

APPENDIX J

SPECIAL COMPONENT MODELS

Some 1D components in TRAC perform functions that cause the 1D fluid equations to be solved in a different way, either changing the equations' normal finite differencing, introducing source terms to the equations, or using complex logic to alter the closure relations to accomplish a specific modeling requirement. The PUMP component produces a momentum source term in the form of an additional Δp on the right-hand side of the momentum equation. The steam/water separator component (SEPD) permits the user to specify the separation efficiency in such locations as the separator at the top of a normal U-tube steam generator. The PLENUM component is a special, single-cell, 1D component that allows the user to connect as many normal 1D components as desired. To accomplish this objective, the PLENUM component interacts with the networking logic in the code and changes the form of the momentum flux terms. The FILL and BREAK components provide boundary conditions to the normal 1D components by allowing the user to specify either flows (FILL) or pressures (BREAK) at the boundary together with the fluid state if an inflow condition exists. The turbine (TURB) component models the removal of energy from the fluid by work performed by turbines. PWR accumulators are modeled by special logic that is built into TRAC's PIPE component, which is triggered by one of two PIPE input options. The PRIZER component models a PWR pressurizer. The VALVE component has a variety of options to adjust the flow area at a user-specified cell face.

Note: Separator (SEPD) and Turbine (TURB) Components. The base code for both TRAC-M/F77 and TRAC-M/F90 had components to model steam/water separators (SEPD) and turbines (TURB). The TRAC-P SEPD component permits the user to specify the separation efficiency in such locations as the separator at the top of a normal U-tube steam generator. The TRAC-P SEPD component received minimal support over the years, and its general use was not, and is not, recommended. (It should be used with caution.) Both TRAC-M/F77, Version 5.5.2, and TRAC-M/F90 (Version 3.0), have a SEPD component that was brought over directly from TRAC-P, and the same caution applies. The separator capability is to be improved in a future TRAC-M/F90 version. In the TURB component, energy is removed from the fluid as a result of work performed by the turbine. The TURB component received minimal support over its years in TRAC-P, and its general use was not, and is not, recommended. (It should be used with caution.) The TRAC-P TURB was brought over directly to TRAC-M. Currently, the TURB component is only available in TRAC-M/F77, and the same caution applies. The TRAC-P TURB has been removed entirely from TRAC-M/F90. An improved turbine-modeling capability is to be included in a future TRAC-M/F90 version. Section J.2. describes the current SEPD component. Section J.6. describes the current TURB component.

NOMENCLATURE

A :	area
A_{norm} :	normalized flow area of valve
AVLVE:	fully open valve flow area
b :	ratio of valve flow area to pipe flow area
C_0, C_1, C_2, C_3 :	proportionality constants in Eqs. (I-15) and (I-16)
C_b :	bucket velocity coefficient
C_n :	nozzle velocity coefficient
C_r :	fraction of total reaction energy delivered
d_j :	momentum source cell-edge diameter
D_H :	hydraulic diameter
DPMAX:	pressure offset at which the heaters or sprays achieve their maximum effectiveness
FA:	flow area
FAVLVE:	valve flow-area fraction
FRIC:	additive-loss coefficient
g :	gravitational acceleration constant
h :	normalized pump head (Section I.1.) or specific enthalpy (Section I.6.) or characteristic dimension for valve component (Fig. I-17.)
h_1 :	nondimensional head from the single-phase homologous head curves
h_2 :	nondimensional head from the fully degraded homologous head curves
H :	pump head or parameter defined by Eq. (I-44)
H_1 :	single-phase pump head
H_2 :	fully degraded pump head
I :	moment of inertia
K :	form-loss coefficient
$K0$:	fully open valve form-loss coefficient
$K_1 - K_3$:	coefficients of frictional torque
K_{eff} :	effective form-loss coefficient
LMT:	total liquid mass in pressurizer
m :	total number of stages belonging to a turbine
\dot{m} :	mass-flow rate
\dot{m}_{total} :	total mass-flow rate
\dot{m}_{vapor} :	vapor mass-flow rate
M :	momentum

$M(\alpha)$:	head degradation multiplier (<u>Section I.1.</u>) or momentum (<u>Section I.6.</u>)
n :	number of moving blade rows
N :	number of cells
$N(\alpha)$:	torque degradation multiplier
p :	pressure
p_1 :	pressure at the top of the pressurizer in <u>Eq. (I-37)</u> or pressure in cell 1 in <u>Eq. (I-2)</u>
p_2 :	pressure in cell 2
p_{set} :	pressure from which the system is controlled
P :	stage power
P_{enr} :	power extracted from energy equation
P_{demand} :	generator power demand
P_f :	liquid mass weighted fraction of power to be added or subtracted from liquid in each pressurizer cell
$P_{\text{gen,input}}$:	power input to generator
$P_{\text{gen,output}}$:	power output by the generator
P_{HEAT} :	maximum effective power that can be added to or subtracted from the liquid
P_{ideal} :	ideal stage power
P_{in} :	total power to be added or subtracted from the liquid
P_{total} :	total turbine power
q :	normalized volumetric flow
Q :	volumetric flow
r :	fraction of stage energy transferred
R :	valve radius
RFMX:	maximum rate of change of the FILL-component parameter
T :	torque
T_1 :	single-phase torque
T_2 :	fully degraded torque
T_{demand} :	torque due to external load
T_f :	torque caused by friction and by the bearing and windage
T_{hy} :	hydraulic torque
T_{loss} :	loss in torque
T_{turb} :	turbine torque
TWTOLD:	time-step weighting factor for the FILL-component parameter
vol:	cell volume

