{mix1} is the mixing volume temperature at the beginning of a time step, the T_{mix2} is the value at the end, then

$$T_{\text{mix}2} = T_{eq} + \left(T_{mix1} - T_{eq}\right) e^{-\frac{\Delta t}{\tau}} \tag{3.3-98}$$

For the outlet plenum,

$$\frac{1}{\tau} = \frac{1}{\tau_{\text{mix}}} + \frac{|\overline{w}|}{M_{\text{mix}}} \tag{3.3-99}$$

where $\overline{w}$ is the average coolant flow rate at the subassembly outlet. The equilibrium temperature is

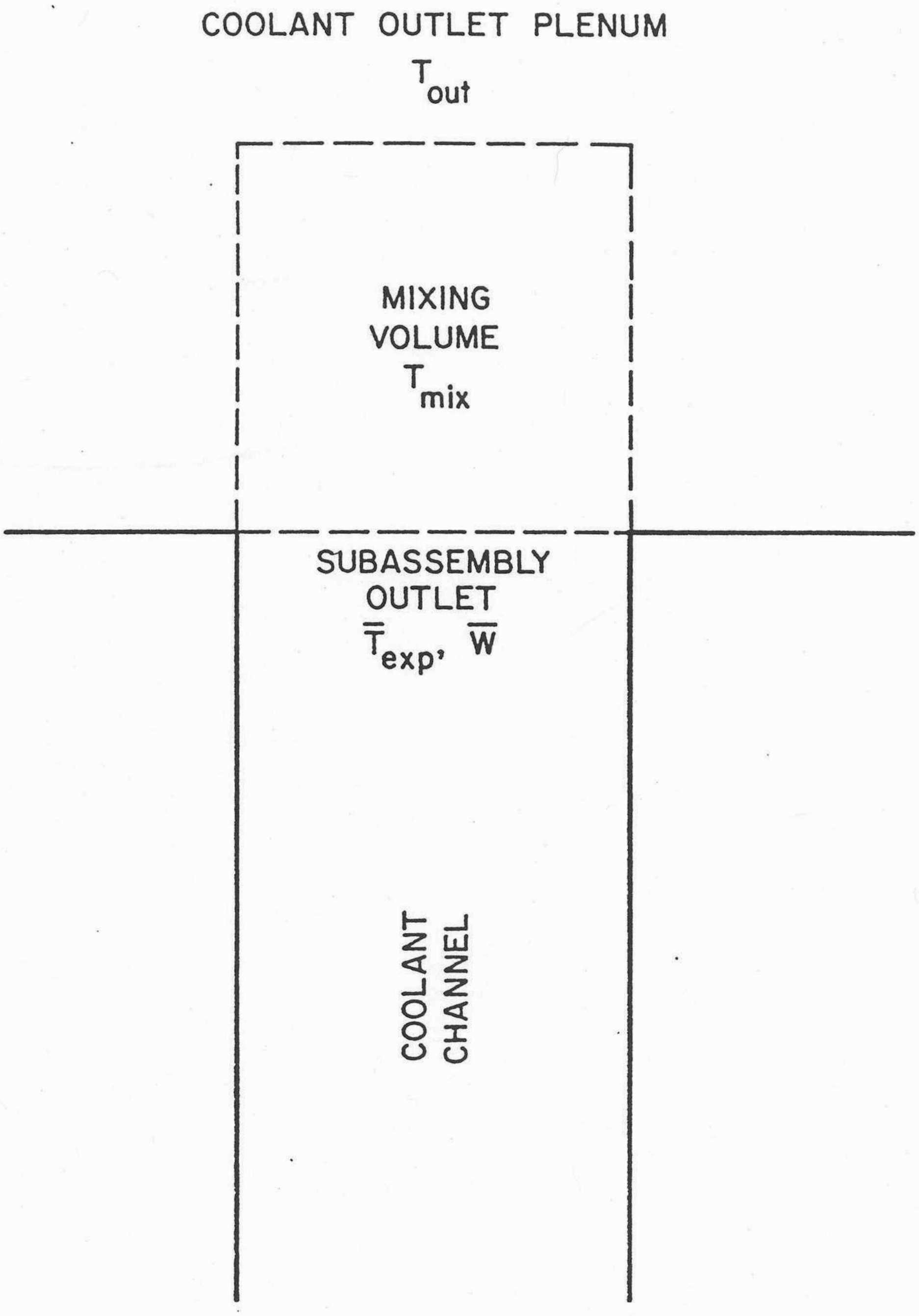


Figure 3.3–1: Coolant Re-entry Temperature Model

$$T_{eq} = \begin{cases} T_{\text{out}} & \text{if } \overline{w} \le 0 \\ \left(\dfrac{T_{\text{out}}}{\tau_{\text{mix}}} + \dfrac{\overline{w}\,\overline{T}_{\text{exp}}}{M_{\text{mix}}} \right) \tau & \text{if } \overline{w} > 0 \end{cases} \qquad (3.3\text{-}100\text{a-b})$$

where $\overline{T}_{\text{exp}}$ is the average temperature of the sodium expelled from the top of the subassembly into the mixing volume during the time step. T_{out} is the bulk temperature in the outlet plenum. For the inlet plenum, the same equations are used except that $\overline{w}$ changes sign because positive flow results in flow from the inlet mixing volume into the subassembly inlet. For the inlet, Eqs. 3.3-100a and 3.3-100b become

$$T_{\text{eq}} = \begin{cases} T_{\text{in}} & \text{if } \overline{w} \ge 0 \\ \left(\dfrac{T_{\text{in}}}{\tau_{\text{mix}}} - \dfrac{\overline{w}\,\overline{T}_{\text{exp}}}{M_{\text{mix}}} \right) \tau & \text{if } \overline{w} < 0 \end{cases} \qquad (3.3\text{-}101\text{a-b})$$

where T_{in} is the bulk temperature in the inlet plenum.

The coolant inlet and reentry temperature calculations described above are used in both the pre-voiding module and the boiling module except when a vapor bubble has blown out of the top of the channel. In this case, the condensation of vapor in the outlet plenum raises the temperature in the mixing volume. A two-step process is used for each time step. In the first step, vapor condensation raises the mixing volume temperature. In the second step, heat transfer or mixing with the bulk plenum coolant is accounted for. For the first step, a condensation heat-transfer time constant, τ_c, is calculated as

$$\tau_c = \frac{M_{\text{mix}}\, C_{\text{mix}}}{h_{\text{cond}}\, \Delta Z_v\, p_{er}} \qquad (3.3\text{-}102)$$

where h_{cond} is the vapor condensation heat-transfer coefficient, ΔZ_v is the length of the vapor bubble beyond the subassembly outlet, and p_{er} is the bubble perimeter or surface area per unit length. The value of p_{er} is taken from the coolant channel dimensions in the top node below the subassembly outlet. For step 1, the new mixing volume temperature, T_{new}, is calculated as

$$T_{\text{new}} = \overline{T}_v + \left(T_{\text{mix}1} - \overline{T}_v \right) e^{-\frac{\Delta t}{\tau_c}} \qquad (3.3\text{-}103)$$

where $\overline{T}_v$ is the average vapor temperature in the part of the bubble above the subassembly outlet. In the second step, the final mixing volume temperature is calculated as

$$T_{\text{mix2}} = T_{\text{new}} + \left(T_{\text{out}} - T_{\text{new}}\right)\left(1 - e^{-\frac{\Delta t}{\tau_{\text{mix}}}}\right) \tag{3.3-104}$$

3.4 Steady-State Thermal Hydraulics

The steady-state thermal hydraulics calculations for a channel using the single pin per subassembly option consist of direct solutions of the relevant steady-state equations, rather than running the transient calculations until they converge to a steady-state solution.

For the steady-sate calculations, the user specifies the coolant flow rate for each channel, the coolant inlet temperature and exit pressure, and the power in each node of each channel. The code then calculates the remaining coolant temperatures and pressures, as well as the temperatures in the fuel, cladding, structure, and reflectors. First, the coolant temperatures in a channel are calculated, starting at the inlet and working upward. The steady-state coolant temperature calculation requires only the coolant flow rate, the total power in each axial node, and the coolant heat capacity; so coolant temperatures can be calculated before the fuel and cladding temperatures are known. The second step is to calculate the coolant pressures, starting at the subassembly outlet and working down. Inlet orifice coefficients are adjusted so that all channels have the same total pressure drop. The pressure calculations are described in Section 3.9. The third step is to set the structure and reflector temperatures equal to the coolant temperatures everywhere except in the core and axial blankets. The gas plenum temperatures are also set equal to the coolant temperature in this region, and the cladding temperature in the gas plenum region is also set equal to the coolant temperature. Fourth, the fuel-pin temperatures are calculated for each axial node in the core and axial blankets, starting at the cladding outer surface and working inward. Last, the structure temperatures in the core and axial blankets are calculated.

3.4.1 Basic Equations

The basic heat-transfer equations used in the steady-state calculations are the same as those used for the transient solution, except that all of the time derivatives are dropped in the steady-state solution. These equations include Eq. 3.3-1 and Eqs. 3.4-4 to 3.3-10. Also, the spatial finite differencing used in the steady-state is the same as that used in the transient.

For the steady-state calculations, eq. 3.3-5 becomes

$$\frac{d}{dz}\left(wc_c T\right) = Q_{ct} A_c \tag{3.4-1}$$

where the total heat source per unit volume, Q_{ct}, at node *jc* is

$$Q_{ct}(jc) = Q_c(jc) + Q_{ec}(jc) + Q_{sc}(jc) = \frac{\overline{P}(jc)}{A_c \Delta z(jc)} \tag{3.4-2}$$

and $\overline{P}(jc)$ is the total steady-state power (watts) in the node. For this equation, it is assumed that all heat generated in the fuel, cladding, and structure ends up in the coolant. Note that outside the core and axial blankets $\overline{P}(jc)$ and $Q_{ct}(jc)$ are zero.

For the steady-state fuel and cladding calculations, Eq. 3.3-1 is multiplied by $2\pi r$ and integrated from the fuel inner surface, r_{if}, to give

$$2\pi kr \frac{dT}{dr} = -2\pi \int_{r_{if}}^{r} r' Q(r') dr' \tag{3.4-3}$$

where the adiabatic boundary condition at r_{if} has been used.

3.4.2 Coolant Temperatures

The finite difference form for Eq. 3.4-1 may be written as

$$w\overline{c}_c(j) \frac{\left[T_c(jc+1) - T_c(jc)\right]}{\Delta z(jc)} = \frac{\overline{P}(jc)}{\Delta z(jc)} \tag{3.4-4}$$

or

$$T_c(jc+1) = T_c(jc) + \frac{\overline{P}(jc)}{w\overline{c}_c(jc)} \tag{3.4-5}$$

where $\overline{c}_c(jc)$ is the specific heat evaluated at the average temperature, $\overline{T}_c(jc)$, given by

$$\overline{T}_c(jc) = \frac{T_c(jc) + T_c(jc+1)}{2} \tag{3.4-6}$$

Also, $T_c(1)$ is equal to the inlet temperature:

$$T_c(1) = T_{in} \tag{3.4-7}$$

Starting from jc = 1, Eq. 3.4-5 is used to match up the channel. An iteration is used to obtain consistency between $\overline{c}_c(jc)$ and $\overline{T}_c(jc)$.

3.4.3 Fuel and Cladding Temperatures in the Core and Axial Blankets

At each axial node, the radial node powers, $Q(i)$, are calculated using Eqs. 3.3-22 to

3.3-25. Note that the Q in Eq. 3.4-3 is a power per unit volume, whereas $Q(i)$ is an integral value for a node:

$$Q(i) = \int_{z(j)}^{z(j+1)} \int_{r_i}^{r_{i+1}} 2\pi r\, Q\, dr\, dz \tag{3.4-8}$$

The sums, $Q_{sm}(i)$, are calculated as

$$Q_{sm}(i) = \sum_{ii=1}^{i} Q(ii) \tag{3.4-9}$$

Equation 3.4-3 becomes

$$2\pi \bar{k}_{i,i+1} \frac{r(i+1)\left[T(i+1)-T(i)\right]}{\Delta r_{i,i+1}} = \frac{Q_{sm}(i)}{\Delta z} \tag{3.4-10}$$

or

$$T(i) = T(i+1) + \frac{\Delta r_{i,i+1}\; Q_{sm}(i)}{2\pi k_{i,i+1}\; r(i+1)\Delta z} \tag{3.4-11}$$

where $\Delta r_{i,i+1}$ and $\bar{k}_{i,i+1}$ are given by Eqs. 3.3-21 and 3.3-26.

The calculations for an axial node start with the coolant temperature that has already been calculated, as in the section above:

$$T(\mathrm{NC}, j) = \bar{T}(jc) \tag{3.4-12}$$

Then the cladding surface temperature is given by

$$T(\mathrm{NE'}) = T(\mathrm{NC}) + \frac{Q_{sm}(\mathrm{NE'})}{2\pi r(\mathrm{NE'})\Delta z h_c} \tag{3.4-13}$$

Cladding temperatures at nodes NE and NE″ are calculated using Eq. 3.4-11. Since $\bar{k}_{i,i+1}$ can be a function of T_i, a simple iteration between Eq. 3.4-11 and Eq. 3.3-26 is used.

The equation used for the fuel surface temperature is

$$2\pi r(\mathrm{NR}) \left\{ h_b\left[T(\mathrm{NT}) - T(\mathrm{NE''})\right] + \varepsilon\sigma\left[T(\mathrm{NT})^4 - T(\mathrm{NE''})^4\right] \right\} = \frac{Q_{sm}(\mathrm{NT})}{\Delta z} \tag{3.4-14}$$

or

$$T\left(\mathrm{NT}\right)=d_1-d_2\,T\left(\mathrm{NT}\right)^4 \tag{3.4-15}$$

where

$$d_1=T\left(\mathrm{NE''}\right)+\frac{Q_{sm}\left(\mathrm{NT}\right)}{2\pi r\left(\mathrm{NR}\right)\Delta z h_b}+\frac{\varepsilon\sigma T\left(\mathrm{NE''}\right)^4}{h_b} \tag{3.4-16}$$

and

$$d_2=\frac{\varepsilon\sigma}{h_b} \tag{3.4-17}$$

Equation 3.4-15 is solved by iteration.

After the fuel surface temperature has been calculated, the inner fuel node temperatures are calculated one at a time, starting at the outside and working inward, by iterating between Eqs. 3.4-11 and 3.3-26. In this procedure, *T*(*i*) is to be found after *T*(*i*+1) is known. First, *T*(*i*) is set equal to $T(i+1)$. Second, *k*(*i*) is to be found after *T*(*i*+1) is known. First, *T*(*i*) is set equal to $T(i+1)$. Second, *k*(*i*) is calculated as a function of the temperature, *T*(*i*). Third, $\bar{k}_{i,i+1}$is calculated using Eq. 3.3-26. Fourth, a new value for *T*(*i*) is calculated, using Eq. 3.4-11. Fifth, the new *T*(*i*) from the fourth step is compared with old value used in the second step. If the two values differ by less than a user-specified convergence criterion, then the iteration is finished, and the code goes on to the next node. Otherwise, the code goes back to the second step, using the new value of *T*(*i*), and repeats the process.

3.4.4 Structure Temperatures in the Core Axial Blankets

The inner structure node temperature is calculated using

$$\Delta z\,S_{pr}\,H_{sic}\left[T\left(\mathrm{NSI}\right)-T\left(\mathrm{NC}\right)\right]=\gamma_s\bar{P}\left(j\right) \tag{3.4-18}$$

or

$$T\left(\mathrm{NSI}\right)=T\left(\mathrm{NC}\right)+\frac{\gamma_s\bar{P}\left(j\right)}{\Delta z S_{pr}\,H_{sic}} \tag{3.4-19}$$

The outer structure node is then calculated using

$$T\left(\mathrm{NSO}\right)=T\left(\mathrm{NSI}\right)+\frac{\gamma_{s}\bar{P}\left(j\right)d_{sto}}{\Delta zS_{pr}\,H_{stio}\left(d_{sti}+d_{sto}\right)} \tag{3.4-20}$$

3.4.5 Reflector, Structure, Cladding, and Gas Plenum Temperature Outside the Core and Axial Blankets

Outside the core and axial blankets no power sources are considered, so the reflector and structure temperatures at an axial node are the same as the coolant temperature for the steady-state. The coolant temperatures are the same at all axial nodes in the gas plenum region, and the cladding and gas temperatures in this region are equal to the coolant temperatures.

3.5 Transient Heat Transfer after the Start of Boiling

After the start of boiling, the coolant temperatures are calculated in the coolant routines, rather than being calculated simultaneously with fuel, cladding, and structure temperatures. Coupling between the boiling calculations and the non-coolant heat-transfer calculation takes place in two parts for each time step. First, the boiling routines us extrapolated cladding and structure temperatures to calculate the heat fluxes to the coolant for the boiling calculation. Then the heat fluxes actually used in the coolant routines are passed to the heat-transfer routines to be used as boundary conditions at the cladding, structure, and reflector surfaces. The net results of this procedure are that energy is conserved, and a fully implicit boiling calculation can be made without requiring a direct simultaneous solution of all of the fuel-pin temperatures in the boiling model. The coupling through extrapolated cladding and structure temperatures and heat fluxes at the cladding and structure surfaces imposes numerical stability limitations on the heat-transfer time-step sizes. Currently, fuel-pin temperatures are calculated at the end of the fuel-pin heat-transfer time step, whereas structure and reflector temperatures are calculated at every coolant time step. The coolant time step can be no longer than the heat-transfer step, and the coolant step is often much shorter. For typical fuel pins, the stability limit for the heat-transfer time step is of the order of .02 s. With a thin structure, the stability limit for structure temperature calculations could be less than .02 s, although for typical duct wall thicknesses (.12 in. or .003 m) the stability limit would be closer to one second. Should timing studies indicate that the structure and reflector temperature calculations account for a significant fraction of the total computing time, then the code will be modified so as to do these calculations less often than once every coolant time step.

3.5.1 Fuel and Cladding Temperatures in the Core and Axial Blanket

The equations used for fuel and cladding temperatures after the switch to the boiling module are the same as those used in the non-voiding module, except that in the boiling module the fuel-pin heat-transfer calculations stop at the cladding outer surface rather than carrying through to the structure outer node. The finite difference equations for radial nodes 1-NE are the same as Eqs. 3.3-16 to 3.3-36. For node NE', the cladding outer node, the heat flux to the coolant at the cladding surface must be accounted for. Also, in a boiling region, a film of liquid sodium can be left on the cladding. Since the film is in intimate contact with the cladding, the heat capacity of the film is added to the heat capacity of the cladding outer node, rather than being accounted for in the boiling calculation. Thus, the finite difference equation for node *NE′* becomes

$$\begin{aligned}&\left[\frac{m_e c_e}{4}+2\pi r\left(\mathrm{NE'}\right)\rho_c c_c w_{fe}\Delta z\left(j\right)\right]\cdot\left[\frac{T_2\left(\mathrm{NE'}\right)-T_1\left(\mathrm{NE'}\right)}{\Delta t}\right]\\&\quad=-\frac{2\pi\overline{r}_{\mathrm{NE'}}\Delta z\left(j\right)\overline{k}_{\mathrm{NE,\,NE'}}}{\Delta r_{\mathrm{NE,\,NE'}}}\left\{\theta_{21}\left[T_1\left(\mathrm{NE'}\right)-T_1\left(\mathrm{NE}\right)\right]\right.\\&\qquad\left.+\theta_2\left[T_2\left(\mathrm{NE'}\right)-T_2\left(\mathrm{NE}\right)\right]\right\}-2\pi r\left(\mathrm{NE'}\right)\Delta z\left(j\right)\frac{E_{ec}\left(j\right)}{\Delta t}\\&\qquad+Q\left(\mathrm{NE'}\right)\end{aligned}\tag{3.5-1}$$

where w_{fe} is the thickness of liquid sodium film left on the cladding after voiding occurs, and E_{ec} is the integrated heat flux from cladding to coolant.

The value of E_{ec} is computed in the coolant routines as

$$E_{ec}\left(jc\right)=\int_{t}^{t+\Delta t}\frac{T_{eex}\left(jc\right)-\overline{T}_c\left(jc\right)}{R_{ec}\left(jc\right)}dt'\tag{3.5-2}$$

where

T_{eex} = extrapolated cladding temperature at a point ¼ of the way from the outer cladding surface to the inner cladding surface:

$$\begin{aligned}T_{eex}\left(j,t'\right)=f_1&\frac{\left[T\left(\mathrm{NE},j,t_1\right)+T\left(\mathrm{NEP},j,t_1\right)\right]}{2}\\+f_2&\frac{\left[T\left(\mathrm{NE},j,t_2\right)+T\left(\mathrm{NEP},j,t_2\right)\right]}{2}\end{aligned}\tag{3.5-3}$$

$$f_1=\frac{t_2-t'}{t_2-t_1}\tag{3.5-4}$$

$$f_1=1-f_1\tag{3.5-5}$$

$$R_{ec} = \frac{1}{h_c} + R_{ehf} \tag{3.5-6}$$

and

$$R_{ehf} = \frac{\left[r(\mathrm{NE'}) - r(\mathrm{NE}) \right]}{k_e} \left\{ \frac{r(\mathrm{NE'}) \left[1 - \frac{\gamma_e}{4(1-\gamma_c-\gamma_s)} \right]}{r(\mathrm{NE}) + 3r(\mathrm{NE'})} \right\} \tag{3.5-7}$$

If a node is partly voided, then E_{ec} is averaged over the length of the node as well as integrated over time.

The $\rho_c c_c w_{fe}$ term is supplied by the boiling routines. It represents the heat capacity of any liquid film left on the cladding after voiding occurs. This term is zero unless voiding has occurred at this axial node. Since the film temperature tends to follow the cladding surface temperature more closely than it follows the vapor temperature, the film heat capacity is lumped with the cladding outer node, and film temperatures are not explicitly calculated in the boiling routines.

The finite difference equations are again put in a matrix form like Eq. 3.3-48, except that in the voiding case there are only NE′ elements. The definitions of α, β, and D are the same as in Eqs. 3.3-49 to 3.3-51 for nodes 1-NE. For node NE′, α is still given by Eq.

$$\beta_{\mathrm{NE'}} = 0 \tag{3.5-8}$$

3.3-49d or 3.3-49b, except that the $\rho_c c_c w_{fe}$ term is added to it. Also,

and

$$D_{\mathrm{NE'}} = \frac{\theta_1}{\theta_2} \beta_{\mathrm{NE}} T_1(\mathrm{NE}) + T_1(\mathrm{NE'}) \left[\alpha_{\mathrm{NE'}} - \frac{\theta_1}{\theta_2} \beta_{\mathrm{NE}} \right] + \psi_{\mathrm{NE'}} - r(\mathrm{NE'}) E_{ec}(j) \tag{3.5-9}$$

The equations are solved in the same manner as in the non-voiding case.

3.5.2 Structure Temperatures

The basic equations used for the two structure radial nodes at an axial node are

$$(\rho c)_{sto} d_{sto} \frac{dT_{sto}}{dt} = H_{stio} \left(T_{sti} - T_{sto} \right) + Q_{st} f_o \tag{3.5-10}$$

and

$$\left[(\rho c)_{sti}\, d_{sti} + \rho_c c_c w_{fst}\right]\frac{dT_{sti}}{dt} = H_{stio}\left(T_{sto} - T_{sti}\right) + Q_{st}\, f_i + \frac{\left(T_c - T_{sti}\right)}{R_{sc}} \tag{3.5-11}$$

where

$$f_i = \frac{d_{sti}}{d_{sti} + d_{sto}} \tag{3.5-12}$$

$$f_o = \frac{d_{sto}}{d_{sti} + d_{sto}} \tag{3.5-13}$$

and w_{fst} = thickness of the liquid-sodium film left on the structure after voiding occurs.

The heat capacity of the film in the boiling region, as represented by the $p_c c_c w_{fst}$ term, is supplied by the boiling routines to be added to the inner structure node.

The structure temperature calculation is either a semi-implicit or a fully implicit calculation, depending on the time-step in relation to an inner structure node heat-transfer time constant, τ_{sti}, calculated as

$$\tau_{sti} = \frac{(\rho c)_{sti}\, d_{sti}^2}{2k_{sti}} \tag{3.5-14}$$

If Δt is less than τ_{sti} then the semi-implicit calculation is used. Otherwise the fully implicit calculation is used.

3.5.2.1 Semi-Implicit Calculations

For the semi-implicit calculation, finite differencing of Eqs. 3.5-10 and 3.5-11 gives

$$(\rho c)_{sto}\, d_{sto}\, \frac{\left(T_{sto2} - T_{sto1}\right)}{\Delta t} = \frac{H_{stio}}{2}\left[T_{sti2} - T_{sto2} + T_{sti1} - T_{sto1}\right] + Q_{st}\, f_0 \tag{3.5-15}$$

and

$$\begin{aligned} &\left[(\rho c)_{sti}\, d_{sti} + \rho_c c_c w_{fst}\right]\frac{\left(T_{sti2} - T_{sti1}\right)}{\Delta t} \\ &\quad = \frac{H_{stio}}{2}\left[T_{sto2} - T_{sti2} + T_{sto1} - T_{sti1}\right] + Q_{st}\, f_i - \frac{E_{sc}}{\Delta t} \end{aligned} \tag{3.5-16}$$

where

$$E_{sc}\left(jc\right) = \int_{t}^{t+\Delta t} \frac{T_{stex}\left(jc\right) - \overline{T}_{c}\left(jc\right)}{R_{sc}\left(jc\right)} dt' \tag{3.5-17}$$

The value of E_{sc} is computed in the boiling routines. T_{stex} is the extrapolated structure inner node temperature, and

$$R_{sc} = \frac{1}{h_c} + \frac{d_{sti}}{2k_{sti}} \tag{3.5-18}$$

Equations 3.5-15 and 3.5-16 are put in the form

$$a_{11}T_{sto2} + a_{12}T_{sti2} = b_1 \tag{3.5-19}$$

$$a_{21}T_{sto2} + a_{22}T_{sti2} = b_2 \tag{3.5-20}$$

with the solutions

$$T_{sto\,2} = \frac{a_{22}b_1 - a_{12}b_2}{a_{11}a_{22} - a_{12}a_{21}} \tag{3.5-21}$$

and

$$T_{sti\,2} = \frac{a_{11}b_2 - a_{21}b_2}{a_{11}a_{22} - a_{12}a_{21}} \tag{3.5-22}$$

where

$$a_{11} = \left(\rho c\right)_{sto} d_{sto} + \frac{\Delta t}{2} H_{stio} \tag{3.5-23}$$

$$a_{12} = -\frac{\Delta t}{2} H_{stio} \tag{3.5-24}$$

$$a_{21} = a_{12} \tag{3.5-25}$$

$$a_{22} = \left(\rho c\right)_{sti} d_{sti} + \frac{\Delta t}{2} H_{stio} + \rho_c c_c w_{fst} \tag{3.5-26}$$

$$b_1 = \left[(\rho c)_{sto} \, d_{sto} - \frac{\Delta t}{2} H_{stio} \right] T_{sto1} + Q_{st} f_o \Delta t - a_{12} \, T_{sti1} \tag{3.5-27}$$

and

$$b_2 = \left[(\rho c)_{sti} \, d_{stio} - \frac{\Delta t}{2} H_{stio} + \rho_c c_c w_{fst} \right] T_{sti1} - a_{12} \, T_{sto1} + Q_{st} f_i \Delta t - \Delta t \, Q_{sc} \tag{3.5-28}$$

3.5.2.2 Fully Implicit Calculations

Since the inner structure node may represent only a small fraction of the total structure thickness, τ_{sti} can be small. If the time-step size is appreciably larger than τ_{sti} then the semi-implicit calculation can become numerically unstable. Therefore, a different algorithm is used for larger time-step sizes. This algorithm uses two steps. First, a fully implicit calculation is made, using a coolant temperature and thermal resistance to the coolant as structure surface boundary conditions, rather than using the integrated heat flux. In this first step, the heat flux from the coolant to the structure will, in general, not match the heat flux form structure to coolant used in the coolant calculations. Therefore, in the second step, the inner node and outer node structure temperatures are both adjusted by the same amount so that the integrated heat flux from structure to coolant is matched.

For the first step the finite difference equations used for the two structure node temperatures are

$$(\rho c)_{sto} \, d_{sto} \frac{(T_{sto2} - T_{sto1})}{\Delta t} = H_{stio} \left(T_{sti2} - T_{sto2} \right) + Q_{st} \, f_o \tag{3.5-29}$$

and

$$\left[(\rho c)_{sti} \, d_{sti} + \rho_c c_c w_{fst} \right] \frac{(T_{sti2} - T_{sti1})}{\Delta t} = H_{stio} \left(T_{sto2} - T_{sti2} \right) + \frac{T_c - T_{sti2}}{R_{sc}} + Q_{st} \, f_i \tag{3.5-30}$$

where the values for T_c and R_{sc} are supplied by the coolant routines. The solutions for these two equations again have the same form as Eqs. 3.5-21 and 3.5-22 except that in this case the coefficients are defined as

$$a_{11} = (\rho c)_{sto} \, d_{sto} + \Delta t \, H_{stio} \tag{3.5-31}$$

$$a_{12} = a_{21} = -\Delta t \, H_{stio} \tag{3.5-32}$$

and

$$a_{22} = (\rho c)_{sti}\, d_{sti} + \rho_c c_c w_{fst} + \Delta t\ H_{stio} + \frac{\Delta t}{R_{sc}} \tag{3.5-33}$$

The temperature difference, ΔT_{st}, between the outer and inner nodes is then defined as

$$\Delta T_{st} = T_{sto2} - T_{sti2} \tag{3.5-34}$$

In the second step, ΔT_{st} is preserved but the temperatures are adjusted so as to match the value of E_{sc} supplied by the coolant routines:

$$\begin{aligned}\left[(\rho c)_{sti} d_{sti} + \rho_c c_c w_{fst}\right](T_{sci2} - T_{sti1}) + (\rho c)_{sto} d_{sto}(T_{sto2} - T_{sto1})\\ = Q_{st} - E_{sc}\end{aligned} \tag{3.5-35}$$

The solution to Eqs. 3.5-34 and 3.5-35 is

$$\begin{aligned}T_{sto2} = \{&Q_{st}\Delta t - E_{sc} + (\rho c)_{sto}\, d_{sto} T_{sto1}\\ &+ \left[(\rho c)_{sti}\, d_{sti} + \rho_c c_c w_{fst}\right] \cdot (T_{sti1} + \Delta T_{st})\}\\ &/\left[(\rho c)_{sto}\, d_{sto} + (\rho c)_{sti} d_{sti} + \rho_c c_c w_{fst}\right]\end{aligned} \tag{3.5-36}$$

and

$$T_{sti2} = T_{sto2} - \Delta T_{st} \tag{3.5-37}$$

Note that the second step can still cause numerical instabilities if the time-step size is too large or the total structure thickness is too small, but in the fully implicit scheme the stability limit is based on the total structure thickness, whereas in the semi-implicit scheme the stability is based mainly on the inner node thickness.

3.5.3 Reflector Temperatures

In the boiling module the treatment of reflector temperatures is almost identical to the structure temperature treatment. The main difference is that in the reflector, the outer node is in contact with the coolant, whereas in the structure, the inner code is in contact with the coolant. Also, in the reflector the density, specific heat, and thermal conductivity are the same for both nodes, whereas in the structure these properties can vary from inner node to outer node.

The basic equations are

$$(\rho c)_r\, d_{ri} \frac{dT_{ri}}{dt} = H_{rio}\left(T_{ro} - T_{ri}\right) \tag{3.5-38}$$

and

$$\left[(\rho c)_r\, d_{ro} + \rho_c c_c w_{fr}\right] \frac{dT_{ro}}{dt} = H_{rio}\left(T_{ri} - T_{ro}\right) + \left(T_c - T_{ro}\right) H_{erc} \tag{3.5-39}$$

where

w_{fr} = thickness of the liquid-sodium film left on the reflector after voiding occurs.

An outer reflector node heat-transfer time constant, τ_{ro}, is calculated as

$$\tau_{ro} = \frac{(\rho c)_r\, d_{ro}^2}{2k_r} \tag{3.5-40}$$

and a fully implicit calculation is used if the time-step size is greater than τ_{ro}. Otherwise a semi-implicit calculation is used.

3.5.3.1 Semi-Implicit Calculations

Finite differencing of Eqs. 3.5-38 and 3.5-39 gives

$$(\rho c)_r\, d_{ri} \frac{\left(T_{ri2} - T_{ri1}\right)}{\Delta t} = \frac{H_{rio}}{2}\left(T_{ro2} - T_{ri2} + T_{ro1} - T_{ri1}\right) \tag{3.5-41}$$

$$\left[(\rho c)_r\, d_{ro} + \rho_c c_c w_{fr}\right] \frac{\left(T_{ri2} - T_{ri1}\right)}{\Delta t} = \frac{H_{rio}}{2}\left(T_{ri2} - T_{ro2} + T_{ri1} - T_{ro1}\right) - \frac{E_{rc}}{\Delta t} \tag{3.5-42}$$

where

$$E_{rc} = \int_{t}^{t+\Delta t} \left(T_{rex} - \overline{T}_c\right) H_{erc}\, dt' \tag{3.5-43}$$

These equations are put in the form

$$a_{11} T_{ri2} + a_{12} T_{ro2} = b_1 \tag{3.5-44}$$

$$a_{21} T_{ri2} + a_{22} T_{ro2} = b_2 \tag{3.5-45}$$

with the solution

$$T_{ri2} = \frac{a_{22}b_1 - a_{12}b_2}{a_{11}a_{22} - a_{12}a_{21}} \tag{3.5-46}$$

and

$$T_{ro2} = \frac{a_{11}b_2 - a_{21}b_1}{a_{11}a_{22} - a_{12}a_{21}} \tag{3.5-47}$$

The coefficients are

$$a_{11} = (\rho c)_r \, d_{ri} + \frac{\Delta t}{2} H_{rio} \tag{3.5-48}$$

$$a_{12} = -\frac{\Delta t}{2} H_{rio} \tag{3.5-49}$$

$$a_{21} = a_{12} \tag{3.5-50}$$

$$a_{22} = (\rho c)_r \, d_{ro} + \frac{\Delta t}{2} H_{rio} + \rho_c c_c w_{fr} \tag{3.5-51}$$

$$b_1 = \left[(\rho c)_r d_{ri} - \frac{\Delta t}{2} H_{rio} \right] T_{ri1} + \frac{\Delta t}{2} H_{rio} \, T_{ro1} \tag{3.5-52}$$

and

$$b_2 = \left[(\rho c)_r d_{ro} + \rho_c c_c w_{fr} - \frac{\Delta t}{2} H_{rio} \right] T_{ro1} + \frac{\Delta t}{2} H_{rio} \, T_{ri1} - E_{rc} \tag{3.5-53}$$

3.5.3.2 Fully Implicit Calculations

As in the structure temperature case, a two-step process is used. In the first step, the finite difference equations used are

$$(\rho c)_r \, d_{ri} \frac{(T_{ri2} - T_{ri1})}{\Delta t} = H_{rio} \left(T_{ro2} - T_{ri2} \right) \tag{3.5-54}$$

and

$$\left[(\rho c)_r\, d_{ro} + \rho_c c_c w_{fr}\right]\frac{\left(T_{ro2} - T_{ro1}\right)}{\Delta t} = H_{rio}\left(T_{ri2} - T_{ro2}\right) + \left(\overline{T}_c - T_{ro2}\right)H_{erc} \qquad (3.5\text{-}55)$$

The solutions again have the same form as Eqs. 3.5-46 and 3.5-47, with the coefficients given by

$$a_{11} = (\rho c)_r\, d_{ri} + \Delta t\, H_{rio} \qquad (3.5\text{-}56)$$

$$a_{12} = -\Delta t\, H_{rio} \qquad (3.5\text{-}57)$$

$$a_{21} = a_{12} \qquad (3.5\text{-}58)$$

$$a_{22} = (\rho c)_r\, d_{ro} + \Delta t\ H_{rio} + \rho_c c_c w_{fr} \qquad (3.5\text{-}59)$$

$$b_1 = (\rho c)_r\, d_{ri}\, T_{ri1} \qquad (3.5\text{-}60)$$

and

$$b_2 = \left[(\rho c)_r d_{ro} + \rho_c c_c w_{fr}\right] T_{rol} + \ \overline{T}_c \Delta t\ H_{erc} \qquad (3.5\text{-}61)$$

The temperature difference between nodes, ΔT_r, is defined as

$$\Delta T_r = T_{ri2} - T_{ro2} \qquad (3.5\text{-}62)$$

In the second step, ΔT_r is preserved and E_{rc} is matched. The energy conservation equation is

$$(\rho c)_r d_{ri}(T_{ri2} - T_{ri1}) + \left[(\rho c)_r d_{ro} + \rho_c c_c w_{fr}\right](T_{ro2} - T_{ro1}) = -E_{rc} \qquad (3.5\text{-}63)$$

The solution to Eq. 3.5-62 and 3.5-63 is

$$T_{ri2} = \frac{\left\{-E_{rc} + (\rho c)_r d_{ri} T_{ri1} + \left[(\rho c)_r d_{ro} + p_c c_c w_{fr}\right] \cdot (T_{rol} + \Delta T_r)\right\}}{\left[(\rho c)_r (d_{ri} + d_{ro}) + p_c c_c w_{fr}\right]} \qquad (3.5\text{-}64)$$

and

$$T_{ro2} = T_{ri2} - \Delta T_r \qquad (3.5\text{-}65)$$

3.5.4 Gas Plenum Region

The basic equations used for the cladding and gas temperatures in the gas plenum region are Eq. 3.3-68 and the following equation:

$$\rho_e c_e A_{ep} \frac{dT_e(jp)}{dt} = 2\pi r_{erp} H_{erc} \left[\bar{T}_c(jc) - T_e(jp) \right] + 2\pi r_{brp} H_{eg} \left[T_g - T_e(jp) \right] \tag{3.5-66}$$

Since Eq. 3.3-68 links all of the cladding nodes in the gas plenum, a direct semi-implicit or implicit solution of Eqs. 3.3-68 and 3.5-66 would require a simultaneous solution for the gas temperature and all of the cladding node temperatures. Instead, the cladding temperatures are calculated first, using the gas temperature at the beginning of the time step. Then the gas temperature is calculated using the newly computed cladding temperatures.

Finite differencing of Eq. 3.5-66 gives

$$\rho_e c_e A_{ep} \frac{\left(T_{e2} - T_{e1}\right)}{\Delta t} = -2\pi r_{erp} \frac{E_{ec}}{\Delta t} + \pi r_{brp} H_{eg} \left(2T_{g1} - T_{e2} - T_{e1}\right) \tag{3.5-67}$$

where E_{ec} is calculated in the coolant routines in the same manner as indicated in Eq. 3.5-2, except that in the gas plenum only one radial node is used in the cladding, and R_{ehf} becomes

$$R_{ehf} = \frac{r_{erp} - r_{brp}}{2k_e} \tag{3.5-68}$$

The solution of Eq. 3.5-67 for T_{e2} gives

$$T_{e2} = \frac{\left\{\left(\rho_e c_e A_{ep} - \pi r_{brp} H_{eg} \Delta t\right) T_{el} - 2\pi r_{erp} E_{cc} + 2\pi r_{brp} H_{eg} \Delta t \, T_{g1}\right\}}{\left(\rho_e c_e A_{ep} + \pi r_{brp} H_{eg} \Delta t\right)} \tag{3.5-69}$$

In the second step, Eq. 3.3-70 is used with $\theta_1 = \theta_2 = 1/2$. The solution for T_{g2} is

$$T_{g2} = \frac{\left[(\rho c)_g A_g - \pi r_{brp} H_{eg} \Delta t\right] T_{g1} + \pi r_{brp} H_{eg} \Delta t \, s_1 / s_2}{(\rho c)_g A_g + \pi r_{brp} H_{eg} \Delta t} \tag{3.5-70}$$

where

$$s_1 = \sum_{jp} \left[T_{e1}(jp) + T_{e2}(jp)\right] \Delta z(jp) \tag{3.5-71}$$

and

$$s_2 = \sum_{jp} \Delta z(jp) \tag{3.5-72}$$

3.5.5 Coolant Temperatures in Liquid Slugs

Before the onset of coolant voiding, coolant temperatures are calculated at all node boundaries, as indicated in Figure 3.2–3. After the start of boiling, liquid coolant temperatures are calculated at all node boundaries outside vapor regions, as well as at moving nodes near the bubble interfaces. Two different types of calculations are made. Eulerian temperature calculations are made for fixed coolant nodes in the inlet and outlet liquid slugs. Lagrangian temperature calculations are made for the moving interface nodes and for any fixed nodes in liquid slugs between bubbles. There is also an option to use Lagrangian temperature calculations for all nodes, both fixed and moving.

The Eulerian calculation is probably more accurate for the fixed nodes. The main disadvantage of this method is that a sudden jump in inlet temperature can lead to a sawtooth temperature pattern, with the temperature high at one node, low at the next, and high again at the third node. The Lagrangian calculation does not exhibit this behavior. This sawtooth behavior is not unstable: the perturbation at any node is no larger than the jump in the inlet temperature, and the perturbations tend to die out in later time steps. Also, the coolant inlet and reentry temperature calculations described in Section 3.3.6 tend to eliminate sudden jumps in inlet and reentry temperatures.

3.5.5.1 Eulerian Temperature Calculation

The basic equation used in this calculation is again Eq. 3.3-5. The heat fluxes Q_{ec} and Q_{sc} are calculated as

$$Q_{ec} = \frac{\left(T_e - \overline{T}_c\right)}{R_{ec}} \frac{2\pi r\left(\mathrm{NE}'\right)}{A_c} \tag{3.5-73}$$

and

$$Q_{sc} = \frac{\left(T_{si} - \overline{T}_c\right)}{R_{sc}} \frac{S_{st}}{A_c} \tag{3.5-74}$$

where R_{ec} and R_{sc} are given by Eqs. 3.5-6 and 3.5-18, T_e is the average of T(NE) and T(NE′), and T_{si} is the inner structure node temperature. In reflector zones, T_e is replaced by the reflector outer node temperature; and in the gas plenum region, the one radial cladding node temperature is used. In the boiling module, the coolant temperatures are calculated before the cladding and structure temperatures are, so linear extrapolation in time is used to obtain values of T_e and T_{si} at the end of a time step.

A semi-implicit finite differencing of Eq. 3.3-5 gives

$$
\begin{aligned}
&\overline{p}(jc)\overline{c}_c(jc)A_c(jc)\frac{\left[T_{c2}(jc+1)+T_{c2}(jc)-T_{c1}(jc+1)+T_{c1}(jc)\right]}{2\Delta t}\\
&\quad+\overline{c}_c(jc)w_1\frac{\left[T_{c1}(jc+1)+T_{c1}(jc)\right]}{2\Delta z(jc)}+\overline{c}_c(jc)w_2\frac{\left[T_{c2}(jc+1)-T_{c2}(jc)\right]}{2\Delta z(jc)}\\
&\quad=Q_c(jc)A_c(jc)+\frac{k_5(jc)A_c(jc)}{4}\left\{\frac{2T_{e2}(jc)-T_{e2}(jc)-T_{c2}(jc+1)}{R_{ec2}(jc)}\right.\\
&\quad+\frac{2T_{e1}(jc)-T_{e1}(jc)-T_{c1}(jc+1)}{R_{ec1}(jc)}+\gamma_2(jc)\left[\frac{2T_{si2}(jc)-T_{c2}(jc)-T_{c2}(jc+1)}{R_{sc2}(jc)}\right.\\
&\quad\left.\left.+\frac{2T_{si1}(jc)-T_{c1}(jc)-T_{c1}(jc+1)}{R_{sc1}(jc)}\right]\right\}
\end{aligned}
\tag{3.5-75}
$$

where

$$
k_5(jc)=\begin{cases}\dfrac{2\pi r(\mathrm{NE}',jc)}{A_c(jc)} & \text{in the core and blankets}\\[2ex] \dfrac{s_{er}(kz)}{A_c(jc)} & \text{in a reflector region}\\[2ex] \dfrac{2\pi r_{erp}}{A_c(jc)} & \text{in the gas plenum region}\end{cases}
\tag{3.5-76a-c}
$$

and

$$
\gamma_2(jc)=\frac{S_{st}(jc)}{k_5(jc)A_c(jc)}
\tag{3.5-77}
$$

Solving for $T_{c2}(jc+1)$ gives

$$
\begin{aligned}
T_{c2}(jc+1) = \Bigg\{ & T_{c1}(jc+1)\left[\bar{\rho}_c(jc) - \frac{\Delta t w_1}{\Delta z(jc) A_c(jc)} - \frac{k_5(jc)\Delta t h_{br}(jc)}{2\bar{c}_c(jc)}\right] \\
& + T_{c2}(jc)\left[-\rho(jc) + \frac{\Delta t w_2}{\Delta z(jc) A_c(jc)} - \frac{k_5(jc)\Delta t}{2\bar{c}_c(jc)} h_{b2}(jc)\right] \\
& + T_{c1}(jc)\left[\rho_c(jc) + \frac{\Delta t w_1}{\Delta z(jc) A_c(jc)} - \frac{k_5(jc)\Delta t}{2\bar{c}_c(jc)} h_{b1}(jc)\right] \\
& + \frac{4\Delta t}{2\bar{c}_c(jc)} k_5(jc)\, \phi_1(jc) + \frac{2\Delta t Q_c(jc)}{\bar{c}_c(jc)}\Bigg\} / \Bigg\{\bar{\rho}_c(jc) \\
& + \frac{\Delta t w_2}{\Delta z(jc) A_c(jc)} + \frac{k_5(jc)\Delta t h_{b2}(jc)}{2\bar{c}_c(jc)}\Bigg\}
\end{aligned}
\tag{3.5-78}
$$

with

$$
h_{b1}(jc) = \frac{1}{R_{ec1}(jc)} + \frac{\gamma_2(jc)}{R_{sc1}(jc)} \tag{3.5-79}
$$

$$
h_{b2}(jc) = \frac{1}{R_{ec2}(jc)} + \frac{\gamma_2(jc)}{R_{sc2}(jc)} \tag{3.5-80}
$$

and

$$
\begin{aligned}
\phi_1(jc) = 1/2 \Bigg\{ & \frac{T_{e2}(jc)}{R_{ec2}(jc)} + \frac{T_{e1}(jc)}{R_{ec1}(jc)} + \gamma_2(jc)\Bigg[\frac{T_{si2}(jc)}{R_{sc2}(jc)} \\
& + \frac{T_{si1}(jc)}{R_{sc1}(jc)}\Bigg]\Bigg\}
\end{aligned}
\tag{3.5-81}
$$

If the inlet flow is positive, then the coolant temperature at node 1 is determined by the inlet temperature. Equation 3.5-78 is then used to march up the channel through the inlet liquid slug, with $T_{c2}(jc+1)$ being computed after $T_{c2}(jc)$. Similarly, if the flow in the upper liquid slug is downward, then the assembly outlet reentry temperature determines the coolant temperature at the last coolant node. Then an equation similar to Eq. 3.5-79 is used to march down through the upper liquid slug, with $T_{c2}(jc)$ being computed after $T_{c2}(jc+1)$.

The Eulerian calculations always go from node to node in the direction of flow. An inlet slug expelling downward and an outlet liquid slug going upward are special cases, since in these cases the calculation starts at a liquid vapor interface rather than an end of the subassembly. The interface liquid temperatures are first calculated using the Lagrangian treatment described below. Then Eq. 3.5-78 or the equivalent equation for downward flow is used to calculate the temperatures at the fixed nodes within the liquid slug. For the first fixed node near the interface, some of the terms in Eq. 3.5-78 are modified. The moving interface node is treated as node *jc*. The interface temperature is used for $T_{c2}\left(jc\right)$, and an interpolated value is used for $T_{c1}\left(jc\right)$. The distance from the fixed node to the interface at the end of the step is used for $\Delta z\left(jc\right)$. Interpolated interface cladding and structure temperatures are used in calculating ϕ_1 for the interface node.

3.5.5.2 Lagrangian Calculations for Interface Temperatures

For every liquid-vapor interface a vapor temperature is calculated at or very near the interface. The liquid temperature right at the interface would be close to the vapor temperature, but there can be strong axial temperature gradients in the liquid near the interface. These strong axial gradients would only extend a short distance into the liquid. The heat flow through the interface into a small vapor bubble is accounted for, as described in Chapter 12; but since only one liquid temperature node is used near the interface, the axial temperature distribution neat the interface is not represented. A liquid temperature is calculated for each interface, but axial conduction is neglected in this calculation. Thus, the liquid interface temperature can be considered as either the interface temperature that would occur if there were no axial conduction or the temperature a short distance from the interface where axial conduction is negligible.

A Lagrangian formulation, moving with the liquid, is used for the interface temperature calculation. The basic equation used is

$$\rho c \frac{DT_c}{Dt} = Q_c + Q_{ec} + Q_{sc} \tag{3.5-82}$$

where the Lagrangian total derivative is used. After finite differencing this equation gives

$$\begin{aligned}
&\rho_{ci} c_{ci} \frac{T_{\ell i2}(k,L) - T_{\ell i1}(k,L)}{\Delta t} \\
&\quad = Q_c(jc) + \frac{k_5(jc)}{2} \Bigg\{ \frac{T_{ei1}(k,L) - T_{\ell i1}(k,L)}{R_{eci1}(k,L)} \\
&\qquad + \frac{T_{ei2}(k,L) - T_{\ell i2}(k,L)}{R_{sci1}(k,L)} \Bigg\} \Bigg] \\
&\qquad + \gamma_2(jc) \Bigg\{ \frac{T_{si1}(k,L) - T_{\ell i1}(k,L)}{R_{sci1}(k,L)} + \frac{T_{si2}(k,L) - T_{\ell i2}(k,L)}{R_{sci2}(k,L)} \Bigg] \Bigg\}
\end{aligned} \tag{3.5-83}$$

or

$$\begin{aligned}
T_{\ell i2}(k,L) = \Big\{ & T_{\ell i1}(k,L) \left[1 - d_1 h_{bi1}(k,L) \right] + d_1 \big[\phi_{1i}(k,L) \\
& + 2\, Q_c(jc) / k_5(jc) \big] \Big\} / \left[1 + d_1 h_{bi2}(k,L) \right]
\end{aligned} \tag{3.5-84}$$

where

$$d_1 = \frac{k_5(jc) \Delta t}{2 \rho_{ci}\, c_{ci}} \tag{3.5-85}$$

$$h_{bi1}(k,L) = \frac{1}{R_{eci1}(k,L)} + \frac{\gamma_2(jc)}{R_{sci1}(k,L)} \tag{3.5-86}$$

$$h_{bi2}(k,L) = \frac{1}{R_{eci2}(k,L)} + \frac{\gamma_2(jc)}{R_{sci2}(k,L)} \tag{3.5-87}$$

k = bubble number

L = 1 for lower bubble interface, 2 for upper bubble interface

jc = coolant node containing the interface

$$\theta_{1i}(k,L) = \frac{T_{ei2}(k,L)}{R_{eci2}(k,L)} + \frac{T_{ei1}(k,L)}{R_{eci1}(k,L)} + \gamma_2(jc) \left[\frac{T_{si2}(k,L)}{R_{sci2}(k,L)} + \frac{T_{si1}(k,L)}{R_{sci1}(k,L)} \right] \tag{3.5-88}$$

T_{ei2}, T_{si2}=cladding and structure interface temperatures at the end of the step, extrapolated in tir

T_{ei1}, T_{si1} = same at the beginning of the time step.

R_{eci2}, R_{sci2} = values of R_{ec} and R_{sc} at the interface at the end of the time step.

R_{eci1}, R_{sci1} = same at the beginning of the time step.

3.5.5.3 Lagrangian Calculation for Fixed Nodes

The Lagrangian temperature calculations for fixed coolant nodes are similar to those for interface nodes. The fluid particle that ends up at coolant node *jc* at the end of a time step is considered. During the time step, the particle travelled a distance

$$\Delta z' = \frac{(w_1 + w_2)\Delta t}{2\rho_c(jc)A_c(jj)} \tag{3.5-89}$$

where

$$jj = \begin{cases} jc & \text{if } w_1 + w_2 < 0 \\ jc-1 & \text{otherwise} \end{cases} \tag{3.5-90}$$

At the beginning of the time step, the particle was at z', given by

$$z' = z_c(jc) - \Delta z' \tag{3.5-91}$$

The coolant temperature, T'_{c1}, at z' at the beginning of the step is obtained by linear interpolation between the nodes on either side of z'. Also, the cladding and structure temperatures, T'_{e1} and T'_{s1}, at z' at the beginning of the step are obtained by linear interpolation. The cladding and structure temperatures, T'_{e2} and T'_{s2}, at $z_c(jc)$ at the end of the time step are also obtained by linear interpolation between the cladding and structure nodes.

The result of finite differencing of Eq. 3.5-82 for the particle at node *jc* is

$$T_{c2}(jc) = \left\{T'_{c1}\left[1 - d'_1 h_{b1}(jj)\right] + d'_1\left[\theta'_1 + 2Q_c(jj)/k_5(jj)\right]\right\} / \left[1 + d'_1 h_{b2}(jj)\right] \tag{3.5-92}$$

where

$$d'_1 = \frac{k_5(jj)\Delta t}{\rho_c(jc)\bar{c}_c(jj)} \tag{3.5-93}$$

and

$$\phi_1' = \frac{T_{e2}'}{R_{ec2}(jj)} + \frac{T_{e1}'}{R_{ec1}(jj)} + \gamma_2(jj)\left[\frac{T_{s2}'}{R_{sc2}(jj)} + \frac{T_{s1}'}{R_{sc1}(jj)}\right] \qquad (3.5\text{-}94)$$

Again, h_{b1} and h_{b2} are given by Eqs. 3.5-79 and 3.5-80.

3.6 Fuel-Cladding Bond Gap Conductance

A number of gap-size-dependent bond gap correlations are available in SAS4A/SASSYS-1. The bond gap conductance depends on two main factors: the gap size or the contact pressure between fuel and cladding after the gap has closed, and the correlation for bond gap conductance as a function of gap size or contact pressure. Since small differences in differential expansion between fuel and cladding can make the difference between an open gap and a closed gap, and since gap conductance correlations are strongly dependent on gap size, the models used for fuel and cladding thermal expansion and swelling might have a much larger impact on computed bond gap conductances than the choice of the particular correlation used for bond gap conductance as function of gap size.

There are a number of options for computing the gap size. One common option for oxide fuel is to use DEFORM-IV to compute the steady-state and transient dimensions. Chapter 8 describes DEFORM-IV and the bond gap conductance correlations that can be used with it. A second option would be to use DEFORM-IV for the steady-state but not for the transient. In this case, the gap size and gap conductance determined in the steady-state calculations would be constant during the transient. A third option is not to use DEFORM-IV at all. In this case the gap size is constant, based on the user-specified pin dimensions, and the bond gap conductance is constant. The fourth option is to use a simple thermal expansion model for the transient bond gap size. For a metal fuel, the DEFORM-5 model described in Chapter 9 can be used to obtain the bond gap conductance.

The simple thermal expansion model applies only to the transient calculation. It can be used either with or without the DEFORM-IV steady-state calculations, but it cannot be used with the transient DEFORM-IV. In this model, it is assumed that the gap size, Δr_g, is determined by simple thermal expansion of the fuel and cladding from their steady-state dimensions:

$$\begin{aligned}\Delta r_g = {} & r_o(\mathrm{NE}) - r_o(\mathrm{NR}) \\ & + \frac{\left[r_o(\mathrm{NE}) + r_o(\mathrm{NE'})\right]}{2}\alpha_e\left[T(\mathrm{NE}) - T_o(\mathrm{NE})\right] \\ & - r_o(\mathrm{NR})\alpha_f\left(\bar{T}_f - \bar{T}_{fo}\right)\end{aligned} \qquad (3.6\text{-}1)$$

where

r_o = steady-state radii,

$\overline{T}_o$ = steady-state temperature,

$\overline{T}_f$ = average fuel temperature, mass-weighted average,

$\overline{T}_{fo}$ = average steady-state fuel temperature,

α_e = cladding thermal expansion coefficient, and

α_f = fuel thermal expansion coefficient

The bond gap conductance then has the form

$$h_b = \frac{\overline{h}_b}{\Delta r_g} \tag{3.6-2}$$

or

$$h_b = A_g + \frac{1}{B_g + \dfrac{\Delta r_g + C_g}{\overline{h}_b}} \tag{3.6-3}$$

depending on the correlation chosen. In these correlations $\overline{h}_b$, A_g, B_g, and C_g are user-supplied correlation coefficients. For either correlation, the bond gap conductance is also constrained to lie between user-supplied minimum and maximum values; so if a value outside this range is calculated using Eq. 3.6-2 or 3.6-3, the minimum or the maximum value is used instead.

3.7 Fuel Pin Heat-transfer After Pin Disruption or Relocation of Fuel or Cladding

The preceding sections describe fuel pin heat transfer with intact fuel pins and no relocation of fuel of cladding. After pin disruption or the relocation of fuel or cladding, the heat transfer calculations are modified. The modifications after the start of in-pin fuel relocation in the PINACLE module are described in Chapter 15. The modification after the start of cladding melting and relocation in the CLAP module are described in Chapter 13. The modifications after pin disruption are described in Section 3.7.1 below and in Chapters 14 and 16.

3.7.1 Fuel-pin Heat Transfer After Pin Disruption in PLUTO2 or LEVITATE

When PLUTO2 or LEVITATE is active, the PLHTR subroutine calculates the heat conduction in all solid fuel (including axial blankets) and also in the cladding which is in contact with the lower and upper coolant slugs. The heat conduction calculation of the cladding in the interaction region, which is between the lower and upper coolant slug, (see Fig. 14.1-4) is performed in the PLUTO2 or LEVITATE modules (see Sections 14.5.2 and 16.5.7) using a shorter time step than the PLHTR calculation.

Along the interaction region, the heat flow rate from the cladding inner surface to the fuel outer surface is calculated in PLUTO2 or LEVITATE assuming a constant gap conductance of the value in existence at the time of initiation. The PLUTO2 or LEVITATE calculated heat flow rates are integrated over a PLHTR time step in order to provide PLHTR with the total heat added during a heat-transfer time step. Outside the interacting region the heat flow rate between the liquid sodium flow and the cladding outer surface is calculated in the PLCOOL subroutine of PLUTO2. The latter subroutine mainly determines the liquid sodium temperatures in the coolant slugs. It uses the same time step as the PLHTR subroutine.

The temperature calculations in the molten fuel cavity in the pins are performed by PLUTO2 or LEVITATE and are part of the in-pin fuel motion calculation in these modules. The heat flow rates from each molten cavity node to the surrounding sold fuel are also calculated in PLUTO2 or LEVITATE. Since the time steps of the latter modules are shorter than the PLHTR time steps, the PLUTO2 or LEVITATE heat flow rates have to be integrated over the whole heat-transfer time step, because the total heat transferred to the cavity wall during a heat-transfer time step is required by PLHTR.

The initial configuration of the molten pin cavity at the time of pin failure is determined in the PLUTO2 and LEVITATE initialization routines PLINPT and PLSET (see Section 14.2.2). PLINPT initializes the integer array IXJ(K) for each axial node K with the index of the innermost radial fuel node whose melt fraction has not yet exceeded the input value FNMELT. This array IZJ(K) thus determines the initial molten cavity configuration.

When PLUTO2 or LEDVITATE are active, additional fuel can melt into the cavity and thereby enlarge it. The integer array IZJ(K) is updated for each axial node K whenever another radial node exceeds the input value FNMELT. However, such a radial node is only gradually added to the molten cavity (see Eqs. 14.2-10 to 14.2-12). The heat conduction calculation in PLHTR includes this partial node.

Figure 3.7–1 shows the radial grid used in PLHTR. The heat conduction calculation covers the radial region from I = INDBOT to I = NTHELP. The latter can be the outermost radial fuel node (for axial nodes in the interaction region) or the outer cladding node (for axial nodes outside the interaction region). Temperatures and heat sources are defined at the midpoints of the grid in Figure 3.7–1.

PLHTR is a modified version of the TSHTRV subroutine that calculates the fuel-pin heat transfer during coolant boiling and the reader is referred to Section 3.5 for a detailed presentation of the equations. One of the main differences is that the conduction calculation is done only in the solid fuel region and in the cladding outside the interaction region. This is achieved by having the calculational loops go from I = INDBOT to I = NTHELP (see Fig. 14.2-1) and by adding or subtracting the integrated heat flux to or from the solid fuel nodes at the boundaries in the form of heat sources or sinks, respectively. The integrated heat flux at the outer pin boundary is obtained from

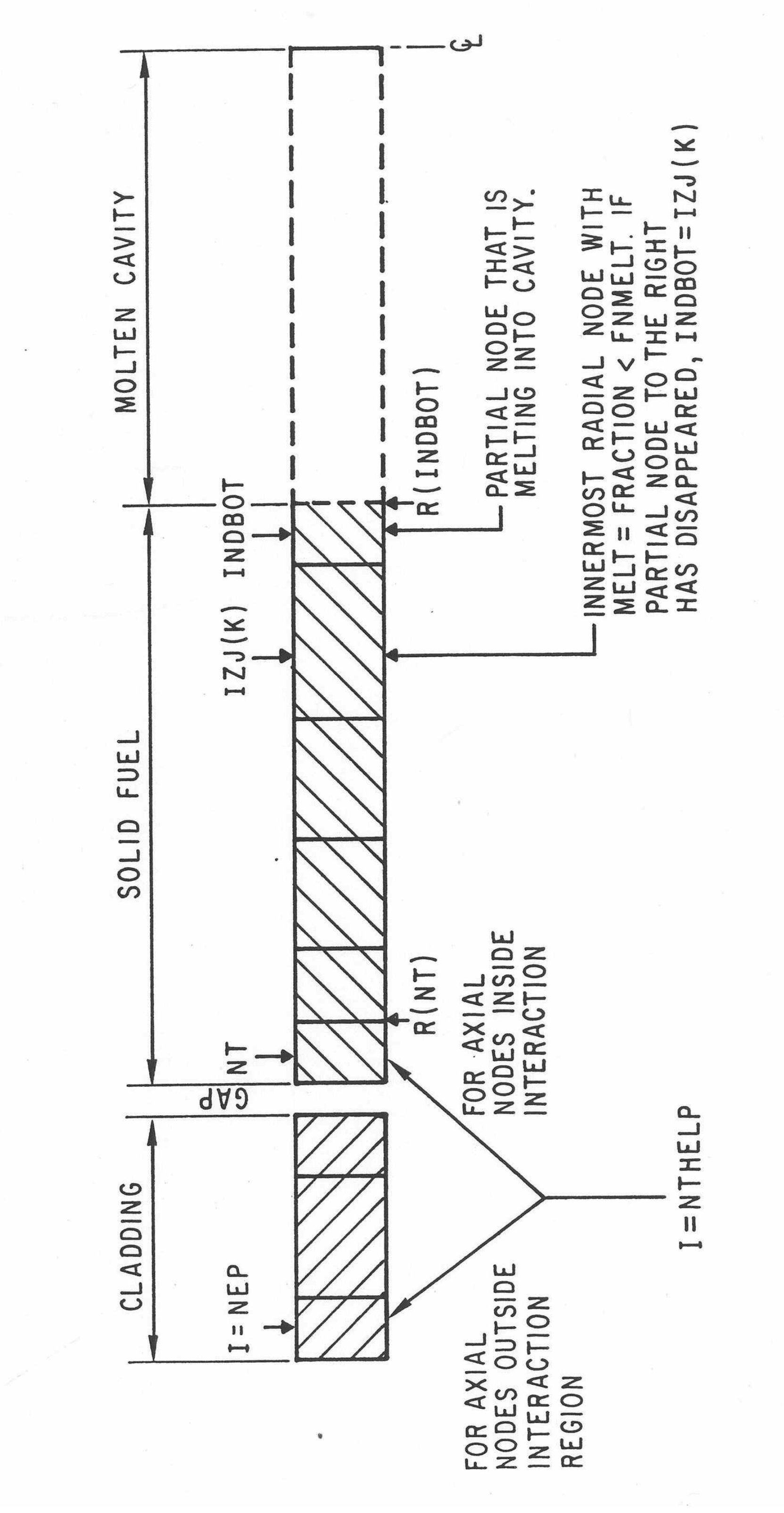

MOLTEN CAVITY
SOLID FUEL
GAP
CLADDING
℄
INDBOT
IZJ(K)
NT
I = NEP
R(INDBOT)
PARTIAL NODE THAT IS MELTING INTO CAVITY.
INNERMOST RADIAL NODE WITH MELT = FRACTION < FNMELT. IF PARTIAL NODE TO THE RIGHT HAS DISAPPEARED, INDBOT = IZJ(K)
R(NT)
FOR AXIAL NODES INSIDE INTERACTION
I = NTHELP
FOR AXIAL NODES OUTSIDE INTERACTION REGION

Figure 3.7–1: Radial Grid for the PLHTR Calculation

the array HFPICL, calculated in the PLUTO routine PLMISE or in the LEVITATE routine LESDEN. The integrated heat flux at the cavity boundary is obtained from the array HFCAWA; calculated in the PLUTO routine PC1PIN or the LVITATE routine LE1PIN. Moreover, the heat conduction terms at the fuel surface, which are necessary in the TSHTRV calculation, had to be set to zero. This meant setting the term BETA (NTHELP) in Eq. 3.3-48 to zero and ignoring all equations related to the cladding in Eq. 3.3-48.

3.8 Heat-transfer Time Step Control

Each channel uses its own separate heat-transfer time step size, so that channels in which temperatures are changing rapidly or in which boiling is occurring can use small heat-transfer time steps while other channels use larger steps.

After each heat-transfer time step, the size of the next step is determined, based on number of criteria. Most of these criteria are based on user-supplied values for maximum time step sizes and maximum temperature changes per step. The time step used in the smallest of the various criteria. If not other factor is more limiting, then the next heat-transfer step size is set equal to the initial main time-step size. This is usually in the range of .05-1.0 s. After the onset of boiling in a channel, a maximum boiling heat-transfer time step size, typically .01-.02 s, is used. In addition, a heat-transfer time step cannot go past the end of a main time step. Also, if the channel has not started boiling yet, then an attempt is made to end a heat-transfer time step right at, or very close to, the time when the first bubble is formed. For this purpose, the pre-boiling coolant routines make an estimate of the boiling time at the end of each coolant time step. The boiling time estimate is based on linear extrapolations in time for the coolant temperature and the saturation temperature at each axial node. The other criteria are based on the rate of change of the temperatures. The user supplies values of the maximum change per time step for the fuel and clad temperatures. Typical values are in the range of 30-50K. The rate of change of the fuel center-line temperature, the fuel surface temperature, and the clad mid-point temperature at each axial node are used to determine maximum time step sizes. After the minimum of the various time-step criteria has been found, the time step size is rounded to eight decimal places to minimize differences in results caused by different round-off errors on different computers. These criteria are summarized in Table 3.8-1.

Table 3.8-1: Criteria for Heat-Transfer Time Step Sizes

1.	Initial time step size
2.	Maximum boiling time step size, if boiling
3.	End of a main time step
4.	Initiation of boiling
5.	Maximum fuel center-line temperature change
6.	Maximum fuel surface temperature change
7.	Maximum clad mid-point temperature change

3.9 Steady-State and Single-Phase Transient Hydraulics

3.9.1 Introduction

The core assembly hydraulics treatment in SAS4A/SASSYS-1 includes the calculation of coolant flow rates and pressure distributions within each core channel. Coolant flow rates and pressures are calculated in a number of different places in the code. They are used in the steady-state thermal hydraulics initialization; the pre-voiding thermal hydraulics module calculates coolant flows and pressures; and after the onset of voiding, they are calculated in the boiling module. This section describes only the steady-state and pre-voiding calculations. Chapter 12 describes the hydraulics calculations after the onset of voiding. Chapters 14 and 16 describe the hydraulics calculations after pin disruption.

The coolant flow provides the heat removal from the fuel pins, and so the main reason for calculating coolant flow rates before the onset of voiding is to provide information for the fuel-pin temperature calculations. Core channel flow rates are also needed for the PRIMAR-4 primary loop thermal hydraulics calculations. The pressure distribution within a channel is calculated mainly to obtain the pressure-dependent coolant saturation temperature used to determine the onset of boiling.

For the steady-state initialization the user specifies the initial coolant flow rate for each channel and the outlet plenum pressure. The code then calculates the pressure distribution in each channel, starting from the outlet and working down to the inlet. The channel with the largest steady-state pressure drop is used to determine the steady-state inlet plenum pressure; and the inlet orifice coefficients in all other channels are adjusted so that all channels have the same total steady-state pressure drop.

In the transient hydraulic calculations, both the coolant flow rates and the pressure distributions are calculated for each channel. Pressure boundary conditions are used for the transient coolant flow rate calculations. The driving pressures for these calculations are the inlet and outlet coolant plenum pressures supplied by the PRIMAR-4 module. Since all channels use the same inlet and outlet pressures, flow redistribution between channels is automatically accounted for as temperatures and flows change. Extrapolated coolant temperatures are used to evaluate coolant properties in the transient hydraulics calculations, since coolant flow rates for a time step are calculated before temperatures are calculated.

Figure 3.1–2 shows the logic flow in subroutine TSCL0, the driver for pre-voiding core channel thermal hydraulics. As indicated previously, a multi-level time step approach is used for the transient calculations in SAS4A/SASSYS-1. In general, the routines in a transient module start with all quantities knows at the beginning of a time step, and a pass through a module results in calculating the values of relevant parameters of the module for one channel at the end of the time step. Different time-step sizes can be used for different phenomena. In particular, the pre-voiding coolant hydraulics time step can be smaller than, but not larger than, the heat-transfer time step or the PRIMAR time step. One pass through TSCL0 calculates one coolant time step for one channel. If the coolant time step finishes a heat-transfer step, then the temperature calculations for the heat-transfer step are also carried out.

Figure 3.9–1 shows the logic flow in subroutine TSCNV1. This subroutine and the routines called by it carry out the pre-viding coolant flow and pressure calculations. Subroutine TSCNV1 also checks on whether to switch to the boiling model.

As indicated in Figure 3.9–2, the pre-voiding transient hydraulics module interacts with PRIMAR and TSHTRN. PRIMAR supplies the inlet and outlet pressures that drive the coolant flows in the core channels. In return, TSCL0 supplies PRIMAR with the current channel flow rates, as well as hydraulic parameters to use in estimating future flow rates for the next PRIMAR time step. TSCL0 supplies the coolant flow rates to TSHTRN, and TSHTRN computes the coolant temperatures used by TSCL0.

3.9.2 Basic Equations

Before the onset of voiding, the coolant is treated as incompressible, and the basic equation used for liquid coolant flow in non-voided channels is

$$\frac{1}{A_c}\frac{\partial w}{\partial t}+\frac{\partial p}{\partial z}+\frac{1}{A_c}\frac{\partial(wv)}{\partial z}= -\left(\frac{\partial p}{\partial z}\right)_{fr}-\left(\frac{\partial p}{\partial z}\right)_{K}-\rho_c g \qquad (3.9\text{-}1)$$

with

w = $w(t)$-independent of z

w = coolant flow rate $(kg/s) = \rho_c\ v\ A_c$

Ac = coolant flow area

p = pressure

ρ_c = density

v = coolant velocity (m/s)

z = axial position

t = time

$\left(\frac{\partial p}{\partial z}\right)_{fr}$ = friction pressure drop

$$\left(\frac{\partial p}{\partial z}\right)_{fr}=\mathrm{f}\frac{\mathrm{w}^2}{2\rho_{\mathrm{c}}A_c^2D_h} \qquad (3.9\text{-}2)$$

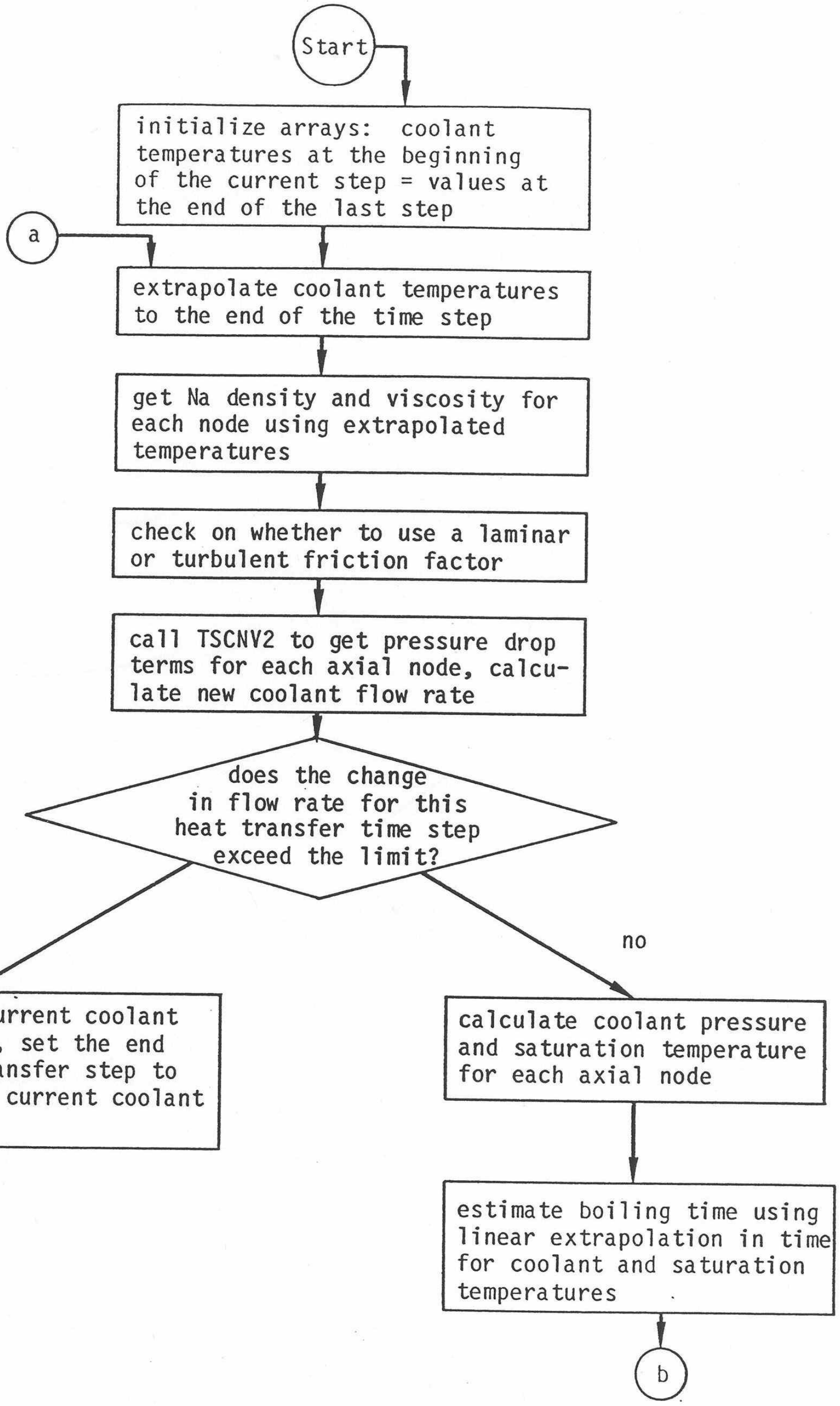

Start
initialize arrays: coolant temperatures at the beginning of the current step = values at the end of the last step
a
extrapolate coolant temperatures to the end of the time step
get Na density and viscosity for each node using extrapolated temperatures
check on whether to use a laminar or turbulent friction factor
call TSCNV2 to get pressure drop terms for each axial node, calculate new coolant flow rate
does the change in flow rate for this heat transfer time step exceed the limit?
yes
no
cut back the current coolant time step size, set the end of the heat transfer step to the end of the current coolant step
a
calculate coolant pressure and saturation temperature for each axial node
estimate boiling time using linear extrapolation in time for coolant and saturation temperatures
b

Figure 3.9–1: Subroutine TSCNV1, Pre-Boiling Coolant Flow Rates and Pressure Distribution

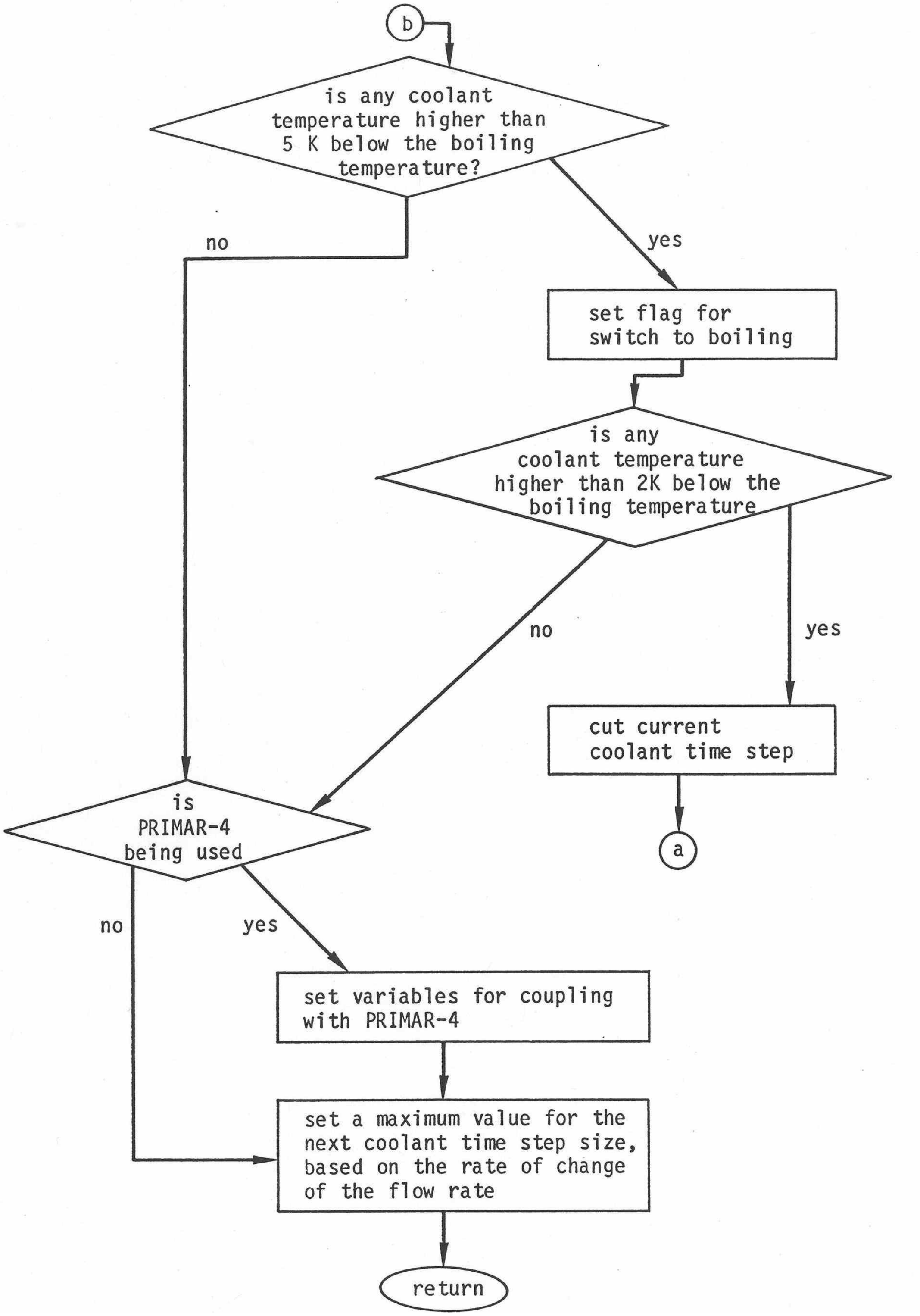

b
is any coolant temperature higher than 5 K below the boiling temperature?
no
yes
set flag for switch to boiling
is any coolant temperature higher than 2K below the boiling temperature
no
yes
cut current coolant time step
a
is PRIMAR-4 being used
no
yes
set variables for coupling with PRIMAR-4
set a maximum value for the next coolant time step size, based on the rate of change of the flow rate
return

Figure 3.9-1: Subroutine TSCNV1, Pre-Boiling Coolant Flow Rates and Pressure Distribution (Cont'd)

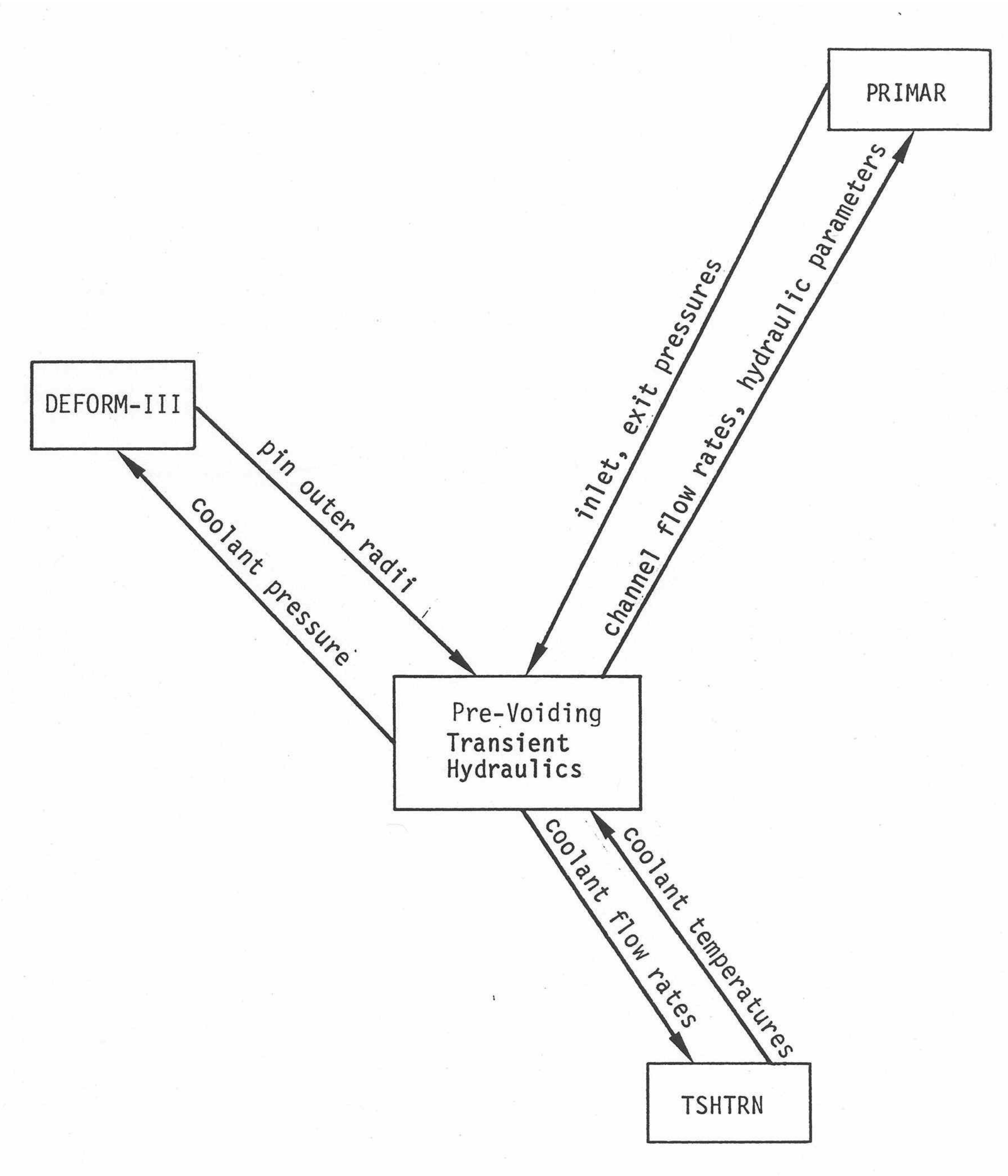


Figure 3.9–2: Interactions Between Pre-Voiding Transient Hydraulics and Other Modules

f= friction factor, approximated either as

$$f = \begin{cases} A_{fr}(\mathrm{Re})^{b_{fr}} \text{ if Re} \geq R_{eL} \\ A_{fL}/_{\mathrm{Re}} \text{ if Re} < R_{eL} \end{cases} \tag{3.9-3a}$$

or as

$$f = A_{fr}(\mathrm{Re})^{b_{fr}} + A_{fL} / \mathrm{Re} \tag{3.9-3a}$$

where A_{fr} b_{fr}, and A_{fL} are user-supplied correlation coefficients

$$\mathrm{Re} = \frac{D_h w}{\mu A_c} = \text{Reynolds number} \tag{3.9-4}$$

R_{eL} = Reynolds number for the transition from turbulent to laminar flow

μ = viscosity

D_h = hydraulic diameter

$\left(\frac{\partial p}{\partial z}\right)_K$ = orifice pressure drop

g = acceleration of gravity

ρ = density

The axial node structure shown in Figure 3.2–3 is used for the coolant. Figure 3.9–3 shows which variables are defined at node boundaries and which are averages or integrals over the length of a node. With this mesh structure, integrating Eq. 3.9-1 over the length of the channel gives

$$I_1 \frac{\partial w}{\partial t} + p_t - p_b + w^2 I_2 + A_{fr} w\, |w|^{I+b_{fr}} I_3 + w|w|\, I_4 + g\, I_5 = 0 \tag{3.9-5}$$

where

$$I_1 = \int \frac{dz}{A_c} = \left(\frac{\Delta z_i}{A}\right)_b + \left(\frac{\Delta z_i}{A}\right)_t + \sum_{\mathrm{JC}=1}^{\mathrm{MZC}-1} X_{I1}(\mathrm{JC}) \tag{3.9-6}$$

and

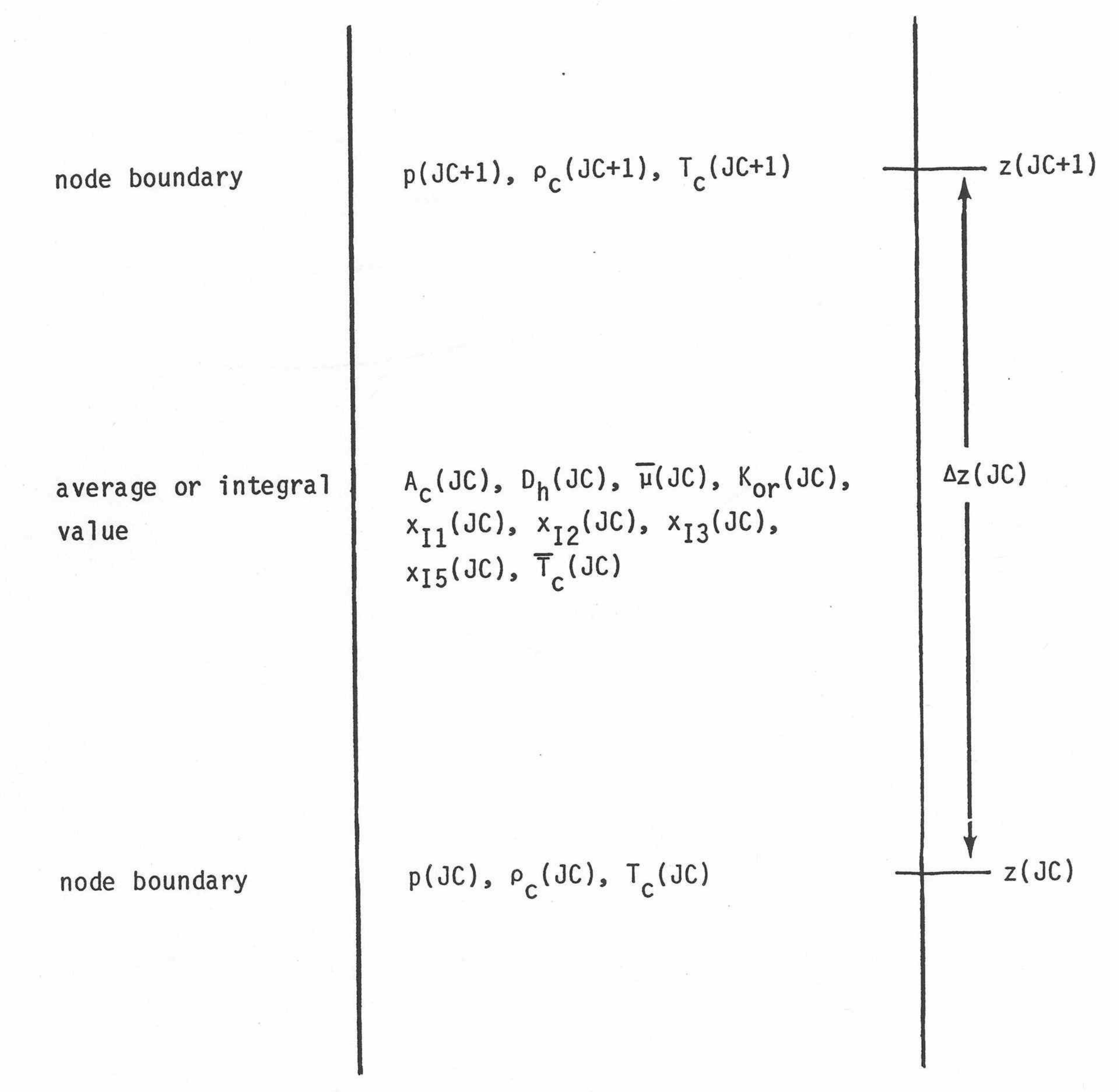

node boundary
$p(JC+1)$, $\rho_c(JC+1)$, $T_c(JC+1)$
$z(JC+1)$
average or integral value
$A_c(JC)$, $D_h(JC)$, $\overline{\mu}(JC)$, $K_{or}(JC)$,
$x_{I1}(JC)$, $x_{I2}(JC)$, $x_{I3}(JC)$,
$x_{I5}(JC)$, $\overline{T}_c(JC)$
$\Delta z(JC)$
node boundary
$p(JC)$, $\rho_c(JC)$, $T_c(JC)$
$z(JC)$

Figure 3.9–3: Interactions Between Pre-Voiding Transient Hydraulics and Other Modules

$$X_{I1}\left(\text{JC}\right)=\frac{\Delta z\left(\text{JC}\right)}{A_c\left(\text{JC}\right)} \tag{3.9-7}$$

$\left(\frac{\Delta z_i}{A}\right)_b$ and $\left(\frac{\Delta z_i}{A}\right)_t$ are effective inertial terms at the bottom and top of the subassembly.

$$I_2=\sum_{JC=1}^{\text{MZC-1}} X_{I2}\left(\text{JC}\right) \tag{3.9-8}$$

$$X_{I2}\left(\text{JC}\right)=\frac{1}{A_c\left(\text{JC}\right)^2}\left[\frac{1}{\rho_c\left(\text{JC+1}\right)}-\frac{1}{\rho_c\left(\text{JC}\right)}\right] \tag{3.9-9}$$

$$I_3=\int\frac{1}{2\rho_c A_c^2 D_h}\left[\frac{D_h}{\mu A_c}\right]^{b_{fr}}dz=\sum_{JC=1}^{\text{MXC}-1} X_{I3}\left(\text{JC}\right) \tag{3.9-10}$$

$$X_{13}(\text{JC})=\frac{\Delta z(\text{JC})}{[\rho_c(\text{JC})+\ \rho_c(\text{JC}+1)]A_c(\text{JC})^2 D_h(\text{JC})}\left[\frac{D_h(\text{JC})}{\bar{\mu}(\text{JC})A_o(\text{JC})}\right]^{b_{fr}} \tag{3.9-11}$$

$$I_4=\sum_{\text{JC}=1}^{\text{MZC}-1} K_{or}\left(\text{JC}\right) \tag{3.9-12}$$

$$I_5=\int\rho_c\ dz=\sum_{\text{JC}=1}^{\text{MXC}-1} X_{I5}\left(\text{JC}\right) \tag{3.9-13}$$

$$X_{I5}\left(\text{JC}\right)=.5\left[\rho_c\left(\text{JC}\right)+\rho_c\left(\text{JC}+1\right)\right]\Delta z\left(\text{JC}\right) \tag{3.9-14}$$

p_b = pressure at bottom of channel, and p_t – pressure at top of channel.

3.9.3 Flow Orifices

As mentioned in the section on the SAS4A/SASSYS-1 channel treatment in Section 3.2, the channel is divided axially into a number of zones. One zone represents the fuel-pin region. Other zones represent regions above and below the fuel pins. At the bottom of each zone, a flow direction dependent orifice coefficient can be specified by the user. Another orifice coefficient can be used at the top of the upper zone. These orifices can represent orifice blocks, subassembly inlet and exit losses, and the pin support grid at the bottom of the pins. For each orifice, the user supplies one orifice coefficient for positive flow and another for negative flow. In addition, in the fuel-pin region, orifice coefficients representing equally spaced grid spacers can be specified. These orifice coefficients are used as the values of K_{or}(JC) in Eq. 3.9-12 for the appropriate axial nodes. The effect of each orifice is spread evenly over an axial node, rather than concentrated at a point.