V :	velocity
V_a :	axial velocity component
$V_B, V_{B'}$:	turbine exit velocity
W :	blade velocity
x :	independent variable
x_i :	thermodynamic steam quality
XCO:	carryover parameter
XCU:	carryunder parameter
XPOS:	valve-stem position
y :	dependent variable
z :	collapsed liquid level
α :	void fraction
β :	dimensionless hydraulic torque
β_1 :	dimensionless hydraulic torque from the single-phase homologous torque curves
β_2 :	dimensionless hydraulic torque from the fully-degraded homologous torque curves
Δp :	pressure drop or pressure rise through the pump
Δt :	time-step size
Δx or ΔX :	cell length
ϕ_m :	moisture loss
ϕ_{rem} :	miscellaneous losses
η_{corr} :	correction efficiency
η_{gen} :	generator efficiency
η_{nb} :	nozzle bucket efficiency
η_{stg} :	stage efficiency
θ :	inclination angle from vertical or characteristic angle for valve component defined by <u>Eq. (J-45)</u>
ρ :	density
ω :	pump speed [<u>Eq. (J-1)</u>] or turbine-generator angular speed [<u>Eq. (J-28)</u>] or normalized pump speed [<u>Eq. (J-6)</u>]
ω_s :	specific pump speed
Ω :	pump-impeller angular velocity

Subscripts

g :	gas
l :	liquid

liq: liquid
m: two-phase mixture
R: rated

Superscripts

n : current-time value
 $n + 1$: new-time value

J.1. PUMP Component

The PUMP component describes the interaction of the system fluid with a centrifugal pump. The model calculates the pressure differential across the pump and its angular velocity as a function of the fluid-flow rate and the fluid properties. The model can treat any centrifugal pump and allows for the inclusion of two-phase effects.

The pump model is represented by a 1D component with N cells ($N > 1$). The pump momentum is modeled as a source at the interface between cells 1 and 2. This interface will be referred to as the pump interface for the remainder of this discussion. The source is positive for normal operation so that a pressure rise occurs from cell 1 to cell 2. Therefore, it is necessary to construct the cell noding such that the cell number increases in the normal flow direction. This latter requirement is equivalent to requiring the first cell of the component to be in the pump suction and the second cell in the pump discharge for normal operation.

The following considerations were important in creating the PUMP component:

1. compatibility with adjacent components should be maximized,
2. choking at the pump inlet or outlet should be predicted automatically, and
3. the calculated pressure rise across the pump should agree with that measured at steady-state conditions.

The first two criteria precluded the use of a lumped-parameter model. The PUMP component, therefore, combines the PIPE component with pump correlations.

The pump model in TRAC is very similar to that included in RELAP4/MOD5 (Ref. J-1.), although some additional assumptions were made to incorporate the momentum source into the momentum equations. Also, the details of the input for the homologous curves are somewhat different in that the eight curve segments defined in RELAP4/MOD5 are combined into only four segments in TRAC. Reference J-2. (Chapter 9) provides a good discussion of pump operation; Section J.2. of the same reference describes the single-phase homologous curve of a pump. Reference J-3. is a general text on pumps and provides much information about pumps and their operating characteristics. Runstadler (Ref. J-4.) provides an overview of the state of the art in pump modeling in the

mid-1970s, about the time that work began on TRAC and when there was a need for a pump model in the code. Several organizations worked on pump models during the late 1970s and the work continued through the 1980s. Furuya (Ref. J-5.) developed an analytical pump model that yields the two-phase performance characteristics based on single-phase characteristics and the details of the pump geometry.

The similarity factor for pumps that is most often discussed is the specific speed, ω_s , defined in the following equation (Ref. J-3., Eq. 5.9):

$$\omega_s = \frac{\omega Q^{\frac{1}{2}}}{(gH)^{\frac{3}{4}}} \quad , \quad (J-1)$$

where ω is the pump speed, Q is the volumetric flow, g is the acceleration of gravity, and H is the pump head. This specific speed ω_s is dimensionless only if the units of the other parameters are consistent: ω in radians per second ($\text{rad} \cdot \text{s}^{-1}$) or revolutions per second (rps), Q in $\text{m}^3 \cdot \text{s}^{-1}$ or $\text{ft}^3 \cdot \text{s}^{-1}$, H in meters (m) or feet (ft), and g in $\text{m} \cdot \text{s}^{-2}$ or $\text{ft} \cdot \text{s}^{-2}$. Stepanoff (Ref. J-3., p. 27) points out that ω_s is constant for all similar pumps and ideally does not change with speed for a given pump. However, when it is used as a similarity parameter, ω_s should be calculated at the highest efficiency point of operation. Stepanoff also casts ω_s in another dimensionless form (Ref. J-3., Eq. 5.35), which shows the importance of maintaining certain geometric ratios in similar pumps. This discussion is intended to provide a guide to help the code user determine if a set of homologous curves can be used to describe this pump.

J.1.1. Pump Governing Equations

The pump model is identical to the 1D pipe model except that the momentum equations between cells 1 and 2 are rewritten as

$$\frac{V_g^{n+1} - V_g^n}{\Delta t} = \frac{\left[p_1^{n+1} - p_2^{n+1} + \Delta p^n + \left(\frac{\partial \Delta p}{\partial V} \right)^n (V_g^{n+1} - V_g^n) \right]}{(\bar{\rho}_m)_{3/2}^n \Delta x_{3/2}} - g \cos \theta \quad (J-2)$$

and

$$V_\ell = V_g \quad , \quad (J-3)$$

where Δp is the pressure rise through the pump evaluated from the pump correlation. Equation (J-2) is implemented in subroutines StbVel1D (FEMOM in TRAC-M/F77) and TFIDS1. The steady-state solution of Eq. (J-2) is

$$\Delta p = p_2 - p_1 + g \cos \theta \quad ,$$

which is the desired result. Friction does not enter explicitly into the pump motion equation because we assume that the friction effects are normally included in the homologous curves defining the pump head. Therefore, additive friction is not allowed between cells 1 and 2 [FRIC(2) = 0.0].

It is necessary to evaluate Δp and its derivative with respect to velocity for a pump cell only once each time step. The source is needed only in routines StbVel1D (FEMOM in TRAC-M/F77) and TF1DS1. This evaluation is performed by subroutine PUMPSR.

J.1.2. Pump Head and Torque from Homologous Curves

The pump correlation curves describe the pump head and torque response as a function of fluid volumetric flow rate and pump speed. Homologous curves (one curve segment represents a family of curves) are used for this description because of their simplicity. These curves describe, in a compact manner, all operating states of the pump obtained by combining positive or negative impeller velocities with positive or negative flow rates.

The following definitions are used in the subsequent development:

- H = the pump head = $(\Delta p) / \bar{\rho}_m$,
- Q = the pump volumetric flow, and
- Ω = the pump impeller angular velocity,

where Δp is the pressure rise across the pump and $\bar{\rho}_m$ is the pump-average mixture density in the cells immediately adjacent to the pump interface. The code user should note that the definition of pump head H above differs from the standard definition by a factor g , the acceleration caused by gravity. To allow one set of curves to be used for a variety of pumps, the following normalized quantities are used

$$h \equiv \frac{H}{H_R} \quad , \quad \text{(J-4)}$$

$$q \equiv \frac{Q}{Q_R} \quad , \quad \text{(J-5)}$$

and

$$\omega \equiv \frac{\Omega}{\Omega_R} \quad , \quad \text{(J-6)}$$

where H_R is the rated head (RHEAD) for the pump, Q_R is the rated volumetric flow (RFLOW), and Ω_R is the rated pump speed (ROMEGA). The pump similarity relations show that

$$\frac{h}{\omega^2} = f\left(\frac{q}{\omega}\right) \quad (J-7)$$

For small ω this correlation is not satisfactory and the following combination of variables is used:

$$\frac{h}{q^2} = f\left(\frac{\omega}{q}\right) \quad (J-8)$$

Correlation Eq. (J-7) is used in the range $0 \leq |q/\omega| \leq 1$ and results in two separate curves, one for $\omega > 0$ and one for $\omega < 0$. Correlation Eq. (J-8) is used in the range $0 \leq |\omega/q| < 1$ and yields two separate curves, one for $q > 0$ and one for $q < 0$. The four resulting curve segments, as well as the curve-selection logic used in TRAC, are shown in Table J-1.

To account for two-phase effects on pump performance, the pump curves are divided into two separate regimes. Data indicate that two-phase pump performance in the vapor-fraction range of 20-80% is degraded significantly in comparison with its performance at vapor fractions outside of this range. One set of curves describes the pump performance for single-phase fluid (0 or 100% vapor fraction), and another set describes the two-phase, fully degraded performance at some void fraction between 0 and 100%. For single-phase conditions, the curve segments for correlation Eq. (J-7) are input as HSP1 for $\omega > 0$ and HSP4 for $\omega < 0$, and for correlation Eq. (J-8) curve segments are input as HSP2 for $q > 0$ and HSP3 for $q < 0$. The fully degraded version of correlation Eq. (J-7) is input as curve HTP1 for $\omega > 0$ and HTP4 for $\omega < 0$. The fully degraded version of correlation Eq. (J-8) is input as HTP2 for $q > 0$ and HTP3 for $q < 0$.

TABLE J-1.
Definitions of the Four Curve Segments that Describe the Homologous Pump Head Curves^a

Curve Segment	$\frac{ q }{ \omega }$	ω	q	Correlation
1 4	$\left[\begin{array}{l} \leq 1 \\ \leq 1 \end{array} \right]$	$\left[\begin{array}{l} > 0 \\ < 0 \end{array} \right]$	$\left[\begin{array}{l} \\ \end{array} \right]$	$\left[\frac{h}{\omega^2} = f\left(\frac{q}{\omega}\right) \right]$
2 3	$\left[\begin{array}{l} > 1 \\ > 1 \end{array} \right]$		$\left[\begin{array}{l} > 0 \\ < 0 \end{array} \right]$	$\left[\frac{h}{q^2} = f\left(\frac{\omega}{q}\right) \right]$

a. For the special case of both $\omega = 0.0$ and $q = 0.0$, TRAC sets $h = 0.0$

The pump head at any vapor fraction is calculated from the relationship

$$H = H_1 - M(\alpha)(H_1 - H_2), \quad (J-9)$$

where

- H = the total pump head;
- H_1 = the single-phase pump head = $h_1 H_R$ where h_1 is the nondimensional head from the single-phase homologous head curves;
- H_2 = the fully degraded pump head = $h_2 H_R$, where h_2 is the nondimensional head from the fully degraded homologous head curves;
- $M(\alpha)$ = the head degradation multiplier (input as HDM); and
- α = the donor-cell void fraction.

To this point, no knowledge of density is required to calculate H from the homologous head curves. The average mixture density in the cells immediately adjacent to the pump interface is always used to convert the total pump head H to Δp , the pressure rise through the pump, by the definition $\Delta p = \rho_m H$.

The development of homologous torque curves parallels the previous development for homologous head curves. The dimensionless hydraulic torque is defined by

$$\beta \equiv \frac{T_{hy}}{T_R}, \quad (J-10)$$

where T_{hy} is the hydraulic torque and T_R is the rated torque (RTORK). The convention used is that a positive T_{hy} works to retard positive pump angular velocity. The dimensionless torque β is correlated as either β / ω or β / q , just as the dimensionless head was correlated. For single-phase conditions, the correlations yield the corresponding four curve segments TSP1, TSP2, TSP3, and TSP4. The fully degraded correlations produce four corresponding curves: TTPI, TTP2, TTP3, and TTP4. The homologous torque curve segments are correlated in the same manner as the head curve segments shown in Table J-1. (replace h with β). For the special case of $\omega = q = 0.0$, the code sets $\beta_1 = \beta_2 = 0.0$.