3.9.4 Finite Difference Equations – Coolant Flow Rates

If $w_1 = w(t), w_2 = w(t + \Delta t), \Delta w = w_2 - w_1$, then we approximate Δw as

$$\Delta w = \left[\theta_1 \frac{\partial w}{\partial t}\bigg|_t + \theta_2 \frac{\partial w}{\partial t}\bigg|_{t+\Delta t}\right] \Delta t \qquad (3.9\text{-}15)$$

and

$$\theta_1 + \theta_2 = 1.0 \qquad (3.9\text{-}16)$$

in which θ_2 is the degree of implicitness. For small time steps a semi-implicit calculation is used, with $\theta_1 = \theta_2 = .5$. For a fully implicit calculation for large time steps, $\theta_1 = 0$ and $\theta_2 = 1.0$. The value used for θ_2 is

$$\theta_2 = \frac{6.12992 + 2.66054x + x^2}{2x6.12992 + 3.56284x + x^2} \qquad (3.9\text{-}17)$$

where

$$x = \Delta t/\tau \qquad (3.9\text{-}18)$$

and τ is the time constant for flow rate changes given by

$$\tau = I_1/d_1 \qquad (3.9\text{-}19)$$

The definition of d_1 is given by Eq. 3.9-25 below. This correlation for θ_2 is discussed in Appendix 3.1. The additional approximations are made that

$$(w_1 + \Delta w)^2 \simeq w_1^2 + 2\Delta w\, w_1 \qquad (3.9\text{-}20)$$

$$(w_1 + \Delta w)\,|w_1 + \Delta w| \simeq w_1\,|w_1| + 2|w_1|\Delta w \qquad (3.9\text{-}21)$$

$$(w_1 + \Delta w)|w_1 + \Delta w|^{1+b_{fr}} \simeq w_1|w_1|^{1+b_{fr}} + \left(2 + b_{fr}\right)|w_1|^{1+b_{fr}}\,\Delta w \qquad (3.9\text{-}22)$$

These equations are combined to give

$$\begin{aligned} I_1 \frac{\Delta w}{\Delta t} &+ \theta_1(p_{t1} - p_{b1}) + \theta_2(p_{t2} - p_{b2}) + I_2(w_1^2 + 2\theta_2 w_1 \Delta w) \\ &+ A_{fr} I_3 \left[w_1 |w_1|^{1+b_{fr}} + \theta_2 \left(2 + b_{fr}\right) |w_1|^{1+b_{fr}} \Delta w \right] \\ &+ I_4 (w_1 |w_1| + 2\theta_2 |w_1| \Delta w) + g I_5 = 0 \end{aligned} \tag{3.9-23}$$

where p_{t1} and p_{t2} are the pressures at the top of the subassembly at the beginning and end of the time step. Similarly, p_{b1} and p_{b2} are the pressures at the bottom of the subassembly at the beginning and end of the time step, respectively.

Solving for the coolant flow rate change, Δw, gives

$$\begin{aligned} \Delta w = \Delta t \Big[& \theta_1 \left(p_{b1} - p_{t1}\right) + \theta_2 \left(p_{b2} - p_{t2}\right) - I_2 w_1^2 - A_{fr} I_3 w_1 \left|w_1\right|^{1+b_{fr}} \\ & - I_4 w_1 \left|w_1\right| - g\ I_5 \Big] / \left(I_1 + \Delta t\, \theta_2 d_1\right) \end{aligned} \tag{3.9-24}$$

where

$$d_1 = 2\, w_1\, I_2 + \left(2 + b_{fr}\right) |w_1|^{1+b_{fr}}\, A_{fr}\, I_3 + 2\, |w_1|\, I_4 \tag{3.9-25}$$

The temperatures used to evaluate ρ_c and $\overline{\mu}$ in Eqs. 3.9-9, 3.9-11, and 3.9-14 are extrapolated coolant temperatures, extrapolated to the end of the coolant step.

3.9.5 Coolant Pressures

After the flow rate has been calculated, the coolant pressures in a channel at the end of a time step are calculated. First, the pressure at node MZC, the last node at the top of the subassembly, is calculated as

$$p\left(\mathrm{MXC}\right) = p_{t2} + \left[\frac{\Delta z_i}{A}\right]_t \frac{\partial w}{\partial t} \tag{3.9-26}$$

where

$$\mathrm{p}_{t2} = p_{x2} + \rho_{cout}\, g \left[z_{plu} - z_c(\mathrm{MZC})\right] \tag{3.9-27}$$

z_{phi} = outlet plenum elevation

ρ_{cout} = density at outlet

p_{x2} = coolant outlet plenum pressure at the end of the time step

Note that p_{x2}, the outlet plenum pressure supplied by the primary loop module, is defined at an elevation z_{plu}, the elevation of the upper plenum, whereas $p(\mathrm{MZC})$ is calculated at $z(\mathrm{MZC})$, so the gravity head term occurs in Eq. 3.9-27.

After $p(\mathrm{MZC})$ has been calculated, the other pressures are calculated, starting at node MZC-1 working down, using

$$\begin{aligned} p(\mathrm{JC}) = p(\mathrm{JC}+1) + X_{I1}(\mathrm{JC})\frac{\partial w}{\partial t} + w^2 X_{I2}(\mathrm{JC}) + A_{fr}\, w|w|^{1+b_{fr}} X_{I3}(\mathrm{JC}) \\ + |w|\,w\,K_{or}(\mathrm{JC}) + g\,X_{I5}(\mathrm{JC}) \end{aligned} \qquad (3.9\text{-}28)$$

In Eqs. 3.9-25 and 3.9-28, the value used for $\partial w/\partial t$ is calculated at the end of the time step for the transient calculation as

$$\frac{\partial w}{\partial t} = \left[p_{b2} - p_{t2} - w_2^2 I_2 - A_{fr} w_2 |w_2|^{1+b_{fr}} I_3 - w_2|w_2| I_4 - g I_5\right] / I_1 \qquad (3.9\text{-}29)$$

The pressure, p_{b2} at the bottom of the subassembly at the end of the time step is calculated as

$$p_{b2} = p_{in2} - \rho_{cin}\, g\left[z_c(1) - z_{p\ell\ell}\right] \qquad (3.9\text{-}30)$$

$z_{p\ell\ell}$ = plenum location. Note that p_{in2}, the inlet plenum pressure supplied by he primary loop module, is defined at an elevation $z_{p\ell\ell}$, the elevation of the lower plenum.

The pressure distribution in the steady state is calculated the same as in the transient, except that $\partial w/\partial t$ is zero in the steady-state calculation.

3.10 Multiple-Pin Model

3.10.1 Introduction

The new multiple pin treatment was added to the code to account for pin-to-pin variations within a subassembly. This new multiple pin treatment can be used in at least two different ways. One approach is to use the new treatment to compute nominal or "best estimate" variations within a subassembly. Another approach is to compute hot channel behavior due to postulated deviations in coolant flow rate, coolant flow area, and pin power. When doing a whole-core analysis, one probably would not want the hot channel temperatures used in reactivity feedback calculation or in the core outlet temperatures that feed into the primary loop calculations. Therefore, one would probably use parallel treatments for the same subassembly: a nominal one-pin or multi-pin treatment used for reactivity feedback, plus a hot channel treatment decoupled from the reactivity feedback and from the primary loop calculation. The hot channel treatment would be de-coupled by setting reactivity feedback coefficients to zero and by setting the number of subassemblies represented by this treatment to zero.

The new multiple pin treatment allows a number of coupled channels to be used to model a single subassembly. Thus a channel can represent a part of a subassembly instead of the whole subassembly. Peaking factors can be mechanistically calculated by reducing coolant flow areas and flow rates or increasing pin power levels in some channels.

As previously mentioned, the multiple pin option is currently only available for steady-state and single phase transient calculations.

3.10.2 Physical Model

In the new multiple pin option, the regions above and below the pin section of a subassembly are still represented by a single channel; but a number of channels can be used to represent the pin section. Each channel in the pin section can represent one or more concentric rows of pins and their associated coolant. It is also possible for channels to represent slices of pins for a subassembly with a strong lateral power skew or for one with a hot subassembly on one side and a cool subassembly on the opposite side. Figure 3.10–1 illustrates one way that a number of channels can be used to model the pin section of a subassembly as concentric rings of pins and coolant subchannels. The thimble flow region in this figure is a feature of the EBR-II XX09 instrumented subassembly and is not found in ordinary subassemblies. This case concentrates on the coolant subchannels and splits the pins. It is also possible to shift the channel boundaries half a pin and use a pin-based representation with intact pins and split coolant subchannels. Currently up to 56 channels can be used to represent a subassembly.

The new option accounts for coolant-to-coolant heat transfer between adjacent channels, including the effects of both conduction and turbulent mixing. It also accounts for subassembly-to-subassembly heat transfer from the duct wall of a subassembly, through the interstitial sodium, to the duct wall of a neighboring subassembly. In addition, axial conduction in the coolant is accounted for.

Figure 3.10–2 illustrates the axial representation of the subassembly flow, with parallel flow paths through the pin section. Each channel used to represent the pin section of subassembly has its own separate time-dependent flow rate, and the flow rates in all channels in a subassembly are driven by common pressures at the inlet and outlet of the pin section. Thus, transient flow redistribution among channels is accounted for. The single flow rate in the regions above and below the pin section is the sum of the pin section flow rates, so the subassembly flow orifice sees the correct total flow rate. Even though the coolant-to-coolant heat transfer coefficient includes a term for turbulent mixing between coolant subchannels, one effect that the flow calculation does not account for is mass flow between channels in the pin section, although cross-flow at the ends of the pin section is allowed. Therefore, if recirculation loops occur within a subassembly at low flows, the model would calculate them; but the recirculation loops would go to the ends of the pin section.

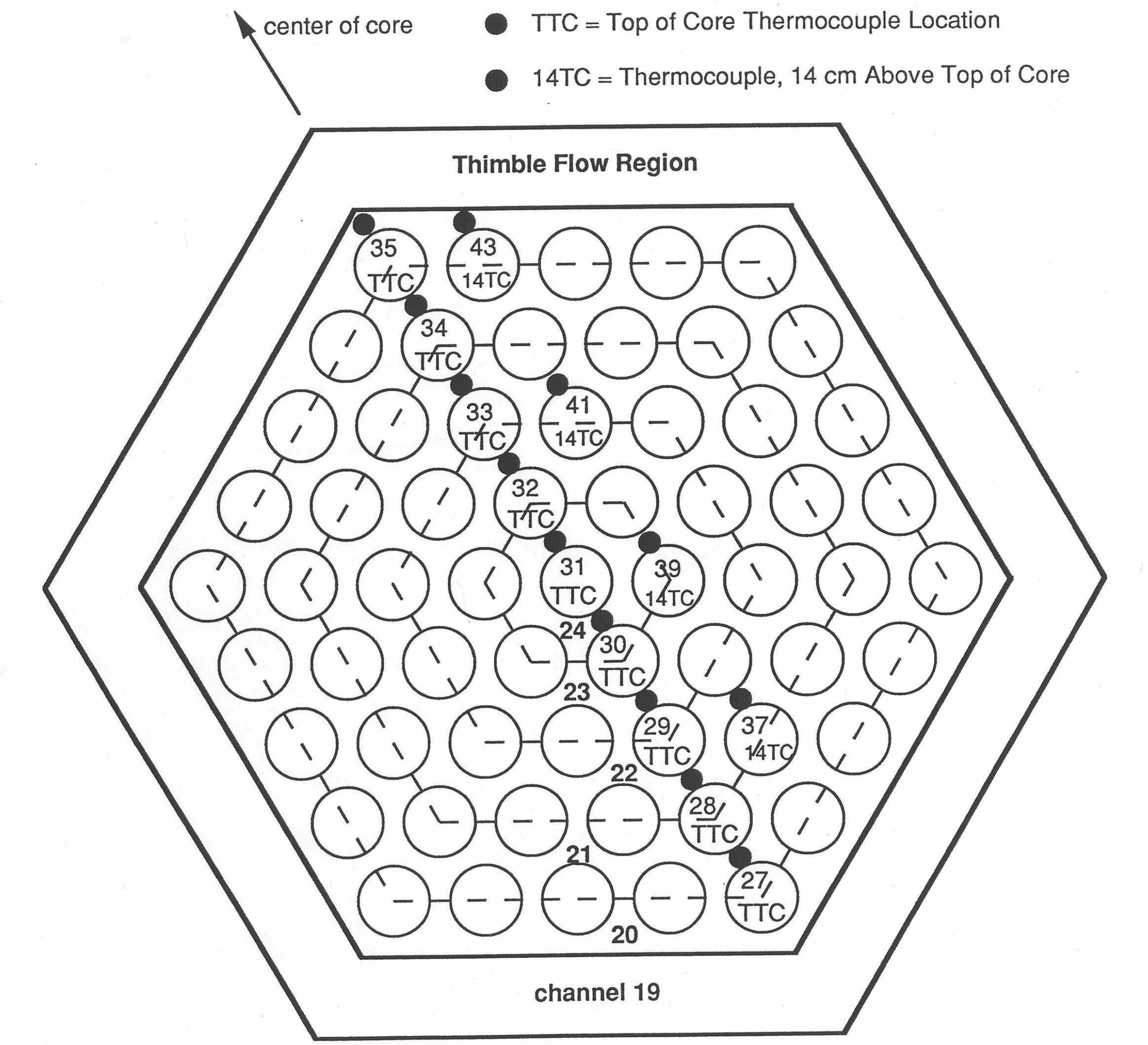


Figure 3.10–1: SAS4A/SASSYS-1 Multiple Pin Representation and Thermocouple Locations for the EBR-II XX09 Instrumented Subassembly

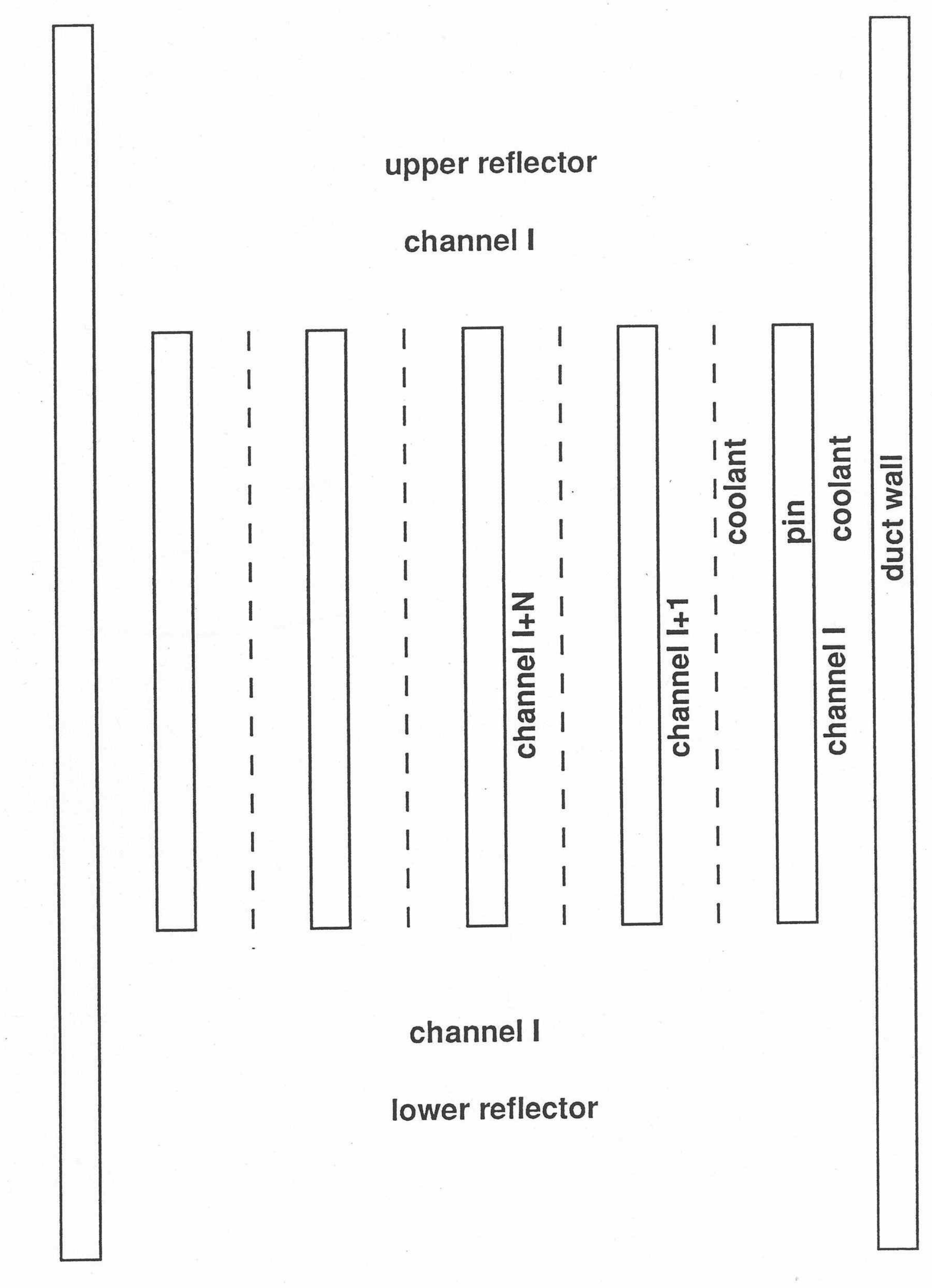


Figure 3.10–2: SAS4A/SASSYS-1 Multiple Pin Treatment of a Subassembly

In the new model the coolant in channel I can transfer heat directly to the coolant in channel I-1 and channel I+1. Using correlations of the same form as those used in THI3D code [3-6] and the HOTCHAN code [3-7], the channel-to-channel heat flow per pin per unit height from channel I to channel I+1 is calculated as:

$$Q_{I,I+1} = \left[U_1\bar{k} + U_2\ \bar{C}\left(w_1 + w_{I+1}\right)\right]\left(T_{I-T_{I+1}}\right) \tag{3.10-1}$$

where

$\bar{k}$ = average thermal conductivity of the coolant

$\bar{C}$ = average specific heat

w_I = coolant mass flow per pin (kg/s) in channel I,

and

T_I = coolant temperature

In this equation, U_I is a geometry factor for thermal conduction, and U_2 is a product of a turbulent-mixing coefficient and a geometry factor for turbulent mixing between channels. Since a SAS4A/SASSYS-1 channel usually models a group of coolant subchannels, the values used for U_1 and U_2 must account for a combination of parallel and series heat flow paths between the middle of channel I and the middle of channel $I+1$. Subassembly-to-subassembly heat transfer is handled in a somewhat simpler manner. For heat transfer from the outer surface of the structure in channel I to the outer surface of the structure in channel J, a constant value is used for the produce of the heat transfer coefficient and the heat transfer area per unit height.

3.10.3 Numerical Methods

Most of the transient heat transfer calculations and flow rate calculations in SASSYS-1 use semi-implicit time differencing in order to obtain stable and accurate solutions with reasonably long time steps. Before the onset of coolant boiling or pin disruption, time step sizes of a second or more are commonly used; and the code usually runs significantly faster than real time on a Cray XMP computer.

From a numerical computation point of view, the two main tasks in adding the multiple pin model to the code were the coolant-to-coolant heat transfer calculation and the coolant flow rate calculation with parallel flow paths in the pin section. Both of these calculations use semi-implicit time differencing. In the single-pin model, coolant temperatures for all of the radial temperature nodes in the pin, coolant, and structure at one axial node are solved for simultaneously in order to obtain a semi-implicit time differencing solution without iteration. In the new multiple-pin treatment, this concept is carried one step further. At a given axial node, temperature at all of the radial nodes for all channels representing a subassembly are solved for simultaneously. In the heat transfer calculations, the axial conduction terms, which are small, are treated with explicit forward time differencing so that axial nodes are decoupled and can be treated separately except for the coolant convection terms. The axial coupling due to the coolant convection terms is handled by starting at one end of the subassembly and solving for axial nodes one at a time in the direction of the flow. If flow has reversed in some channels but not in others, the calculation progresses in the direction of the dominate flow; and explicit forward differencing is used for the coolant convection terms in the non-dominate flow direction channels. The subassembly-to-subassembly heat fluxes are calculated with explicit forward differencing in time, and this does impose a stability limit on the time step size. For typical subassembly duct wall thicknesses, the explicit subassembly-to-subassembly heat flux calculation limits the maximum time step size to a value in the range from .25 to .5 seconds. For the coolant flow rate calculations, the incompressible flow momentum equations are linearized about conditions at the beginning of the time step. Then, flow at the end of the step are calculated for all channels in a subassembly simultaneously.

A null transient is used to obtain steady-state temperatures at the start of the regular transient. First all coolant, pin, structure, and reflector temperatures in all subassemblies are set to the coolant inlet temperature. Then, the power levels and coolant flow rates are held constant while a number of transient heat transfer time steps are made. Since the pin thermal time constant and the coolant transit time through a subassembly are both less than a second, the null transient results converge rapidly if reasonably large time steps are used.

3.10.4 Detailed Mathematical Treatment

The main computational parts of the new multiple pin model are the heat transfer calculations in the pin section of a subassembly, the coolant flow rate calculations for a subassembly, the subassembly-to-subassembly heat transfer, and changes to the driver routines to call the appropriate new routines at the proper times. The pin section heat transfer calculations are done in two new subroutines: TSHTM3 calculates temperatures in the core and axial blankets, and TSHTM2 calculates temperatures in the gas plenum region. The existing TSHTN1 still calculates temperatures in the reflector zones above and below the pin section, but it was modified. TSHTN1 had to be modified slightly to pick up mixed mean outlet temperatures from the multiple channels representing the pin section. Also, axial conduction in the coolant was added to TSHTN1. The new TSCLM1 routine calculates transient coolant flow rates for a subassembly. The subassembly-to-subassembly heat fluxes are calculated in CHCHFL.

3.10.4.1 Heat Transfer Calculations in the Core and Axial Blankets: Subroutine TSHTM3

The multiple pin heat transfer calculations for the core and axial blankets in subroutine TSHTM3 are similar to the single pin calculations in the existing subroutine TSHTN3, as described in Section 3.3.1. One difference is that TSHTN3 does one time step for one channel each time it is called, whereas TSHTM3 solves for temperatures in all pins or channels representing a subassembly when it is called. Also, TSHTM3 adds extra terms for coolant-to-coolant heat transfer; and TSHTM3 includes axial conduction in the coolant.

Within a fuel pin, one-dimensional radial heat transfer is used. The basic heat transfer equation is Eq. 3.3-1. This part of the calculation in TSHTM3 is identical to the corresponding calculation in TSHTN3.

For the coolant in channel I, the heat transfer calculations are basically one-dimensional in the axial direction, with extra terms for coolant-to-coolant heat transfer from channel I-1 and I+1. The basic heat transfer equation is:

$$\rho c A_c \frac{\partial T}{\partial t} + \frac{\partial}{\partial z}\left(wcT\right) = Q_T A_c \qquad (3.10\text{-}2)$$

where

A_c = coolant flow area

z = axial position

w = coolant mass flow rate

and

Q_r = total heat source per unit volume

The heat source contains a number of terms:

$$Q_T = Q_c + Q_{ec} + Q_{sc} + Q_{ax} + Q'_{I-1,I} - Q'_{I.I+1} \qquad (3.10\text{-}3)$$

where

Q_c = source due to direct heating of the coolant by neutrons and gamma rays,

Q_{ec} = heat flow from cladding to coolant,

Q_{sc} = heat flow from structure to coolant,

Q_{ax} = axial conduction source, given by

$$Q_{ax} = \frac{\partial}{\partial z} k \frac{\partial T}{\partial z} \qquad (3.10\text{-}4)$$

and

$$Q'_{I,I+1} = \frac{Q_{I,I+1}}{A_c} \qquad (3.10\text{-}5)$$

where $Q'_{I,I+1}$ is given by equation 3.10-1. Note that Eq. 3.10-2 is the same as Eq. 3.3-5, except that additional source terms are included in the multiple pin model.

Finite differencing in space and time is used to solve these heat transfer equations. Figure 3.2–3 shows the axial and radial mesh used for a single channel. In the multiple pin model the pin section mesh is repeated for each channel, but the axial reflector zones are only used in the first channel used to represent a subassembly. At each axial node in the core and axial blankets a number of radial nodes are used for the fuel, three radial nodes are used for the cladding, one node is used in the coolant, and two nodes are used in the structure. All channels, representing a subassembly must use the same axial mesh. Also, all subassemblies connected by subassembly-to-subassembly heat transfer must use the same axial mesh.

For a given time step, Eqs. 3.3-1 and 3.10-2 become linear finite difference equations. The temperatures are known at the beginning of the time step, and the temperature at the end of the step are the unknowns to be solved for. Semi-implicit time differencing is used for all terms except the axial conduction terms Q_{ax}, and the coolant-to-coolant terms $Q_{I-1,I}$ and $Q'_{I,I+1}$. The Q_{ax} axial conduction term is calculated explicitly based on temperatures at the beginning of the time step. The coolant-to-coolant terms are calculated fully implicitly, using the coolant flow rates and temperatures at the end of the step.

After finite differencing for one axial node, one obtains N simultaneous liner equations in N unknowns, where the unknowns are the temperatures at the end of the time step for all radial nodes in all channels representing the subassembly. These equations are solved by Gaussian elimination. The equations are basically tri-diagonal, with extra non-tri-diagonal terms for coolant-to-coolant heat transfer between channels. No full N by N matrix is ever set up by TSHTM3, and a general full matrix solution package is not used. Instead, only non-zero terms are computed and stored; and the Gaussian elimination solution was written specifically for this set of equations. The result is that the storage requirements vary linearly, rather than quadratically, with the maximum allowable value for N; and the computation time varies linearly, rather than quadratically, with the actual value of N.

3.10.4.2 Heat Transfer Calculations in the Gas Plenum Region: Subroutine TSHTM2

Subroutine TSHTM2 does the heat transfer calculations for one time step for all axial nodes in the gas plenum region for all channels representing a subassembly when it is called. TSHTM2 is similar to TSHTM3, but TSHTM2 deals with fewer radial nodes. As shown in Figure 3.2–5, in the plenum region there are still two radial nodes in the structure and one in the coolant; but there is only one in the cladding; and there are no fuel nodes. Instead of fuel temperatures there is one gas temperature common to all of the axial nodes in a pin. As in TSHTM3, for a given axial node TSHTM2 solves simultaneously for temperatures at all radial modes in all channels representing a subassembly.

One special problem in TSHTM2 is the gas temperature, which is common to all axial nodes in the gas plenum region of a pin. Axial nodes are handled one at a time, rather than simultaneously, whereas a semi-implicit time differencing treatment would require solving for all axial nodes simultaneously. The new multiple-pin TSHTM2 routine handles this problem in a simpler manner than the old single-pin TSHTN2. The way that the gas temperature is handled is TSHTM2 is to calculate a separate gas temperature for each axial node. At the beginning of the time step, the gas temperatures at all axial nodes are set to one common value. Then separate values are calculated for each axial node at the end of the step. Finally, an average of the separate values is calculated for the one common value at the ed of the step.

3.10.4.3 Coolant Flow Rates: Subassembly TSCLM1

The coolant for rate calculation for a subassembly in the multiple-pin routine TSCLM1 is much more complex than the corresponding calculation in the single pin routine TSCNV1. As shown in Figure 3.10–2, the multiple pin calculation involves multiple parallel flow paths in the pin section, in series with single flow paths in the reflector regions. Therefore, the multiple-pin routine TSCLM1 was written from scratch, rather than starting form the single pin TSCNV1 routine.

Figure 3.10–3 illustrates the main variables used in TSCLM1.

Incompressible flow is used for these calculations, so conservation of mass gives

$$N_1 \, w_r = \sum_k w_{pk} \, N_k \tag{3.10-6}$$

where

w_r = coolant mass flow rate per pin in the reflector zones,

w_{pk} = pin section coolant mass flow rate per pin in channel k,

N_k = number of pins in channel k,

and

I = channel number of the first channel representing the subassembly.

The momentum equation for the pin section is

$$\frac{L_p}{A_{cpk}} \frac{dw_{pk}}{dt} = p_b - p_c - \Delta p_{pk} \tag{3.10-7}$$

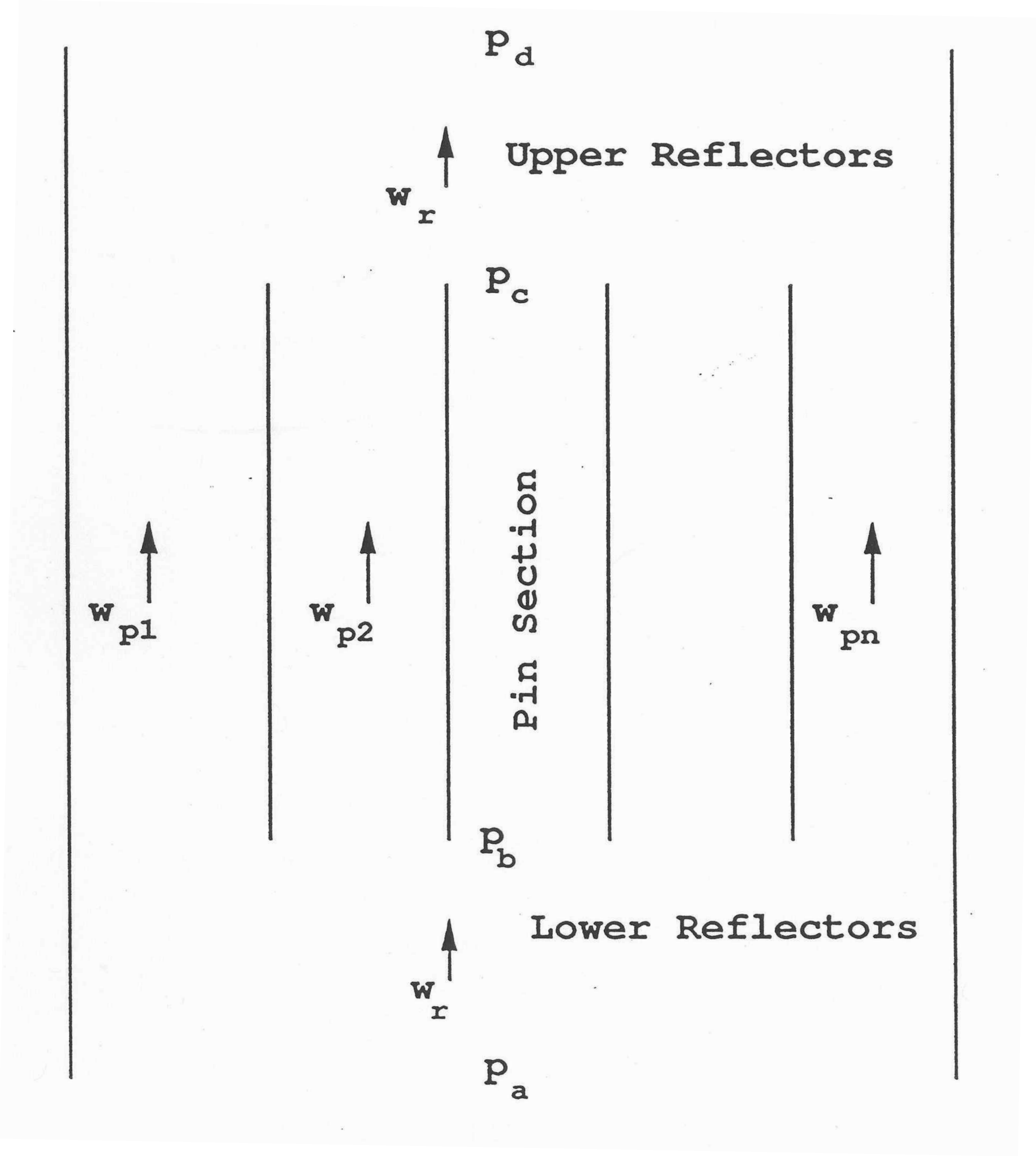


Figure 3.10–3: Subassembly Coolant Flows

where

L_p = length of pins,

A_{cpk} = coolant flow area in channel k in the pin section,

t = time,

p_b = coolant pressure at the bottom of the pins,

p_c = coolant pressure at the top of the pins,

$$\Delta p_{pk} = \text{pressure loss} = \sum_{jv=\text{JCPNBT}}^{\text{JCPNTM}} \Delta p_{kjc} \tag{3.10-8}$$

jc = axial coolant node,

JCPNBT = first axial node in the pin section

JCPNTM = last axial node in the pin section,

and

Δp_{kjc} = pressure loss in axial node jc of channel *k*.

The pressure loss contains a number of terms:

$$\Delta p_{kjc} = \Delta p_{frkjc} + \Delta p_{grkjc} + \Delta p_{orkjc} + \Delta p_{acckjc} \tag{3.10-9}$$

where

$$\Delta p_{frkjc} = \text{friction loss} = \frac{f_{kjc}\,\Delta z_{jc}\, w_{pk} \left| w_{pk} \right|}{2\bar{\rho}_{ckjc}\, A_{cpk}\, D_{hpk}} \tag{3.10-10}$$

$$\Delta p_{grpjc} = \text{gravity head} = g\,\bar{p}_{ckjc}\,\Delta z_{jc} \tag{3.10-11}$$

$$\Delta p_{orfkjc} = \text{orifice loss} = \frac{K_{orkjc}\, w_{pk} \left| w_{ph} \right|}{2\bar{\rho}_{ckjc}\, A_{cpk}^{2}} \tag{3.10-12}$$

$$\Delta p_{acckjc} = \text{acceleration term} = \frac{w_{pk}^2}{A_{cpm}^2}\left(\frac{1}{\rho_{ckjc+1}} - \frac{1}{\rho_{ckjc}}\right) \tag{3.10-13}$$

Δz_{jc} = axial node length,

f_{kjc} = friction factor in channel k at node *jc*,

ρ_{ckjc} = coolant density at bottom of node *jc* in channel *k*,

$\bar{\rho}_{ckjc}$ = average coolant density in node *jc* in channel *k*,

D_{hpk} = Hydraulic diameter,

and

K_{orkjc} = orifice coefficient in node *jc* of channel *k*

The friction factor is calculated as in Eq. 3.9-3.

using

$$R_e = \text{Reynolds number} = \frac{D_{hpk}\left|w_{pk}\right|}{\mu_{kjc} A_{cpk}} \tag{3.10-14}$$

and

μ_{kjc} = viscosity of the coolant.

After semi-implicit finite differencing in time and linearization of the pressure drop terms, applying Eq. 3.10-7 to a time step from *t* to *t* + Δ*t* gives

$$\frac{L_p}{A_{cpk}} \frac{\Delta w_{pk}}{\Delta t} = p_b(t) + \theta_2 \Delta p_b - p_c(t) - \theta_2 \Delta p_c - \Delta p_{pk}(t) - \theta_2 \, \Delta \, w_{pk} K_{wpk} \tag{3.10-15}$$

where

$$\Delta w_{pk} = w_{pk}(t + \Delta t) - w_{pk}(t) \tag{3.10-16}$$

$$\Delta p_b = p_b(t + \Delta t) - p_b(t) \tag{3.10-17}$$

$$\Delta p_c = p_c\left(t+\Delta t\right) - p_c\left(t\right) \tag{3.10-18}$$

θ_2 = degree of implicitness

$$\theta_1 = 1 - \theta_2 \tag{3.10-19}$$

$$K_{wpk} = \sum_{jc=\text{JCPNBT}}^{\text{JCPNTM}} K_{wpkjc} \tag{3.10-20}$$

$$K_{wpkjc} = \frac{\partial \Delta p_{kjc}}{\partial w_{pk}} = K_{frkjc} + K_{wcrkjc} + K_{acckjc} \tag{3.10-21}$$

$$K_{frkjc} = \begin{cases} \left(2+b_{fr}\right)\dfrac{\Delta p_{frkjc}\left(t\right)}{w_{pk}\left(t\right)} & \text{if } R_{ckjc} \geq R_{et} \\ \dfrac{\Delta p_{frkjc}\left(t\right)}{w_{pk}\left(t\right)} & \text{if } R_{ckjc} < R_{et} \end{cases} \tag{3.10-22a}$$

or

$$K_{frkjc} = \left[\left(2+b_{fr}\right) A_{fr}\, A\left(\text{RE}\right)^{b_{fr}} + A_{fL} / \text{RE}\right] \frac{\Delta z_{jc}\left|w_{pk}\left(t\right)\right|}{2 p_{ckjc} D_{hpk} A_{cpk}} \tag{3.10-22b}$$

$$K_{wpkjc} = \frac{2\Delta p_{orfkjc}\left(t\right)}{w_{pk}\left(t\right)} \tag{3.10-23}$$

and

$$k_{acckjc} = \frac{2\Delta p_{acckjc}\left(t\right)}{w_{pk}\left(t\right)} \tag{3.10-24}$$

Equation 3.10-15 can then be written as

$$\Delta p_b - \Delta p_c = d_{0pk} + d_{0pk} + d_{1pk}\Delta w_{pk} \tag{3.10-25}$$

where

$$d_{0pk} = \frac{p_c(t) - p_b(t) + \Delta p_{pk}(t)}{\theta_2} \tag{3.10-26}$$

and

$$d_{1pk} = \frac{L_p / A_{cpk} + \theta_2 \Delta t \, K_{wpk}}{\theta_2 \Delta t} \tag{3.10-27}$$

Applying a similar process to the upper and lower reflector regions gives:

$$\Delta p_b = D_{0\ell} + D_{1\ell} \sum_k \Delta w_{pk} \tag{3.10-28}$$

and

$$\Delta p_c = D_{0u} + D_{1u} \sum_k \Delta w_{pk} \tag{3.10-29}$$

where

$$D_{0\ell} = \frac{\theta_1 p_a(t) + \theta_2 \, p_a(t + \Delta t) - p_b(t) - \Delta p_\ell(t)}{\theta_2} \tag{3.10-30}$$

$$D_{1\ell} = \frac{-\sum_{\mathrm{KA=1}}^{\mathrm{KZPIN-1}} L_{kz} / A_{cKZ} - \theta_2 \Delta t \, K_{w\ell}}{\theta_2 \Delta t} \tag{3.10-31}$$

$$D_{0u} = \frac{\theta_1 p_d(t) + \theta_2 \, p_d(t + \Delta t) - p_c(t) + \Delta p_u(t)}{\theta_2} \tag{3.10-32}$$

$$D_{1u} = \frac{\sum_{\mathrm{KZ-KZPIN+1}}^{\mathrm{KZM}} L_{\mathrm{KZ}} / A_{\mathrm{cKZ}} + \theta_2 \Delta t \, K_{wu}}{\theta_2 \Delta t} \tag{3.10-33}$$

L_{KZ} = length of zone *kz*

$A_{c\mathrm{KZ}}$ = coolant flow area in zone *kz*

KZPIN = axial zone number of the pin section

KZM = last zone number

$$\Delta p_{\ell} = \sum_{jc=1}^{\text{JCPNBT}-1} \Delta p_{rjc} \tag{3.10-34}$$

$$\Delta p_{u} = \sum_{jc=\text{JCPNTP}}^{\text{MZCM1}} \Delta p_{rjc} \tag{3.10-35}$$

Δp_{rjc} = pressure loss in node *jc* of a reflector zone

JCPNBT = first axial node in pin section

JCPNTP = first axial node above the pin section

MZCM1 = last axial node

$$K_{w\ell} = \frac{\partial \Delta p_{\ell}}{\partial w_r} \tag{3.10-36}$$

and

$$K_{wu} = \frac{\partial \Delta p_{u}}{\partial w_u} \tag{3.10-37}$$

Then, combining Eqs. (3.10-25) and (3.10-28) gives

$$\Delta p_b = B_o + B_1 \Delta p_c \tag{3.10-38}$$

where

$$B_o = \frac{D_{o\ell} - D_{1\ell} S_1}{1 - D_{1f} S_o}, \tag{3.10-39}$$

$$B_1 = \frac{-D_{1l} S_o}{1 - D_{1l} S_o} \tag{3.10-40}$$

$$S_o = \sum_k \frac{1}{d_{1pk}}, \tag{3.10-41}$$

and

$$S_1 = \sum_k \frac{d_{opk}}{d_{1pk}}, \tag{3.10-42}$$

Similarly, combining Eqs. (3.10-25) and (3.10-29) gives

$$\Delta p_c = C_o + C_1 \Delta p_b \tag{3.10-43}$$

where

$$C_o = \frac{D_{ou} - D_{1u} S_1}{1 + D_{1u} S_o}, \tag{3.10-44}$$

and

$$C_1 = \frac{D_{1u} S_o}{1 + D_{1u} S_o}. \tag{3.10-45}$$

Combining Eqs. (3.10-38) and (3.10-43) gives

$$\Delta p_c = \frac{C_o + C_1 B_o}{1 - C_1 B_1} \tag{3.10-46}$$

Then Δp_b can be obtained from Eq. (3.10-38) and Δw_{bk} can be obtained from Eq. (3.10-25) for each channel.

After p_b, p_c, and the coolant flow rates have been calculated for the end of the time step it is possible to calculate the pressures. The pressure is calculated at the axial node boundaries. p_{kjc} is the pressure at the bottom (inlet end) of node *jc* in channel *k*. First, the nodal pressure loss at $t + \Delta t$ is calculated as

$$\Delta p_{pkjc}(t + \Delta t) = \Delta p_{kjc}(t) + K_{wpkjc}\, \Delta w_{pk} \tag{3.10-47}$$

and Δp_{pk} $(t + \Delta t)$ is calculated from Eq. (3.10-8). Then Eq. (3.10-7) is used to obtain $\frac{dw_{pk}(t+\Delta t)}{dt}$.

Integrating the momentum equation over one axial node gives

$$\frac{\Delta z_j}{A_{cpk}} \frac{dw_{pk}}{dt} = p_{pkjc} - p_{pkjc+1} - \Delta p_{kjc} \tag{3.10-48}$$

Starting by setting

$$p_{pkjJCPNTP} = P_c \tag{3.10-49}$$

the code marches down the channel, using Eq. (3.10-48) to obtain p_{pkjc} after p_{pkjc+1} has been calculated. A similar procedure is used for calculating the pressures in the upper and lower reflector zones.

The equation used to compute the degree of implicitness as a function of time step size in TSCMV1 is

$$\theta_2 = \frac{a + bx + x^2}{2a + cx + x^2} \tag{3.10-50}$$

where

$$x=\Delta t/\tau \tag{3.10-51}$$

τ = a time constant

a = 6.12992

b = 2.66054

and

c = 3.56284

The basis for this expression is given in Appendix 3.1. For a single channel treatment, the time constant, τ would be

$$\tau = \frac{\sum_{KZ=1}^{KZM} \frac{L_{KZ}}{A_{cKZ}} + \left(\frac{L_i}{A}\right)_i + \left(\frac{L_i}{A}\right)_x}{K_{wpk}\, K_{w1} + K_{wu}} \tag{3.10-52}$$

where

$\left(\frac{L_i}{A}\right)_i$ =extra coolant inertia term at the subassembly inlet to account for inertia of the coolant in the inlet plenum,

and

$\left(\frac{L_i}{A}\right)_x$ =same for the subassembly outlet.

Since a simultaneous solution of flows in all channels of an assembly plus the lower and upper reflector zones is required, the overall time constant is calculated as

$$\tau = \frac{\dfrac{L_p}{\sum\limits_k A_{cpk}} + \sum\limits_{kZ \neq KZPIN} \dfrac{L_{KZ}}{A_{cKZ}} + \left(\dfrac{L_i}{A}\right)_i + \left(\dfrac{L_i}{A}\right)_x}{K_{w1} + K_{wu} + \dfrac{1}{\sum\limits_k \dfrac{1}{k_{wpk}}}} \tag{3.10-53}$$

3.10.5 Relationship Between Single Pin and Multiple Pin Models

The single pin treatment and the multiple pin treatment coexist in the SAS4A/SASSYS-1 code; it is possible to use single pin treatments for some subassemblies and multiple pin treatments for other subassemblies in the same problem. There are now two separate sets of subroutines in the code for computing single phase thermal hydraulics: the single pin routines and multiple pin routines. Since the multiple pin model can handle the case of a single pin per subassembly, the single pin single phase thermal hydraulics subroutines are now redundant; and at some point in the future they will probably be removed from the code.

As previously mentioned, currently the multiple pin model is only available for steady state and single phase transient calculations; no multiple pin treatment is available in the code for coolant boiling, pin disruption, or relocation of fuel or cladding. In the future, the multiple pin treatment will probably be extended to coolant boiling and to relocation of fuel and cladding.

3.11 Subassembly-to-subassembly Heat Transfer

SAS4A/SASSYS-1 contains a model for transferring heat from the duct wall of one subassembly, through the interstitial sodium, to the duct wall of an adjacent subassembly. During normal, full power operation the subassembly-to-subassembly heat flow is usually small compared to the power generation rate, but at decay heat power levels the heat flow between a subassembly and its neighbors can be comparable to the power generation rate within the subassembly. This heat flow between adjacent subassemblies will affect subassembly-to-subassembly flow re-distribution at low flow rates; and at low powers and flow rates it will tend to cause all subassemblies to have similar temperature rises.

The subassembly duct wall is normally represented by the structure in a SAS4A/SASSYS-1 channel; so this mode is implemented by transferring heat from the outer structure nodes of a SAS4A/SASSYS-1 channel to the outer structure nodes of other SAS4A/SASSYS-1 channels or to a constant temperature heat sink. Also, it is possible to transfer heat from the outer structure node of one channel to the coolant of another channel. This option was included in the code in order to treat the thimble flow region shown in Figure 3.10–1. The equation for the structure to structure heat flux,

$Q_{I,M}(j)$, from channel I to channel M at axial node j is

$$Q_{I,M}\left(j\right)=\frac{\left(HA\right)_{I,M}}{N_{PI}N_{SI}S_{st1}}\left[T_{sto}\left(I,J\right)-T_{sto}\left(M,j\right)\right] \tag{3.11-1}$$

where

$(HA)_{I,M}$ = heat transfer coefficient times area per unit height, between channels I and M

T_{sto} = structure outer node temperature

N_{PI} = number of pins per subassembly in channel I

N_{SI} = number of subassemblies represented in channel I

S_{stI} = structure perimeter per pin in channel I

The SAS4A/SASSYS-1 channel treatment only accounts for one structure per channel; so the channel-to-channel heat flux seen by the structure in a channel will be a sum of the heat fluxes to all adjacent channels. This heat flux is

$$Q_{I}\left(j\right)=\sum_{M} Q_{I,M}\left(j\right) \tag{3.11-2}$$

Each SAS4A/SASSYS-1 channel can transfer heat to up to eight other channels. A subassembly has six nearest neighbors, but a channel can represent more than one subassembly. Therefore, a channel can be in contact with more than six other channels.

If the constant temperature heat sink option is used, then the user supplies values of T_{snk} and H_{snk} for each axial zone in the channel, and these values are held constant during the calculations.

In contrast to many of the other temperature calculations in the code, explicit forward time differencing is used for the channel to channel heat transfer. The channel to channel heat fluxes are calculated at the beginning of each main time step using the temperatures at that time. These heat fluxes are then held constant during a time step. This explicit forward differencing imposes stability limits on the time step size. The stability limit is typically in the range of .2-.5 seconds.

At this point, the subassembly-to-subassembly heat transfer model has only been implemented in the transient single phase heat transfer calculations. It is necessary to run a null transient to obtain the current steady-state temperatures when subassembly-to-subassembly heat transfer is used.

3.12 Interaction with Other Models

As mentioned in Section 3.1, the heat-transfer routines interact with a number of other models. These interactions are indicated in Figure 3.1–1.

If PRIMAR-4 is being used, then the subassembly coolant outlet temperatures calculated in the heat-transfer routines are used by PRIMAR-4 to calculate the outlet plenum temperature. Also, if flow reversal occurs in a channel, then the temperature calculated in the heat-transfer routines for the coolant leaving the bottom of the subassembly is used by PRIMAR-4 in the calculation of the inlet plenum temperature. Section 3.3.6 describes how the inlet and outlet plenum temperatures are used in the calculations of the subassembly coolant inlet and reentry temperatures.

Before the onset of voiding, TSHTRN calculates the coolant temperatures used in the coolant channel hydraulics calculations, and the hydraulics routines calculate the coolant flow rates used by TSHTRN.

After the onset of voiding, coolant temperatures are calculate din TSBOIL. This module supplies the heat flux at the cladding outer surface, as defined in Eq. 3.5-2 and used in Eq. 3.5-9 to TSHTRV. TSBOIL uses the calculated cladding temperatures form TSHTRV to obtain extrapolated cladding temperatures, as in Eq. 3.5-3, for use in its coolant temperature calculations.