The single-phase torque T_1 is dependent upon the fluid density and is calculated from

$$T_1 = \beta_1 T_R \left(\frac{\rho_m}{\rho_R} \right) \quad (J-11)$$

where β_1 is the dimensionless hydraulic torque from the single-phase homologous torque curves, ρ_m is the pump-average mixture density in the cells immediately adjacent to the pump interface, and ρ_R is the rated density (RRHO). The density ratio is needed to correct for the density difference between the pumped fluid and the rated condition. Similarly, the fully degraded torque T_2 is obtained from

$$T_2 = \beta_2 T_R \left(\frac{\rho_m}{\rho_R} \right), \quad (J-12)$$

where β_2 is the dimensionless hydraulic torque from the fully degraded homologous torque curves. For two-phase conditions, the impeller torque is calculated from

$$T = T_1 - N(\alpha)(T_1 - T_2) \quad , \quad (J-13)$$

where T is the total impeller torque and $N(\alpha)$ is the torque degradation multiplier (input as TDM).

In addition to the homologous head and torque curves, the head and torque degradation multipliers defined in Eqs. (J-9) and (J-13) are required. These functions of void fraction are nonzero only in the vapor-fraction range where the pump head and torque are either partially or fully degraded.

J.1.3. Pump Speed

The pump component treats the pump angular velocity as a constant (input) while the motor is energized. After a drive-motor trip, the time rate of change for the pump angular velocity Ω is proportional to the sum of the moments acting on it and is calculated from

$$I \frac{d\Omega}{dt} = -\sum_i T_i = -(T + T_f) \quad , \quad (J-14)$$

where I is the combined impeller, shaft, and motor-assembly moment of inertia (EFFMI); T is the hydraulic torque on the impeller; and T_f is the torque caused by friction and by the bearing and windage. We assume that

$$T_f = C_0 + C_1 \frac{\Omega}{\Omega_R} + C_2 \frac{\Omega|\Omega|}{\Omega_R^2} + C_3 \frac{\Omega^3}{\Omega_R^3} \quad , \quad (J-15)$$

where $C_0, C_1, C_2,$ and C_3 are input constants (TFR0, TFR1, TFR2, and TFR3, respectively). The user may specify a different set of coefficients (TFRL0, TFRL1, TFRL2, and TFRL3) for low pump speed. These coefficients will be applied at speeds below the user-specified value TFRB. As rotational speed approaches zero, the coefficient C_0 linearly approaches zero so that the frictional torque will be vanished at zero speed.

The hydraulic torque T is evaluated using the homologous torque curves and Eq. (J-13). It is a function of the volumetric flow, the upstream void fraction, the average density in the cells immediately adjacent to the pump interface, and the pump angular velocity. For time step $(n + 1)$, Eq. (J-14) is evaluated explicitly,

$$\Omega^{n+1} = \Omega^n - \frac{\Delta t}{I} \left[T(Q, \alpha, \rho_m, \Omega) + C_0 + C_1 \frac{\Omega^n}{\Omega_R} + C_2 \frac{\Omega^n |\Omega^n|}{\Omega_R^2} + C_3 \frac{(\Omega^n)^3}{\Omega_R^3} \right]. \quad (J-16)$$

So that all torque losses go to zero as Ω approaches zero, C_0 is ramped to zero at low pump speed.

J.1.4. Pump Input Options

To invoke the pump model in TRAC, the code user has to specify the following options: pump type (IPMPTY), motor action (IPMPTR), reverse speed option (IRP), degradation option (IPM), and pump curve option (OPTION). The input variables IPMPTR and NPMPTX specify, respectively, the trip number for the pump-trip initiation and the number of pairs of points in the pump-speed table (array SPTBL). If IPMPTR = 0, no pump-trip action occurs, and the pump runs for the entire calculation at the constant speed (OMEGAN) specified in the input.

If the pump motor is energized (trip IPMPTR set OFF), the angular velocity is assumed to be the constant value specified (OMEGAN). If the motor is not energized (trip IPMPTR set ON), the pump speed is allowed to vary.

Two types of pumps are available. For pump type 1 (IPMPTY = 1), the pump-speed variation is specified by the input table. The pump is energized initially at a constant speed specified by input (OMEGAN). Trip IPMPTR may initiate a pump trip, after which the pump speed is taken from a pump-speed table (array SPTBL). The independent variable for the pump-speed table may be elapsed time since trip initiation or any other signal variable. For pump type 1 the torque calculation is not used. Pump type 2 (IPMPTY = 2) is similar to option 1 except that a speed table is not input. Instead, the pump speed is calculated from Eq. (J-16) after a trip has occurred.

If the reverse speed option is specified (IRP = 1), the pump is allowed to rotate in both the forward and reverse directions. If reverse speed is not allowed (IRP = 0), the pump will rotate in the forward direction only. For this case, if negative rotation is calculated (after trip with pump type 2), the speed will be set to zero. If IRP = 0 and a negative pump speed is encountered (i.e., from the speed table), error messages will be printed by subroutines PUMPD, PUMPX, and PUMPSR. (This is a fatal error.)

If the degradation option is turned on (IPM = 1), the degraded pump head and torque are calculated from Eqs. (J-9) and (J-13). If the degradation option is turned off (IPM = 0), only the single-phase head and torque homologous curves are used [equivalent to setting $M(\alpha)$ and $N(\alpha)$ to zero in Eqs. (J-9) and (J-13)].

J.1.5. Pump Homologous Curves

The user may specify pump homologous curves in the input (OPTION = 0) or alternatively may use the built-in pump curves (OPTION = 1 or 2). The first set (OPTION = 1) of built-in pump curves is based on the Semiscale Mod-1 system pump (Refs. J-1. and J-6. through J-8.). The Semiscale pump curves for single-phase homologous head (HSP), fully degraded two-phase homologous head (HTP), head degradation multiplier (HDM), single-phase homologous torque (TSP), and torque degradation multiplier (TDM) are provided in Figs. J-1. through J-5., respectively. The second set (OPTION = 2) of built-in pump curves is based on the Loss-of-Fluid Test (LOFT) system pump (Refs. J-9. through J-11.). The LOFT pump curves for HSP, HTP, HDM, TSP, and TDM are shown in Figs. J-6. through J-10., respectively. For lack of data, the fully degraded two-phase homologous torque curves (TTP) for both Semiscale and LOFT are zero. Where applicable, the curves are numbered corresponding to the conditions provided in Table J-1.

Because these homologous curves are dimensionless, they can describe a variety of pumps by specifying the desired rated density, head, torque, volumetric flow, and angular velocity as input. We recommend that for full-scale PWR analyses, plant-specific pump curves be input; however, if such data are unavailable, the LOFT pump curves (OPTION = 2) generally should be used.

Several restrictions and limitations are present in the current version of the PUMP component. Because there is no pump motor torque versus speed model, the pump speed is assumed to be input if the motor is energized. Pump noding is restricted such that the cell numbers increase in the normal flow direction (NCELLS \geq 2), the pump momentum source is located between cells 1 and 2 of the pump model, and the additive friction (loss coefficient) between cells 1 and 2 is 0.0 [FRIC (2) = 0.0]. A flow-area change should not be modeled between cells 1 and 2. Finally, the head degradation multiplier $M(\alpha)$ and the torque degradation multiplier $N(\alpha)$ are assumed to apply to all operating states of the pump.

The PUMP component input consists of the same geometric and hydrodynamic data and initial conditions that are required for the PIPE component. In addition, information specific to the PUMP is required. The speed table (SPTBL), as well as the homologous pump curve arrays, must be input in the following order:

$$x(1), y(1), x(2), y(2), \dots, x(n), y(n) \ .$$

Here x is the independent variable and y is the dependent variable. Furthermore, the independent variables must increase monotonically on input, that is,

$$x(1) < x(2) < \dots < x(n-1), x(n) \ .$$

Linear interpolation is used within the arrays.

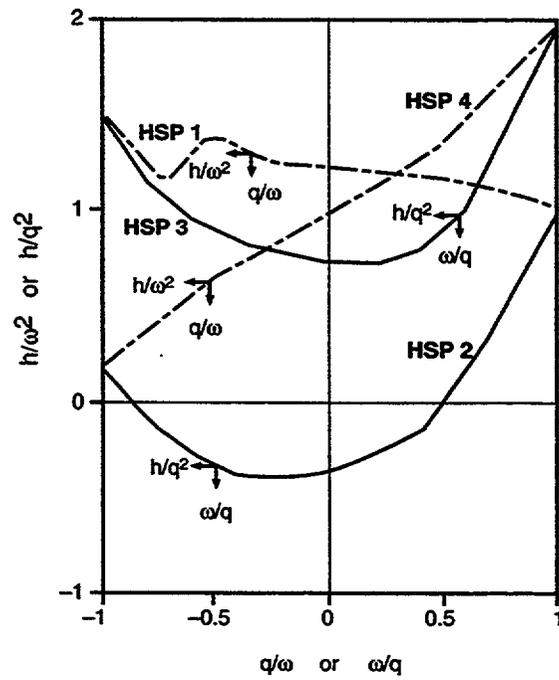


Fig. J-1. Semiscale single-phase homologous head curves.

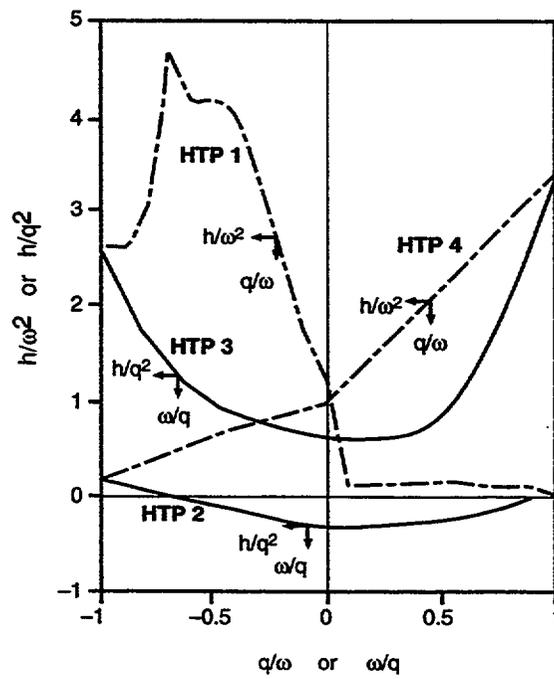


Fig. J-2. Semiscale fully degraded homologous head curves.

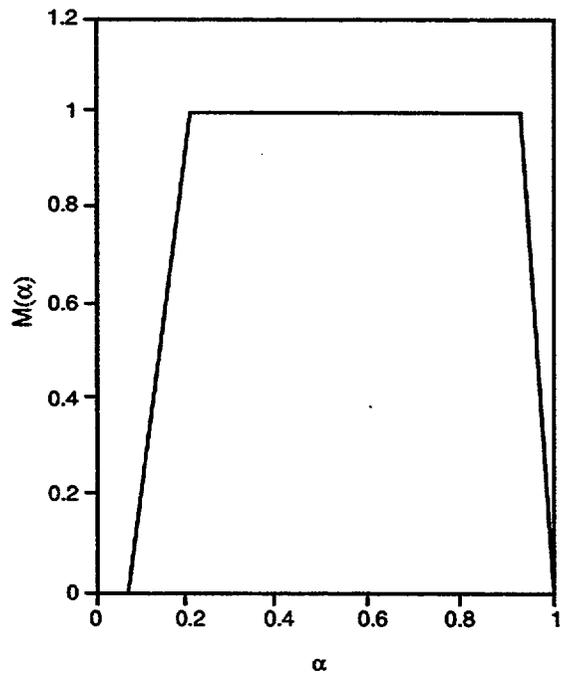


Fig. J-3. Semiscale head degradation multiplier curve.