The point kinetics model supplies the power level used in the heat-transfer routines. In return, the heat-transfer routines supply the temperatures used to calculate reactivity feedback.

3.12.1 Reactivity Feedback

The temperatures calculated in the core thermal hydraulics routines are used to calculate various components of reactivity feedback. These reactivity components include Doppler feedback, axial expansion of the fuel and cladding, density changes in the sodium, core radial expansion and control rod drive expansion. These reactivity feedbacks are described in Chapter 4.

3.12.2 Coupling Between Core Channels and PRIMAR-4

As described in Chapter 5, the PRIMAR-4 calculations for a PRIMAR time step are carried out before the core channel coolant calculations. PRIMAR-4 must make estimates of the core flows for a new time step, and it also makes corrections for the differences between its estimates for the previous step and values computed by the core channel coolant routines. The coolant routines supply information to PRIMAR-4 for use in these estimates and corrections. Also, PRIMAR-4 supplies inlet and outlet coolant plenum pressures and temperatures for the use in the core channel calculations.

The information supplied by PRIMAR-4 is

$p_{in}(t_{p1})$ = inlet plenum pressure at the beginning of the PRIMAR time step

$p_x(t_{p1})$ = outlet plenum pressure at the beginning of the PRIMAR time step

$\frac{dp_{in}}{dt}$ =time derivative of the inlet plenum pressure

$\frac{dp_x}{dt}$ =time derivative of the outlet plenum pressure

ρ_{cin} =coolant density in the inlet plenum

ρ_{cout} =coolant density in the outlet plenum

T_{cin} =coolant temperature in the inlet plenum

and

T_{cout} = coolant temperature in the outlet plenum

The pressure p_{in} is at an elevation $z_{p\ell\ell}$ and p_x is at z_{plu}. At any time, *t*, during the PRIMAR time step, the inlet plenum pressure is

$$p_{in}(t) = p_{in}(t_{p1}) + (t - t_{p1})\frac{dp_{in}}{dt} \tag{3.12-1}$$

and the exit plenum pressure is

$$p_x(t) = p_x(t_{p1}) + (t - t_{p1})\frac{dp_x}{dt} \tag{3.12-2}$$

The information supplied to PRIMAR-4 by the core coolant routines for channel *ic* at the subassembly inlet *(L=*1*)* or outlet *(L=*2*)* is the following:

$$\Delta m_c(L) - \sum_{ic} N_{ps}(ic) \int_{p1}^{t_{p2}} w(L,ic)\,dt, \tag{3.12-3}$$

$$\Delta m_c T_c(L) - \sum_{ic} N_{ps}(ic) \int_{p1}^{t_{p2}} w(L,ic) T_{ex}(L,ic)\,dt, \tag{3.12-4}$$

$w(L,ic,t = t_{p2})$ = computed flow rate at t_{p2}

$T_{ex}(L,ic)$ = coolant temperature at the subassembly inlet or outlet

$\Delta E_v(L,ic)$ = heat added to the inlet or outlet plenum by condensing sodium vapor. (This term is zero before the onset of boiling.)

and the coefficients $C_o(L,ic)$, $C_i(L,ic)$, $C_2(L,ic)$, and $C_3(L,ic)$ used by PRIMAR-4 to estimate the core channel flow. PRIMAR-4 estimates the flow into or out or each subassembly using

$$\frac{dw_e(L,ic)}{dt} = C_0(L,ic) + C_1(L,ic)\,p_{in} + C_2(L,ic)\,p_x + C_3(L,ic)\,w_e(L,ic)\left|w_e(L,ic)\right| \tag{3.12-5}$$

The core channel calculations use *w* as the flow rate per pin, whereas PRIMAR-4 estimates the total flow represented by a channel, so $N_{ps}(ic)$, the number of pins per subassembly times the number of subassemblies represented by the channel, comes into Eqs. 3.12-3, 3.12-4 and the calculations of the coefficients C_0, C_1, C_2, and C_3. In the pre-voiding module the coefficients are calculated as

$$C_0 = -\frac{gN_{ps}\left\lfloor I_5 + \rho_{cin}\left(z_c(1) - z_{p\ell\ell}\right) + \rho_{cout}\left(z_{p\ell u} - z(MZC)\right)\right\rfloor}{I_1} \tag{3.12-6}$$

$$C_1 = \frac{N_{ps}}{I_1} \tag{3.12-7}$$

$$C_2 = -C_1 \tag{3.12-8}$$

and

$$C_3 = -\frac{\left(I_2 w_2 + A_{fr} I_3 |w_2|^{1+b_{fr}} + I_4 |w_2|\right)}{|w_2| I_1 N_{ps}} \tag{3.12-9}$$

In this case, the coefficients for L=2 are equal to those for L=1.

3.13 Subroutine Descriptions and Flowcharts

The subroutines used in the core-channel thermal hydraulics calculations are described below, grouped by the phase of the calculation where they are used.

Steady-state Thermal Hydraulics:

SSTHRM-- Driver for the steady-state calculations:

SSCOOL-- Calculations steady-state coolant, structure, reflector, and gas plenum temperature for each axial node. Also calculates coolant pressure and saturation temperature at each axial node.

SSCLM1-- Calculates steady-state coolant pressures and saturation temperatures for the multiple pin option.

SSHTR-- Calculates steady-state fuel and cladding temperatures in the core and blankets.

SSNULL-- Driver for the steady-state null transient.

Pre-Voiding Transient Temperature Calculations:

TSCL0-- Driver of the pre-voiding thermal hydraulics.

TSHTRN-- Driver for the pre-voiding transient temperature calculations. Calls TSHTN1, TSHTN2, TSHTN3, TSHTN4, TSHTM2, and TSHTM3 as appropriate.

TSHTN1-- Calculates reflector, coolant, and structure temperatures in a reflector zone.

TSHTN2-- Calculates cladding, coolant, structure, and plenum gas temperatures in the gas plenum region.

TSHTN3-- Calculates fuel, cladding, coolant, and structure temperatures in the core and axial blankets.

TSHTN4-- Calculates the axial node size and the coolant density and specific heat for each axial code.

TSHTN5-- Called from TSHTN3 to adjust fuel and cladding temperatures to account for the heat of fusion if melting is occurring.

TSHTM2-- Multiple pin version of TSHTN2.

TSHTM3-- Multiple pin version of TSHTN3.

Fuel and Cladding Temperatures During Boiling:

TSBOIL-- Driver for the boiling module.

TSHTRV-- Calculates fuel and cladding temperatures in the core and axial blankets.

Pre-voiding Coolant Flow Rates and Pressures:

TSCHV1-- Extrapolates coolant temperatures, computes coolant flow rate and pressures. Also tests for start of boiling. Sets variables for coupling with PRIMAR-4. Called by TSCLO every coolant step.

TSCNV2-- Called by TSCNV1 to compute flow-rate coefficients x_{I1}(JC), x_{I2}(JC), x_{I3}(JC), and x_{I5}(JC).

TSCNV3-- Called by TSCLO, only at the end of a heat-transfer step, to get heat-transfer coefficients H_{erc}(JC) and H_{sic}(JC). Calls TSCNV8 to get liquid heat-transfer coefficients h_c(JC).

TSCNV7-- Computes ρ_c, μ_c for liquid sodium.

TSCNV8-- Computes liquid heat-transfer coefficient h_c (JC).

TSCLM1-- Multiple pin version of TSCNV1.

Auxiliary Routines:

CFUEL-- Calculates the specific heat of the fuel as a function of temperature. For coding efficiency, one call to CFUEL gives the specific heats for all radial nodes at an axial node, rather than using as separate call to CFUEL for each radial node of each axial node.

CCLAD-- Calculates the specific heat of the cladding as a function of temperature. One call to CCLAD returns the values for all axial nodes int eh core and axial blankets.

KFUEL-- Calculates the thermal conductivity of the fuel as a function of temperature. One call to KFUEL provides the values for all radial nodes at an axial node.

KCLAD-- Calculates cladding thermal conductivity as a function of temperature. One call to KCLAD returns the values for all axial nodes in the core and axial blankets.

INTRP-- General interpolation routine that uses linear interpolation from a table to obtain y as a function of x. A single call to INTRP with an array of x's will return an array of resulting y's.

SHAPE-- Multiplies the array of axial power shapes by the current time-dependent power to obtain the current power, in watts, at each axial node in a channel.

HBSMPL-- Calculates the bond gap conductance at each axial node if the simple bond gap conductance model is used. Otherwise, DEFORM-4 calculates the bond gap conductance; and HBSMPL is not called.

INVRT3-- Solves a tri-diagonal matrix equation of arbitrary size.

Figure 3.1–2 is a flowchart for subroutine TSCLO, Figure 3.13–1 is a flowchart for subroutine TSHTRN, and Figure 3.13–2 is a flowchart for subroutine TSHTN3. The logic in the other pre-voiding single pin transient heat-transfer routines is very simple and straightforward. Subroutine TSHTRV is similar to TSHTN3, except that TSHTRV does not calculate coolant and structure temperatures. Also, in TSHTRV the axial nodes are completely de-coupled, so the order in which they are treated is immaterial. For simplicity, the axial node loop in TSHTRV always starts at the bottom and works up.

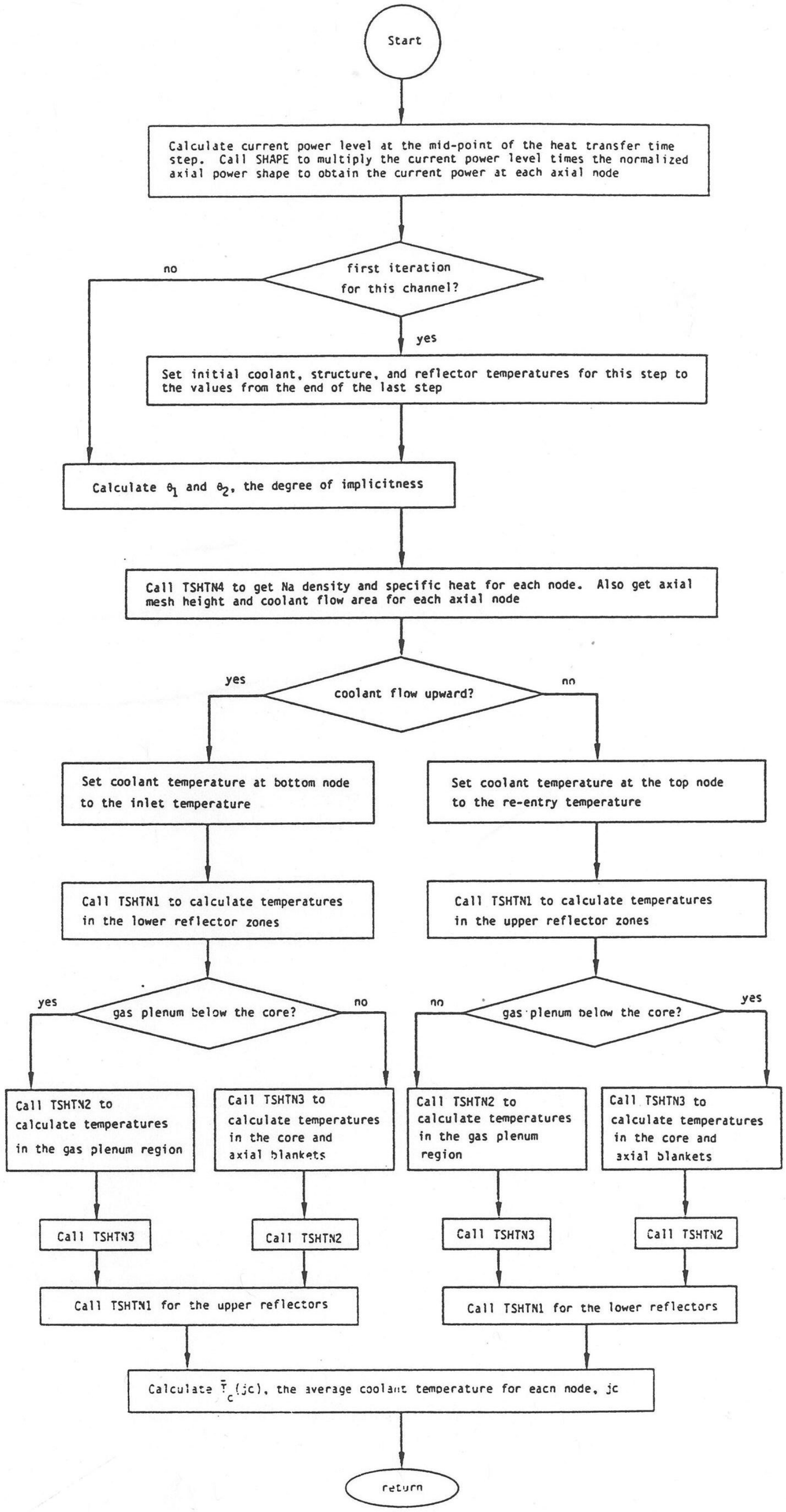

Start
Calculate current power level at the mid-point of the heat transfer time step. Call SHAPE to multiply the current power level times the normalized axial power shape to obtain the current power at each axial node
first iteration for this channel?
no
yes
Set initial coolant, structure, and reflector temperatures for this step to the values from the end of the last step
Calculate θ_1 and θ_2, the degree of implicitness
Call TSHTN4 to get Na density and specific heat for each node. Also get axial mesh height and coolant flow area for each axial node
coolant flow upward?
yes
no
Set coolant temperature at bottom node to the inlet temperature
Set coolant temperature at the top node to the re-entry temperature
Call TSHTN1 to calculate temperatures in the lower reflector zones
Call TSHTN1 to calculate temperatures in the upper reflector zones
gas plenum below the core?
yes
no
gas plenum below the core?
no
yes
Call TSHTN2 to calculate temperatures in the gas plenum region
Call TSHTN3 to calculate temperatures in the core and axial blankets
Call TSHTN2 to calculate temperatures in the gas plenum region
Call TSHTN3 to calculate temperatures in the core and axial blankets
Call TSHTN3
Call TSHTN2
Call TSHTN3
Call TSHTN2
Call TSHTN1 for the upper reflectors
Call TSHTN1 for the lower reflectors
Calculate $\bar{T}_c(jc)$, the average coolant temperature for each node, jc
return

Figure 3.13–1: Flowchart for Subroutine TSHTRN

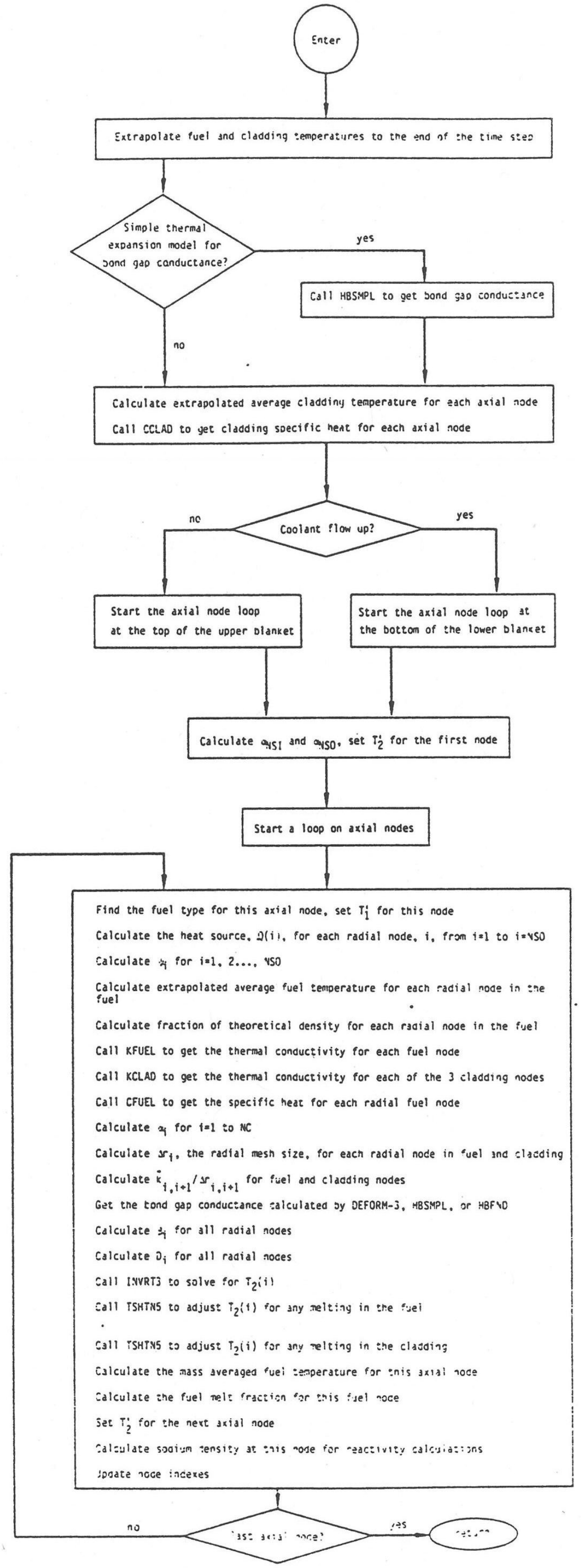

Enter
Extrapolate fuel and cladding temperatures to the end of the time step
Simple thermal expansion model for bond gap conductance?
yes
Call HBSMPL to get bond gap conductance
no
Calculate extrapolated average cladding temperature for each axial node
Call CCLAD to get cladding specific heat for each axial node
no
Coolant flow up?
yes
Start the axial node loop at the top of the upper blanket
Start the axial node loop at the bottom of the lower blanket
Calculate α_{NSI} and α_{NSO}, set T_2' for the first node
Start a loop on axial nodes
Find the fuel type for this axial node, set T_1' for this node
Calculate the heat source, Q(i), for each radial node, i, from i=1 to i=NSO
Calculate α_i for i=1, 2..., NSO
Calculate extrapolated average fuel temperature for each radial node in the fuel
Calculate fraction of theoretical density for each radial node in the fuel
Call KFUEL to get the thermal conductivity for each fuel node
Call KCLAD to get the thermal conductivity for each of the 3 cladding nodes
Call CFUEL to get the specific heat for each radial fuel node
Calculate α_i for i=1 to NC
Calculate Δr_i, the radial mesh size, for each radial node in fuel and cladding
Calculate $\bar{k}_{i,i+1}/\Delta r_{i,i+1}$ for fuel and cladding nodes
Get the bond gap conductance calculated by DEFORM-3, HBSMPL, or HBFND
Calculate a_i for all radial nodes
Calculate D_i for all radial nodes
Call INVRT3 to solve for $T_2(i)$
Call TSHTN5 to adjust $T_2(i)$ for any melting in the fuel
Call TSHTN5 to adjust $T_2(i)$ for any melting in the cladding
Calculate the mass averaged fuel temperature for this axial node
Calculate the fuel melt fraction for this fuel node
Set T_2' for the next axial node
Calculate sodium density at this node for reactivity calculations
Update node indexes
no
last axial node?
yes
return

Figure 3.13–2: Flowchart for Subroutine TSHTN3

3.14 Subchannel Treatment Details

Argonne and the Korea Atomic Energy Research Institute (KAERI) cooperated on an International Nuclear Energy Research Initiative on “Passive Safety Optimization in Liquid Sodium-Cooled Reactors.” Part of this work involved the development, implementation, and testing of a detailed whole core coolant subchannel model for inclusion in the Argonne SAS4A/SASSYS-1 LMR safety analysis code and the KAERI SSC-K systems code [3-8]. The purpose of this work was to increase the accuracy and decrease the uncertainties in calculations of reactor safety margins in accident situations. Steady-state and transient hot channel factors are computed mechanistically with a detailed subchannel model for the core, coupled with a thermal hydraulic treatment of the primary and intermediate heat transport loops. It should be noted here that some of the notation used in this section does not match that used in the preceding sections of this chapter. Therefore, all variables are defined in order to avoid confusion.

3.14.1 Model Features

The new model uses a coolant subchannel treatment similar to that used by the COBRA-4 code [3-9] or the SUPERENERGY-2 code [3-10]. The subchannel treatment includes axial coolant flow parallel to the pins and cross flow between coolant subchannels driven by pressure differences and wire wrap sweeping. Heat flow between adjacent coolant subchannels is calculated, including effects of turbulent mixing. The channel treatment includes the whole length of the subassembly, with the detailed subchannel treatment in the pin section and a simpler treatment above and below the pins.

Figure 3.14–1 shows a typical subchannel treatment for fuel pins with a triangular arrangement in a hexagonal duct. In this arrangement, there are interior subchannels, edge subchannels and corner subchannels. If the pins are wrapped with spacer wires, then there would be wire wrap flow sweeping along the duct walls in the edge and corner subchannels. This is the type of geometry the model is intended for, but the geometry is not hard-wired into the code. The model is flexible enough to model other geometries, such as pins in a square arrangement with grid spacers and no wire wraps. The new model couples with existing models for the remainder of the primary loop, intermediate loop and balance of plant.

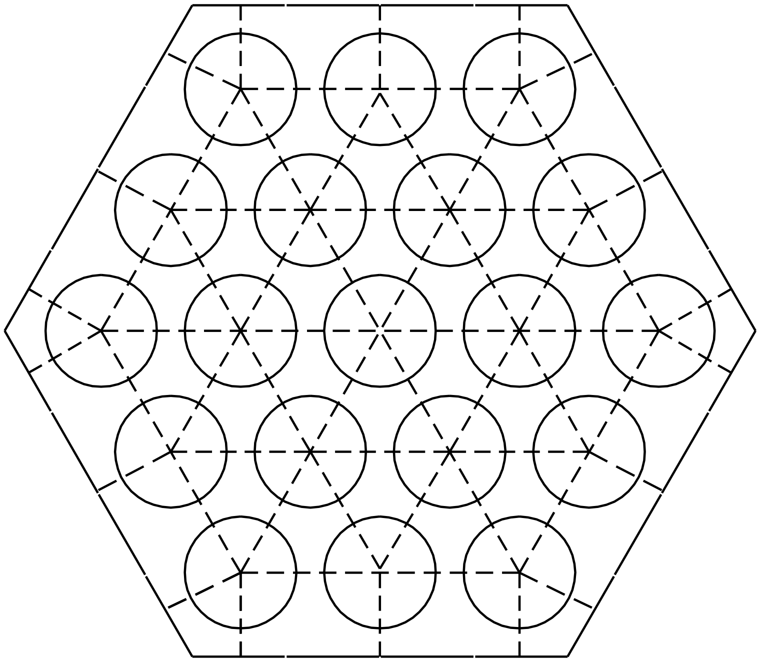


Figure 3.14–1 Coolant Subchannels in a 19 Pin Hex

3.14.2 Basic Equations

Figure 3.14–2 shows the main coolant subchannel variables used in the model for an axial node.

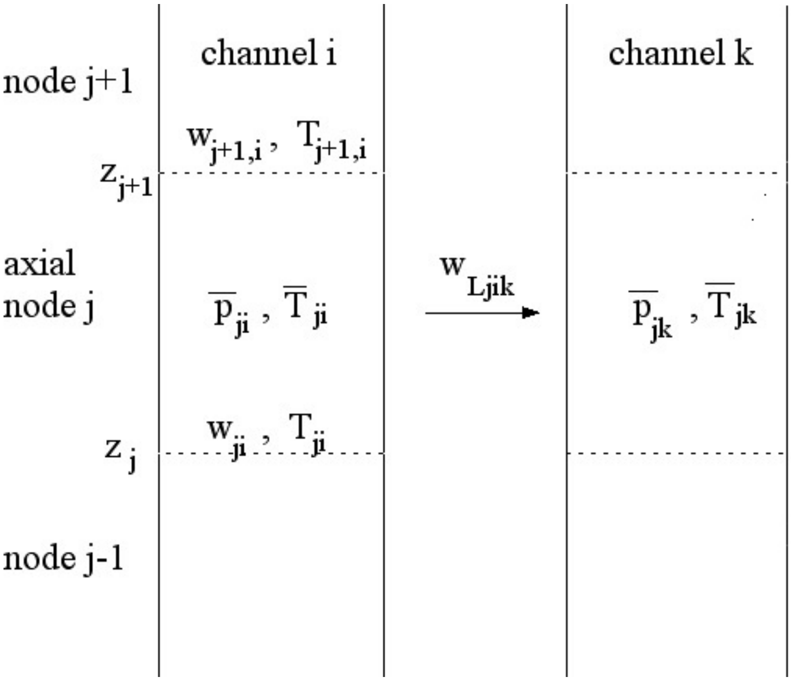

channel i
channel k
node j+1
$w_{j+1,i}$, $T_{j+1,i}$
z_{j+1}
axial node j
$\overline{p}_{ji}$, $\overline{T}_{ji}$
w_{Ljik}
$\overline{p}_{jk}$, $\overline{T}_{jk}$
w_{ji} , T_{ji}
z_j
node j-1

Figure 3.14–2 Coolant Subchannel Variables

The coolant variables in Figure 3.14–2 are defined as:

$\overline{p}_{ji}$ = coolant pressure at the middle of node j in channel i

$\overline{T}_{ji}$ = average coolant temperature for node j in channel i

w_{ji} = coolant flow rate in the axial direction at the bottom of node j in sub-channel i

w_{Ljik} = lateral flow rate from subchannel i to subchannel k at node j

z_j = axial location at the bottom of node j

The basic continuity equation for the coolant in an axial node of a subchannel is

$$\frac{d}{dt}(\overline{\rho}_{ji}A_{ji}\Delta z_j) = \quad w_{ji} - w_{j+1,i} - \sum_k w_{Ljik} \tag{3.14-1}$$

The coolant momentum equation is

$$\frac{1}{2}(\frac{\Delta z_j}{A_{ji}} + \frac{\Delta z_{j-1}}{A_{j-1,i}})\frac{dw_{ji}}{dt} = \overline{p}_{j-1,i} - \overline{p}_{ji} - \Delta p_{frji} - \frac{w_{ji}\,|\,w_{ji}\,|\,K_{orji}}{2\rho_{ji}A_{ji}^2} - \rho_{ji}g(\Delta z_j + \Delta z_{j-1})/2$$

$$-\frac{\overline{w}_{ji}^2}{\overline{\rho}_{ji}A_{ji}^2} + \frac{w_{ji}^2}{\rho_{ji}A_{ji}^2} - \frac{w_{ji}^2}{\rho_{ji}A_{j-1,i}^2} + \frac{\overline{w}_{j-1,i}^2}{\overline{\rho}_{j-1,i}A_{j-1,i}^2} - \frac{1}{2}\sum_k \; [(1 - S_{wjik})w_{Ljik}(\frac{w_{jk}}{\overline{\rho}_{jik}A_{ji}A_{jk}} \tag{3.14-2}$$

$$-\frac{w_{ji}}{\overline{\rho}_{jik}A_{ji}^2}) + (1 - S_{wj-1,ik})w_{Lj-1,ik}(\frac{w_{j-1,k}}{\overline{\rho}_{j-1,ik}A_{j-1,i}A_{j-1,k}} - \frac{w_{j-1,k}}{\overline{\rho}_{j-1,ik}A_{j-1,i}^2})]$$

The energy equation for the coolant is

$$V_{cji}C_{prefji}\frac{d}{dt}[\overline{\rho}_{ji}(\overline{T}_{ji} - T_r)] = \sum_{in} w_{inji}(T_{inji} - T_r)C_{prefji} - \sum_{out} w_{outji}(T_{out} - T_r)C_{prefji} + \varphi_{ji} \tag{3.14-3}$$

The energy equation in the fuel is

$$\rho_f C_f \frac{dT_f}{dt} = \frac{1}{r}\frac{d}{dr}(k_f r \frac{dT_f}{dr}) + Q \tag{3.14-4}$$

A similar equation is used for the cladding, and a bond gap conductance is used between the fuel outer surface and the cladding inner surface. In the above equations:

A = coolant flow area in the subchannel

C_f = heat capacity of the fuel

C_{pref} = coolant heat capacity, evaluated at T_r

g = acceleration of gravity

i = subchannel number

j = axial node number

k = subchannel number of a connecting subchannel

k_f = fuel thermal conductivity

K_{or} = orifice coefficient

Δp_{frji} = friction pressure drop in the bottom half of node j and the top half of node j-1

Q = heat source per unit volume in the fuel

S_{wjik} = 1 if $w_{ljik} \geq 0$,
0 otherwise

t = time

T_f = fuel temperature

T_{in} = temperature of coolant entering the node, either from another axial node in the same subchannel or from a connecting subchannel

T_{out} = temperature of coolant leaving the node

T_r = reference temperature, $\overline{T}$, at the beginning of the time step is used for T_r.

V_c = coolant volume in the axial node

z = axial distance

ρ_j = coolant density at the bottom of node j

$\overline{\rho}_j$ = coolant density at the middle of node j

ρ_f = fuel density

φ_{ji} = heat source to the coolant node

$$\phi_{ji} = \phi_{cji} + \phi_{\gamma ji} + \phi_{scji} + \phi_{chchji} \tag{3.14-5}$$

ϕ_{cji} = heat source from the cladding and structure surfaces to the coolant

$\phi_{\gamma ji}$ = direct heat source to the coolant from neutrons and gamma rays

ϕ_{scji} = heat source from turbulent mixing and conduction from adjacent sub-channels

ϕ_{chchj} = heat source to the coolant from channel-to-channel heat transfer

Also,

$$\phi_{scji} = \sum_{k} [u_{1ik} k_j + u_{2ik} C_j (\overline{w}_{ji} + \overline{w}_{jk})](\overline{T}_{jk} - \overline{T}_{ji}) \Delta \mathrm{z_j} \tag{3.14-6}$$

where

u_{1ik} = geometry factor for conduction from subchannel i to subchannel k

u_{2ik} = turbulent mixing factor

$\Delta \mathrm{z_j}$ = axial node length

The parameters u_{1ik} and u_{2ik} are geometry-dependent factors supplied by the user for the particular geometry being used.

Based on a numerical simulation [3-11] with the CFX code for a triangular array of rods, the conduction geometry factor can be obtained from

$$u_{1ik} = 0.7774 \left(\frac{s}{L_c}\right)\left(\frac{P}{D}\right)\left(\frac{s}{D}\right)^{-0.2627} \Delta \mathrm{z_j} \tag{3.14-7}$$

where

s = gap size between subchannels

L_c = centroid distance between subchannels

This correlation only accounts for thermal conduction in the coolant. It may be desirable to increase u_{1ik} to account for circumferential thermal conduction in the clad and possibly conduction in the fuel.

The value for u_{2ik} can be obtained from the parameter ε^* used by Cheng and Todreas [3-12].

$$u_{2ik} = \varepsilon^*{}_{\mathrm{i}\acute{\eta}}\mathrm{c}/(\mathrm{A_{ci}} + \mathrm{A_{ck}}) \tag{3.14-8}$$

c = s = gap width

A_{ci} = coolant flow area in subchannel *i*

The coolant lateral flow rate is given by

$$w_{Ljik} = K_{sik} \Delta z_j \frac{(w_{ji} + w_{j+1,i})}{2} + w_{Lpjik} \tag{3.14-9}$$

K_{sik} = wire wrap sweeping factor

In this equation the first term is the net flow due to wire wrap sweeping, and the second term is the pressure driven subchannel-to-subchannel flow. Note that K_{sik} = 0 unless there is net sweeping from subchannel i to subchannel k. For a gap between an inner subchannel and another inner subchannel or between an inner subchannel and an edge subchannel there are two wrapper wires involved; one on each pin. The two wires sweep in opposite directions, so the net sweep flow is approximately zero. Normally the wire wrap sweeping factor is zero unless i and are both edge or corner subchannels.

The second term in equation 3.14-9 is determined by

$$(\frac{L}{A})_{Lat} \frac{dw_{Lpjik}}{dt} = \overline{p}_{ji} - \overline{p}_{jk} - \frac{w_{Lpjik} \mid w_{Lpjik} \mid}{2\overline{\rho}_{jik} A_L^2} K_{Lat} \quad \text{if} \mid \overline{p}_{ji} - \overline{p}_{jk} \mid > \Delta p_{lam} \tag{3.14-10}$$

or

$$(\frac{L}{A})_{Lat} \frac{dw_{Lpjik}}{dt} = \overline{p}_{ji} - \overline{p}_{jk} - K_{Lat} \frac{w_{lam} w_{Lpjik}}{2\overline{\rho}_{jik} A_L^2} \quad \text{otherwise} \tag{3.14-11}$$

where

K_{Lat} = effective lateral flow orifice coefficient

$(\frac{L}{A})_{Lat}$ = lateral inertia term

Δp_{lam} = user specified transition pressure difference

and

$$w_{lam} = \frac{2\overline{\rho}_{jik} A_L^2}{K_{Lat}} \Delta p_{lam} \tag{3.14-12}$$

K_{sik} is supplied by the user. It can be obtained from the parameter C_{1L} used by Cheng and Todreas:

$K_{sik} = c\ C_{1L}/A_c$

Cylindrical geometry with radial heat conduction is used for calculating the fuel pin temperatures. Figure 3.14–3 shows the radial node structure used for a fuel pin.

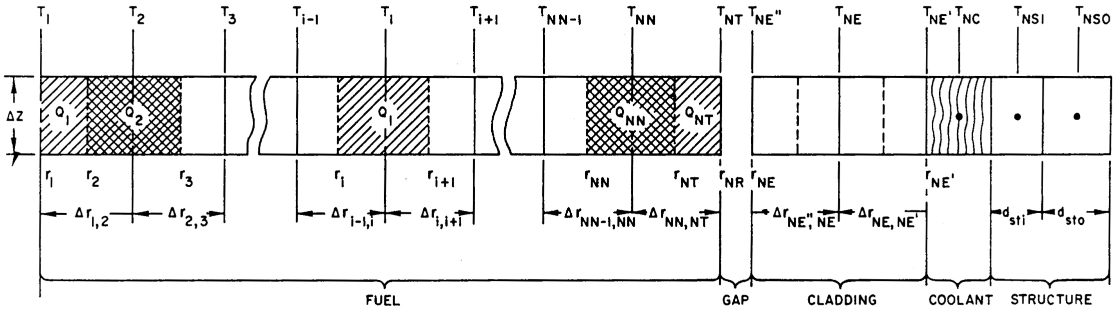


Figure 3.14–3 Radial Node Structure Used for a Fuel Pin, with Coolant and Structure

For an interior fuel node, 1 < i < NT, the conduction equation (equation 3.14-4) becomes:

$$m_{fi} C_{fi} \frac{dT_{fi}}{dt} = 2\pi r_{i+1} \bar{k}_{i,i+1} (T_{i+1} - T_i) + 2\pi r_i \bar{k}_{i-1,i} (T_{i-1} - T_i) + Q_i \tag{3.14-13}$$

where

m_{fi} = fuel mass per unit height in node i

Q_i = heat source per unit height in node i

$\bar{k}_{i,i+1}$ = effective average thermal conductivity for heat flow from node i to i+1

Similar equations are used for node 1, node NT and the cladding.

3.14.3 Numerical Procedures

The solution methods used for the subchannel treatment are dictated by stability requirements and by the features that the model includes. In the coolant, a compressible treatment is desired to provide the pressures to drive subchannel-to-subchannel flow rates. Equation 3.14-1 above implies a compressible treatment. With a compressible treatment and using explicit time differencing, the Courant limit would apply. The sonic speed in sodium is about 2400 m/s. A typical axial node size is 0.03 m. Thus the Courant limit would restrict the time step size to about 10^{-5} seconds or less. To avoid such a small limit, an implicit solution is used. For one time step, the pressures and flows in all channels representing a subassembly are solved for simultaneously.

For the coolant temperatures a similar limit applies. The stability limit for an explicit solution is given by the axial node size divided by the coolant velocity. A node size of 0.03 m and a coolant velocity of 7 m/s corresponds to a limit of about 0.004 seconds. Thus, implicit time differencing is also used in this case and the coolant temperatures in a subassembly are solved for simultaneously. Also, it is probably necessary to solve for coolant temperatures simultaneously with the pressures and flows in order to obtain a stable solution. Therefore, all coolant temperatures, pressures, and flows in a subassembly are solved for simultaneously. Fortunately the situation with subassembly-to-subassembly heat transfer is different. In the PRISM design the hex can wall thickness is 0.0039 m, and the duct gap is 0.0043 m, giving a stability limit of about 2.6 seconds for explicit can wall-to-can wall heat transfer. Therefore, the subassembly-to-subassembly heat transfer can be treated explicitly, using conditions at the beginning of the time step. This means that for a given time step each subassembly can be treated separately using time step sizes of 1.0 second or more.

The steady-state calculation begins with an initial approximation of the steady-state coolant temperatures, pressures, and flow rates for each subassembly where the heat flux from each fuel pins to the coolant is determined by the steady-state pin power. Then a null transient is run for each subassembly separately, neglecting subassembly-to-subassembly heat transfer. Finally, a null transient with subassembly-to-subassembly heat transfer is run for all subassemblies. During both null transients the powers and subassembly coolant inlet flows are held constant. After the coolant conditions are calculated, the fuel pin and structure temperatures are calculated.

A time step approach is used both for the steady-state null transient and for the regular transient. Conditions are known at the beginning of the time step, and the main computational task is to determine the conditions at the end of the step. The equations are linearized about values at the beginning of the time step, and fully implicit finite differencing in time is used for the basic conservation equations. This leads to N linear equations in N unknowns. The unknowns are solved for by iteration. Explicit time differencing is used for the subassembly-to-subassembly heat transfer, so the calculations for one time step for each subassembly can be done independently.

For a transient time step the heat flux from the fuel pins and structure to the coolant is approximated as

$$\phi_{cji} = \phi_{c1ji} + \phi_{c2ji} \Delta \overline{T}_{ji} \tag{3.14-14}$$

for the coolant calculations. The coefficients ϕ_{c1} and ϕ_{c2} are calculated in the fuel pin heat transfer routines. In this equation ΔT is the change in coolant temperature during the time step. After the coolant temperatures are calculated, the fuel pin and structure temperatures are calculated for the time step.

3.14.4 Interfacing with Other Models in the Code

This section summarizes the interfacing between the existing SAS4A/SASSYS-1 code and the new three-dimensional thermal hydraulics core model. The main interactions between the new model and the rest of the code are with the input module, output module, neutronics module, DEFORM-5 fuel pin mechanics module for metal fuel, simple PRIMAR-1 primary loop module, and detailed PRIMAR-4 module for the primary and intermediate heat transfer loops. These interactions and the interface data requirements are discussed in the sections below.

Note that the new subchannel model is only applicable to single-phase coolant with intact fuel pins. The SAS4A/SASSYS-1 sodium boiling model and PLUTO and LEVITATE disrupted fuel pin modules cannot be used in the same subassembly as the coolant subchannel module.

3.14.4.1 Input Module

The SAS4A/SASSYS-1 code reads input into numbered locations in input blocks. Extra space has been provided in these input blocks for future use by new modules. Some of this extra space is used by the new input variables for the new three dimensional thermal hydraulics core module. The new module also uses many existing input variables. There is enough extra space in each input block to accommodate the new variables. No modifications are required to the input module for use with the new module. The new input variables are listed in the user guide section below.

3.14.4.2 Output Module

The output files for a whole-core case using the new model can get very large because of the large number of channels involved. Therefore, some modification of the output module has been necessary. For the printed output one change that was necessary was to print reactivity feedback by subassembly rather than by channel. Output on CHANNEL.dat is binary data intended for input to plotting programs. Previously one record per channel per time step was put on CHANNEL.dat. This has been modified to output every N time steps for specified channels or subassemblies.

3.14.4.3 Neutronics Module

The neutronics module uses fuel, cladding, and coolant temperatures and coolant densities from the core thermal hydraulics module to obtain reactivity feedback components at the end of each transient main time step. For each transient step, the neutronics module supplies the core thermal hydraulics module with the power level. The coupling with the neutronics module is the same for the new three dimensional thermal hydraulics core module as for the older core thermal hydraulics module.

3.14.4.4 DEFORM-5 Fuel Pin Mechanics for Metal Fuel

The DEFORM-5 module uses fuel and cladding temperatures from the core thermal hydraulics module for its prediction of pin failure. The coupling with the DEFORM-5 module is the same for the new three-dimensional thermal hydraulics core module as for the older core thermal hydraulics module.

3.14.4.5 PRIMAR-1

SAS4A/SASSYS-1 uses either the simple PRIMAR-1 primary loop module or the detailed PRIMAR-4 module for the primary and intermediate heat transport systems. When PRIMAR-1 is used, the initial steady-state subassembly inlet pressure and average core outlet temperature calculated by the core thermal hydraulics module are used by PRIMAR-1 to obtain the initial steady-state primary loop gravity head and pump head. During the transient calculation, PRIMAR-1 provides the subassembly inlet pressure and inlet temperature to the core thermal hydraulics module. No transient information from the core thermal hydraulics module is used by PRIMAR-1.

3.14.4.6 PRIMAR-4

The core thermal hydraulics module and PRIMAR-4 are tightly coupled, and the transient coupling with PRIMAR-4 required more effort in the development of the new core thermal hydraulics module than the coupling with any other module. This coupling also contributes significantly to the running time of the new module.

The steady-state coupling between the core thermal hydraulics and PRIMAR-4 is similar to the coupling with PRIMAR-1. The user specifies the subassembly outlet pressure and inlet temperature in the input. The initial coolant flow rate in each channel is also specified by the user. Then the core thermal hydraulics module calculates the outlet temperature and inlet pressure for each subassembly. Inlet orifice coefficients are adjusted so that the inlet pressures for all subassemblies match the maximum inlet pressure. The subassembly inlet and outlet pressures, inlet temperatures and average outlet temperature are used in the steady-state PRIMAR-1 and PRIMAR-4 calculations. This steady-state coupling is the same with the new core thermal hydraulics model.

During the transient calculation, PRIMAR-4 provides the subassembly inlet pressure and inlet temperature to the core thermal hydraulics module in the same way that PRIMAR-1 does. The big difference between the coupling with PRIMAR-4 and PRIMAR-1 is that during a transient time step, PRIMAR-4 estimates the core flows and outlet temperatures before they are calculated by the core thermal hydraulics module. With the previous thermal hydraulics module at the end of each PRIMAR time step, the core module calculates the coefficients $C_0, C_1, C_{2,}$ and C_3 for each channel for use by PRIMAR-4. At the beginning of the next PRIMAR time step, PRIMAR-4 gets the channel flow rates from the core thermal hydraulics module and estimates the channel flow at the end of the step using

$$\frac{dw_{Li}}{dt} = C_{0Li} + C_{1Li} p_{in} + C_{2Li} p_x + C_{3Li} w_{Li} \mid w_{Li} \mid \tag{3.14-15}$$

where

w = coolant flow rate

t = time

L = 1 for the subassembly inlet, 2 for the subassembly exit

i = channel number

With the new core thermal hydraulic treatment, at the beginning of each transient PRIMAR-4 time step, after the time step size has been determined, the core thermal hydraulics module is called to provide new coefficients a_0, a_1 and a_2 for each subassembly. PRIMAR-4 then estimates the coolant flow at the end of the PRIMAR step using

$$w_{Ln}(t_p + \Delta t) = w_{Ln}(t_p) + (a_{0Ln} + a_{1Ln}\Delta p_{in} + a_{2Ln}\Delta p_x)\Delta t \tag{3.14-16}$$

where

n = subassembly number

t_p = time at beginning of the PRIMAR step

Δ t = PRIMAR time step size

Δp_{in} = change in inlet pressure during the PRIMAR step

Δp_x = change in outlet plenum pressure during the PRIMAR step

Note that with the proper choice of values for the coefficients, Equation 3.14-15 and Equation 3.14-16 can be made approximately equivalent, but there are two significant differences between them. First when using the previous core thermal hydraulics module, the C's are calculated for each channel, based on the assumption that one channel represents one or more subassembly. With the new core thermal hydraulics module, a number of channels can be used to represent one subassembly, and the a's of Equation 3.14-16 are calculated for each subassembly rather than for each channel. Second, the coefficients in Equation 3.14-15 are calculated before the new PRIMAR time step size is known; and the coefficients are independent of time step size. On the other hand, the coefficients in Equation 3.14-16 are calculated after the new PRIMAR time step size is known, and they may depend on the size of the time step. The previous core thermal hydraulics module uses incompressible coolant flow, whereas the new module uses a compressible treatment for the coolant. With an incompressible treatment, it is possible to obtain C's for Equation 3.14-15 that are approximately correct for any reasonable time step size. On the other hand, with a compressible treatment the values of the coefficients in Equation 3.14-16 will depend strongly on whether or not the time step is large enough for a sound wave to travel from one end of the subassembly to the other and back in the time step.

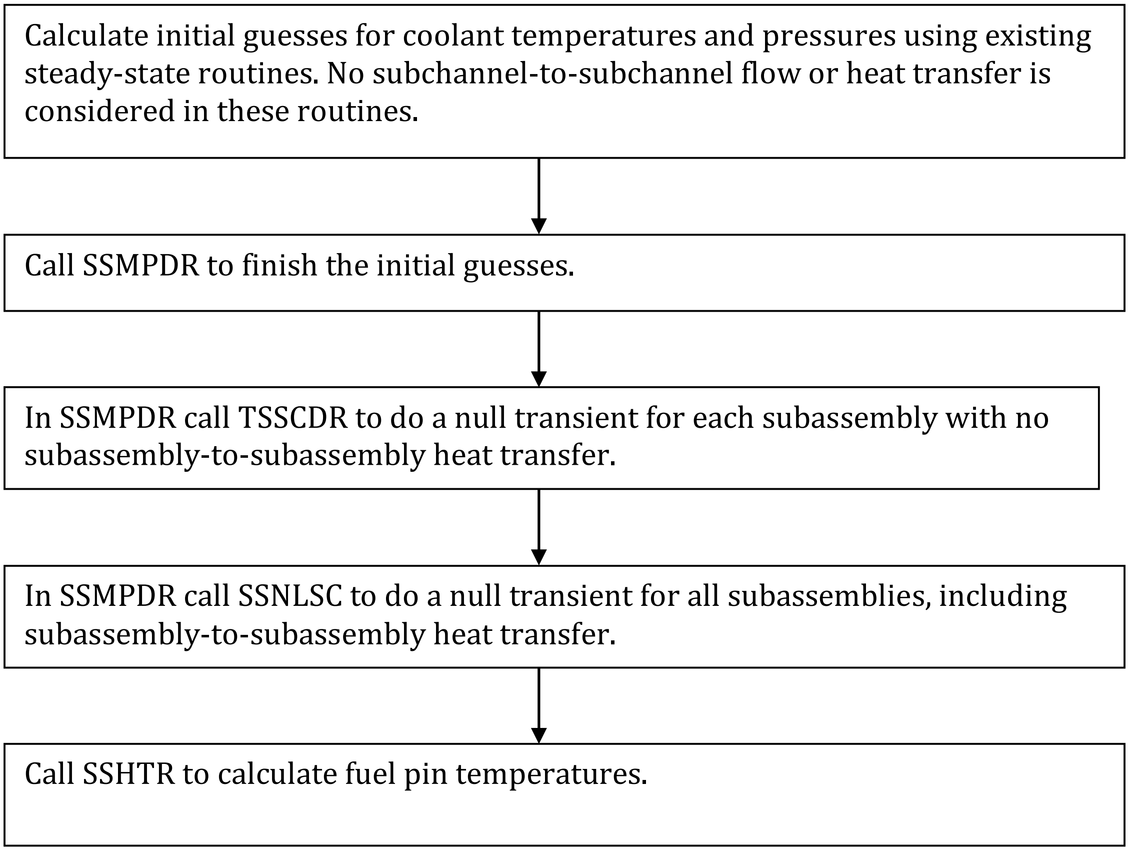
3.14.5 Data Management

A very flexible data management scheme is required for the detailed subchannel model because of the range of size of the cases that will be run and because a lot of the data is needed simultaneously in the solution. The range of sizes of the cases expected to be run with the model is very large: from a single subassembly containing a few pins and a few coolant subchannels to a large whole core case containing hundreds of subassemblies with hundreds of channels per subassembly. Also, in the calculations for a time step the equations for all channels in a subassembly are solved simultaneously, requiring simultaneous access to the data for all of the channels in the subassembly.