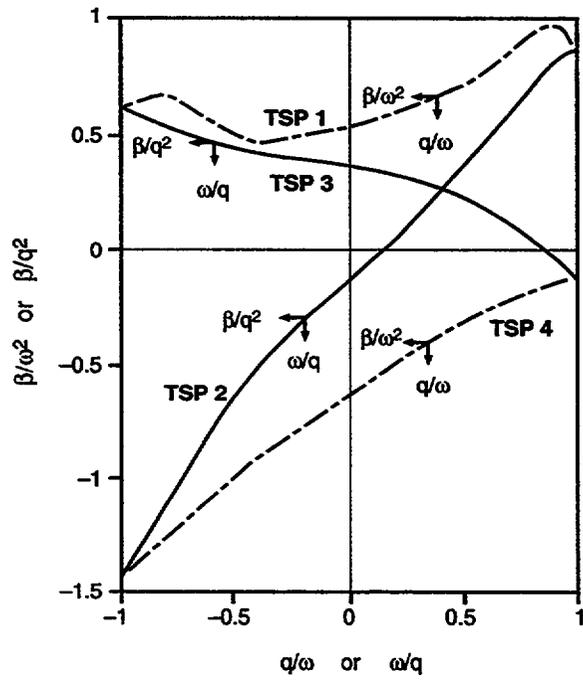


Fig. J-4. Semiscale single-phase homologous torque curves.

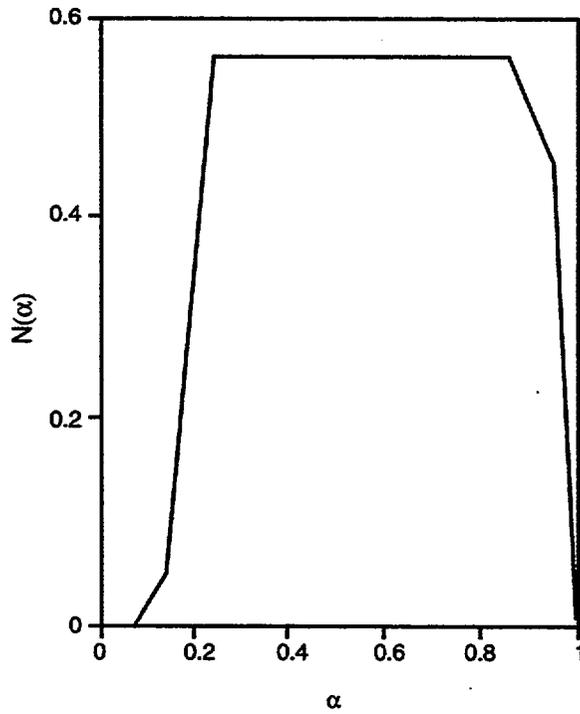


Fig. J-5. Semiscale torque degradation multiplier curve.

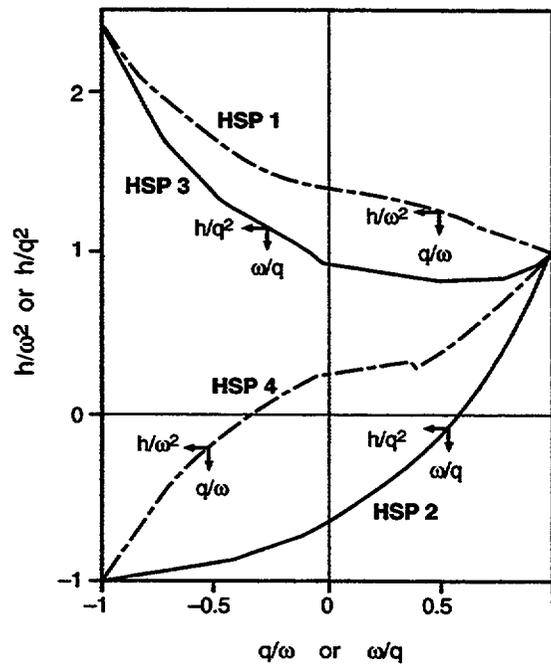


Fig. J-6. LOFT single-phase homologous head curves.

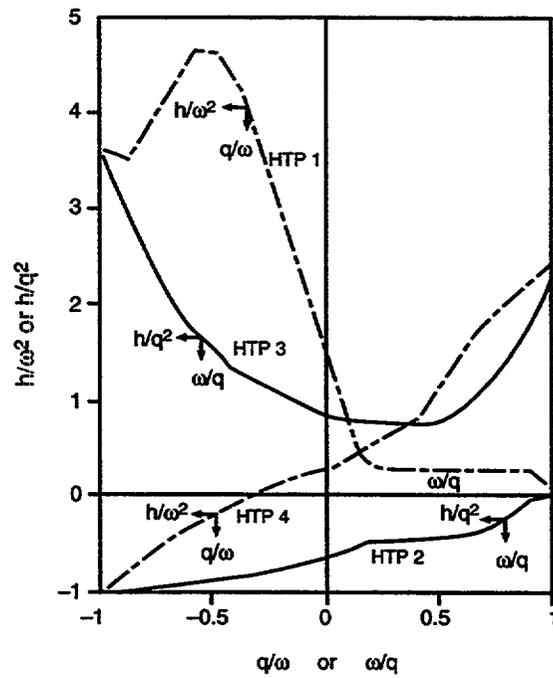


Fig. J-7. LOFT single-phase homologous head curves.

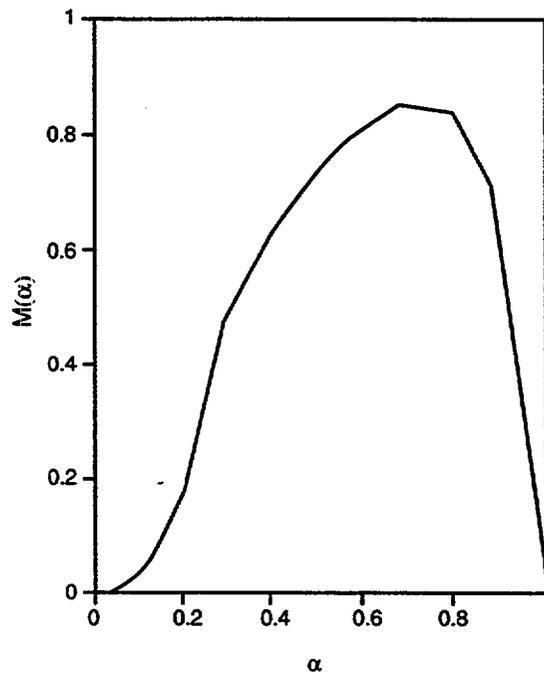


Fig. J-8. LOFT head degradation multiplier curve.