In order to provide the flexibility needed by the model, most of the data is stored in dynamically allocated storage containers whose sizes are determined at run time based on the size of the case being run. A structured pointer system is used to access the data. The permanent channel dependent variables, which are saved from one time step to the next, are stored in a large container and accessed with channel-dependent pointers. The temporary variables used in the simultaneous solution for all of the channels in a subassembly are stored in a number of containers also accessed with pointers. Thus small cases can be run with a small amount of memory, and the available memory can be apportioned efficiently to run large cases without re-compiling the code.

3.14.6 Coding Overview and Subroutines

Figure 3.14–4 shows an overview of the steady-state calculations.



Calculate initial guesses for coolant temperatures and pressures using existing steady-state routines. No subchannel-to-subchannel flow or heat transfer is considered in these routines.

Call SSMPDR to finish the initial guesses.

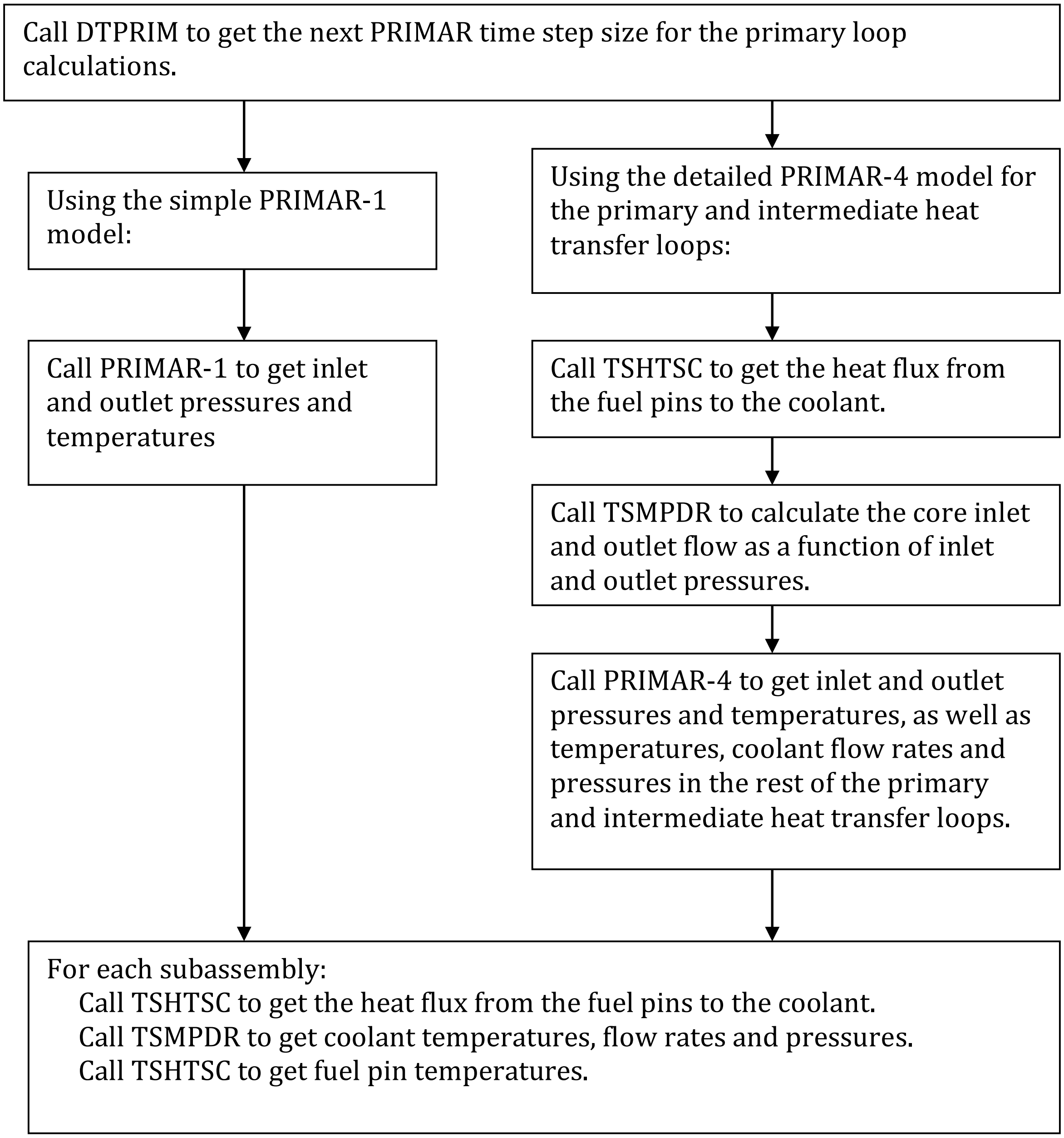
In SSMPDR call TSSCDR to do a null transient for each subassembly with no subassembly-to-subassembly heat transfer.

In SSMPDR call SSNLSC to do a null transient for all subassemblies, including subassembly-to-subassembly heat transfer.

Call SSHTR to calculate fuel pin temperatures.

Figure 3.14–4 Coding Overview of the Steady-State Calculation for the Detailed Subchannel Model

Figure 3.14–5 shows an overview of the transient calculations for the model. It should be noted that SSHTR is a previously existing routine in the SAS4A/SASSYS-1 code. Also, TSHTSC is based mainly on the previously existing routine TSHTRV, with some modifications. Listings of the subroutines used in the new model are given in Table 3.14-1, Table 3.14-2, and Table 3.14-3.



Call DTPRIM to get the next PRIMAR time step size for the primary loop calculations.

Using the simple PRIMAR-1 model:

Call PRIMAR-1 to get inlet and outlet pressures and temperatures

Using the detailed PRIMAR-4 model for the primary and intermediate heat transfer loops:

Call TSHTSC to get the heat flux from the fuel pins to the coolant.

Call TSMPDR to calculate the core inlet and outlet flow as a function of inlet and outlet pressures.

Call PRIMAR-4 to get inlet and outlet pressures and temperatures, as well as temperatures, coolant flow rates and pressures in the rest of the primary and intermediate heat transfer loops.

For each subassembly:
Call TSHTSC to get the heat flux from the fuel pins to the coolant.
Call TSMPDR to get coolant temperatures, flow rates and pressures.
Call TSHTSC to get fuel pin temperatures.

Figure 3.14–5 Coding Overview for One Time Step of the Transient Calculation for the Detailed Subchannel Model

Table 3.14-1: New Subroutines – Detailed Coolant Subchannel Model

Subroutine	Description
CHCHFL	Calculates subassembly-to-subassembly heat flow from the outer structure node of channel i to the outer structure node of channel j. Also contains an option to transfer heat from the outer structure node of channel i to the coolant of channel j. In the future this subroutine will be modified to include an option to transfer heat from the outer structure node of channel i to a constant temperature heat sink. This is a modified version of a SAS4A/SASSYS-1 subroutine.
COOLFN	Finishes the coolant calculations for one time step for one group of channels after SOLVIT has calculated the changes in the main variables.
COOLPT	Prints the coolant results for one channel at the end of a time step.
DYNAL2	Dynamic storage allocation for containers for temporary storage for the coolant model.
INVTRI	Solves a general tri-diagonal matrix equation.
LATPRT	Prints coolant lateral flow rates.
SBFT24	Writes selected binary output on unit 24. Section 3.15.3.1.4 has more detail on the unit 24 output.
SCCOEF	Calculates the coefficients of the linearized equations for the coolant model for one group of channels for one time step.
SOLVIT	Solves the equations set up by SCCOEF to obtain the changes in the coolant independent variables for a time step.
SSMPDR	Steady-state driver for the detailed coolant subchannel model. Calculates initial approximate values for coolant variables. Then runs a null transient for each subassembly without subassembly-to-subassembly heat transfer. Finally runs a null transient for the whole core with subassembly-to-subassembly heat transfer.
SSNLSC	Driver for the steady-state null transient with subassembly-to-subassembly heat transfer.
SSSTR	Calculates steady-state temperatures in the structure and in the reflectors.
TSHTSC	Calculates transient temperatures in the fuel pins. Also calculates structure and reflector temperatures. A modified version of TSHTRV from SAS4A/SASSYS-1.
TSMPDR	Transient driver for the detailed subchannel model. Calculates one coolant time step for one group of channels.

Table 3.14-2: SAS4A/SASSYS-1 Subroutines Used with Detailed Subchannel Model

Subroutine	Description
CCLAD	Cladding heat capacity.
CFUEL	Fuel heat capacity.
DATMOV	Copies data packs from the dynamically allocated memory container to the regular common blocks and moves them back.
HBSMPL	Simple model for the bond gap conductance between the fuel and the cladding.
INVRT3	Solves a tri-diagonal matrix equation for temperatures. Note, INVRT3 and INVTRI have similar uses, but INVTRI can handle a more general tri-diagonal matrix.
KFUEL	Calculates fuel thermal conductivity.
LINES	Counts the number of lines that have been printed and puts page headings in the output.
POINST	Calculates pointers used by DATMOV and in the memory management.
READIN	Reads the input data
RESTAR	Reads or writes a restart file.
SHAPE	Sets the axial power shape in the fuel pin section of a channel.
SSCOOL	Calculates steady-state coolant temperatures and pressures ignoring subchannel-to-subchannel cross flow and heat transfer. Used for initial conditions for the steady-state null transient.
SSHRT	Steady-state fuel pin temperatures.
TSHTN5	Calculates melting of the cladding.
TSHTN6	Calculates melting of the fuel.
TSPRNT	Prints thermal hydraulic output for selected channels.

Table 3.14-3: SAS4A/SASSYS-1 Drivers that Call Detailed Subchannel Model

Subroutine	Description
INPDRV	Input driver, calls READIN
SSTHRM	Steady-state thermal hydraulics driver for the core channel calculations.
TSTHRM	Transient thermal hydraulics driver for the core channel calculations.

3.14.7 Required Computer Capabilities

One question that has been addressed is whether it is practical to run a detailed whole core case on currently available computers. There are two main computer-related considerations. One consideration is whether currently available computers have enough memory for a whole core case. The other consideration is whether computers are fast enough to run a large case in a reasonable amount of time.

3.14.7.1 Memory Requirements

For the detailed subchannel model the SAS4A/SASSYS-1 code requires 0.13 or 0.17 MB of main memory per channel, depending on whether or not the fuel pin mechanics model in the code is used in addition to the subchannel thermal hydraulics model. An additional 8 MB are used for coding, and a few MB are used for non-channel dependent memory.

To determine how many channels are required for a detailed treatment, consider the ALMR mod B reactor design, which is a moderately sized reactor, with a power rating of 840 MWt. This reactor contains 108 driver subassemblies with 271 pins and 546 coolant subchannels per subassembly, 84 blanket subassemblies with 127 pins and 258 coolant subchannels per subassembly, 66 shield subassemblies with 7 pins and 18 coolant subchannels per subassembly, and a few other subassemblies. With a one computational channel per subchannel treatment, a whole core case could therefore use about 81,000 channels. This would require 10.5 – 13.8 GB of memory, although if the core loading is symmetrical the number of computational channels and the memory requirements could be reduced by making use of the symmetry. Liquid metal reactor designs about four times as large as the ALMR reactor have been considered. A detailed analysis of such a design might require 40 – 50 GB of memory. Some current computers are limited to a maximum of 2 GB of memory, although many computers have more memory. Accessing more than 2 GB of memory would probably require the use of 64 bit integers.

3.14.7.2 Speed Considerations

Timing data for the new detailed subchannel model indicates that the time required on an old Sun Blade computer with a processor speed of 500 MHz is about 8 ms per time step per channel. Thus, even if this computer had enough memory to run the 81,000 channel ALMR case discussed above, the time required for a solution for 1000 time steps could be 650,000 seconds, or 7.5 days. Computers with processor speeds significantly greater than 500 MHz are currently available, but it appears that running a large case will require significant amounts of computing power and running time.

3.14.7.3 Use of a Multi-Processor Cluster

Because of the memory and speed considerations listed above, it might appear that the most practical way to run large cases with the new model is with the use of a cluster of processors, with one or more subassemblies run on each processor. The model was developed with that in mind. However, the older modules in SAS4A/SASSYS-1 were written before clusters of processors were being considered, and converting the whole code to run on multiple processors would require a great effort. Currently the approach is to model part of the core with the detailed subchannel model and to model the rest of the core with the older, less detailed models which require considerably less memory and computing time. This approach works well with the current code and current computers. High-end currently available computers have enough memory and speed to run large detailed subchannel cases on a single computer.

3.15 Input and Output

3.15.1 Input Description

The input variables used in the pre-voiding subassembly thermal hydraulics calculations are listed in Table 3.14-1. Some additional comments on some of these input variables are listed below.

3.15.1.1 Per Pin Basis

All of the core channel thermal hydraulic input to SAS4A/SASSYS-1 is on a per pin basis rather than a per subassembly basis. Thus, the initial flow rate, W0, is kg/s per pin; and the perimeters SRFSTZ and SER are perimeters per pin. Also, ACCZ is a coolant flow area per pin, and the variables DZIAB and DZIAT are ratios of inertial lengths to flow areas per pin.

3.15.1.2 Structure/Duct Wall and Wrapper Wires

Typically a SAS4A/SASSYS-1 channel represents an average pin in a subassembly. In this case, the structure normally represents one pin's share of the duct wall, and it may also include the wrapper wire. If there are N pins in the subassembly, then the thickness used for the structure is the actual duct wall thickness, and the perimeter used for the structure is the duct wall perimeter divided by N. The wrapper wire can be either lumped with the cladding or included in the structure. Since wrapper wires are in much better contact with the cladding than with the duct wall, the most accurate treatment of the wrapper wire is probably to lump it with the cladding by increasing the specific heat of the cladding by enough to account for the total heat capacity of the cladding plus the wrapper wire. The cladding dimensions would not be changed.

The problem with lumping the wrapper wires with the duct wall to produce a single "structure" is that typically the duct wall has a much larger ratio of volume to wetted surface area than a wrapper wire, so the wrapper wire temperature will respond much more rapidly than the duct wall temperature to a change in coolant temperature. The heat capacity of the duct wall is considerably greater than the heat capacity of all of the wrapper wires in a subassembly, but the total perimeter of the wrapper wires is greater than the perimeter of the duct wall. If the duct wall and wrapper wires are lumped together, then the thickness of the structure should be determined by the thickness of the duct wall, since the duct wall contains most of the heat capacity. Then the perimeter of the structure should be chosen to conserve the total volume or total heat capacity of the duct wall plus wrapper wires.

A SAS4A/SASSYS-1 channel can be used to represent a central pin rather than an average pin in a subassembly. In this case, the duct wall would probably be ignored, and the "structure: would represent only the wrapper wire. The perimeter of the structure would equal the perimeter of a wrapper wire, and the thickness of the structure would equal one half of the wrapper wire radius in order to conserve volume.

Table 3.15-1: Subassembly Thermal Hydraulics Input Variables

Variable	Reference Eq. No.	Input Variable	Input Block	Location Number	Suggested Value	External Reference
		NCHAN	1	1	1-34	
		IFUEL1	1	3	1-8	
		ICLAD1	1	4	1-3	
		ITKEL	1	7	0 or 1	
		IPOWOP	1	9	1	
		MAXSTP	1	11	--	
		IPO	1	12	20-50	
		IPOBOI	1	13	20-50	
		IBLPRT	1	14	0	
		INAS3D	1	29	0	
		ISSNUL	1	87	--	
		IPRSNL	1	88		
		EPSTEM	11	1	.1 or less	
		DTMXB	11	6	.01	
		DTFUEL	11	10	50.	
		DTCLAD	11	11	30.	
		POW	12	1		
ρ	3.3.1	COEFDS(1)	13	1	11080.	
		COEFDS(2)	13	2	$2.04\text{x}10^{-5}$	
		COEFDS(3)	13	3	$8.70\text{x}10^{-9}$	
k	3.3.1	COEFK	13	4		
k	3.3.1	EXKTB	13	11		
		EXKTM	13	71		
ρ	3.3.1	RHOTAB	13	91		
		RHOTEM	13	251		
k	3.3.1	XKTAB	13	420		
		XKTEM	13	580		
$\bar{c}_f$	3.3-16	CPFTAB	13	606		
		CPFTEM	13	766		
T_{sol}	3.3-92	TFSOL	13	786		
T_{liq}	3.3-93	TFLIQ	13	794		

Variable	Reference Eq. No.	Input Variable	Input Block	Location Number	Suggested Value	External Reference
U_{melt}	3.3-3	UFMELT	13	802		
U_{melt}	3.3-3	UEMELT	13	816		
c_e	3.3-32	CPCTAB	13	819		
$\rho_e c_e$		COCTEM	13	879		
		CROETB	13	990		
		CROETM	13	1050		
p_x	3.12-1	PX	14	1		
T_{in}	3.3-101	TOTAB	14	45		
		TOTME	14	65		
$z_{p\ell\ell}$	3.9-30	ZPLENL	14	87		
$z_{p\ell u}$	3.9-27	ZPLENU	14	88		
M_{mix}	3.3-99	XMXMSI	14	93		
M_{mix}	3.3-99	XMXMS0	14	94		
τ_{mix}	3.3-101	TIMMIX	14	95		
		IRHOK	51	3		
		NPLN	51	4	2-6	
		NREFB	51	5	1-5	
		NREFT	51	6	1-5	
		NZNODE	51	7		
NT		NT	51	14	4-11	
		IFUELV	51	15	1-IFUEL1	
		IFUELB	51	16	1-IFUEL1	
		ICLADV	51	17	1-ICLAD1	
		NGRDSP	51	18	0-10	
		IHGAP	51	24	0 or 1	
		NPIN	51	25	1 or more	
		NSUBAS	51	26	1 or more	
		MZUB	51	27	0-24	
		MZLB	51	28	0-24	
		ILAG	51	34	0	
		IEQMAS	51	118	0	
		IFRFAC	51	186		
		NCHCH	51	205		
		ICHCH	51	206		
		JJMLTP	51	214		

Variable	Reference Eq. No.	Input Variable	Input Block	Location Number	Suggested Value	External Reference
		IAXCON	51	284		
A_c	3.3-5	ACCZ	61	1	>0.	
Δ_z	3.3-6	AXHI	61	8	>0.	
D_h	3.3-9	DHZ	61	32	>0.	
d_{sti}	3.3-45	DSTIZ	61	39	>0.	
d_{sto}	3.3-47	DSTOZ	61	46	>0.	
		PLENL	61	53	>0.	
		RBR	61	54	>_ROUTFP	
		RER	61	78	>RBR	
r_{brp}	3.3-68	RBRPL	61	102	>0.	
r_{crp}	3.3-73	RERPL	61	103	>RBRPL	
		RINFP	61	104		
		ROUTFP	61	128	>RINFP	
		ZONEL	61	152	>0.	
S_{pt}	3.3-40	SRFSTZ	61	159		
		AREAPC	61	166	>0.	
d_{ro}	3.3-59	DRFO	61	169	>0.	
		RBRO	61	180	>0.	
		RERO	61	181	>0.	
S_{cr}	3.3-62	SER	61	182	>0.	
d_{ri}	3.3-56	DRFI	61	189	>0.	
N_{PI}	3.11-1	XXNPIN	61	274	>0.	
γ_s	3.3-22	GAMSS	62	2		
γ_c	3.3-6	GAMTNC	62	4		
γ_e	3.3-23	GAMTNE	62	5		
		PSHAPE	62	6		
P_r	3.3-22	PSHAPR	62	30		
		AHBPAR	63	2		
		BHBPAR	63	3		
		CHIBPAR	63	4		
		HBMAX	63	5		
		HBMIN	63	6		
		HBPAR	63	7		
k_{si}	3.3-43	XKSTIZ	63	11		
k_{so}	3.3-46	XKSTOZ	63	18		

Variable	Reference Eq. No.	Input Variable	Input Block	Location Number	Suggested Value	External Reference
$\varepsilon\sigma$	3.3-4	DEL	63	25		
k_r	3.3-57	XKRF	63	28		
$(\rho c)_{sti}$	3.3-45	RHOCSI	63	37	>0.	
$(\rho c)_{sto}$	3.3-47	RHOCSO	63	44	>0.	
$(\rho c)_r$	3.3-56	RHOCR	63	51	>0.	
$(\rho c)_g$	3.3-68	RHOCG	63	58	>0.	
R_g	3.3-71	RG	63	59	>0.	
α_f	3.10-4	FUELEX	63	73		
α_e	3.10-5	CLADEX	63	74		
Y_f	3.10-8	YFUEL	63	75		
Y_e	3.10-9	YCLAD	63	76		
		FULREX	63	77		
		CLDREX	63	78		
$(HA)_{I,M}$	3.11-1	HACHCH	63	82		
$1/\gamma_{ht}$	3.3-116	TAVINV	63	104	.5-5	
A_{fr}	3.8-3	AFR	64	1	.1875	
b_{fr}	3.8-3	BFR	64	2	-.2	
c_1	3.3-9	C1	64	3	.025	2-5
c_2	3.3-9	C2	64	4	.8	2-5
c_3	3.3-9	C3	64	5	4.8	2-5
		DWMAX	64	6	.2	
R_{eL}	3.9-3	RELAM	64	7	2100.	
A_{fL}	3.9-3	AFLAM	64	8	64.	
		W0	64	47		
K_{or}	3.9-12	XKORV	64	48		
		XKORGD	64	64		
$\left(\frac{\Delta z_i}{A}\right)_b$	3.9-6	DZIAB	64	65	>0.	
$\left(\frac{\Delta z_i}{A}\right)_f$	3.9-6	DZIAT	64	66		
T_{out}	3.3.-100	DTLMAX	64	69	15.	
U_1	3.10-1	TUPL	64	74		
U_2	3.10-2	UACH1	64	189		
		UACH2	64	190		

3.15.1.3 Empty Reflector Region

Sometimes a reflector zone is used to represent an empty section of a subassembly. The duct wall is usually represented by the structure; and since there is nothing but sodium inside the duct wall, there is nothing for the reflector to represent. The code requires a “reflector” in every axial zone except the pin section, so some input for the reflector must be included. The reflector perimeter, SER, cannot be zero; but it can be set to a very small value, such as 10^{-9}, so that it would have no impact on coolant temperatures. Also, the total reflector thickness, DRFI plus DRFO, must be a reasonable value, 0.003 m or more, to prevent numerical instabilities in the reflector temperature calculations in the boiling module. The pre-voiding reflector temperature calculations are numerically stable for any reflector greater than zero.

3.15.1.4 Structure and Reflector Node Thickness

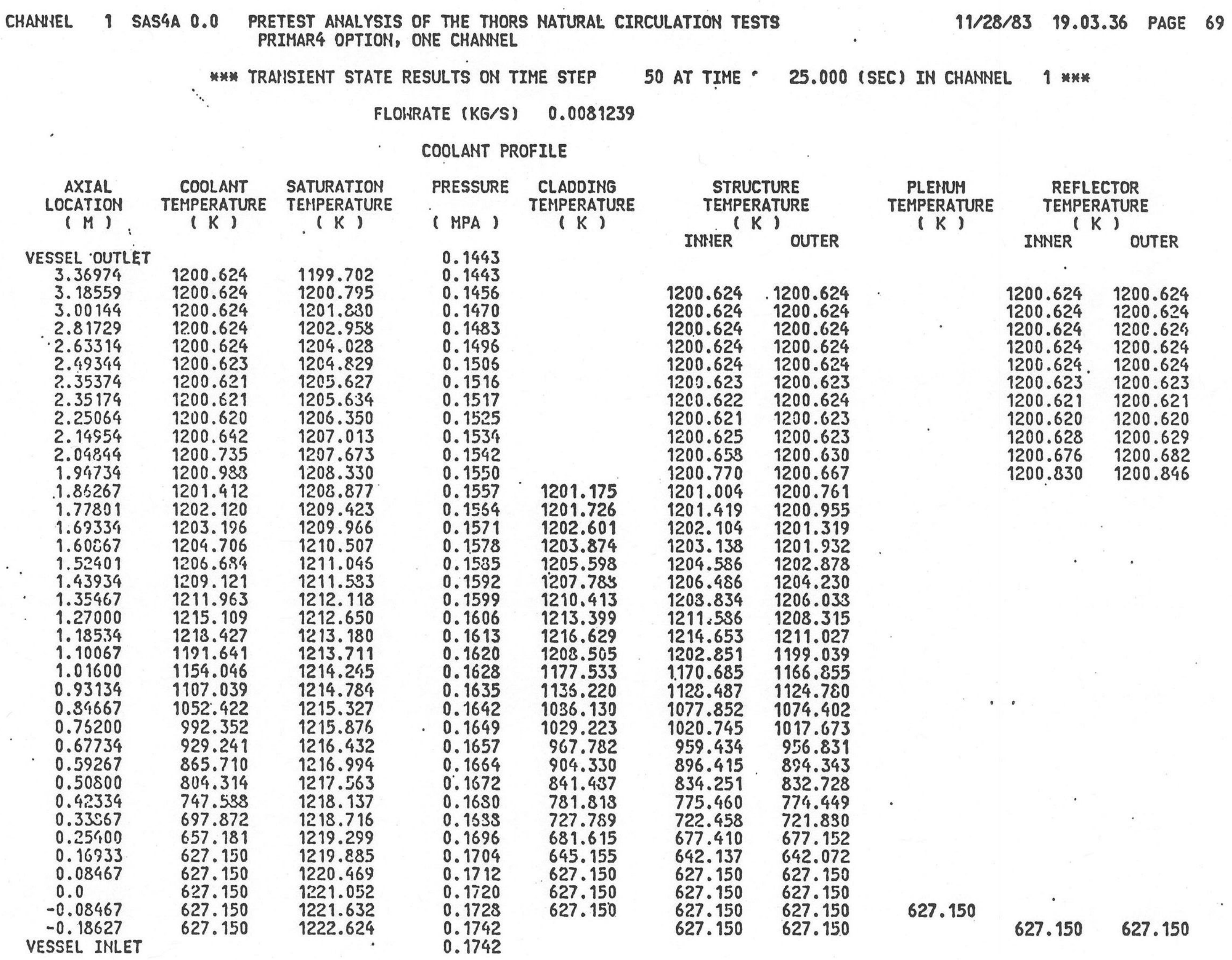
Two radial nodes are used in the reflectors and in the structure. The only restrictions on node thicknesses are that the total reflector thickness and the total structure thickness must be reasonable; 0.003 m or more, to prevent numerical instabilities in the temperature calculations in the boiling module. Usually, the node in contact with the coolant, the inner structure node or the outer reflector node, represents approximately 10% of the total thickness; and the other node represents the rest. Then the small node in contact the coolant will react rapidly to rapid changes in the coolant temperatures, whereas the larger node will dominate the longer-term response to the slow changes.

3.15.1.5 Coolant Re-entry Temperature

If PRIMAR-1 is being used, then the re-entry temperature calculation described in Section 3.3.6 uses the input variable TUPL for the bulk temperature of the coolant in the outlet plenum. If PRIMAR-4 is being used, then the outlet plenum temperature is computed by PRIMAR-4; and TUPL is not used.

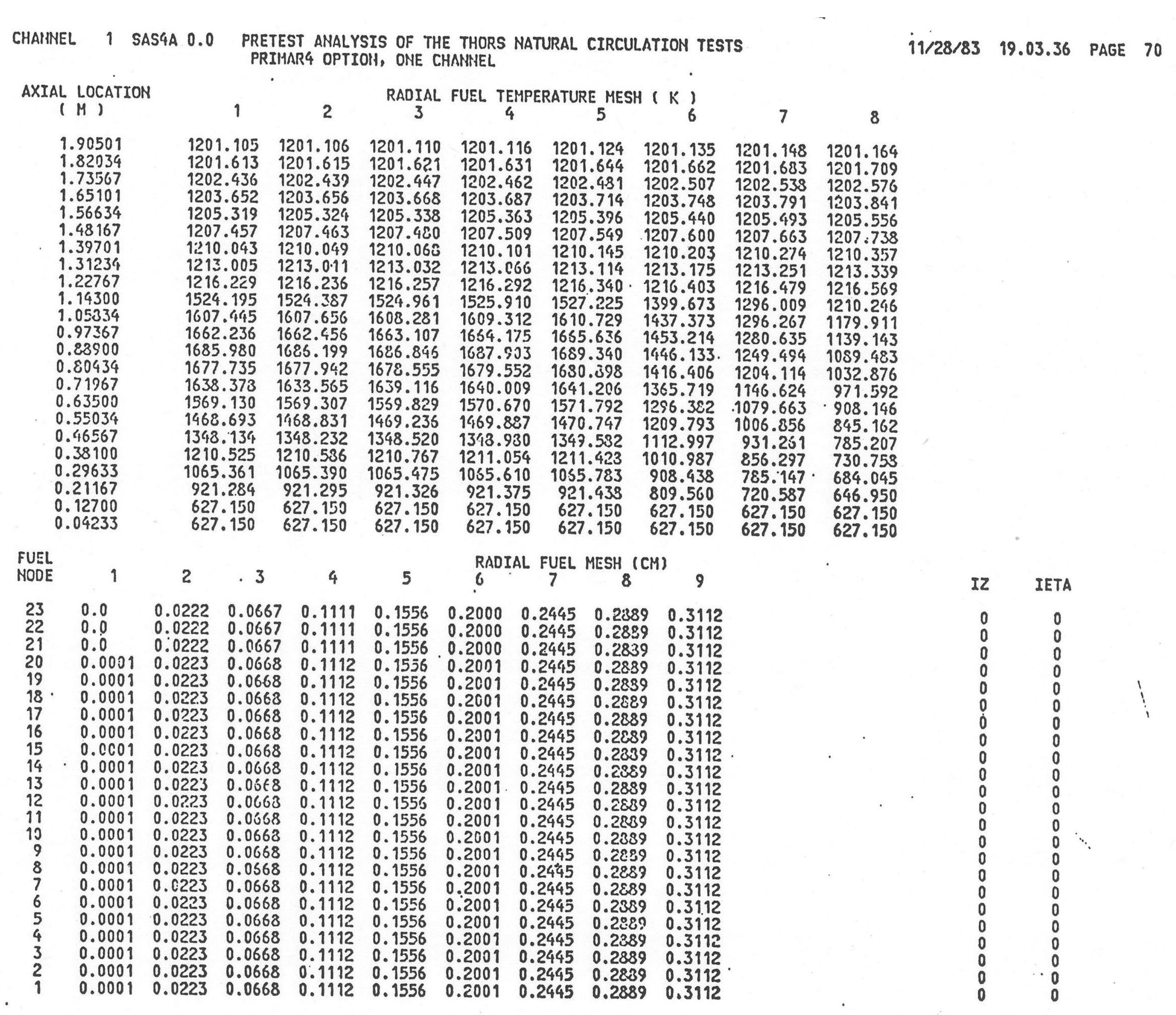
3.15.2 Output Description

Figure 3.14–1 shows a typical thermal hydraulic output for one step of the transient. The output is largely self-explanatory. Note that on the first page of this output the axial location, coolant temperature, saturation temperature, and pressure are the values at node boundaries; whereas the cladding, structure, plenum, and reflector temperatures are the values at node mid-points. Also, on the second page the radial fuel temperatures are the values at node mid-points except for the inner and outer fuel temperature, as indicated in Figure 3.2–4; whereas the radii are the values of radial node boundaries.



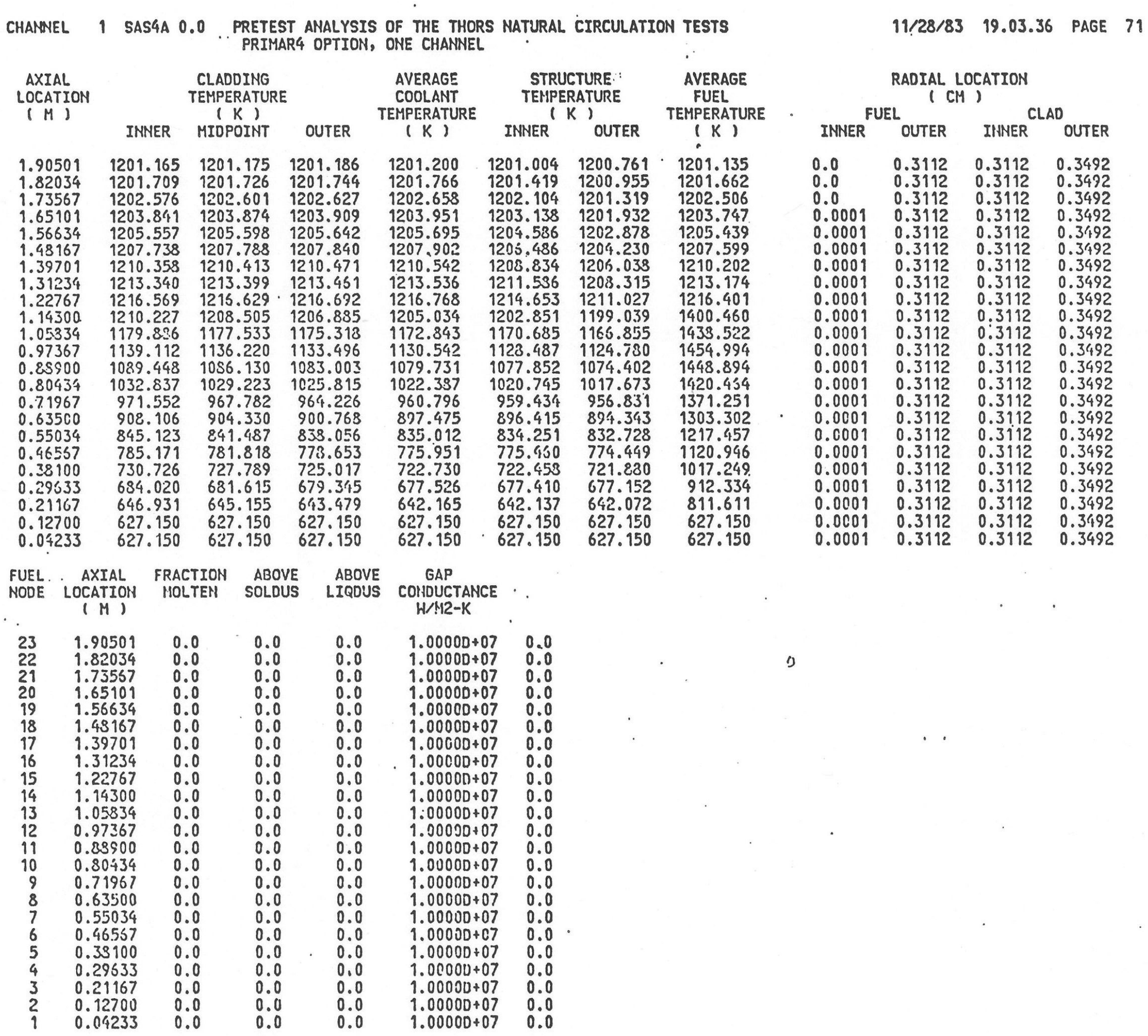
```
CHANNEL    1  SAS4A 0.0    PRETEST ANALYSIS OF THE THORS NATURAL CIRCULATION TESTS                11/28/83   19.03.36   PAGE  69
                             PRIMAR4 OPTION, ONE CHANNEL

                     *** TRANSIENT STATE RESULTS ON TIME STEP      50 AT TIME    25.000 (SEC) IN CHANNEL    1 ***

                                    FLOWRATE (KG/S)    0.0081239

                                         COOLANT PROFILE

   AXIAL      COOLANT     SATURATION    PRESSURE   CLADDING        STRUCTURE             PLENUM        REFLECTOR
  LOCATION  TEMPERATURE  TEMPERATURE             TEMPERATURE      TEMPERATURE        TEMPERATURE     TEMPERATURE
   ( M )       ( K )        ( K )       ( MPA )     ( K )            ( K )              ( K )           ( K )
                                                                INNER      OUTER                    INNER      OUTER
VESSEL OUTLET                               0.1443
    3.36974    1200.624     1199.702     0.1443
    3.18559    1200.624     1200.795     0.1456                1200.624   1200.624                  1200.624   1200.624
    3.00144    1200.624     1201.830     0.1470                1200.624   1200.624                  1200.624   1200.624
    2.81729    1200.624     1202.958     0.1483                1200.624   1200.624                  1200.624   1200.624
    2.63314    1200.624     1204.028     0.1496                1200.624   1200.624                  1200.624   1200.624
    2.49344    1200.623     1204.829     0.1506                1200.624   1200.624                  1200.624   1200.624
    2.35374    1200.621     1205.627     0.1516                1200.623   1200.623                  1200.623   1200.623
    2.35174    1200.621     1205.634     0.1517                1200.622   1200.624                  1200.621   1200.621
    2.25064    1200.620     1206.350     0.1525                1200.621   1200.623                  1200.620   1200.620
    2.14954    1200.642     1207.013     0.1534                1200.625   1200.623                  1200.628   1200.629
    2.04844    1200.735     1207.673     0.1542                1200.658   1200.630                  1200.676   1200.682
    1.94734    1200.988     1208.330     0.1550                1200.770   1200.667                  1200.830   1200.846
    1.86267    1201.412     1208.877     0.1557    1201.175    1201.004   1200.761
    1.77801    1202.120     1209.423     0.1564    1201.726    1201.419   1200.955
    1.69334    1203.196     1209.966     0.1571    1202.601    1202.104   1201.319
    1.60867    1204.706     1210.507     0.1578    1203.874    1203.138   1201.932
    1.52401    1206.684     1211.046     0.1585    1205.598    1204.586   1202.878
    1.43934    1209.121     1211.583     0.1592    1207.783    1206.486   1204.230
    1.35467    1211.963     1212.118     0.1599    1210.413    1208.834   1206.033
    1.27000    1215.109     1212.650     0.1606    1213.399    1211.586   1208.315
    1.18534    1218.427     1213.180     0.1613    1216.629    1214.653   1211.027
    1.10067    1191.641     1213.711     0.1620    1208.505    1202.851   1199.039
    1.01600    1154.046     1214.245     0.1628    1177.533    1170.685   1166.855
    0.93134    1107.039     1214.784     0.1635    1136.220    1128.487   1124.780
    0.84667    1052.422     1215.327     0.1642    1036.130    1077.852   1074.402
    0.76200     992.352     1215.876     0.1649    1029.223    1020.745   1017.673
    0.67734     929.241     1216.432     0.1657     967.782     959.434    956.831
    0.59267     865.710     1216.994     0.1664     904.330     896.415    894.343
    0.50800     804.314     1217.563     0.1672     841.487     834.251    832.728
    0.42334     747.588     1218.137     0.1680     781.818     775.460    774.449
    0.33867     697.872     1218.716     0.1688     727.789     722.458    721.880
    0.25400     657.181     1219.299     0.1696     681.615     677.410    677.152
    0.16933     627.150     1219.885     0.1704     645.155     642.137    642.072
    0.08467     627.150     1220.469     0.1712     627.150     627.150    627.150
    0.0         627.150     1221.052     0.1720     627.150     627.150    627.150
   -0.08467     627.150     1221.632     0.1728     627.150     627.150    627.150      627.150
   -0.18627     627.150     1222.624     0.1742                 627.150    627.150                    627.150    627.150
VESSEL INLET                                0.1742
```

Figure 3.15–1: Sample Subassembly Thermal Hydraulics Output



```
CHANNEL    1  SAS4A 0.0    PRETEST ANALYSIS OF THE THORS NATURAL CIRCULATION TESTS                     11/28/83  19.03.36  PAGE  70
                            PRIMAR4 OPTION, ONE CHANNEL

 AXIAL LOCATION                                   RADIAL FUEL TEMPERATURE MESH ( K )
    ( M )                  1          2          3          4          5          6          7          8

    1.90501            1201.105   1201.106   1201.110   1201.116   1201.124   1201.135   1201.148   1201.164
    1.82034            1201.613   1201.615   1201.621   1201.631   1201.644   1201.662   1201.683   1201.709
    1.73567            1202.436   1202.439   1202.447   1202.462   1202.481   1202.507   1202.538   1202.576
    1.65101            1203.652   1203.656   1203.668   1203.687   1203.714   1203.748   1203.791   1203.841
    1.56634            1205.319   1205.324   1205.338   1205.363   1205.396   1205.440   1205.493   1205.556
    1.48167            1207.457   1207.463   1207.480   1207.509   1207.549   1207.600   1207.663   1207.738
    1.39701            1210.043   1210.049   1210.068   1210.101   1210.145   1210.203   1210.274   1210.357
    1.31234            1213.005   1213.011   1213.032   1213.066   1213.114   1213.175   1213.251   1213.339
    1.22767            1216.229   1216.236   1216.257   1216.292   1216.340   1216.403   1216.479   1216.569
    1.14300            1524.195   1524.387   1524.961   1525.910   1527.225   1399.673   1296.009   1210.246
    1.05834            1607.445   1607.656   1608.281   1609.312   1610.729   1437.373   1296.267   1179.911
    0.97367            1662.236   1662.456   1663.107   1664.175   1665.636   1453.214   1280.635   1139.143
    0.88900            1685.980   1686.199   1686.846   1687.903   1689.340   1446.133   1249.494   1089.483
    0.80434            1677.735   1677.942   1678.555   1679.552   1680.898   1416.406   1204.114   1032.876
    0.71967            1638.378   1633.565   1639.116   1640.009   1641.206   1365.719   1146.624    971.592
    0.63500            1569.130   1569.307   1569.829   1570.670   1571.792   1296.382   1079.663    908.146
    0.55034            1468.693   1468.831   1469.236   1469.887   1470.747   1209.793   1006.856    845.162
    0.46567            1348.134   1348.232   1348.520   1348.980   1349.582   1112.997    931.261    785.207
    0.38100            1210.525   1210.586   1210.767   1211.054   1211.423   1010.987    856.297    730.758
    0.29633            1065.361   1065.390   1065.475   1065.610   1065.783    908.438    785.147    684.045
    0.21167             921.284    921.295    921.326    921.375    921.438    809.560    720.587    646.950
    0.12700             627.150    627.150    627.150    627.150    627.150    627.150    627.150    627.150
    0.04233             627.150    627.150    627.150    627.150    627.150    627.150    627.150    627.150

FUEL                                         RADIAL FUEL MESH (CM)
NODE     1        2       3       4       5       6       7       8       9                            IZ     IETA

 23    0.0      0.0222  0.0667  0.1111  0.1556  0.2000  0.2445  0.2889  0.3112                           0       0
 22    0.0      0.0222  0.0667  0.1111  0.1556  0.2000  0.2445  0.2889  0.3112                           0       0
 21    0.0      0.0222  0.0667  0.1111  0.1556  0.2000  0.2445  0.2839  0.3112                           0       0
 20    0.0001   0.0223  0.0668  0.1112  0.1556  0.2001  0.2445  0.2889  0.3112                           0       0
 19    0.0001   0.0223  0.0668  0.1112  0.1556  0.2001  0.2445  0.2889  0.3112                           0       0
 18    0.0001   0.0223  0.0668  0.1112  0.1556  0.2001  0.2445  0.2889  0.3112                           0       0
 17    0.0001   0.0223  0.0668  0.1112  0.1556  0.2001  0.2445  0.2889  0.3112                           0       0
 16    0.0001   0.0223  0.0668  0.1112  0.1556  0.2001  0.2445  0.2889  0.3112                           0       0
 15    0.0001   0.0223  0.0668  0.1112  0.1556  0.2001  0.2445  0.2839  0.3112                           0       0
 14    0.0001   0.0223  0.0668  0.1112  0.1556  0.2001  0.2445  0.2889  0.3112                           0       0
 13    0.0001   0.0223  0.0668  0.1112  0.1556  0.2001  0.2445  0.2889  0.3112                           0       0
 12    0.0001   0.0223  0.0668  0.1112  0.1556  0.2001  0.2445  0.2889  0.3112                           0       0
 11    0.0001   0.0223  0.0668  0.1112  0.1556  0.2001  0.2445  0.2889  0.3112                           0       0
 10    0.0001   0.0223  0.0668  0.1112  0.1556  0.2001  0.2445  0.2889  0.3112                           0       0
  9    0.0001   0.0223  0.0668  0.1112  0.1556  0.2001  0.2445  0.2839  0.3112                           0       0
  8    0.0001   0.0223  0.0668  0.1112  0.1556  0.2001  0.2445  0.2889  0.3112                           0       0
  7    0.0001   0.0223  0.0668  0.1112  0.1556  0.2001  0.2445  0.2889  0.3112                           0       0
  6    0.0001   0.0223  0.0668  0.1112  0.1556  0.2001  0.2445  0.2889  0.3112                           0       0
  5    0.0001   0.0223  0.0668  0.1112  0.1556  0.2001  0.2445  0.2889  0.3112                           0       0
  4    0.0001   0.0223  0.0668  0.1112  0.1556  0.2001  0.2445  0.2889  0.3112                           0       0
  3    0.0001   0.0223  0.0668  0.1112  0.1556  0.2001  0.2445  0.2889  0.3112                           0       0
  2    0.0001   0.0223  0.0668  0.1112  0.1556  0.2001  0.2445  0.2889  0.3112                           0       0
  1    0.0001   0.0223  0.0668  0.1112  0.1556  0.2001  0.2445  0.2889  0.3112                           0       0
```

Figure 3.15-1: Sample Subassembly Thermal Hydraulics Output (Cont'd.)



```
CHANNEL    1  SAS4A 0.0    PRETEST ANALYSIS OF THE THORS NATURAL CIRCULATION TESTS                 11/28/83   19.03.36   PAGE  71
                              PRIMAR4 OPTION, ONE CHANNEL

  AXIAL                 CLADDING                  AVERAGE        STRUCTURE          AVERAGE                    RADIAL LOCATION
 LOCATION              TEMPERATURE                COOLANT       TEMPERATURE          FUEL                          ( CM )
  ( M )                   ( K )                 TEMPERATURE        ( K )          TEMPERATURE             FUEL                 CLAD
              INNER     MIDPOINT     OUTER        ( K )        INNER      OUTER       ( K )          INNER      OUTER      INNER      OUTER

  1.90501   1201.165   1201.175   1201.186     1201.200     1201.004   1200.761     1201.135        0.0        0.3112     0.3112     0.3492
  1.82034   1201.709   1201.726   1201.744     1201.766     1201.419   1200.955     1201.662        0.0        0.3112     0.3112     0.3492
  1.73567   1202.576   1202.601   1202.627     1202.658     1202.104   1201.319     1202.506        0.0        0.3112     0.3112     0.3492
  1.65101   1203.841   1203.874   1203.909     1203.951     1203.138   1201.932     1203.747        0.0001     0.3112     0.3112     0.3492
  1.56634   1205.557   1205.598   1205.642     1205.695     1204.586   1202.878     1205.439        0.0001     0.3112     0.3112     0.3492
  1.48167   1207.738   1207.788   1207.840     1207.902     1206.486   1204.230     1207.599        0.0001     0.3112     0.3112     0.3492
  1.39701   1210.358   1210.413   1210.471     1210.542     1208.834   1206.038     1210.202        0.0001     0.3112     0.3112     0.3492
  1.31234   1213.340   1213.399   1213.461     1213.536     1211.536   1208.315     1213.174        0.0001     0.3112     0.3112     0.3492
  1.22767   1216.569   1216.629   1216.692     1216.768     1214.653   1211.027     1216.401        0.0001     0.3112     0.3112     0.3492
  1.14300   1210.227   1208.505   1206.885     1205.034     1202.851   1199.039     1400.460        0.0001     0.3112     0.3112     0.3492
  1.05834   1179.886   1177.533   1175.318     1172.843     1170.685   1166.855     1438.522        0.0001     0.3112     0.3112     0.3492
  0.97367   1139.112   1136.220   1133.496     1130.542     1128.487   1124.780     1454.994        0.0001     0.3112     0.3112     0.3492
  0.88900   1089.448   1086.130   1083.003     1079.731     1077.852   1074.402     1448.894        0.0001     0.3112     0.3112     0.3492
  0.80434   1032.837   1029.223   1025.815     1022.387     1020.745   1017.673     1420.434        0.0001     0.3112     0.3112     0.3492
  0.71967    971.552    967.782    964.226      960.796      959.434    956.831     1371.251        0.0001     0.3112     0.3112     0.3492
  0.63500    908.106    904.330    900.768      897.475      896.415    894.343     1303.302        0.0001     0.3112     0.3112     0.3492
  0.55034    845.123    841.487    838.056      835.012      834.251    832.728     1217.457        0.0001     0.3112     0.3112     0.3492
  0.46567    785.171    781.818    778.653      775.951      775.460    774.449     1120.946        0.0001     0.3112     0.3112     0.3492
  0.38100    730.726    727.789    725.017      722.730      722.458    721.880     1017.249        0.0001     0.3112     0.3112     0.3492
  0.29633    684.020    681.615    679.345      677.526      677.410    677.152      912.334        0.0001     0.3112     0.3112     0.3492
  0.21167    646.931    645.155    643.479      642.165      642.137    642.072      811.611        0.0001     0.3112     0.3112     0.3492
  0.12700    627.150    627.150    627.150      627.150      627.150    627.150      627.150        0.0001     0.3112     0.3112     0.3492
  0.04233    627.150    627.150    627.150      627.150      627.150    627.150      627.150        0.0001     0.3112     0.3112     0.3492

 FUEL    AXIAL    FRACTION    ABOVE     ABOVE        GAP
 NODE   LOCATION   MOLTEN     SOLDUS    LIQDUS   CONDUCTANCE
          ( M )                                     W/M2-K

  23    1.90501     0.0        0.0       0.0     1.0000D+07    0.0
  22    1.82034     0.0        0.0       0.0     1.0000D+07    0.0
  21    1.73567     0.0        0.0       0.0     1.0000D+07    0.0
  20    1.65101     0.0        0.0       0.0     1.0000D+07    0.0
  19    1.56634     0.0        0.0       0.0     1.0000D+07    0.0
  18    1.48167     0.0        0.0       0.0     1.0000D+07    0.0
  17    1.39701     0.0        0.0       0.0     1.0000D+07    0.0
  16    1.31234     0.0        0.0       0.0     1.0000D+07    0.0
  15    1.22767     0.0        0.0       0.0     1.0000D+07    0.0
  14    1.14300     0.0        0.0       0.0     1.0000D+07    0.0
  13    1.05834     0.0        0.0       0.0     1.0000D+07    0.0
  12    0.97367     0.0        0.0       0.0     1.0000D+07    0.0
  11    0.88900     0.0        0.0       0.0     1.0000D+07    0.0
  10    0.80434     0.0        0.0       0.0     1.0000D+07    0.0
   9    0.71967     0.0        0.0       0.0     1.0000D+07    0.0
   8    0.63500     0.0        0.0       0.0     1.0000D+07    0.0
   7    0.55034     0.0        0.0       0.0     1.0000D+07    0.0
   6    0.46567     0.0        0.0       0.0     1.0000D+07    0.0
   5    0.38100     0.0        0.0       0.0     1.0000D+07    0.0
   4    0.29633     0.0        0.0       0.0     1.0000D+07    0.0
   3    0.21167     0.0        0.0       0.0     1.0000D+07    0.0
   2    0.12700     0.0        0.0       0.0     1.0000D+07    0.0
   1    0.04233     0.0        0.0       0.0     1.0000D+07    0.0
```

Figure 3.15-1: Sample Subassembly Thermal Hydraulics Output (Cont’d.)

3.15.3 Subchannel Information

This section provides additional information that is necessary to implement the subchannel model described in Section 3.14. This includes a brief user guide, followed by a sample problem, then a detailed overview of the required input. More information about the SAS4A/SASSYS-1 preprocessor, which is available to expedite the input creation process for the subchannel model, can be found in Appendix 3.2.

3.15.3.1 User Guide

This user guide provides a quick summary of the grouping of channels and subassemblies and how to properly prepare the required input. Information about standard output and a binary plotting option is also given.

3.15.3.1.1 Channels, Subassemblies, and Groups

A basic feature of the core treatment is the concept of a channel. In simple terms a channel consists of a fuel pin and its associated coolant and structure. As used in the SAS4A/SASSYS-1 code, a channel is a bit more involved. Figure 3.15–2 shows some details of a SAS4A/SASSYS-1 channel if the three dimensional thermal hydraulics model is *not* being used. In the axial direction the whole length of the subassembly is modeled. A channel includes the whole fuel pin, including the fuel region, axial reflectors in the fuel pin, a gas plenum region at the top or bottom of the fuel pin. Also, a channel includes axial reflector zones above and below the fuel pins.

When the three dimensional thermal hydraulics model is used, the concept of a channel is even more involved. The three-dimensional treatment is used only in the pin section. Multiple channels are used in the pin section of a subassembly, but in the axial reflector zones a single channel treatment is used, as shown in Figure 3.15–3. The first channel used to represent a subassembly includes the pin section and the axial reflectors. All other channels used to represent the subassembly only include the pin section, as seen in Figure 3.15–4.

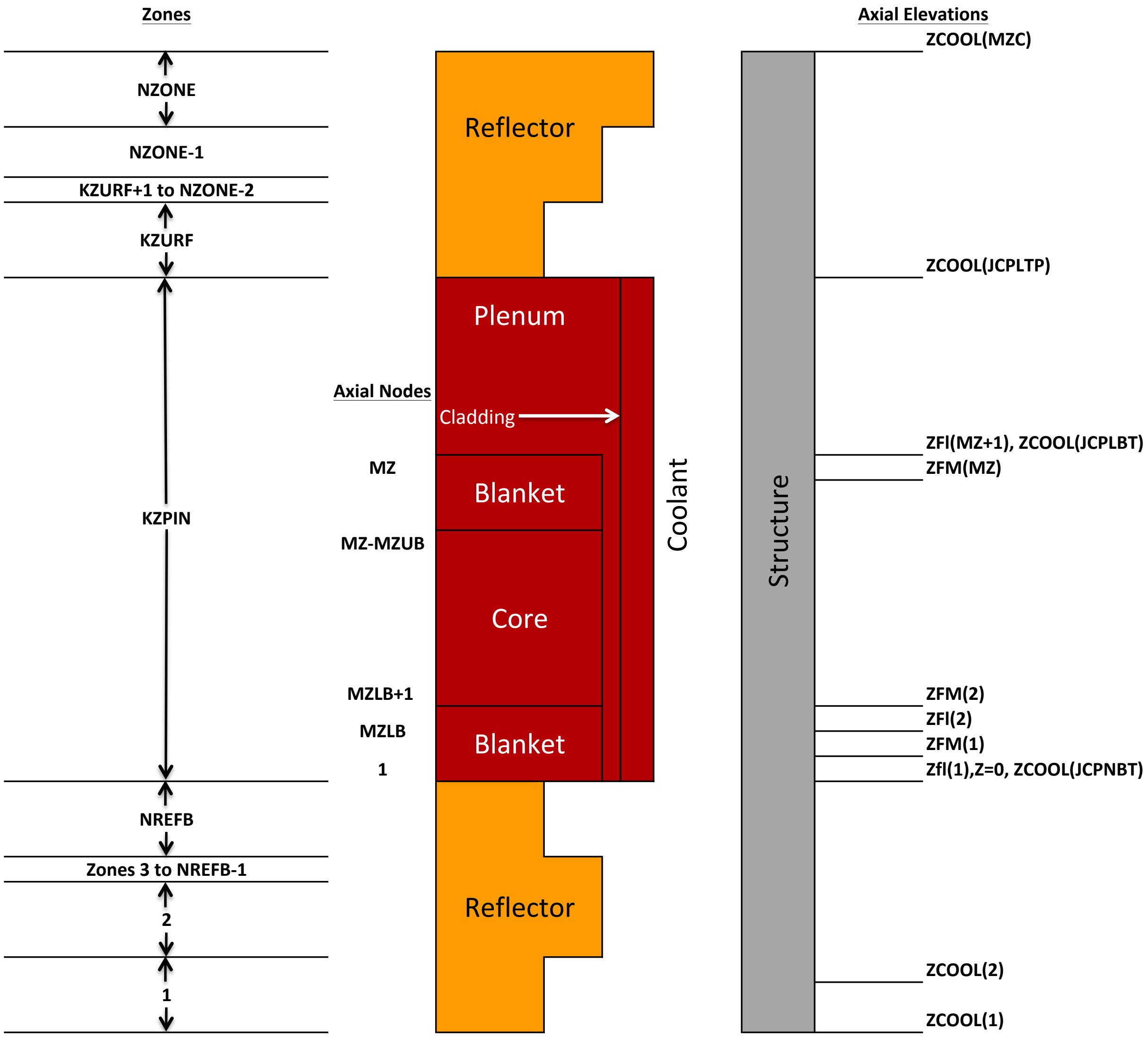


Figure 3.15–2: Details of SAS4A/SASSYS-1 Axial and Radial Nodes (if 3-D Thermal Hydraulics Model is not Being Used)

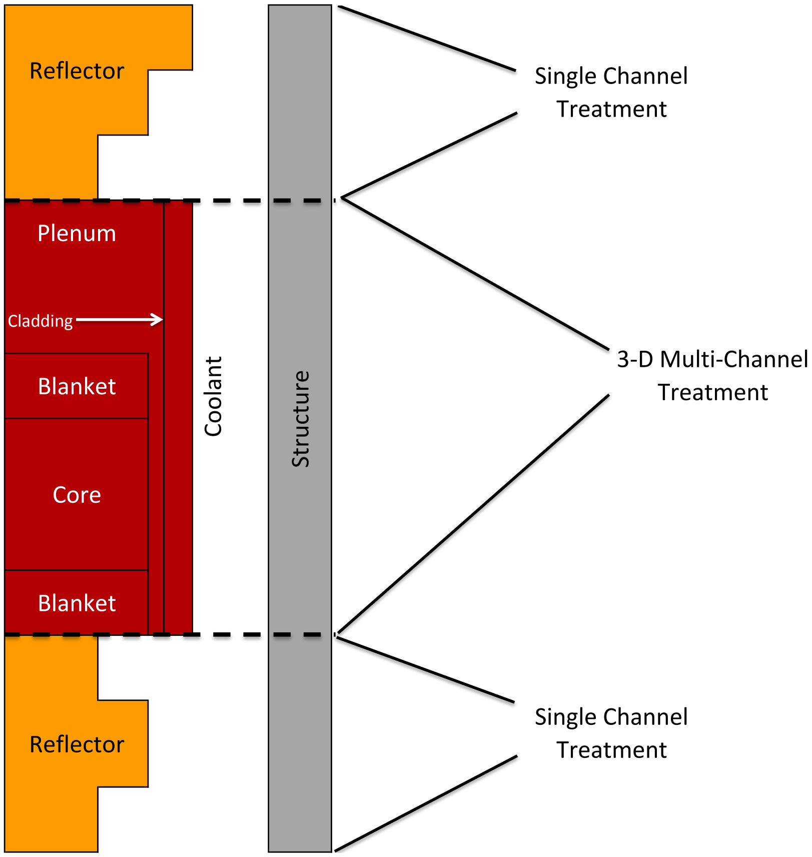


Figure 3.15–3: Breakdown of Axial Regions and Treatments

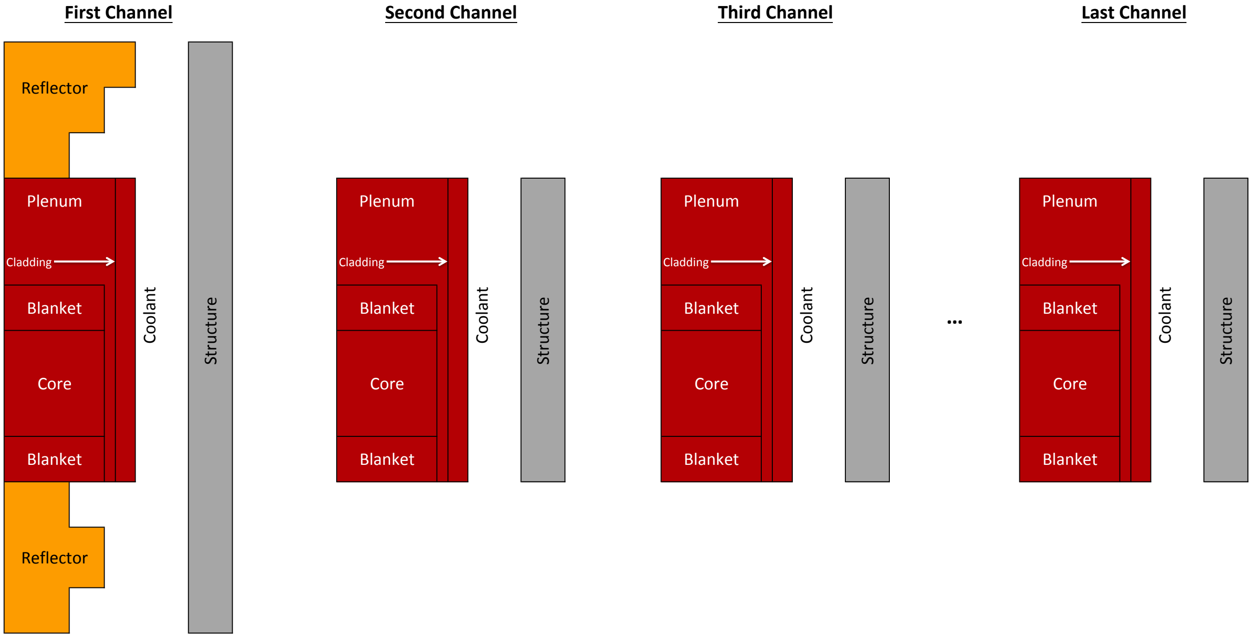


Figure 3.15–4: 3-D Thermal Hydraulics Input Channel Structure

Coolant flow enters a single channel at the bottom of the subassembly, flows through a single channel in the lower reflectors, splits into multiple flow paths in the pin section, and re-combines into a single flow path for the upper reflectors, as shown in Figure 3.15–5. In the pin section, cross-flow between the channels representing a subassembly is accounted for.

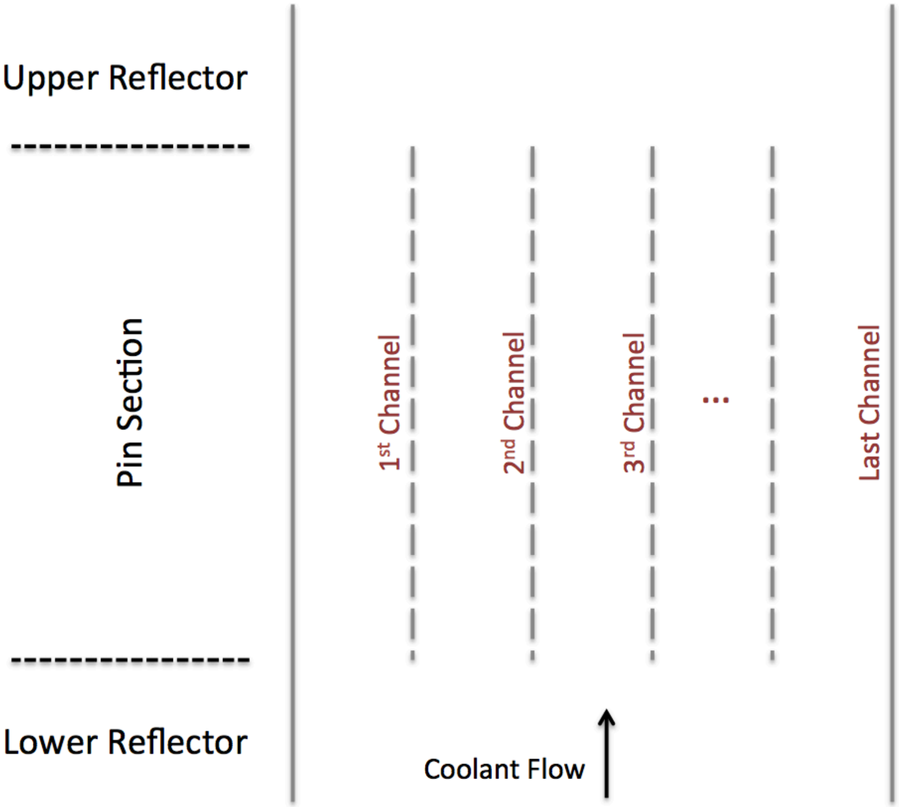


Figure 3.15–5: Subchannel Model Flow Diagram

If one is modeling a reactor without solid duct walls, then accounting for cross-flow between subassemblies is desirable. In order to handle this situation, the concept of a group of channels was incorporated in the model. If solid duct walls are used, then there will be a one-to-one correspondence between subassemblies and groups. If solid duct walls are not used then a group can include many subassemblies, maybe even the whole core. Cross flow between adjacent channels in the whole group is accounted for, but again cross flow is only accounted for in the pin section. The reflector zones are included only in the first channel in a group. Channels are still combined into subassemblies for some edits, even if multiple subassemblies are included in a group. The maximum number of channels that can be included in a group is limited only by the amount of computer memory available for temporary storage of coolant variables during the simultaneous solution of the coolant conditions in all channels in the group. This temporary storage memory is allocated dynamically at run time based on the maximum group size.

Subassembly numbers are assigned by the user by setting the input variable ISUBAS (location 89 of block 51). Group numbers are assigned by the code on the basis of input variable JJMLTP (location 214 of block 51). If JJMLTP for a channel is greater than or equal to zero, then a new group is started. If JJMLTP < 0, then the channel is part of the current group.

3.15.3.1.2 Preparing Input for a Case

One issue to be addressed when preparing input for a case using the new three dimensional thermal hydraulics core model is the amount of detail to be used for various parts of the core. If enough computer memory and computer speed are available, then one could analyze every coolant subchannel and every fuel pin in the core. It is also possible to analyze some subassemblies with the detailed subchannel treatment shown in Figure 3.14–1, to analyze other subassemblies with a somewhat less detailed treatment such as that shown in Figure 3.15–6, and to analyze the rest of the core using one channel to represent a subassembly or a group of similar subassemblies.

The amount of input required for even a moderately sized case can be considerable. Some input is required for each channel, and the number of channels can easily be very large. Fortunately, most of the input for the new model is fairly repetitive, and much of it can be automated by a preprocessor. Appendix 3.2-3 describes a preprocessor that has been written for the SAS4A/SASSYS-1 code with the new model.

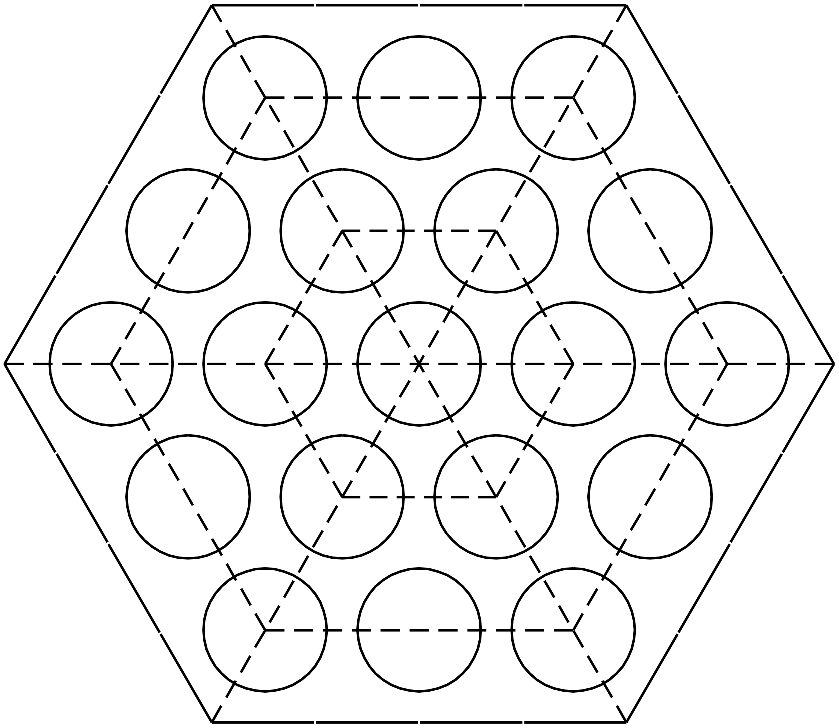


Figure 3.15–6: Subchannels Grouped in Rings

3.15.3.1.3 Controlling the Amount of Standard Output

The standard output from SAS4A/SASSYS-1 includes tables of temperatures, pressures and coolant flow rates for each axial node and each radial node in each channel. These tables are printed out for the steady-state results and for every IPO (location 12, input block 1) time steps during the transient. Even before the new three-dimensional thermal hydraulic model was added to the code, the problem of excessive amounts of output from the code was recognized. An input variable, IPRSKP (location 274, block 51) can be used to skip output from specified channels. Also, before the new model was added to the code, reactivity feedback components for each channel were printed out for each step of the transient. Now the reactivity feedback components are printed out for each subassembly rather than for each channel.

3.15.3.1.4 Binary Output for Plotting

The new model includes a provision for writing a binary output file on unit 24. This output file can contain user specified values calculated by the new model. It is anticipated that plotting programs will use this file to plot results from the new model. The user can specify which data is included on unit 24, how often that data is written, and for which channels.

The SAS4A/SASSYS-1 code writes a number of binary output files other than the unit 24 file mentioned above. The CHANNEL.dat binary file reports reactor power and reactivity feedback in addition to temperatures of fuel, clad, and structure, and coolant flow rates for all channels. The CHANNEL.dat file can be extremely large if using the subchannel model, as a full thermal hydraulic dataset is provided for each subchannel in the binary output. While the user cannot control what channels are included in CHANNEL.dat, the user can control the frequency of output using the MSTPLA, MSTPLB, MSTPL1, MSTPL2, and MSTPL3 (INPCOM:107-111) input fields.

3.15.3.2 Sample Problem

Figure 3.15–7 shows output from a sample problem run with the new model. This test case uses two 19 pin subassemblies with subassembly-to-subassembly heat transfer. Selected coolant temperatures from one of the subassemblies are shown in Figure 3.15–7. The transient was a simulated scram with a delayed pump trip and little natural circulation head in the primary loop. Temperatures drop rapidly due to the scram and then rise after the pump trip until enough buoyancy is established in the subassemblies to increase the flow rate and stop the temperature rise. Flow reversal occurred in the center subchannels from 22.5 seconds to 83.5 seconds in the transient. Flow reversal occurred in the edge subchannels from 22.0 seconds to 111.25 seconds.

The case described above used two subassemblies, each containing 19 pins and 42 coolant subchannels in the pin section. Four axial nodes were used in a lower reflector below the pin section, 24 axial nodes were used in the core, 6 axial nodes were used in the gas plenum region above the core, and 5 axial nodes were used in a reflector above the pins. A null transient of 50 steps with a time step of 1.0 second was used for each subassembly separately. Then a 50 step null transient using both subassemblies and subassembly-to-subassembly heat transfer was run with a time step of 0.5 seconds. Finally the regular transient was run for 1200 steps using a time step of 0.25 seconds.

The null transients started with 35 iterations per step and got down to 2 iterations per step as they converged on a steady-state solution. The regular transient required between 12 and 21 iterations per step. The total running time was 867 seconds on a Sun Blade computer with a 500 MHz processor.

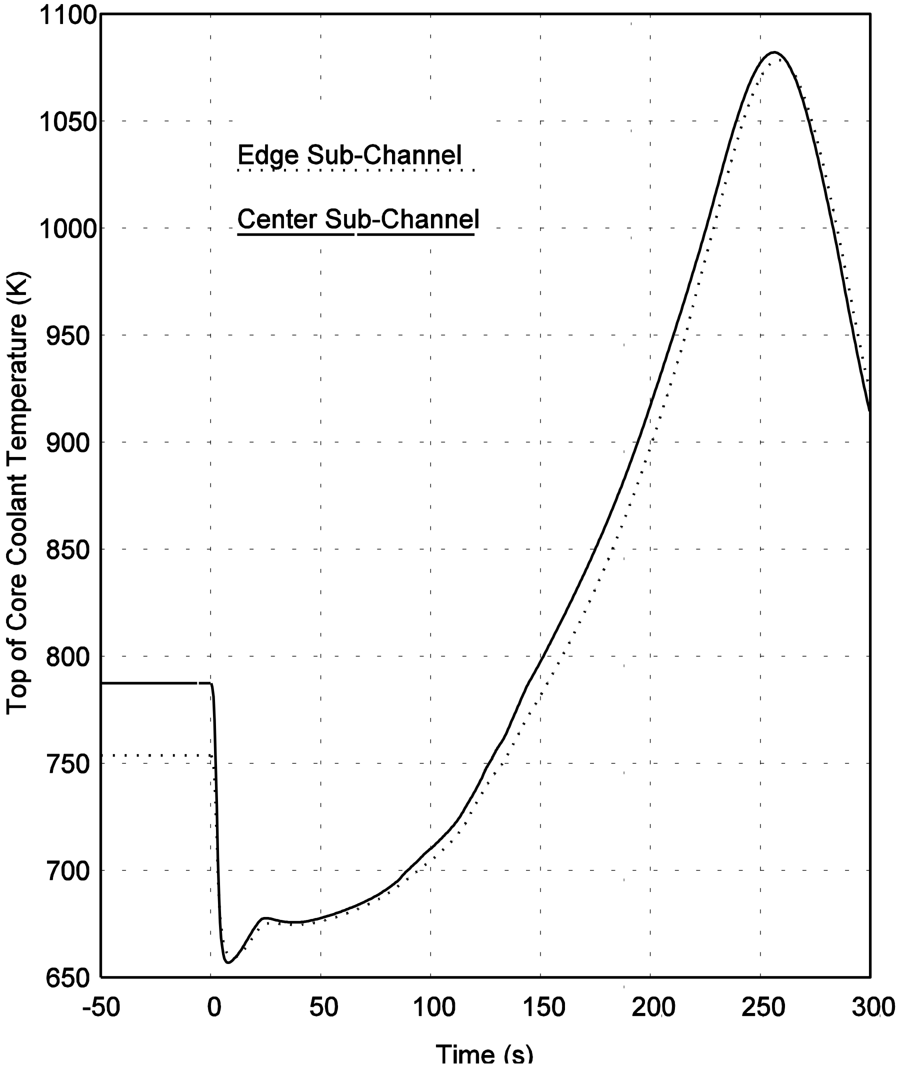


Figure 3.15–7: Results for a Scram with a Delayed Pump Trip

3.15.3.3 Input Description

Input for the SAS4A/SASSYS-1 code is read into numbered locations in input blocks. Each input block has a name and a number. Below is a listing of the additional input variables that are used for the new three-dimensional thermal-hydraulic core model. A description of each input can be found in the User Guide in Chapter 2.

Table 3.15-2: Detailed Subchannel Model Input Variables

Variable	Reference Eq. No.	Input Variable	Input Block	Location Number	Suggested Value	External Reference
		ISSNUL	1	87		
		ISCH	1	115		
		ISUBAS	51	89		
		JJMLTP	51	214		
		JCHMPN(K)	51	215-218		
		NUMKLT	51	219		
		KSWIRL	51	220		
		NULSTI	51	221		
		IFT24	51	495		
		ILATF	51	496		
		NFT24	51	501		
		JCFT24(K)	51	502-521		
		ITYPE24(K)	51	522-541		
		NULPT1	51	542		
	3.14-7	UACHM1(K)	64	197-200		
	3.14-8	UACHM2(K)	64	201-204		
		ALATRL(K)	64	205-208		
		XKLAT(K)	64	209-212		
		DPLTLM	64	213		
		XKSWRL	64	214		
		DTNUL1	64	215		
		EPSFLW	64	216		
		EPSTMP	64	217		
		EPSPRS	64	218	1.0	
		XCMPRS	64	219		
		XLINRT	64	220		

3.16 Thermal Properties of Fuel and Cladding

All material property data for fuel and cladding are cast as functions or subroutines to allow for modularization and the ease of making changes. This also allows for the incorporation of different materials data in a straightforward manner. In a number of the correlations used, the units are inconsistent with the SI unit system adopted by SAS4A/SASSYS-1. The routines that use these correlations carry out the appropriate units conversions internally.

The thermal properties for fuel and cladding are described in this section. Sodium properties are described in Chapter 12.

3.16.1 Fuel Density

The fuel density can be obtained either form a user-supplied table of density vs temperature or from a correlation with

$$\rho_f = \frac{\rho_o}{1 + C_1(T - 273) + C_2(T - 273)^2} \tag{3.16-1}$$

where

ρ_o = The theoretical density at 273 K, kg/m^3

C_1,C_2 = Input coefficients

T = Temperature, K

This applies between 273 K and the solidus temperature.

The liquid fuel density is given by

$$\rho_\ell = \frac{\rho_o}{1 + C_3(T - 273)} \tag{3.16-2}$$

where

C_3 = Input coefficient

This applies to temperatures above the liquidus. For the range between the solidus and liquidus temperatures, a linear interpolation is performed.

These equations are found in the function RHOF. Suggested values of coefficients are from the Nuclear Systems Materials Handbook [3-13].

ρ_c = COEFDS(1) = 11.05x10^3 kg/m^3 (mixed oxide)

C_1 = COEFDS(2) = 2.04x10^{-5} K^{-1}

C$_2$ = COEFDS(3) = 8.70x10^{-9} K^{-2}

C_3 = COEFDS(2) = 9.30x10^{-5} K^{-1}

3.16.2 Fuel Thermal Conductivity

Four different options exist for the fuel thermal conductivity. These are controlled through the input parameter IRHOK.

IRHOK = 0

The thermal conductivity as function of temperature is input in table form through the variable arrays XKTAB and XKTEM.

IRHOK =1

For this option, the conductivity equations are given by:

$$k_1(T) = 1.1 + \frac{1x10^2}{T(.4888 - .4465 f_D)} \quad (3.16\text{-}3)$$

$$\text{for } 800°C \le T \le 2000°C$$

$$k_2(T) = k_1(800)\frac{168.844}{12.044 + (0.196)T} \quad (3.16\text{-}4)$$

$$\text{for } T \le 800°C$$

$$k_3(T) = k_1(2000) \quad \text{for } T > 2000°C \quad (3.16\text{-}5)$$

where

k_1,k_2,k_3 = Fuel thermal conductivity, W/m-k

T = Temperature, ºC

f_D = Fuel fraction of theroretical density

IRHOK = 2

This form of the conductivity is given by

$$k_1(T) = \left[(C_1 - f_D)\, f_D - 1\right]\left[\frac{1}{(C_2 + C_3 T)} + C_4 T^3\right] \quad (3.16\text{-}6)$$

$$\text{for } 0.75 \le f_D \le 0.95$$

$$k_2(T) = (3f_D - 1)\left[\frac{1}{C_5 + C_6 T} + C_7 T^3\right] \qquad (3.16\text{-}7)$$

for $f_D > 0.95$

where

$C_1, C_2, C_3, C_4, C_5, C_6, C_7$ = Input variables

k_1, k_2 = Fuel conductivity W/m-k

T = Temperature, K

If T is greater than the melting temperature, it is set to the melting temperature.

Suggested values:

$C_1 =$ COEFK(1) = 2.1

$C_2 =$ COEFK(2) = 2.88x10^{-3}

$C_3 =$ COEFK(3) = 2.52x10^{-5}

$C_4 =$ COEFK(4) = 2.83x10^{-10}

$C_5 =$ COEFK(5) = 5.75x10^{-2}

$C_6 =$ COEFK(6) = 5.03x10^{-4}

$C_7 =$ COEFK(7) = 2.91x10^{-11}

IRHOK = 3

This conductivity form is [3-14]

$$k_1(T) = \frac{4.005x10^3}{(T - 273) + 402.4} + 0.6416x10^{-10}T^3 \qquad (3.16\text{-}8)$$

where

T = Temperature, K

k = Conductivity in W/m-k

This is the correlation for UO_2 and is converted to mixed oxide by subtracting 0.2.

$$k_2(T) = k_1(T) - 0.2 \qquad (3.16\text{-}9)$$

The porosity correction term was derived for use in the COMETHE-IIIJ [3-15] code and is given by

$$f_p = 1 - 1.029\,\varepsilon - 3.2\,\varepsilon^2 - 40.1\,\varepsilon^3 + 158\,\varepsilon^4 \tag{3.16-10}$$

where

f_p = Porosity multiplier

ε = 1 - ρ_f = Fractional porosity

ρ_f = fractional fuel density = actual density/theoretical density

The conductivity is therefore given by

$$k(T) = f_p\, k_2(T). \tag{3.16-11}$$

Two different routines contain the above correlations, FK and KFUEL. The function FK returns a single value of the conductivity for a single invocation and is used in the steady-state calculation. The subroutine KFUEL returns the conductivity values for each radial node in the current axial segment. It is used in the transient calculational procedure.

REFERENCES

3-1. M. Lees, "An Extrapolated Crank-Nicholson Difference Scheme for Quasilinear Parabolic Equations" *Proc. Symp. On Nonlinear Partial Differential Equations*, W. F. Ames, Ed., Academic Press, 1966.

3-2. F. E. Dunn, et al., "The SAS2A LMFBR Accident Analysis Computer Code," ANL-8138, Argonne National Laboratory, 1974.

3-3. "A Preliminary User's Guide to Version 1.0 of the SAS3D LMFBR Accident Analysis Computer Code," J. E. Cahalan and D. R. Ferguson, Eds.; Available through the Reactor Analysis Division, Argonne National Laboratory, 1977.

3-4. W. M. Rohsenow and H. Y. Choi, *Heat, Mass and Momentum Transfer*, p. 189, Prentice-Hall, 1961.

3-5. W. M. Rohsenow and J. P. Hartnett, Eds., *Handbook of Heat Transfer*, p. 7-33, McGraw-Hill Book company, New York, 1973.

3-6. William T. Sha, Frank J. Goldner, Paull R. Heubotter, and Robert C. Schmitt, "Thermal Hydraulic Multichannel Analysis," *Proceedings of the International Meeting on Reactor Heat Transfer*, Karlsruhe, Germany, p. 180, October 9-11, 1973.

3-7. D. Mohr, L. Chang, and H. P. Planchon, "Validation of the HOTCHAN Code for Analyzing the EBR-II Core Following an Unprotected Loss of Flow," *Trans. Am. Nucl. Soc.*, vol. 57, p. 318, 1988.

3-8. Y. M. Kwon, Y. B. Lee, W. P. Chang and D. Hahn, "SSC-K Code User's Manual (Rev. 1)", KAERI/TR-2014, KAERI, Daejeon, Korea, 2002.

3-9. C. W. Stewart, C. L. Wheeler, R. J. Cena, C.A. McMonagle, J.M. Cuta and D.S. Trent, "COBRA-IV: the Model and the Method," BNWL-2214, Battelle Pacific Northwest Laboratories, Richland, Washington, 1977.

3-10. K. L. Basehore and N. E. Todreas, "SUPERENERGY-2: A MULTIASSEMBLY Steady-State Computer Code for LMFBR Core Thermal-Hydraulics Analysis," PNL-3379, Pacific Northwest Laboratory, Hanford, Washington, 1980.

3-11. P. M. Magee, A. E. Dubberley, A. J. Lipps and T. Wu, "Safety Performance of the Advanced Liquid Metal Reactor," *Proceedings of ARS '94 International Topical Meeting on Advanced Reactors Safety,* Pittsburg, Pennsylvania, April 17-21, vol. 2, pp. 826-833, 1994.

3-12. S. K. CHENG and N. E. Todreas, "Hydrodynamic Models and Correlations for Bare and Wire-Wrapped Hexagonal Rod Bundles – Bundle Friction Factors, Subchannel Friction Factors and Mixing Parameters," *Nucl. Eng. And Design*, vol. 92, p. 227-251, 1986.

3-13. "Nuclear Systems Materials Handbook," TID-26666, Hanford Engineering Development Laboratory.

3-14. S. Y. Ogawa, E. A. Lees and M. F. Lyons, "Power Reactor High Performance UO_2 Program, Fuel Design Summary and Progress Status," GEAP-5591, General Electric

Company, 1968.

3-15. P. Verbeek and H. Hoppe, “COMETHE-IIIJ: A Computer Code for Predicting Mechanical and Thermal Behavior of a Fuel Pin, Part 1: General Description,” INBFR-107 (original report BN-7609-07), Belgonucleaire, Brussels, Belgium, 1977.

APPENDIX 3.1:
DEGREE OF IMPLICITNESS FOR FLOW AND TEMPERATURE CALCULATIONS

A typical calculation in SASSYS-1 involves finding a temperature or flow rate at the end of a time step given the values of relevant parameters at the beginning of the step, as well as the values of driving functions at the beginning and end of the step. Linearized finite difference approximations to the relevant differential equations are usually used, and the accuracy of the finite differencing in time will depend both on the size of the time step and on the degree of implicitness in the solution. By picking appropriate values for the degree of implicitness, it is possible to improve the accuracy of the solution.

The differential equations used for temperatures and flow rates are of the form

$$\frac{dy}{dt} = f(y,t) \tag{A3. 1-12}$$

Where y is the flow rate or temperature being calculated, and f is a function of y and time. The finite difference approximation used for Eq. A3.1-1 is

$$\frac{y_2 - y_1}{\Delta t} = \theta_1 f(y_1,t_1) + \theta_2 f(y_2,t_2) \tag{A3. 1-13}$$

where

t_1 = time at the beginning of the time step

t_2 = time at the end of the time step

y_1 = y at t_1

y_2 = y at t_2

$$\Delta t = t_2 - t_1 \tag{A3. 1-14}$$

$$\theta_1 + \theta_2 = 1.0 \tag{A3. 1-15}$$

The parameters θ_1 and θ_2 determines the degree of implicitness of the solution: for a fully explicit solution θ_1 = 1.0 and θ_2 = 0.0, whereas for a fully implicit solution θ_1 = 0.0 and θ_2 = 1.0.

After linearizing, equation A3.1-1 can be put in the form

$$\frac{dy}{dt} = f(y_1,t_1) + (t - t_1)\frac{df}{dt} + (y - y_1)\frac{df}{dy} \tag{A3. 1-16}$$

or

$$\frac{dy}{dt} = A + B\ (t - t_1) + C(y - y_1) \qquad \text{(A3. 1-17)}$$

where

$$A = f(y_1, t_1) \qquad \text{(A3. 1-18)}$$

$$B = df\ /\ dt \qquad \text{(A3. 1-19)}$$

$$C = df\ /\ dy \qquad \text{(A3. 1-20)}$$

For instance, Eq. 3.80t for the coolant flow rate in a channel is

$$I_1 \frac{dw}{dt} = p_b - p_t - w^2 I_2 - A_{fr} w\ |w|^{I+b_{fr}} I_3 - w|w| I_4 - g I_5 \qquad \text{(A3. 1-21)}$$

Linearizing the w^2, $w|w|^{1+b_{fr}}$, and $w|w|$ terms gives

$$w^2 \simeq w_1^2 + 2w_i(w - w_1) \qquad \text{(A3. 1-22)}$$

$$w|w|^{1+b_{fr}} \simeq w_1|w_1|^{1+b_{fr}} + (2 + b_{fr})|w_1|^{1+b_{fr}}(w - w_1) \qquad \text{(A3. 1-23)}$$

and

$$w|w| \simeq w_1|w_1| + |w_1|(w - w_1) \qquad \text{(A3.1-24)}$$

Also, p_b and p_t are approximated as

$$p_b(t) = p_b(t_1) + (t - t_1)\frac{dp_b}{dt} \qquad \text{(A3. 1-25)}$$

$$p_t(t) = p_t(t_1) + (t - t_1)\frac{dp_t}{dt} \qquad \text{(A3. 1-26)}$$

where

$$\frac{dp_b}{dt} = \frac{p_b(t_2) - p_b(t_1)}{t_2 - t_1} \tag{A3. 1-27}$$

and

$$\frac{dp_t}{dt} = \frac{p_t(t_2) - p_t(t_1)}{t_2 - t_1} \tag{A3. 1-28}$$

Then Eq. A3.1-10 has the form of Eq. A3.1-6 if

$$\begin{aligned} A = \Big\{ & p_b(t_1) - p_t(t_1) - w_1^2 I_2 - A_{fr}\ w_1 |w_1|^{1+b_{fr}}\ I_3 \\ & - w_1 |w_1| I_4 - g I_5 \Big\} / I_1 \end{aligned} \tag{A3. 1-29}$$

$$B = \frac{\dfrac{dp_b}{dt} - \dfrac{dp_t}{dt}}{I_1} \tag{A3. 1-30}$$

and

$$C = -\frac{\left[2w_1\ I_2 - (2 + b_{fr}) A_{fr} |w_1|^{1+b_{fr}} - |w_1| I_4 \right]}{I_1} \tag{A3. 1-31}$$

The exact solution for Eq. A3.1-6 is

$$y - y_1 = \left(A\tau - B\tau^2\right)\left(1 - e^{-(t-t_1)/\tau}\right) + B\tau(t - t_1) \tag{A3. 1-32}$$

where

$$\tau = -\frac{1}{C} \tag{A3. 1-33}$$

The finite difference approximation used for Eq. A3.1-6 is

$$\frac{y_2 - y_1}{\Delta t} = A + \theta_2 B \Delta t + \theta_2 C (y_2 - y_1) \tag{A3. 1-34}$$

where

$$\Delta t = t_2 - t_1 \tag{A3. 1-35}$$

The solution of Eq. A3.1-23 for y_2 is

$$y_2 - y_1 = \frac{\mathrm{A}\Delta t + \theta_2 B \Delta t^2}{1 + \theta_2 \Delta t / \tau} \tag{A3. 1-36}$$

Figure A3.1–1 shows y_2 - y_1, as given by Eq. A3.1-25, as a function of θ_2 for the case where A = 1, B = .5, Δt = 2, and τ = 1. Also shown is the exact solution from Eq. A3.1-21. Note that a value θ_2 can be found such that the finite difference solution of Eq. A3.1-25 exactly matches the differential equation solution, given by Eq. A3.1-21, for this case. Also note that neither a fully explicit solution, θ_2 = 0, nor a fully implicit solution, θ_2 = 1.0, is very accurate for particular case. An explicit solution would be numerically unstable for Δt greater than τ; and values considerably smaller than the time constant τ.

For any values of the parameters in Eq. A3.1-21 and Eq. A3.1-25, a value of θ_2 can be chosen such that the finite difference solution matches the solution of the differential equation by setting t = t_2 in Eq. A3.1-21 and combining this equation with Eq. A3.1-25 to give

$$\left(A\tau - B\tau^2\right)\left(1 - e^{-\Delta t/\tau}\right) + B\,\tau\,\Delta t = \frac{\mathrm{A}\Delta t + \theta_2 B \Delta t^2}{1 + \theta_2 \Delta t / \tau} \tag{A3. 1-37}$$

solving for θ_2 gives

$$\theta_2 = \frac{\dfrac{\Delta t}{\tau} - \left(1 - e^{-\Delta t/\tau}\right)}{\dfrac{\Delta t}{\tau}\left(1 - e^{-\Delta t/\tau}\right)} \tag{A3. 1-38}$$

Note that A and B cancelled out of Eq. A3.1-27, and θ_2 is a function only of $\Delta t/t$.

Figure A3.1–2 shows θ_2, given by Eq. A3.1-27, as a function of $\Delta t/\tau$. This figure also shows some approximations to this function, as discussed below. For small values of $\Delta t/\tau$, θ_2 approaches 0.5, where as for large values of $\Delta t/\tau$, θ_2 approaches 1.0.