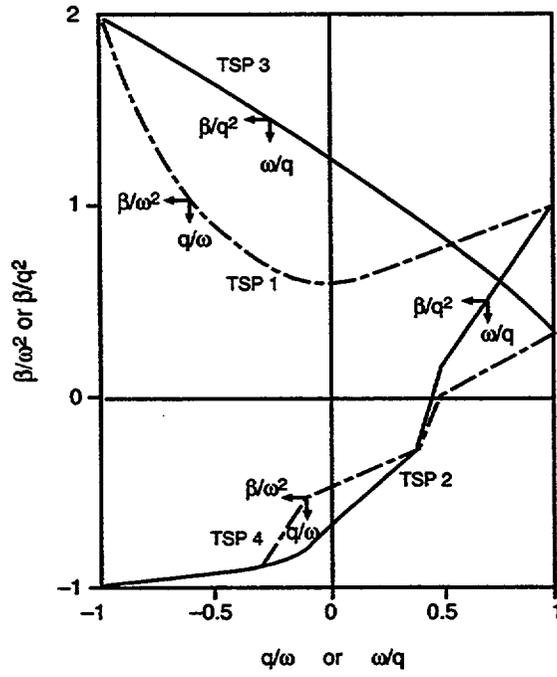


Fig. J-9. LOFT single-phase homologous torque curves.

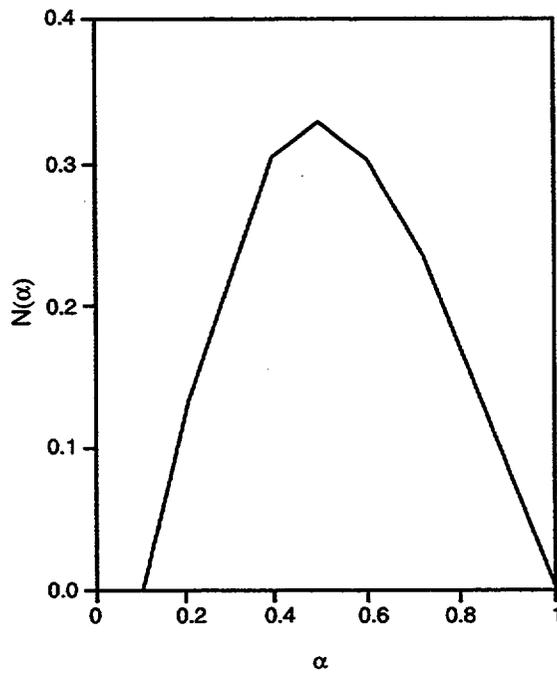


Fig. J-10. LOFT torque degradation multiplier curve.

J.1.6. Pump Conclusions

The pump model included in TRAC has demonstrated a remarkable capability to model reactor pumps under many conditions. Obviously, the quality of the pump simulation is dependent on the quality of the homologous curves used to describe the pump, and we recommend that whenever the data are available, the user input specific curves for the pump under consideration instead of using the built-in curves. An alternative is to use the Tetra Tech model (Ref. J-5.) to generate the fully degraded homologous curves and associated two-phase multiplier curves from the pump geometry and the published single-phase performance curves from the manufacturer. Unless the pump to be modeled is similar to the Semiscale or LOFT pumps, the least desirable option is to select one set of built-in curves.

From a code viewpoint, there is one deficiency regarding the pump model. The assumption of equal phase velocities at the pump interface, although a reasonably good assumption when the pump is operating at a significant rotational speed, breaks down as the pump speed approaches zero. At this point, the homogeneous flow assumption prevents unequal phase velocities and, in particular, prevents countercurrent flow at that one interface. The lack of phase slip can affect the separation of liquid and vapor in the pump suction, and discharge and could result in oscillatory flow to achieve the net effect of countercurrent flow. At the current time, the only solution available to the code user is to replace the PUMP component with an equivalent PIPE component when the pump speed is near zero and the fluid velocities are low.

J.2. Steam/Water Separator Component (SEPD)

Note: Separator (SEPD) Component. The base code for both TRAC-M/F77 and TRAC-M/F90 has a component to model steam/water separators (the SEPD component). The TRAC-P SEPD component permits the user to specify the separation efficiency in such locations as the separator at the top of a normal U-tube steam generator. The TRAC-P SEPD component received minimal support over the years, and its general use was not, and is not, recommended. (It should be used with caution.) Both TRAC-M/F77, Version 5.5.2, and TRAC-M/F90 (Version 3.0), have a SEPD component that was brought over directly from TRAC-P, and the same caution applies. The separator capability is to be improved in a future TRAC-M/F90 version. Section J.2. describes the current (obsolete) SEPD component.

In anticipation of specific needs and in acknowledgment of requests from TRAC users, we determined that a component should be added to the code that would separate an incoming two-phase stream into vapor-enriched and liquid-enriched streams. The USNRC requested that we incorporate the separator component used in the TRAC-BWR computer code into TRAC-PF1/MOD1 and TRAC-PF1/MOD2. After examining the method used in the TRAC-BWR, we determined that this was a possible approach to adding a separator component to TRAC-PF1/MOD2, although not necessarily the one that we would otherwise have taken. Our implementation of the SEPD component essentially incorporates a superset of the TEE component, and in the following

discussion we will frequently make use of the terminology of the TEE component, such as "JCELL" and "side leg."

Several modules of the SEPD component are in essence unchanged from the TRAC-BWR code and, as we did not develop them, we cannot necessarily justify their use. We will therefore limit this discussion to those points at which we know that the version of the SEPD component in TRAC-PF1/MOD2 differs from the version in TRAC-BWR.