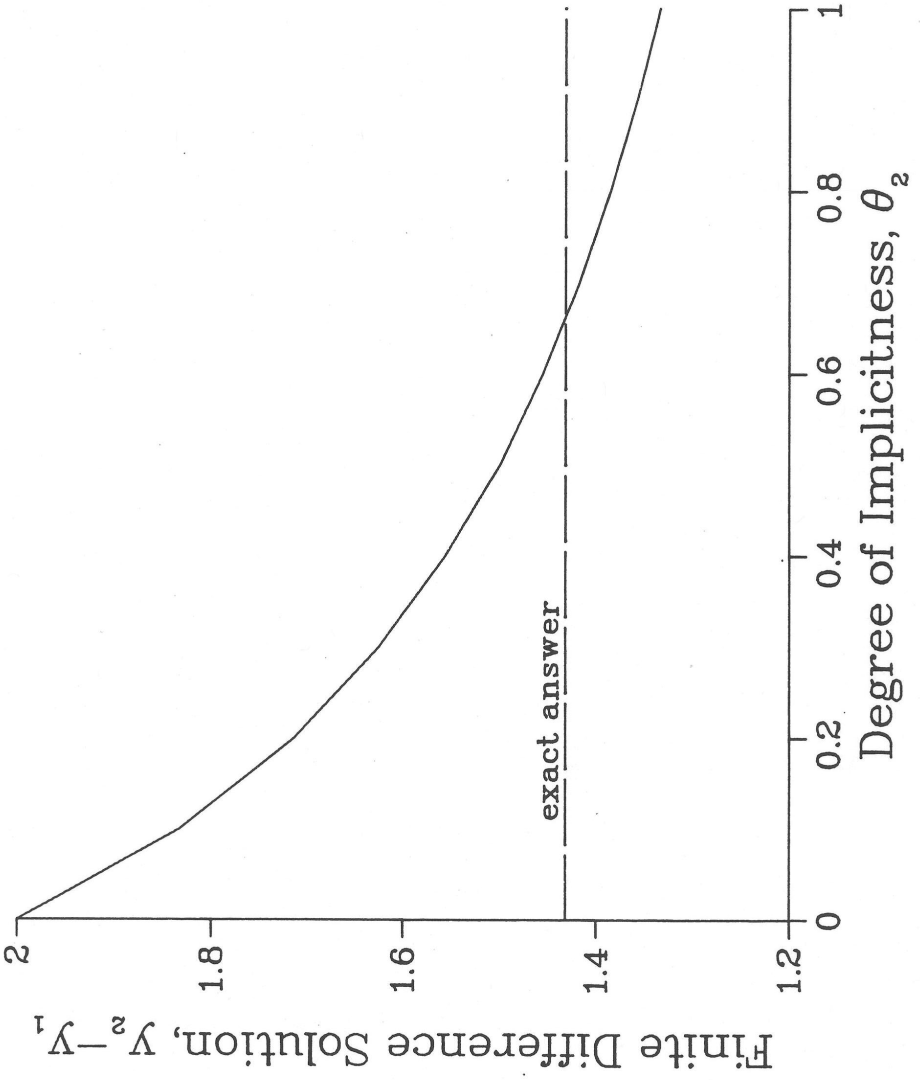


Fig. A3.1-1. Finite Difference Solution as a Function of the Degree of Implicitness

Figure A3.1–1: Finite Difference Solution as a Function of the Degree of Implicitness

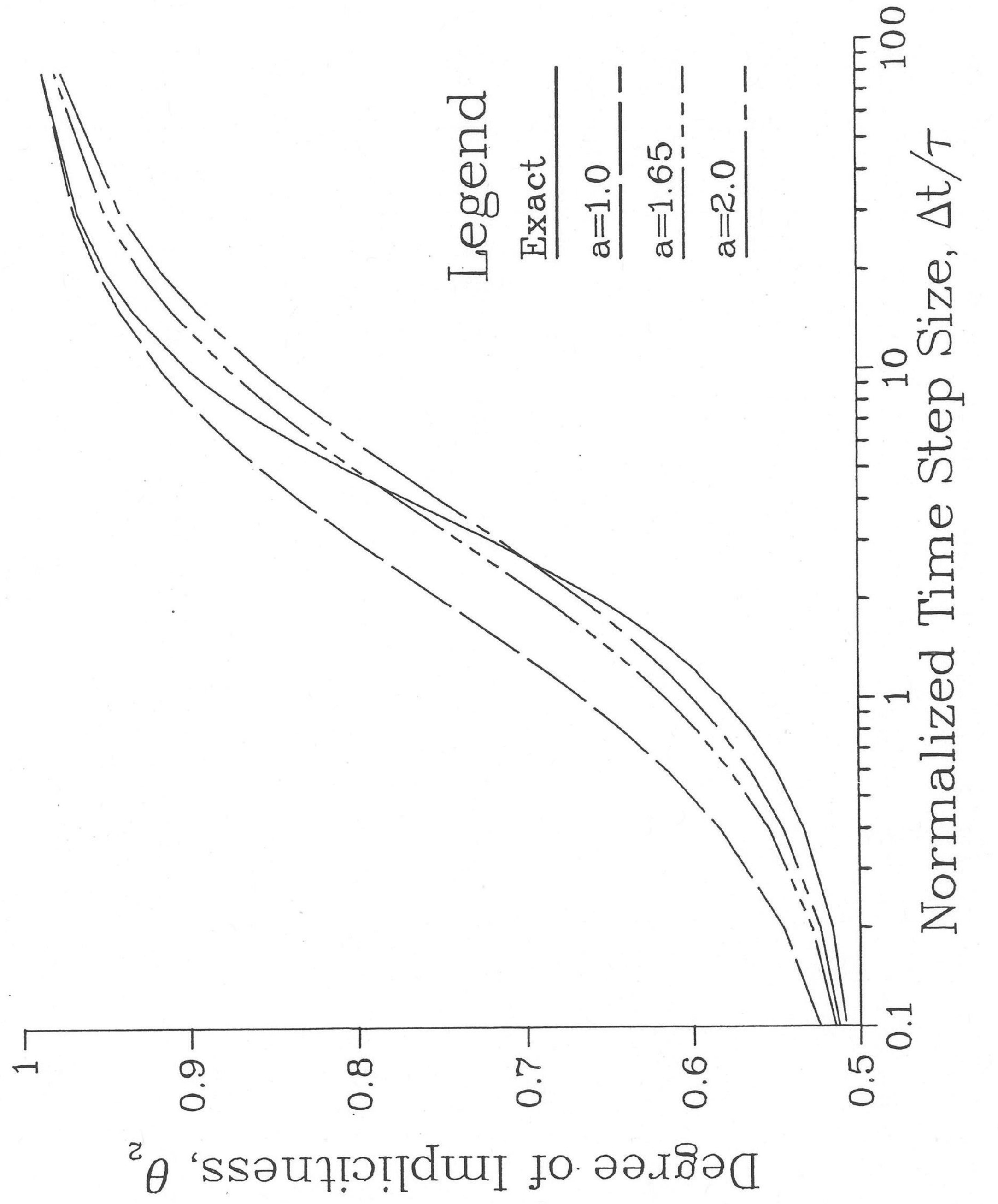


Figure A3.1–2: Degree of Implicitness as a Function of Normalized Time Step Size

For any equation that can be put in the form of Eq. A3.1-5 or A3.1-6, Eq. A3.1-22 can be used to fine τ, and then Eq. A3.1-27 can be used to find the appropriate value of θ_2 for any value of Δt. Note that if driving pressures and coolant flows are being solved for simultaneously, as is done in PRIMAR-4, then the value of B in Eq. A3.1-6 is not known until the pressures have been solved for. In such cases, a direct analytic solution of the differential equation, as in Eq. A3.1-22, could not be made except by iterating between Eq. A3.1-22 and the pressure solution. On the other hand, the calculation of θ_2, as in Eq. A3.1-27, requires only Δt and τ; and the calculation of τ requires only information that is available at the beginning of a time step; so an appropriate value for θ_2 can be found for use in finite difference approximations without iteration, even if driving pressures and flow rates are being solved for simultaneously.

Even if finite differencing in time does not introduce any error into a solution, there are usually other sources of error, such as the linearization approximations of Eq. A3.1-11 through Eq. A3.1-17. Also, the term represented by $B(t\text{-}t_1)$ in Eq. A3.1-6 may in fact not be linear in time. Therefore, simpler and easier to compute approximations to the expression on the right-hand side of Eq. A3.1-27 might sometimes be used without losing much overall accuracy. A simple approximation, θ_2' that approaches the correct limits for very small and very large values of $\Delta t/\tau$ is

$$\theta_2' = \frac{a + \dfrac{\Delta t}{\tau}}{2a + \dfrac{\Delta t}{\tau}} \tag{A3. 1-39}$$

where the parameter a can be chosen to give some best overall fit. Figure A3.1–2 shows this function for various values of the parameter a. The curve for a = 1.65 gives a fairly good fit, although no value of a will give a good fit over the whole range.

A better fit to θ_2 can be obtained by an expression of the form

$$\theta_2' = \frac{a + bx + x^2}{2a + cx + x^2} \tag{A3. 1-40}$$

where

$$x = \frac{\Delta t}{\tau} \tag{A3. 1-41}$$

with a = 6.12992, b = 2.66054, and c = 3.56284, this expression fits the exact value of Eq. A3.1-27 to within 1% over the whole range of $\Delta t/\tau$, as shown in Figure A3.1–3.

In general, when linear approximations are valid, and when a single known time constant dominates the behavior, Eq. A3.1-27 or Eq. A3.1-29 will give a value of θ_2 that will provide accurate finite difference solutions. For cases where a number of different time constants are important, none of these expressions for θ_2 will give a precise finite difference solution for all time-step sizes. Even in this case, though, θ_2 should approach 0.5 for small time steps, and it should approach 1.0 for large time steps. Therefore, an expression of the form given by Eq. A3.1-28 may be appropriate in such cases.

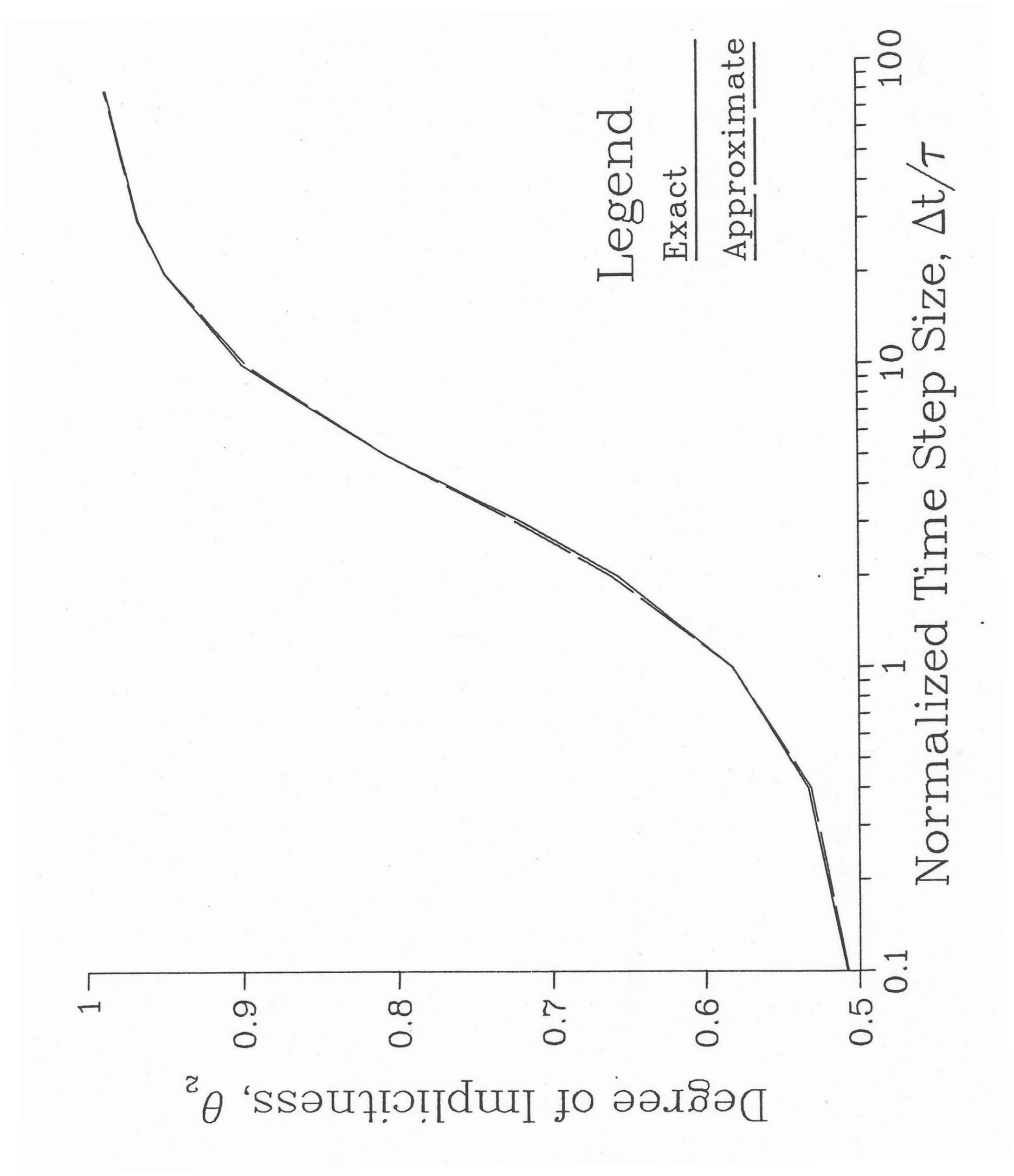


Figure A3.1–3: Approximate Correlation for the Degree of Implicitness

APPENDIX 3.2: INPUT PREPROCESSOR

3.2-1 Introduction

A preliminary version of an input preprocessor has been written for the three-dimensional core thermal hydraulics model described in Section 3.14. This preprocessor reads a relatively small amount of information from the user and produces a large fraction of the input for the new model. This appendix is split into two main sections. The first section details the calculations carried out by the preprocessor, while the second section is a brief user guide, which details the necessary input and its structure.

3.2-2 Calculation Details

The preprocessor calculates and formats many of the variables necessary for the detailed subchannel model. Some of these variables are calculated by the preprocessor, while others are input directly into the preprocessor. The preprocessor functions can be broken down into three main components, as seen in Figure A3.2–1. First, it calculates geometrical parameters such as coolant flow areas and hydraulic diameters. Second, it uses correlations to compute the parameters used for subchannel-to-subchannel heat flow. Lastly, the parameters used in subassembly-to-subassembly heat transfer are determined. This section details each of these calculations

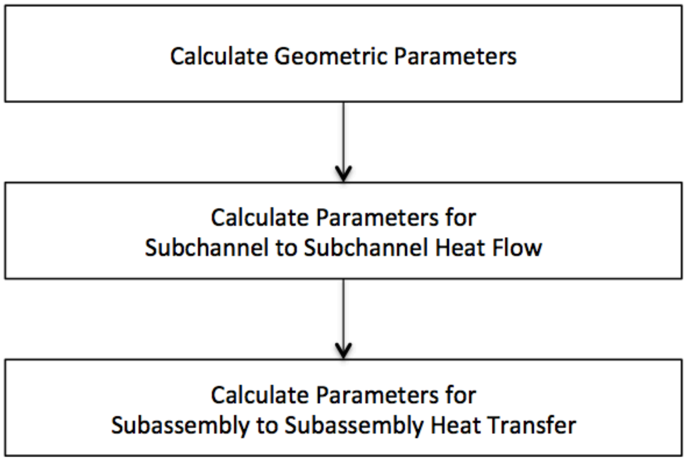

Calculate Geometric Parameters
Calculate Parameters for Subchannel to Subchannel Heat Flow
Calculate Parameters for Subassembly to Subassembly Heat Transfer

Figure A3.2–1: Preprocessor Coding Overview

3.2-2.1 Geometric Properties

This first step to calculating the necessary geometric input for SAS4A/SASSYS-1 is to determine the number of rows[1] of pins in each subassembly. This value is determined from the user-specified number of pins per subassembly n_{SA} (preprocessor variable

[1] A row of pins is the same as a ring of pins

NPINSB), as shown in Table A3.2-1. Note that if the user input for the number of pins per subassembly n_{SA} (NPINSB) is not listed in Table A3.2-1, the preprocessor will return an error.

Table A3.2-1: Number of Rows of Pins

Number of Pins per Subassembly n_{SA}	**Number of Rows of Pins** n_{rows}
1	1
7	2
19	3
37	4
61	5
91	6
127	7
169	8
217	9
271	10

After n_{rows} is calculated, the remaining geometric properties depend on the type of pin treatment specified by the user. As described in Section 3.15.3.1, the subchannel model has several pin options, which are designated using the SAS4A/SASSYS-1 input INPCHN:JJMLTP. These different options are selected using the preprocessor input ISC, as shown in Table A3.2-2.

Table A3.2-2: Preprocessor Pin Options ISC

ISC Option	**Description**
0	JJMLTP = 0, No multi-pin treatment
1	JJMLTP = 1, 1 pin treatment
2	Multi-pin, subchannels grouped by row and sector
3	Multi-pin, same as 2, but bypass flow channel on outside
4	Multi-pin, individual subchannels
5	Multi-pin, individual subchannels, bypass flow channel on the outside
6	JJMLTP = 1, 1 pin treatment, bypass flow channel

The choice of ISC also affects the number of channels used to represent different areas and regions of the subassembly. Table A3.2-3 documents these results, where the values in columns A-E are the number of channels used to represent that region, where Figure A3.2–2 shows the layout of the pin regions. As shown in Table A3.2-3, many of the values for the number of channels in each region depend on n_{rows}, or $n_{c_{ISC}}$, which is the number of channels for the inner subchannels and can take on value 0 or 6, depending on the input for IPDOPT.

Table A3.2-3: Preprocessor Pin Channel Calculation[2]

	Number of Channels				
	A	**B**	**C**	**D**	**E**
ISC	**For Edge SC**	**For Corner SC**	**Thimble Flow Region**	**Represent SA Type**	**Inner SC**
0				1	1
1				1	1
2	6			$6n_{rows}$	D-6-$n_{c_{ISC}}$
3	6		6	$6(n_{rows}$+1)	D-12-$n_{c_{ISC}}$
4	$6(n_{rows}$-1)	6		$2n_{rows}$+4	D-A-$n_{c_{ISC}}$1-B
5	$6(n_{rows}$-1)	6	A-B	4+$8n_{rows}$	D-A-$n_{c_{ISC}}$-1-C-B
6			1	2	1

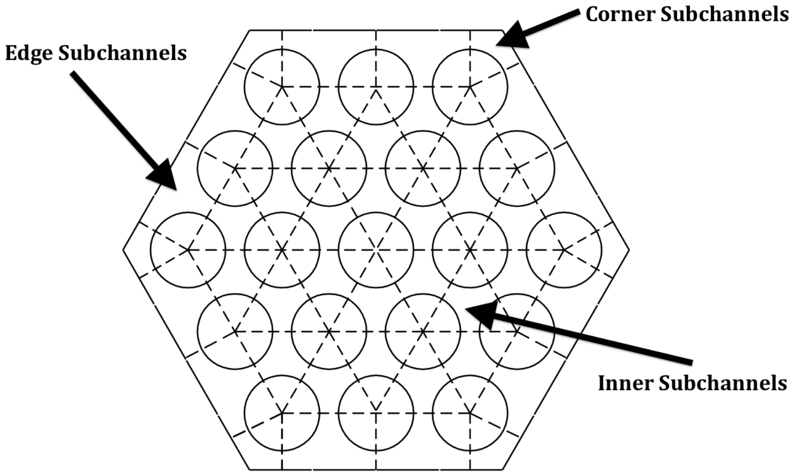


Figure A3.2–2: Pin Region Layout[3]

There are several geometric calculation options available using the input IPDOPT, as seen in Table A3.2-4. These options determine how the distance from the duct wall to the center of the first row d_{DC}, the pitch p, and the flat-to-flat inner diameter D_{ff} are calculated. Figure A3.2–3 presents a diagram of these variables.

[2] Equations reference other variable columns using the assigned letter. SC=Subchannel, SA=Subassembly.

[3] Thimble flow region not shown

Table A3.2-4: IPDOPT Options

IPDOPT	Input	Calculated	Description
0	p, D_{ff}	d_{DC}	Use input p everywhere, calculate d_{DC} based on remainder
1	D_{ff}	p, d_{DC}	Use D_{ff}, calculate uniform p, d_{DC}
2	p	d_{DC}, D_{ff}	Use input p, calculate d_{DC} with same slop
3	D_{ff}, p	d_{DC}	Hot channel option: use: $p = D_{ww} + D_{pin}$* for 1 inner row of subchannels, use p for other inner subchannels, calculate d_{DC} based on remainder
4	p, d_{DC}	D_{ff}	Use input p and d_{DC}, calculate D_{ff}

* D_{ww} = Wire-wrap diameter, D_{pin} = Pin diameter

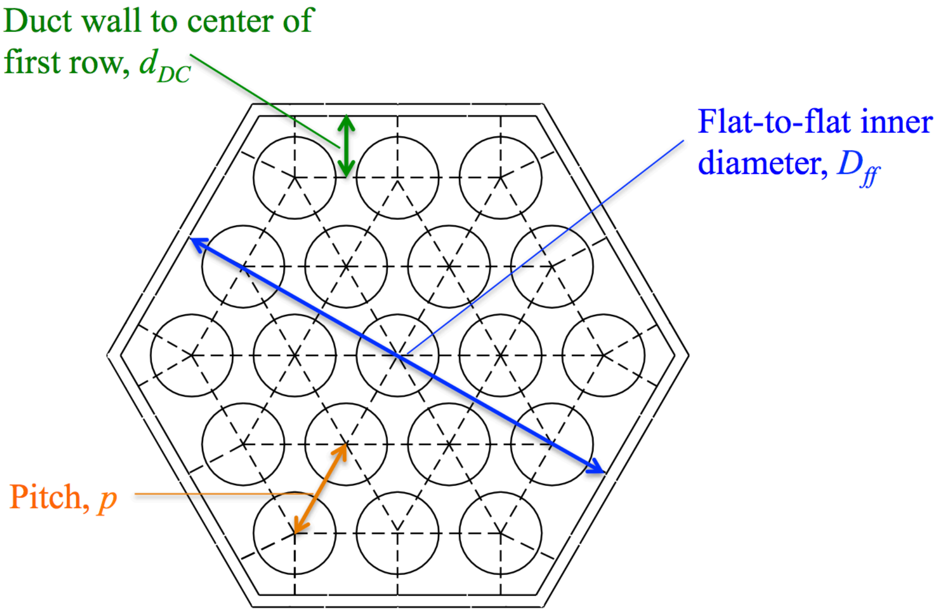

Duct wall to center of first row, d_{DC}
Flat-to-flat inner diameter, D_{ff}
Pitch, p

Figure A3.2–3: Assembly Layout Diagram

3.2-2.1.1 Basic Dimensions Calculation

Depending on the choice of preprocessor inputs IPDOPT and ISC, the SAS4A/SASSYS-1 input for the channel geometry will be calculated differently. In any case, the basic geometric inputs listed in Table A3.2-5 are then used to calculate additional geometric information, such as those variables listed in Table A3.2-6. Some of these variables are needed for direct SAS4A/SASSYS-1 input, such as GEOMIN:ACCZ, while others are used to calculate other variables in the following sections, such as the flow splits or subassembly-to-subassembly heat transfer coefficients.

Table A3.2-5: Preprocessor Input Geometric Variables

Symbol	Input	Description
D_{ff}	FFID	Flat-to-flat inner diameter
d_{DC}	DEDGE	Distance from duct wall to center of 1st row of pins
D_{pin}	DPIN	Diameter of pin
D_{WW}	DWW	Diameter of wrapper wire
p	PITCH	Pin centerline to centerline distance
l_{WW}	HWW	Wrapper wire lead length
t_C	DRE	Thickness of clad
t_{DW}	TDW	Thickness of duct wall
t_{GAP}	TGAP	Thickness of gap between subassemblies

Table A3.2-6: Calculated Geometric Variables

Symbol	Variable	Description
A_{pin}		Area of pin
A_{WW}		Area of wrapper wire
P_{WW}		Perimeter of wrapper wire
A_{hex}		Area of subassembly hex
P_{hex}		Perimeter of subassembly hex
A_{CC}	GEOMIN:ACCZ	Cross-sectional flow area per pin
D_h	GEOMIN:DHZ	Hydraulic diameter

3.2-2.1.2 Flow Splits Calculation

The flow split calculation starts with establishing the exponential variable α, which is calculated using the preprocessor input BFR. BFR is also used in SAS4A/SASSYS-1 for the calculation of the turbulent friction factor.

$$\alpha = \frac{1 - BFR}{2 + BFR} \tag{A3.2-1}$$

From there, the flow split per region f_{XX} can be calculated using the following equation, where XX is the region abbreviation seen in Table A3.2-7.

$$f_{XX} = A_{CC_{XX}} D_{h_{XX}}^{\alpha} \tag{A3.2-2}$$

Table A3.2-7: Region Abbreviations

Abbreviation - XX	Region
INR	Inner Row of Subchannels
INB	Inner Subchannels (except inner row if IPDOPT = 3)
RN	Corner Subchannels
ED	Edge Subchannels

The total flow area can be found from Table A3.2-8, where each value in the right hand column is added to the total flow area.

Table A3.2-8: Total Flow Area Contributions

Region		Amount Added to Total Flow Area
INR		$3f_{INR}$
INB	If $n_{rows} = 1,$	f_{INB}
	If $n_{rows} \neq 1,$	$(3 - 6n_{rows} + 3n_{rows}^2 - 3^*)f_{INB}$
RN		f_{RN}
ED	If $n_{C_{RN}} > 0,$	$\big(3(n_{rows} - 1)\big)f_{ED}$
	If $n_{C_{RN}} = 0,$	$\big(1 + 3(n_{rows} - 1)\big)f_{ED}$

* Subtract 3 only if the number of inner subchannels > 0

Then the normalized initial coolant flow split $w_{N_{XX}}$ is the value of the flow split for region XX divided by the total flow area, determined by Table A3.2-8, times the total core flow w_{core}, which is defined by the preprocessor input WTOTL.

$$w_{N_{XX}} = \left(\frac{f_{XX}}{Total}\right) w_{core} \tag{A3.2-3}$$

If bypass subchannels (thimble flow corners) are present, the same process holds, where XX is the region from Table A3.2-9.

$$f_{XX} = A_{CC_{XX}} D_{h_{XX}}^{\alpha} \tag{A3.2-4}$$

Table A3.2-9: Bypass Region Abbreviations

Abbreviation	Region
BYE	Bypass Edge Subchannels
BYC	Bypass Corner Subchannels

The total bypass flow area can be found from Table A3.2-10

Table A3.2-10: Total Bypass Flow Area Contributions

Region		Amount Added to Total Bypass Flow Area
BY	If ISC = 3	$6f_{BYE}$
	If ISC = 5	$6(n_{rows} - 1)f_{BYE}$
	If ISC = 6	f_{BYE}
NR	If ISC = 3	0
	If ISC = 5	$6f_{NR}$
	If ISC = 6	0

The bypass normalized initial coolant flow split is then just the regional flow split divided by the total bypass flow area, multiplied by the total bypass flow w_{bypass}, which is defined by the preprocessor input WBYFL.

$$w_{N_{XX}} = \left(\frac{f_{XX}}{Total_{bypass}}\right) w_{bypass} \tag{A3.2-5}$$

3.2-2.2 Subchannel-to-Subchannel Heat Flow Coefficients

For the multiple pin options (ISC = 2, 3, 4, or 5), the subchannel-to-subchannel heat flow coefficients must be determined (Eq. 3.14-7 in Section 3.14.2). The SAS4A/SASSYS-1 input COOLIN:UACHM1, which is used in the coolant subchannel-to-subchannel heat flow per pin unit height calculation, can be split into coolant and clad contributions, as explained in Section 3.14.2,

$$UACHM1_{total} = UACHM1_{XXXX} + UACHM1_{Clad} \tag{A3.2-6}$$

where $UACHM1_{XXXX}$ is the coolant contribution from the XXXX region, and $UACHM1_{Clad}$ is the contribution from the cladding. The regions are described in Table A3.2-11.

Table A3.2-11: Region Abbreviations

Abbreviation - XXXX	Region
ININ	Inner row to inner row
IAIA	Inner average row to inner average row
IAE	Inner average row to edge
INIA	Inner row to inner average row
EDED	Edge to edge
EDCN	Edge to corner

The clad contribution is independent of the region, and is defined below, where k_{ww} is the thermal conductivity of the wrapper wire (preprocessor input XKSS), M is a multiplier (preprocessor input FU1E), and k_{gap} is the thermal conductivity of the coolant in the gap (preprocessor input XKGAP).

$$UACHM1_{Clad} = \frac{12k_{ww}t_cM}{\pi k_{gap}D_{pin}} \tag{A3.2-7}$$

The coolant contribution to COOLIN:UACHM1 depends on the region, and uses the following equations for each of the regions defined in Table A3.2-11,

$$UACHM1_{ININ} = \left(\frac{0.774D_{ww}\left(D_{pin} + D_{ww}\right)}{\left(\frac{D_{pin} + D_{ww}}{\sqrt{3}}\right)D_{pin}}\right)\left(\frac{D_{ww}}{D_{pin}}\right)^{-0.2627} \tag{A3.2-8}$$

$$UACHM1_{IAIA} = \left(\frac{0.774\left(p - D_{pin}\right)p}{\frac{p}{\sqrt{3}}D_{pin}}\right)\left(\frac{p - D_{pin}}{D_{pin}}\right)^{-0.2627} \tag{A3.2-9}$$

$$UACHM1_{IAE} = UACHM1_{IAIA} \tag{A3.2-10}$$

$$UACHM1_{INIA} = \left(\frac{0.774D_{ww}\left(D_{pin} + D_{ww}\right)}{\left(\frac{\frac{p}{\sqrt{3}} + \frac{D_{pin} + D_{ww}}{\sqrt{3}}}{2}\right)D_{pin}}\right)\left(\frac{D_{ww}}{D_{pin}}\right)^{-0.2627} \tag{A3.2-11}$$

$$UACHM1_{EDED} = \frac{1}{2}\left(\frac{0.774\left(2\left(p - D_{pin}\right)\right)p}{pD_{pin}}\right)\left(\frac{2\left(p - D_{pin}\right)}{D_{pin}}\right)^{-0.2627} \tag{A3.2-12}$$

$$UACHM1_{EDCN} = \frac{1}{2}\left(\frac{0.774\left(2\left(p - D_{pin}\right)\right)p}{\left(\frac{p}{2} + \frac{d_{DC}}{\sqrt{3}}\right)D_{pin}}\right)\left(\frac{2\left(p - D_{pin}\right)}{D_{pin}}\right)^{-0.2627} \tag{A3.2-13}$$

The SAS4A/SASSYS-1 input COOLIN:UACHM2, which is used in the coolant subchannel-to-subchannel heat flow per pin unit height calculation, can be found by the following (if IPODPT = 3) for inner row to inner row heat flow,

$$UACHM2_{ININ} = \frac{CXTC * D_{ww}}{2A_{CC_{INR}}} \tag{A3.2-14}$$

and for the inner row to the inner average, ($UACHM2_{ININ}$ and $UACHM2_{INIA}$ are zero for IPODPT $\neq$ 3)

$$UACHM2_{INIA} = \frac{CXTC * D_{ww}}{A_{CC_{INR}} + A_{CC_{INB}}} \tag{A3.2-15}$$

For all IPODPT options, the inner average to inner average value follows,

$$UACHM2_{IAIA} = \frac{CX * (p - D_{pin})}{2A_{CC_{INB}}} \tag{A3.2-16}$$

For the inner average to the edge,

$$UACHM2_{IAE} = \frac{CX * (p - D_{pin})}{A_{CC_{INB}} + A_{CC_{ED}}} \tag{A3.2-17}$$

The value for UACHM2 for both edge to edge, and edge to corner heat flow is zero. The placeholder variables are defined below,

$$CXTC = CMMTC\left(TANSQR * \frac{AR1}{A1PTC}\right)^{\frac{1}{2}} \tag{A3.2-18}$$

$$CX = CMT\left(TANSQR * \frac{AR1}{A1PTC}\right)^{\frac{1}{2}} \tag{A3.2-19}$$

$$\begin{aligned} CMMTC &= 0.1\frac{D_{ww}}{D_{pin}}^{-\frac{1}{2}} && \text{If } n_{SA} < 7 \\ CMMTC &= 0.14\frac{D_{ww}}{D_{pin}}^{-\frac{1}{2}} && \text{If } n_{SA} \geq 7 \end{aligned} \tag{A3.2-20}$$

$$\begin{aligned} CMT &= 0.1\left(\frac{p - D_{pin}}{D_{pin}}\right)^{-\frac{1}{2}} && \text{If } n_{SA} < 7 \\ CMT &= 0.14\left(\frac{p - D_{pin}}{D_{pin}}\right)^{-\frac{1}{2}} && \text{If } n_{SA} \geq 7 \end{aligned} \tag{A3.2-21}$$

$$TANSQR = \frac{1}{\left(\frac{h_{ww}}{\sqrt{h_{ww}^2 + \left(\pi\left(D_{pin} + D_{ww}\right)\right)^2}}\right)^2} \tag{A3.2-22}$$

$$AR1 = \frac{\pi\left(D_{pin} + D_{ww}\right)D_{ww}}{6} \tag{A3.2-23}$$

$$A1PTC = \frac{1}{4}p^2\sqrt{3} - \frac{\pi D_{pin}^2}{8} \tag{A3.2-24}$$

For swirl flow, the SAS4A/SASSYS-1 input COOLIN:XKSWRL, which is used to calculate swirl flow between subchannels, can be found using the following for the edge to edge calculation,

$$XKSWRL_{EDED} = \frac{C1L(p - D_{pin})}{A_{CC_{ED}}\beta} \tag{A3.2-25}$$

where,

$$\begin{aligned} &\beta = \frac{1}{2} && \text{If } ISC > 3 \\ &\beta = \frac{1}{6} + \frac{1}{2}(n_{rows} - 1) && \text{If } ISC \leq 3 \end{aligned} \tag{A3.2-26}$$

For XKSWRL from the edge to corner,

$$XKSWRL_{EDCN} = XKSWRL_{EDED} \tag{A3.2-27}$$

Lastly, the value for XKSWRL from the corner to edge depends on the selection for ISC,

$$\begin{aligned} &XKSQRL_{CNED} = \frac{6(C1L)(p - D_{pin})}{A_{CC_{CN}}} && \text{If } ISC > 3 \\ &XKSWRL_{CNED} = XKSWRL_{EDED} && \text{If } ISC \leq 3 \end{aligned} \tag{A3.2-28}$$

Where the placeholder variables are defined below,

$$C1L = CST\left(TANSQR\left(\frac{\left(\frac{3}{2}\right)\pi\left(D_{pin}+D_{ww}\right)\left(\frac{D_{ww}}{6}\right)}{pd_{DC}-\frac{\pi D_{pin}^2}{8}}\right)\right)^{\frac{1}{2}} \tag{A3.2-29}$$

$$CST = 0.1\left(\frac{h_{ww}}{D_{pin}}\right)^{0.3} \qquad \text{If } n_{SA} < 7$$

$$CST = 0.75\left(\frac{h_{ww}}{D_{pin}}\right)^{0.3} \qquad \text{If } n_{SA} \geq 7 \tag{A3.2-30}$$

The calculation of the SAS4A/SASSYS-1 input COOLIN:ALATRL, which is the coolant lateral flow area per pin, depends greatly on the pin options chosen by ISC and IPDOPT, with too many possible combinations to review in this document.

3.2-2.3 Subassembly-to-Subassembly Heat Transfer

Before the subassembly-to-subassembly heat transfer calculations can be conducted, a series of more detailed geometric variables must be found that includes many SAS4A/SASSYS-1 inputs. These variables, shown in Table A3.2-12, are calculated using the basic geometric information found in Section 3.2-2.1.

Table A3.2-12: Detailed Geometric Variables

Block		Variable	Description
COOLIN	61	RER0	Nominal cladding outer radius
	61	RBR0	Nominal cladding inner radius
	61	DRF0	Thickness of the outer reflector node
	61	DRFI	Thickness of the inner reflector node
	61	SER	Reflector perimeter per pin wetted by coolant
	61	RBRPL	Cladding inner radius in gas plenum
	61	RERPL	Cladding outer radius in gas plenum
	61	AREAPC	Coolant plus pin area per pin
	61	SRFSTZ	Structure perimeter
	61	DSTIZ	Thickness of inner structure node
	61	DSTOZ	Thickness of outer structure node
POWIN	62	PRSHAP	Ratio of power per subassembly
PMATCH	63	XKSTIZ	Inner structure thermal conductivity
	63	XKSTOZ	Outer structure thermal conductivity
	63	RHOCSI	Density*heat capacity for the inner structure
	63	RHOCSO	Density*heat capacity for the outer structure
	63	RHOCR	Density*heat capacity for the reflector

	63	XKRF	Thermal conductivity of reflector/cladding (in gas plenum)
COOLIN	64	W0	Steady-state coolant flow rate per fuel-pin

3.2-2.3.1 Single-Pin Options

For the single pin treatment with no bypass channel (ISC = 0 or 1), the preprocessor will only calculate subassembly-to-subassembly heat transfer if the channel represents one subassembly. If the channel represents more than one subassembly, this input must be calculated manually. The first step for calculating subassembly-to-subassembly heat transfer is to determine which subassemblies are in contact. Using the core symmetry option ISYM and the number of rows NROW, the preprocessor finds the six adjacent assemblies. The preprocessor resolves whether a subassembly is at an adiabatic boundary.

Once the adjacent subassemblies are known, the SAS4A/SASSYS-1 input related to subassembly heat transfer, shown in Table A3.2-13, is calculated for each flat of the subassembly. This information is then printed to the preprocessor output in the correct format for insertion into a SAS4A/SASSYS-1 input deck.

Table A3.2-13: Subassembly Heat Transfer SAS Inputs

Block	Variable	Description
51	NCHCH	Number of other channels in contact
51	ICHCH	Channel number that is in contact
51	IOPCH	Subassembly to subassembly heat transfer option
63	HACHCH	Heat-transfer coefficient x area per unit height

For the single pin treatment with bypass channel (ISC = 6), the additional geometric parameters relating to the bypass channel are found, and then the adjacent subassemblies are determined. From there, the process follows a similar path as the single pin options with no bypass channel, where the variables in Table A3.2-13 are found (but taking into account the bypass) and the output is printed.

3.2-2.3.2 Multiple-Pin Options

The calculation process is more complicated for the multi-pin options (ISC = 2, 3, 4, or 5). For each row of pins, the detailed geometric variables in Table A3.2-11 are calculated. The power gradient is found first using the preprocessor input IDRGRD. This information, along with the preprocessor input PRSHP0, is necessary to calculate the SAS4A/SASSYS-1 input POWINC:PRSHAP. The formulas used to calculate the variables in Table A3.2-11 differ depending on the choice of IPDOPT and ISC (whether grouped by sector and row or individual subchannels). Where the channel is located, whether it is an inner, edge, or corner channel, also affects the calculation. The location is determined using the number of rows NROW and the choices for IPDOPT and ISC. This is also used to calculate the SAS4A/SASSYS-1 input INPCHN:JCHMPN, which denotes what channels are in contact.

The heat flow parameters found in Section 3.2-2.2 are then adjusted (normalized) for the different pin nodes. The calculation of the subassembly heat transfer variables in Table A3.2-12 is also more intricate than the single pin models, and depends on whether individual subchannels are being used (ISC = 4, 5) or subchannels grouped by row and sector (ISC = 2, 3).

3.2-3 User Guide

3.2-3.1 Overview

The current preprocessor is useful for a limited subset of the cases that SAS4A/SASSYS-1 can handle, as shown in Table A3.2-14. The current preprocessor was written only for hexagonal subassemblies with solid duct walls and wire wrapped pins. A modified preprocessor would be needed to handle square arrays, pins with grid spacers instead of wire wraps, or ductless subassemblies. Also, the preprocessor does not handle subassembly-to-subassembly heat transfer involving a channel that represents more than one subassembly.

Table A3.2-14: Scenarios Covered by Preprocessor

Covered	**Not Covered**
- Hexagonal Subassemblies	- Square Subassemblies
- Solid Duct Walls	- Ductless Subassemblies
- Wire Wrapped Pins	- Pins with Grid Spacers

The current preprocessor provides input only for the pin section of a subassembly. Input for the reflector zones above and below the pin section must be input into SAS4A/SASSYS-1 manually, as shown in Figure A3.2–4, but such information only needs to be input once for each subassembly type, as explained in Section 3.15.3.1.

The preprocessor numbers the channels in a consistent manner, and determines subchannel connectivity. In the pin zone the preprocessor calculates three types of parameters. As described in Section 3.2-2, it first calculates geometrical parameters, such as coolant flow areas and hydraulic diameters. Second, it uses correlations to calculate the parameters used for subchannel-to-subchannel heat flow. Third, it calculates the parameters used in subassembly-to-subassembly heat transfer.

The preprocessor can also print additional user-specified information directly to the created SAS4A/SASSYS-1 input in order to help reduce the time needed to create the SAS4A/SASSYS-1 input. This allows a block of text to be input once into the preprocessor, which then duplicates that text for all channels of that type. This is done through the ICARD option, and is explained in more detail in Section 3.2-2.3 and 3.2-3.5.

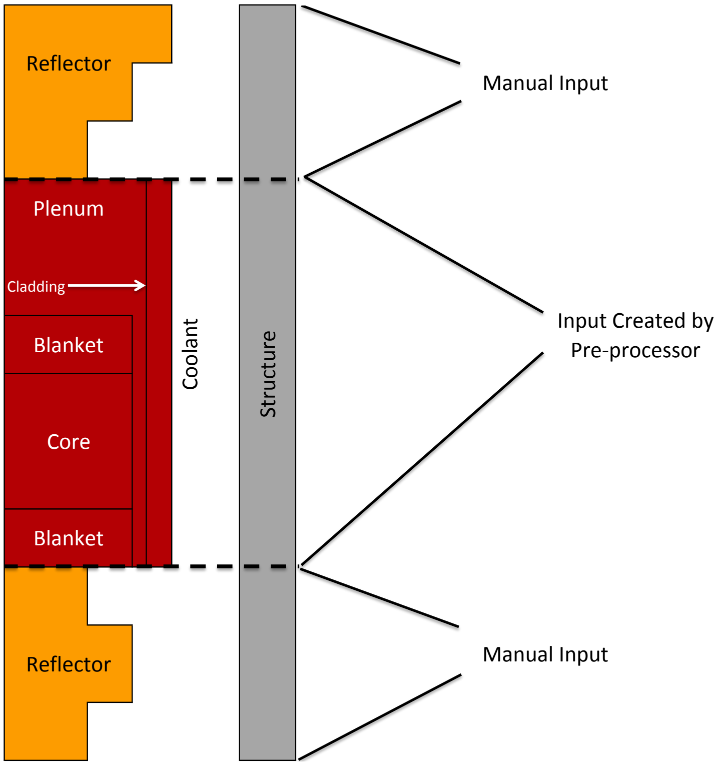


Figure A3.2–4: Preprocessor Coverage

The preprocessor could be of some use even if the three dimensional core thermal hydraulics model is not being used. For a single average channel treatment, the preprocessor can calculate the geometrical parameters such as coolant flow areas and hydraulic diameters for each subassembly. Also, if a separate channel represents each subassembly, then input for subassembly-to-subassembly heat transfer can be calculated by the preprocessor.

3.2-3.2 Input/Output Format

The layout of the input file can be broken into three main sections, as shown in Figure A3.2–5. The general input is first, then the subassembly type(s) information, which can also include the optional visualization input, followed by the individual subassembly information. This last section can also include the ICARD information if the preprocessor options ICD or ICDBFR are being used, which allow additional text to be added to the preprocessor output.

Table A3.2-15 provides more detail about the input structure, beginning with the general input. Each subassembly type must have its ITYPE value in consecutive order. Then, ITYPE = 0 is used to signal the end of the subassembly type input. Also, the optional visualization input is entered independently for each subassembly type. A similar structure is used for the subassembly input, where the value for ISUBAS must be in consecutive order, and ISUBAS = 0 signals the end of the subassembly input. The optional ICARD data is entered independently for each subassembly. The format of the input is described in Table A3.2-16.

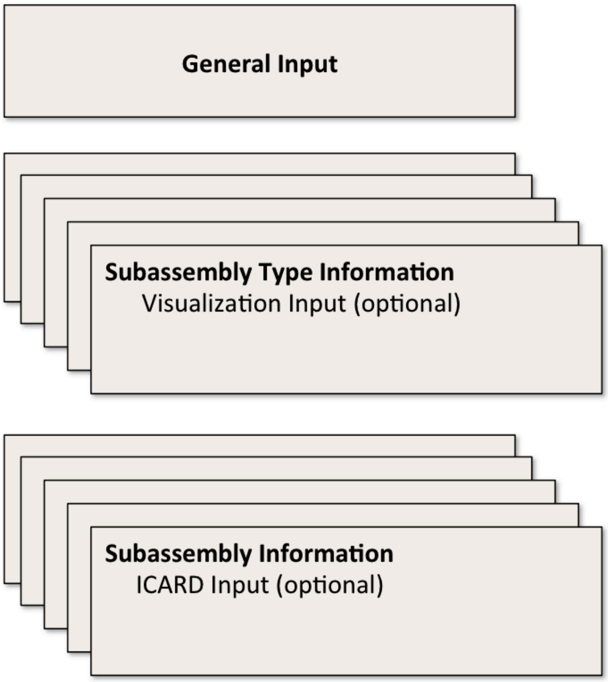


Figure A3.2–5: Preprocessor Input Layout

Table A3.2-15: Preprocessor Input Structure

General Input
1st Subassembly Type Input (ITYPE = 1) Visualization Input (optional) 2nd Subassembly Type Input (ITYPE = 2) Visualization Input (optional) ⋮ Last Subassembly Type Input Visualization Input (optional) ITYPE 0
1st Subassembly Input (ISUBAS = 1) ICARD Input (optional) 2nd Subassembly Input (ISUBAS = 2) ICARD Input (optional) ⋮ Last Subassembly Input ICARD Input (optional) ISUBAS 0

Table A3.2-16: Preprocessor Input Formatting

Section/Variable	Fortran Format
General Input	
ITITLE	(A80)
NROW, ISYM, IFSTCH, IHEXPL, IHEXP2, IDBUG, IFSTSA	7(7X,I5)
Subassembly Type Input	
ITYPTL	(A80)
ITYPE, ISC(ITYPE), NPINSB(ITYPE), IPDOPT(ITYPE), KZPIN(ITYPE), IBYOPT(ITYPE), ISWDIR(ITYPE)	7(7X,I5)
DPIN(ITYPE), DWW(ITYPE), PITCH(ITYPE), DEDGE(ITYPE)	4(10X,F10.5)
FFID(ITYPE), HWW(ITYPE), TBY(ITYPE), TDWBY(ITYPE)	4(10X,F10.5)
AFR(ITYPE), BFR(ITYPE), AFLAM(ITYPE), RELAM(ITYPE)	4(10X,F10.5)
DRE(ITYPE), FU1E(ITYPE), HCBY(ITYPE)	3(10X,F10.5)
TDW(ITYPE), TGAP(ITYPE), RHOCSS(ITYPE), RHOCNA(ITYPE)	4(10X,E10.3)
XKSS(ITYPE), XKGAP(ITYPE), FRSTIN(ITYPE), FRSTED(ITYPE)	4(10X,E10.3)
Visualization Input (optional)	
LDETL(ITYPE), MZ(ITYPE), NT(ITYPE), IEQMAS(ITYPE)	6(7X,I5)
(ZMZ(IZ,ITYPE),IZ=1,MZ)	(6E12.5)
RINFP, ROUTFP	(2E12.5)
Subassembly Input	
ISUBAS, ITYPEV(ISUBAS), KMAX, IDRGRD(ISUBAS), NSUB(ISUBAS), ICD(ISUBAS)	6(7X,I5)
SATITL	(A80)
ICDBFR, IHTLAT(ISUBAS)	2(7X,I5)
WTOTL(ISUBAS), PRSHP0(ISUBAS), PGRAD(ISUBAS), ADOPPN(ISUBAS)	4(10X,F10.6)
BDOPPN(ISUBAS), WBYFL(ISUBAS), HALATP(ISUBAS)	3(10X,F10.6)
(IHXPSS(K), IHXPSE(K),K=1,KMAX)	(20X,12I5)
NPRINT, (ICHPRT(I,ISUBAS),I=1,NPRINT)	(20x,12I5)

3.2-3.3 Input Description

As stated in Section 3.2-3.2 and shown in Table A3.2-17, the input for the preprocessor can be split into three major sections with the visualization input under the subassembly type section. The following tables outline the input for each section. The input needed for the visualization tool, described in Appendix 3.3, is also provided.

Table A3.2-17: Preprocessor Input Sections

Section	Description
General Input	General Input
Subassembly Type Input	Specifies pin treatment and geometric options for each subassembly type
↳ Visualization Input (optional)	Specifies details for creation of visualization mesh, described in Appendix 3.3
Subassembly Input	Subassembly specific data

General Input

Symbol	Description
ITITLE	One line of title information. (80 character limit)
NROW	Number of rows of subassemblies in the model.
ISYM	Symmetry information = 1, Full core = 2, Half core = 3, Third core = 4, Quarter core = 6, Sixth core = 12,1/12 core
IFSTCH	First channel number to use.
IHEXPL	Plot information = 0, Skip plot of subassembly hexes numbered with hex positions for the symmetry and number of rows being used = 1, Plot on fort.98 and continue = 2, Plot on fort.98 and quit
IHEXP2	Plot information = 0, Skip plot of subassembly hexes numbered with subassembly numbers > 0, Plot subassembly hexes numbered with subassembly numbers on fort.99 = 2, Generate visualization geometry file for use with VisIt, additional input must be provided as described in Appendix 3.3
IDBUG	Debugging output > 0, Debugging output
IFSTSA	First subassembly number to use.

Subassembly Type Input

Symbol	Description
ITYPTL	One line of type title information. (80 character limt)
ITYPE	Subassembly type. If ITYPE=0, this is the end of the type input.
ISC (ITYPE)	Pin treatment option = 0, JJMLTP = 0, No multi-pin treatment = 1, JJMLTP = 1, 1 pin treatment = 2, Multi-pin, subchannels grouped by row and sector = 3, Multi-pin, same as 2, but bypass flow channel on outside = 4, Multi-pin, individual subchannels = 5, Multi-pin, individual subchannels, bypass flow channel on the outside = 6, JJMLTP = 1, 1 pin treatment, bypass flow channel
NPINSB (ITYPE)	Number of pins per subassembly.
IPDOPT (ITYPE)	Geometry Calculation Options = 0, Use input PITCH everywhere, calculate DEDGE based on what is left over, use input FFID = 1, Use FFID, calculate uniform PITCH, DEDGE = 2, Use input PITCH, calculate DEDGE with same slop, calculate FFID = 3, Hit channel option: use FFID, use: P = DPIN + DWW for 1 inner row of subchannels, use PITCH for other inner subchannels, calculate DEDGE based on what is left over = 4, Use PITCH and DEDGE, calculate FFID
KZPIN (ITYPE)	Zone KZ

Subassembly Type Input

Symbol	Description
IBYOPT (ITYPE)	Bypass information = 0, Bypass flow the whole length of the subassembly = 1, Bypass flow only in the pin section Note: IBYOPT is only relevant if ISC = 3, 5, or 6. IBYOPT = 1 is not currently implemented
ISWDIR (ITYPE)	Swirl information = 1, Swirl flow is counter-clockwise, when viewed from above = 2, Swirl flow is clockwise
DPIN (ITYPE)	Pin Diameter.
DWW (ITYPE)	Wrapper wire diameter.
PITCH (ITYPE)	Pin Centerline to Centerline Distance.
DEDGE (ITYPE)	Distance from duct wall to center of first row of pins.
FFID (ITYPE)	Flat-to-flat ID.
HWW (ITYPE)	Wire wrap lead length.
TBY (ITYPE)	Thickness of bypass flow gap.
TDWBY (ITYPE)	Thickness of the outer duct wall beyond the bypass gap.
AFR (ITYPE)	Turbulent Friction Factor = AFR*Re**BFR
BFR (ITYPE)	Turbulent Friction Factor = AFR*Re**BFR

Subassembly Type Input

Symbol	Description
AFLAM (ITYPE)	Laminar Friction Factor = AFLAM/Re
RELAM (ITYPE)	Reynolds number for transition from turbulent to laminar. If RELAM = 0, then it is calculated so that turbulent friction factor equals laminar friction factor at RELAM.
DRE (ITYPE)	Clad thickness.
FU1E (ITYPE)	Geometry multiplier for subchannel-to-subchannel heat transfer by conduction through cladding. Typical value near 2.0.
HCBY (ITYPE)	Coolant heat transfer coefficient from outer surface of duct wall to thimble flow coolant. For laminar flow, HCBY = 7.6*k/Dh to 8.23*k/Dh.
TDW	Thickness of duct wall.
TGAP	Thickness of gap between subassemblies.
RHOCSS	Density × heat capacity of the wrapper wire and duct wall.
RHOCNA	Density × heat capacity of the coolant in the gap.
XKSS	Thermal conductivity of the wrapper wire and duct wall.
XKGAP	Thermal conductivity of the coolant in the gap.
FRSTIN	Fraction of structure thickness in the inner node, inner subchannel.
FRSTED	Fraction of structure thickness in the inner node, edge subchannel.

Visualization Input (if IHEXP2 = 2)[4]

Symbol **Description**

LDETL Level of detail in the visualization mesh geometry file.

= 0, Do not generate geometry for this assembly type.
= -1, Only simplified coolant subchannels are defined.
= -2, Simplified coolant subchannels and duct wall structure.
= 1, Detailed coolant subchannels are defined.
= 2, Detailed coolant subchannels and duct wall structure.
= 3, Detailed subchannels, duct wall, and simple fuel pin.
= 4, Detailed subchannels, duct wall, and detailed fule pin.

MZ Number of axial segments in zone KZPIN (e.g., the MZ mesh).

NT Number of radial temperature nodes within the fuel (only relevant for LDETL = 4).

IEQMAS Radial fuel mesh size assumptions (relevant for LDETL=4).

= 0, Equal radial distance.
> 0, Equal cylindrical area.

ZMZ Axial segment boundaries for zone KZPIN, where the lowest boundary is assumed to be zero. Note that these and the above values are only used to generate the geometry file, and are not used to define NZNODE, AXHI, or ZONEL.

RINFP Fuel inner radius.

ROUTFP Fuel outer radius (default = clad inner radius)

[4] These parameters are only used to generate the visualization mesh geometry and, with the exception of LDETL, are not incorporated into the SAS4A/SASSYS-1 input that is generated for the subchannel definitions.

Subassembly Input

Symbol	Description
ISUBAS	Subassembly number. Max = NSBMAX = 1000. If ISUBAS = 0, then this is the end of the subassembly input.
ITYPEV (ISUBAS)	ITYPE = subassembly type.
KMAX	Highest pin axial node, as shown in Figure 3.15–2
IDRGRD (ISUBAS)	Direction of power gradient. = 1, N(high) – S(low) = 2, NNE – SSW = 3, NE – SW = 4, E –W = 5, SE – NW = 6, SSE – NNW
NSUB (ISUBAS)	Number of real subassemblies represented by this ISUBAS
ICD	If > 0, Read in cards, ICARD, to be put at the end of the first channel in ISUBAS
SATITL (ISUBAS)	Subassembly title information.
ICDBFR	If > 0, Read in cards to be put before the first channel in ISUBAS.
IHTLAT	If > 0, Use HALATP(ISUBAS) for lateral heat transfer across pins. Currently, only implemented for the detailed subchannel treatment (ISC = 4 or 5). Not implemented for the one pin per subassembly case.
WTOTL (ISUBAS)	Total subassembly coolant flow, excluding thimble flow.

Subassembly Input

Symbol	Description
PRSHP0 (ISUBAS)	Average PRSHAP for the subassembly. PRSHAP = Ratio of power per subassembly averaged over this channel to the power per subassembly averaged over all channels.
PGRAD (ISUBAS)	Power gradient. $PRSHAP = \frac{PRSHP0(1+PGRAD)XPIN}{NPINSB}$ at SA[5] Edge $PRSHAP = \frac{PRSHP0 \times XNPIN}{NPINSB}$ at SA Center $PRSHAP = \frac{PRSHP0(1-PGRAD)XPIN}{NPINSB}$ at SA Other Edge
ADOPPN	ADOP per pin. ADOP = Doppler coefficient for this channel when part of the core represented by this channel is not voided.
BDOPPN	BDOP per pin. BDOP = Doppler coefficient for this channel when part of the core represented by this channel is fully voided.
WBYFL	Total thimble flow. Used only if ISC = 3, 5, or 6.
HALATP	Heat transfer coefficient × area/height for lateral heat transfer across pins.
IHXPSS (K)	Subassembly ISUBAS represents subassemblies with hex position numbers from IHXPSS(K) to IHXPSE(K)
IHXPSE (K)	See above

[5] SA = Subassembly

Subassembly Input

Symbol	Description
NPRINT	Number of channels in this subassembly which will be printed out in normal prints (set IPRSKP = 0, in all other channels IPRSKP will be set to 1).
ICHPRT	Relative channel number (1 for the first channel in the subassembly).
ICARD	Information to be read and directly printed to fort.1 output. See ICD and ICDBFR.

3.2-3.4 Input Example

An example input is provided in Figure A3.2–6, where "#" is used as a placeholder for user specified values. This example has two subassembly types, and two subassemblies. The visualization tool is also used for both subassembly types. The two subassemblies use the ICARD option, with the first subassembly (ISUBAS = 1) using ICDBFR, and the second subassembly (ISUBAS = 2) using ICD.

```
Title Information (ITITLE)
 NROW      # ISYM      1 IFSTCH    # IHEXPL    1 IHEXP2    2 IDBUG     0 IFSTSA    #
 Subassembly Type 1 (ITYPTL)
 ITYPE     1 ISC       4 NPINSB    # IPDOPT    1 KZPIN     2 IBYOPT    0 ISWDIR    #
 DPIN      #.######  DWW       #.######  PITCH               DEDGE
 FFID      #.######  HWW       #.######  TBY                 TDWBY
 AFR       #.######  BFR      -#.######  AFLAM    #.######   RELAM     #.######
 DRE       #.######  FU1E      #.######  HCBY
 TDW       #.###E#   TGAP      #.###E#   RHOCSS    #.###E#   RHOCNA    #.###E#
 XKSS      #.###E#   XKGAP     #.###E#   FRSTIN    #.###E#   FRSTED    #.###E#
 LDETL     4 MZ       20 NT       10 IEQMAS    0
     0.04000     0.08000     0.12000     0.16000     0.20000     0.24000
     0.28000     0.32000     0.36000     0.40000     0.44000     0.48000
     0.52000     0.56000     0.60000     0.64000     0.68000     0.72000
     0.76000     0.80000
     0.00000     0.00000
 Subassembly Type 2 (ITYPTL)
 ITYPE     2 ISC       4 NPINSB  ### IPDOPT    1 KZPIN     2 IBYOPT    0 ISWDIR    #
 DPIN      #.######  DWW       #.######  PITCH               DEDGE
 FFID      #.######  HWW       #.######  TBY                 TDWBY
 AFR       #.######  BFR      -#.######  AFLAM    #.######   RELAM     #.######
 DRE       #.######  FU1E      #.######  HCBY
 TDW       #.###E#   TGAP      #.###E#   RHOCSS    #.###E#   RHOCNA    #.###E#
 XKSS      #.###E#   XKGAP     #.###E#   FRSTIN    #.###E#   FRSTED    #.###E#
 LDETL     4 MZ       20 NT       10 IEQMAS    0
     0.04000     0.08000     0.12000     0.16000     0.20000     0.24000
     0.28000     0.32000     0.36000     0.40000     0.44000     0.48000
     0.52000     0.56000     0.60000     0.64000     0.68000     0.72000
     0.76000     0.80000
     0.00000     0.00000
 End of Subassembly Type Input:
 ITYPE    0
 ISUBAS    1 ITYPEV    1 KMAX      1 IDRGRD    # NSUB      1 ICD       0
 Subassembly Information 1 (SATITL)
 ICDBFR    1 IHTLAT    #
 WTOTL     #.######  PRSHP0  #.######    PGRAD    #.######   ADOPPN   -#.######
 BDOPPN   -#.######  WBYFL               HALATP   #.######
 IHXPSS, IHXPSE         1    1
 NPRNT, ICHPRT(I)       1    1
 Read in cards, ICARD, to be put before the first channel in ISUBAS (Since ICDBFR = 1)
ENDICD
 ISUBAS    2 ITYPEV    2 KMAX      1 IDRGRD    # NSUB      1 ICD       1
 Subassembly Information 2 (SATITL)
 ICDBFR    0 IHTLAT    #
 WTOTL     #.######  PRSHP0  #.######    PGRAD    #.######   ADOPPN   -#.######
 BDOPPN   -#.######  WBYFL               HALATP   #.######
 IHXPSS, IHXPSE         1    1
 NPRNT, ICHPRT(I)       1    1
 Read in cards, ICARD, to be put at the end of the first channel in ISUBAS (Since ICD =
1)
ENDICD
 End of Subassembly Input:
 ISUBAS    0
```

Figure A3.2–6: Preprocessor Example Input

3.2-3.5 Output Description

Lastly, several output files are created by the preprocessor, as seen in Table A3.2-18. The first output file is the direct output from the preprocessor, which documents notifications printed during its execution. The second file, fort.1 is the SAS4A/SASSYS-1 input created by the preprocessor. If the ICARD option is being used (ICD or ICDBFR > 0), then ICARD information will be printed directly to fort.1. The location of the ICARD information will be before the first channel in ISUBAS if ICDBFR is being used, or at the end of the first channel in ISUBAS if ICD is used.

The preprocessor can optionally produce postscript plots on fort.98 and fort.99. On fort.98 it can make a plot of the subassembly hexes numbered with the position numbers used internally by the preprocessor. On fort.99 it can make a plot of the subassembly hexes numbered with the subassembly numbers assigned by the user. The two plots could be identical if the user picks the same numbering scheme that the preprocessor uses internally. Lastly, the optional visualization output is created. More details on this file can be found in Appendix 3.3.

Table A3.2-18: Output File Description

Output	Description
*.out	Log
fort.1	SAS4A/SASSYS-1 input
fort.98	Plot of the subassembly hexes numbered with the position numbers used internally by the preprocessor (postscript)
fort.99	Plot of the subassembly hexes numbered with the subassembly numbers assigned by the user (postscript)
SAS.sasgeom	Visualization mesh file (optional)

APPENDIX 3.3:
VISUALIZATION

3.3-1 Introduction

An advanced visualization capability has been developed for the SAS4A/SASSYS-1 code [1] that allows for the generation, display, and animation of large datasets that are calculated during a transient simulation. The key features of the new capability include its support for multiple-pin, subchannel modeling, integration with the subchannel input preprocessor, geometry visualization, fully-detailed temperature field support with unlimited dataset sizes, and interactive visualization and animation generation on a desktop PC using the visualization tool VisIt [2].

The motivation for developing an advanced visualization capability results from the ever-increasing fidelity with which fast reactor transient simulations are being carried out and the enormous amounts of data generated during those simulations. For example, the EBR-II SHRT-17 test has recently been modeled [3] with subchannel-level detail and a coarse axial mesh for the experimental assembly XX09 and its six nearest neighbors. For a simulation that lasts 250 seconds with data saved every one-tenth of a second, the full temperature field in the active core region for these seven assemblies is estimated to be approximately 15 GB.[6] While it is trivial to identify maxima and minima within this large dataset, identifying other features is much more difficult. Items of interest might include the identification of power, flow, or temperature fluctuations caused by instabilities in the numerical solution, regions of temperature near (but not at) the peak, flow reversals at the subchannel level, unusual results caused by input error, time-derivatives of solution variables, and so on.

The following sections describe changes made to the subchannel input preprocessor to support creation of visualization mesh geometry and preliminary changes made to the SAS4A/SASSYS-1 code to support large dataset generation. Finally, examples of using the visualization tool VisIt to render three-dimensional images from the SHRT-17 simulation are provided.

3.3-2 Mesh Generation

Channel modeling in the SAS4A/SASSYS-1 code is fundamentally based on a single-pin representation that includes associated coolant and structure. Geometric input to the SAS4A/SASSYS-1 code is represented by generalized parameters such as flow area, hydraulic diameter, heat-transfer area, etc, and details of the problem-specific geometry are not fully known. In order to project computed transient data into three-dimensional space, a visualization mesh representing the actual geometry of the problem needs to be defined.

Detailed core models using the multiple-pin subchannel capability in the SAS4A/SASSYS-1 safety analysis code are usually created using the subchannel input

[6] Animations of the SHRT-17 simulation can be viewed at http://internal.ne.anl.gov/~fanning/SHRT17.html

preprocessor, detailed in Appendix 3.2. The input preprocessor takes subassembly design information and creates detailed SAS4A/SASSYS-1 input descriptions of the multiple-pin subchannels in terms of geometric, thermal, and hydraulic parameters. Because the input preprocessor contains subassembly design information that is not available to the SAS4A/SASSYS-1 code, it was modified to act as the tool to be used in creating the visualization mesh geometry. A visualization mesh is defined so that computed data can be mapped into three-dimensional space. Furthermore, the mesh can be viewed independent of computed data to identify subchannel locations in the global geometry. Details on the format of the visualization mesh geometry file are available at the end of this appendix.

An example of the visualization mesh created for the SHRT-17 simulation is shown in Figure A3.3–1, where the mesh is colored according to the type of geometry present. In this case, gray represents cladding and structure, blue represents sodium coolant, orange is an average fuel region, and red is a detailed fuel region. The bypass flow area of the experimental assembly XX09 is visible in the assembly located in the upper-right portion of the figure.

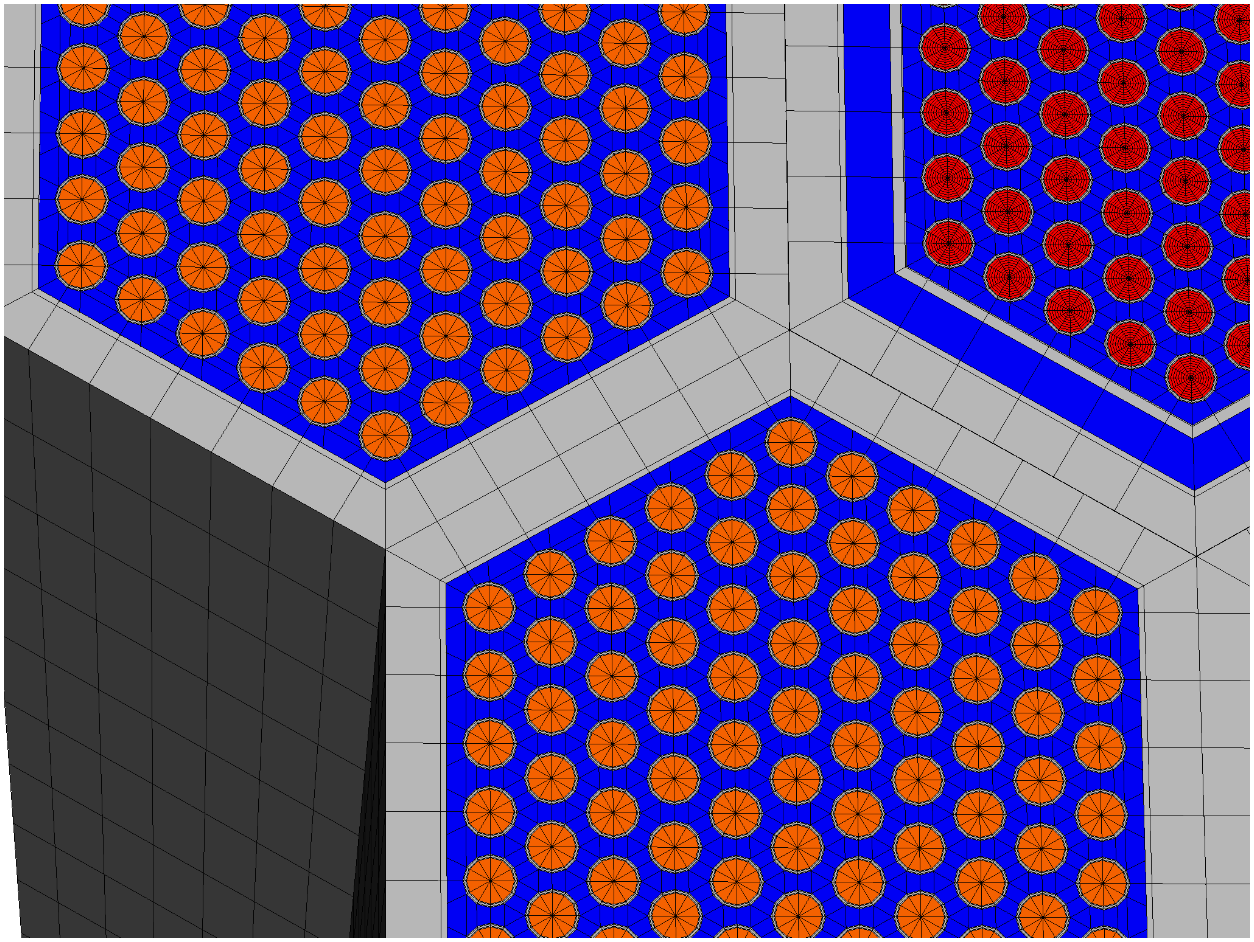


Figure A3.3–1: Portion of the Visualization Mesh Geometry for the EBR-II SHRT-17 Model.

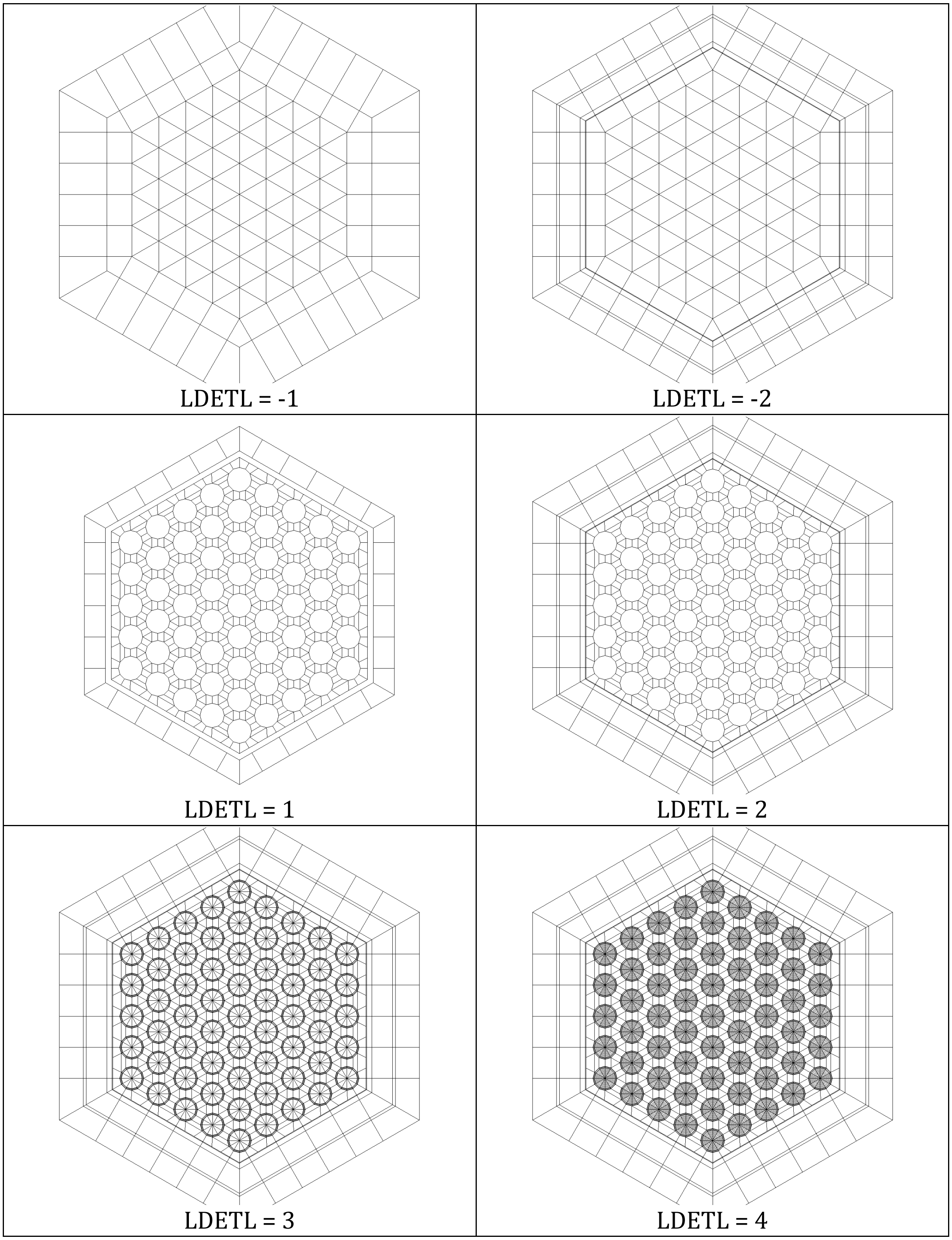
Because of the potential for the enormous amounts of data that need to be generated in order to fill a visualization mesh, an option is provided to control the level of detail represented in the visualization mesh geometry file on an assembly-by-assembly basis. Figure A3.3–1 displays two of the many detail options available. In Figure A3.3–1, experimental assembly XX09 is represented by a detailed radial pin mesh that includes nine radial temperature zones in the fuel region. The surrounding assemblies, however, are represented by an average fuel region that includes only a single radial zone for the bulk average fuel temperature. This reduces the data file for the 250 second simulation from the 15 GB estimate described earlier down to approximately 3.5 GB. The different options for the level of detail in the visualization mesh geometry file are listed in Table A3.3-1 and examples of the mesh for each option are shown in Figure A3.3–2.

The detail options shown in Figure A3.3–2 include two very simplified subchannel representations that use a single cell volume for each subchannel (LDETL = -1 or -2). For the interior subchannels, triangular cells are defined; and for the edge and corner subchannels, quadrilateral cells are defined. For LDETL = -1, these cells (artificially) extend over the fuel pin and structure regions in order to occupy the full assembly volume. For LDETL = -2, the cells only extend over the fuel pin regions, and additional cells are defined to represent the duct wall structure regions.

Positive values of LDETL define multiple coolant cells to approximate the actual shape of the coolant subchannel. LDETL = 1 and LDETL = 2 have void regions in the geometry since neither define regions for fuel pins and the former does not define regions for the duct wall structure. The advantage of LDETL = 1 and LDETL = 2 is that they require the same data as with the corresponding negative values for LDETL but they more accurately represent the physical geometry. LDETL options 3 and 4 introduce regions defined for the fuel pin, including fuel and cladding. The difference between the two options is that LDETL = 3 represents the fuel region by a single radial zone, while LDETL = 4 represents the fuel region by multiple radial zones corresponding to the parameter NT. (See below.)

Table A3.3-1: Geometry Detail Options for Creating Visualization Mesh Geometry Files.

Geometric Detail Option (LDETL Parameter)	Geometric Detail
0	No Geometry Defined
-1	Simplified Coolant Subchannels
-2	Simplified Coolant Subchannels and Duct Wall Structure
1	Detailed Coolant Subchannels
2	Detailed Coolant Subchannels and Duct Wall Structure
3	Detailed Coolant Subchannels, Duct Wall Structure, and Simplified Fuel Pin
4	Detailed Coolant Subchannels, Duct Wall Structure, and Detailed Fuel Pin



LDETL = -1

LDETL = -2

LDETL = 1

LDETL = 2

LDETL = 3

LDETL = 4

Figure A3.3–2: Examples of Geometry Detail Options Applied to the XX09 Experimental Assembly.

It should be noted that even though the coolant subchannels are defined by multiple cells for positive values of LDETL (four in the case of the interior subchannels) SAS4A/SASSYS-1 still only calculates a single temperature for each axial position along a subchannel and that temperature is applied to all cells that belong to a given subchannel. Therefore, it is important to recognize the distinction between the *visualization* mesh described here and the *computation* mesh (i.e. channel model) used by the SAS4A/SASSYS-1 code.

In the multiple-pin subchannel preprocessor, all new parameters associated with the generation of a visualization mesh geometry file are confined to the assembly type definition input. This includes the geometric detail option parameter LDETL. Therefore, even if two assemblies are of the same *type* (both driver assemblies, for example) different types will need to be defined if a different level of geometric detail is desired for each. In the modified preprocessor, the additional parameters are expected only if the existing parameter IHEXP2 = 2. This assures compatibility with existing input files created for earlier versions of the preprocessor. When IHEXP2 = 2, the additional visualization input, shown in Appendix 3.2-3.3, must be added to the end of each assembly type definition.

Using the geometry information contained in the preprocessor input file, along with the additional visualization input shown in Appendix 3.2-3.3, the multiple-pin subchannel preprocessor can create a SAS4A/SASSYS-1 visualization mesh geometry file. By default, the generated file is named “SAS.sasgeom”. Although the basename of the file is not critical, the file extension (*.sasgeom) is used by the VisIt visualization software to identify the correct file reader with which to parse the mesh.

The VisIt visualization software can be used to view the visualization mesh geometry file created by the preprocessor without needing to carry out a simulation to generate data. VisIt has the capability to plot numerous qualities of the visualization mesh, including determining individual cell volumes. In addition to the visualization mesh geometry, the file contains two parameters that can be plotted: channel ID and channel type. Channel ID directly corresponds to the SAS4A/SASSYS-1 channel number, and can be used to identify specific channels in the global geometry. Channel type is somewhat of a misnomer,[7] in that it does not define a type for a given channel but rather a region to which a cell in the channel’s geometry belongs. (Figure A3.3–1 was created by plotting the channel type parameter.) The currently defined types include

- 0: Coolant (always present)
- 1–3: Cladding Radial Components (Inner, Middle, Outer; present when LDETL ≥ 3)
- 4–5: Structure Components (Inner, Outer; present when |LDETL| ≥ 2)
- 10: Bulk Fuel (present when LDETL = 3)
- 11–(10+NT): Radial Fuel Components (present when LDETL = 4)

[7] The misnomer arises because the channel ID and channel type parameters actually exist in the geometry file as a single integer parameter that represents a unique “data channel” that is used to associate simulation data with the visualization mesh.

Data generated during a simulation is associated with a cell in the visualization mesh by matching three parameters: channel ID, axial position, and channel type. This data can then further vary as a function of time. The creation of a SAS4A/SASSYS-1 visualization *data* file is described in the following section.

3.3-3 Data Generation

The visualization capabilities introduced into SAS4A/SASSYS-1 assume that geometry is invariant. This significantly simplifies implementation since the SAS4A/SASSYS-1 code has limited knowledge of the specific geometric details needed to project computed simulation data into three-dimensional space. As described in the previous section, a visualization mesh geometry file is created by the multiple-pin subchannel preprocessor. Modifications have been made to the SAS4A/SASSYS-1 code to produce a separate visualization data file that contains time-dependent simulation data that can be associated with the visualization mesh geometry. Details on the format of the visualization data file are contained in Section 3.3-7.

Two parameters have been added to the SAS4A/SASSYS-1 input description to control the creation of the visualization data file. These two parameters are defined in Table A3.3-2. The first parameter, IVIS3D, indicates how often to write simulation data to the visualization data file. By default, the file created will be named “SAS.sasdata”. Like with the visualization mesh geometry file, the basename of the file is not critical, but the file extension (*.sasdata) is used by the VisIt visualization software to identify the correct file reader with which to parse the data file.

The second parameter, LDETL, is a channel-dependent parameter that controls the extent of data written for each channel. It corresponds to the LDETL parameter defined for the multiple-pin subchannel input preprocessor. When creating channel-dependent input using the input preprocessor, the LDETL parameter will automatically be written to the generated input description. Therefore, a user does not normally need to set this input parameter.

Table A3.3-2: SAS4A/SASSYS-1 Input Parameters for Generation of Visualization Data

Parameter	Description
IVIS3D (Block 1, Location 120)	Visualization Data Generation Flag: = 0, Do not write visualization data > 0, Write visualization data every IVIS3D time steps.
LDETL (Block 51, Location 500)	Visualization Data Detail Flag: = 0, Do not write visualization data for this channel = 1, Write coolant temperatures only = 2, Coolant and duct wall temperatures = 3, Coolant, structure, and average fuel temperatures = 4, Coolant, structure, average fuel, and detailed radial fuel temperatures.

There are two minor differences in the treatment of the LDETL parameter between the multiple-pin subchannel input preprocessor and the SAS4A/SASSYS-1 code. The input preprocessor allows negative values for LDETL, which correspond to coolant subchannel cells that completely occupy the assembly dimensions. However, the data needed for these two options is identical to the data needed for positive values of LDETL that have the same magnitude. Therefore, SAS4A/SASSYS-1 only defines positive values for LDETL. The second difference is that for LDETL = 4, SAS4A/SASSYS-1 also writes out the average fuel temperatures (corresponding to LDETL = 3) in addition to the detailed radial fuel temperatures. This is not done for the visualization mesh because it would define multiple cells for the same physical region. By writing this additional data, a visualization data file created as part of a simulation with LDETL = 4 can be used with a visualization mesh geometry file created with LDETL = 3 without having to re-run the simulation. In fact, once a visualization data file is created, it can be combined with a visualization mesh geometry file with a lower-magnitude LDETL parameter.

3.3-4 Data Visualization

The preceding sections describe the two-step process needed to create a visualization mesh geometry file and a visualization data file respectively. Interactively viewing the results of a simulation contained in these files is accomplished using the VisIt visualization software package freely available from Lawrence Livermore National Laboratory. VisIt is available for multiple computing platforms, including Linux, Windows, Mac OS X, Solaris, and others. In collaboration with LLNL, a Windows-based file reader plug-in for VisIt has been written that can parse the two files described here and import data into VisIt.

Each of the two files requires a unique file extension (*.sasgeom and *.sasdata) that VisIt uses to identify the correct file-reader plug-in to use when reading the data files. While the geometry file can be viewed independently of the data file, the data file requires a matching geometry file so that data can be projected into three-dimensional space. The matching of the geometry and data files is done by comparing a file's basename. Therefore, SAS.sasdata will be associated with SAS.sasgeom (and vice versa). This process is handled automatically by the file reader.

VisIt supports extensive data visualization capabilities, including pseudo color plots, slices, contours, and animations as well as data picks (queries), filters, time derivatives, and the construction of arbitrary expressions based on existing data. The use of the VisIt visualization tool is beyond the scope of this memo. Details on downloading and using VisIt are currently found at http://www.llnl.gov/visit/.

To illustrate some of the visualization capabilities, two (static) examples from the SHRT-17 simulation are shown in Figures A3.3–3 and A3.3–4. Figure A3.3–3 shows a plot of the XX09 assembly and its neighbors at t = 0 (steady state). The plot has been cropped through the center of XX09 to show the interior detail of fuel-pin temperatures. Figure A3.3–4 shows a contour plot of a slice at the top of the active core region at t = 90 seconds into the transient. Animated versions of both of these plots are presently archived at http://internal.ne.anl.gov/~fanning/SHRT17.html.

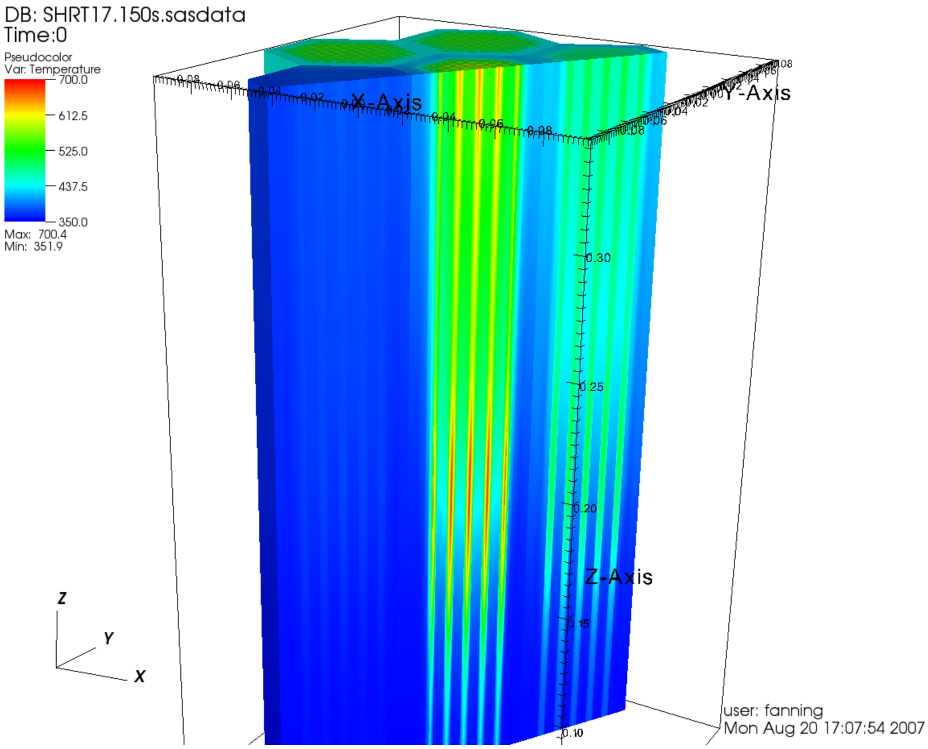


Figure A3.3–3: Pseudo Color Plot of Steady-State Coolant, Structure, and Fuel Temperatures (°C) for XX09 and Surrounding Assemblies.

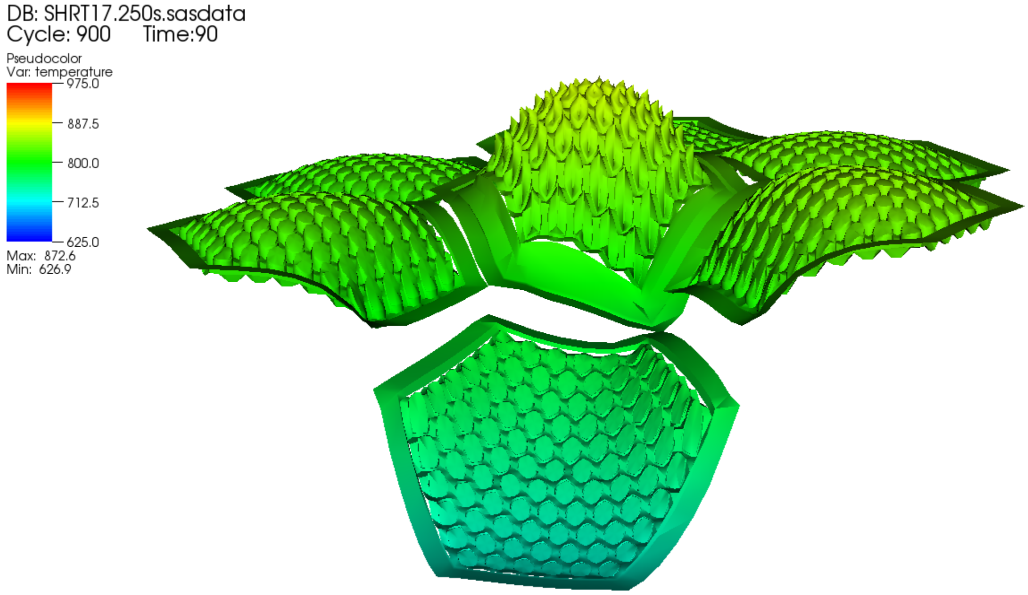


Figure A3.3–4: Contour Plot of Coolant, Structure, and Fuel Temperatures (°K) for XX09 and Adjacent Assemblies. (Top of Core at *t* = 90 seconds)

3.3-5 Summary

An advanced visualization capability has been developed for the SAS4A/SASSYS-1 code that allows for the generation, display, and animation of large datasets that are calculated during a transient simulation. The key features of the new capability include its support for multiple-pin, subchannel modeling, integration with the subchannel input preprocessor, geometry visualization, fully-detailed temperature field support with unlimited dataset sizes, and interactive visualization and animation generation on a desktop PC using the visualization tool VisIt.

3.3-6 References

1. J. E. Cahalan, et al., "Advanced LMR Safety Analysis Capabilities in the SASSYS-1 and SAS4A Computer Codes," *Proc. Int'l Topical Meeting on Advanced Reactor Safety*, American Nuclear Society, Pittsburgh, PA, April 17–21, 1994.

2. *VisIt User's Manual*, UCRL-SM-220449, Lawrence Livermore National Laboratory, October 2005.

3. F. E. Dunn, et al., "Whole Core Subchannel Analysis Verification with the EBR-II SHRT-17 Test," *Proc. ICAPP 2006*, Reno NV, June 4–8, 2006.

3.3-7 File Formats

3.3-7.1 Visualization Mesh Geometry File

The visualization mesh geometry file is an unformatted (binary) Fortran file that contains a file header followed by two data sections. The first data section defines assembly types in a local coordinate system. The second data section defines actual assemblies in the problem geometry in terms of previously-defined types and offsets into a global coordinate system. This arrangement very much mimics the definition of assembly types and positions in the multiple-pin subchannel preprocessor. All numeric data is represented by four byte integers and double-precision (eight-byte) floating point values. Byte order does not matter, since the file reader for VisIt was designed to detect and support both conventions. However byte order must be consistent within a single file.

3.3-7.1.1 File Header

The file header consists of two records that identify file attributes and a problem title. The first record defines the following file attributes:

FILEID File ID. Must be the four characters "SAS ". (The fourth character is a space.)

FILETYPE File Type. Must be the four characters "GEOM".

GEOMVER1 Integer File Version. Currently, GEOMVER1 = 1.

GEOMVER2 Integer File Sub-version. Currently, GEOMVER2 = 0.

CHRDATE File Creation Date. (Eight-byte character string.)

CHRTIME File Creation Time. (Eight-byte character string.)

The first four attributes are used by the VisIt file reader to ensure that the correct reader is being used to parse the data file. If major or minor changes to the file format are introduced, the GEOMVER1 and GEOMVER2 attributes will be changed to identify the correct file format for the reader to parse.

The second record in the file header is an 80-character problem title. The multiple-pin subchannel preprocessor uses the problem title provided as input for this record.

3.3-7.1.2 Assembly Types

The first data section following the file header defines assembly types in a local coordinate system. Since most assemblies (for example, driver assemblies) will have the same geometry, this simplifies (and shortens) the overall visualization mesh geometry file.

The structure of the assembly type definitions is represented by the following pseudo code:

```
NTYPES
DO ITYPE = 1, NTYPES
  TYPE_TITLE(ITYPE)
  ITYPEID(ITYPE), NCHAN(ITYPE), NZ(ITYPE), (Z(K,ITYPE),K=1,NZ(ITYPE))
  DO J = 1, NCHAN(ITYPE)
    IRELCH(J,ITYPE), NP(J,ITYPE), (X(I,J,ITYPE),Y(I,J,ITYPE),I=1,NP(J,ITYPE))
  ENDDO
ENDDO
```

The assembly type definitions begin with a record containing a single integer parameter, NTYPES, defining the number of assembly types contained in this data section. Each of the assembly type definitions that follow contains a title record (TYPE_TITLE, 80 characters) and an attributes record containing the following parameters:

ITYPEID Unique identification number for this assembly type.

NCHAN Number of “data channels” defined for this assembly type.

NZ Number of axial segments boundaries. (Typically, $NZ = MZ + 1$.)

Z Z coordinates of the axial segment boundaries in the local coordinate system.

Following the assembly type attributes record is one record for each of the NCHAN data channels defined for an assembly type. The data channel record defines the attributes and geometric shape (in the x-y plane) of the data channel:

IRELCH Relative data channel identification number. (See ICHOFF below.)

NP Number of vertices in the polygon defining this data channel. (Presently, NP must be 3 or 4.)

X, Y (X,Y) coordinates of the vertices defining the polygon for this data channel in local coordinates.

3.3-7.1.3 Assembly Definitions

The second data section defines "actual" assemblies in the problem geometry by placing assembly types defined earlier into a global coordinate system. The structure of the assembly definitions is represented by the following pseudo code:

```
NSUBS
DO ISUB = 1, NSUBS
  SUB_TITLE(ISUB)
  ISUBID(ISUB), ISUBTYPEID(ISUB), ICHOFF(ISUB), XOFF(ISUB),YOFF(ISUB),ZOFF(ISUB)
ENDDO
```

The subassembly definitions begin with a record containing a single integer parameter, NSUBS, defining the number of assemblies being specified in the visualization mesh geometry file. This need not represent all the assemblies in the problem geometry, only those for which a visualization mesh is needed. Each of the assembly definitions that follow contains a title record (SUB_TITLE, 80 characters) and an assembly attributes record containing the following parameters:

ISUBID Identification number for this subassembly.

ISUBTYPEID Identification number of the assembly type being used to define this assembly.

ICHOFF Integer offset to translate each relative data channel identification number into an absolute (global) data channel identification number.

XOFF, YOFF, ZOFF Coordinate offsets to translate local coordinates of the type definition into global coordinates of the problem geometry.

Absolute data channel identification numbers are constructed from the sum of the relative data channel identification number (IRELCH) and the data channel offset (ICHOFF) for each of the data channels that define an assembly type. For data channels that represent coolant, IRELCH typically starts at 1 and increases by 1 for each coolant subchannel in the assembly type definition. When added to ICHOFF, the resulting data channel identification corresponds to the channel number (ICH) in the SAS4A/SASSYS-1 simulation. Note that because multiple data channels may be used to represent the complex shape of a particular region (the coolant region of a subchannel, for example) absolute data channel identification numbers need not be unique in the visualization mesh geometry file.

For data channels that represent regions other than coolant, an additional term is introduced into the definition of IRELCH. If ICHTYPE represents the channel types as defined by the bulleted list on Page 3-184, then the additional term added to IRELCH is equal to ICHTYPE×10^6. This procedure is also used by SAS4A/SASSYS-1 to create data channel identification numbers based on the channel number, ICH, and guarantees correct association between the visualization mesh and the simulation data. That is, simulation data is correctly associated with the visualization mesh when (IRELCH + ICHOFF) = (ICHTYPE×10^6 + ICH).

3.3-7.2 Visualization Data File

The visualization data file is an unformatted (binary) Fortran file that contains a file header followed by one data section for each time step saved during a simulation. All numeric data is represented by four byte integers and double-precision (eight-byte) floating point values. Byte order does not matter, since the file reader for VisIt was designed to detect and support both conventions. However byte order must be consistent within a single file.

3.3-7.2.1 File Header

The simulation data file header consists of a record that identifies file attributes and two problem title records. The first record defines the following file attributes:

FILEID File ID. Must be the four characters “SAS ”. (The fourth character is a space.)

FILETYPE File Type. Must be the four characters “DATA”.

DATAVER1 Integer File Version. Currently, DATAVER1 = 1.

DATAVER2 Integer File Sub-version. Currently, DATAVER2 = 0.

CHRDATE File Creation Date. (Eight-byte character string.)

CHRTIME File Creation Time. (Eight-byte character string.)

The first four attributes are used by the VisIt file reader to ensure that the correct reader is being used to parse the data file. If major or minor changes to the file format are introduced, the DATAVER1 and DATAVER2 attributes will be changed to identify the correct file format for the reader to parse.

The second and third records in the file header represent two 80-character problem titles. The problem titles are determined by the two title records provided as part of the SAS4A/SASSYS-1 input.

3.3-7.2.2 Simulation Data

The remainder of the visualization data file consists of repeated blocks of simulation data. Data blocks contain simulation data for each time step that is recorded as determined by the value of the IVIS3D input parameter (Block 1, Location 120). The details of the simulation data that is recorded are determined by the channel-dependent input parameter, LDETL (Block 51, Location 500). The arrangement of data in the simulation data block is described by the following pseudo code:

```
TIME, NDATA
```

```
DO J = 1, NDATA
  ICHID(J), ND(J), (DATA(J,K),K=1,ND)
ENDDO
```

The first record in the data block contains the current simulation time (TIME) and the number of data channels that are to follow (NDATA). Although all data blocks from a simulation will have the same length and the same number of data channels, the NDATA parameter is provided to facilitate parsing and verification by file readers.

Following the initial record is a series of NDATA records containing simulation data represented by the following parameters:

ICHID Unique ID for this data channel.

ND Number of axial data values for this data channel. (Typically, ND = MZ.)

DATA Simulation data, with one value for each axial position.

Data channel IDs must be unique within each data block and are associated with zero or more data channels in the visualization mesh geometry in order to project the data into three-dimensional space. The construction of the data channel ID is described in the previous section as ICHID = (ICHTYPE$\times 10^6$ + ICH) and ensures a unique value based on SAS4A/SASSYS-1 channel numbers (ICH) and the type of data being stored.