J.2.1. Basis for the Model

The coding for the SEPD component is broken down into three phases: (i) determination of the appropriate carryover and carryunder for the current conditions; (ii) determination, through quasi-steady-state mass and momentum balances, of the resulting separator exit flows and qualities; and (iii) circumvention of the normal solution method to achieve those flows and qualities. The carryover is defined as the ratio of the liquid mass-flow rate to the total mass-flow rate, both evaluated at the exit of the separating JCELL that passes on through the main leg of the component (Fig. J-11.). The carryunder is defined as the ratio of the vapor mass-flow rate to the total mass-flow rate, both evaluated at the exit of the JCELL that passes into the side leg of the SEPD.

References J-12., J-13., and J-14. describe the TRAC-BWR versions of the SEPD component, and Ref. J-15. describes the General Electric (GE) study that forms the basis for the "mechanistic model" described below. Reference J-14., as amended by material in Ref. J-15., provides a general description of the model and of the input for the SEPD component of TRAC.

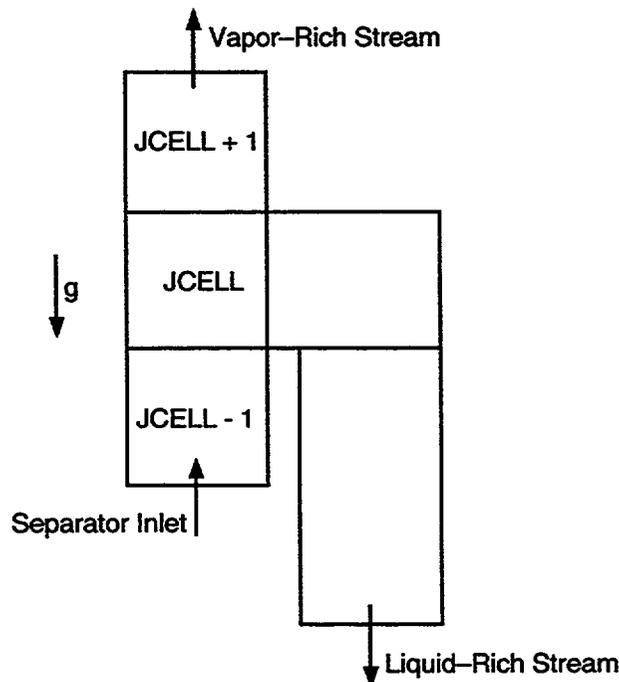


Fig. J-11. Typical separator component nodalization.

We did not perform comparisons with data for our implementation of this existing model. We relied on the correctness of the model development performed by INEEL and GE and did not perform an extensive review of that work. The correlations were not developed at LANL. Refer to Refs. J-10. through J-13.

J.2.2. Assumptions Made in Implementing the Correlation in the Code

Although a detailed discussion of this topic is outside the scope of this document (refer to Refs. J-10. through J-13.), some major points will be discussed so that TRAC users will know under what conditions the use of the SEPD component is recommended and justified.

A primary assumption of the SEPD component as it was implemented in TRAC-BWR and as it is implemented in TRAC-PF1/MOD2 is that the determination of the desired flows and qualities exiting the separating JCELL is based on a quasi-steady-state analysis. As a result, when the flow conditions entering the separator become sufficiently rapidly varying, one should expect the separation calculated by the SEPD component to deviate from reality. Indeed, numerical instabilities could result when conditions are sufficiently far from steady state.

If the "mechanistic" separator is used (the options available to the SEPD component are discussed in Section J.2.4.), it is assumed that the physical separator being modeled is very similar geometrically, if not identical, to a GE BWR steam/water separator. This is a result of the use of a module for this option that was developed at GE specifically for their hardware. The mechanistic option of the SEPD component is not recommended for non GE separators. Other SEPD options are available for arbitrary separator designs for which the user knows the performance data (Ref. J-14., as amended by material in Ref. J-15.).

For any option of the SEPD component, we assume that the separator is oriented vertically ($GRAV = +1$; Fig. J-11.). This is a relic of the TRAC-BWR ancestry of the SEPD component, and may be relaxed in future versions.

J.2.3. Constants

Aside from the GE-supplied default geometric data (which may be overridden) used for the mechanistic option of the SEPD component (Table J-2.), few constants are used in the model. Subroutine SEPDI uses two constants in its logic to determine whether the SEPD should be activated. These are discussed in Section J.2.4.2. in conjunction with an outline of how that logic operates.

SEPDI also uses a constant factor to determine how much the values of the FRICs can change from one time step to the next. This is discussed in Section J.2.5.

**TABLE J-2.
Default Values for the SEPD Component**

	IF ISTAGE=0				
		XCO	0.05		
		XCU	0.003		
	IF ISTAGE=2 or 3				
		AI	0.018637		
		AN	0.014411		
		RH	0.0809585		
		THETA	48.0		
		RR1	0.0857208		
	2-Stage SEPD		3-Stage SEPD		
	1st Stage	2nd Stage	1 Stage	2nd Stage	3rd Stage
RWS	0.10794	0.06985	0.10794	0.10794	0.10794
RRS	0.06985	0.06032	0.08572	0.095245	0.0984201
ADS	0.041578	0.0029133	0.009627	0.009627	0.009627
DDS	0.045558	0.0121699	0.025399	0.025399	0.025399
HBS	0.877845	0.16255	1.0699	0.384156	0.384156
HSK	0.2127	0.0	0.45083	0.0	0.0
CKS	10.0	0.5	2.5	1.429	2.563
EFFLD	450.0	95.85	53.44	1.9464	424.96
AA	110.0	20.0	110.0	20.0	20.0
BB	0.5	0.25	0.5	0.25	0.55

J.2.4. Model as Coded

Here we present a summary of the coding used to implement the SEPD component in TRAC. Much of the coding was provided to LANL by INEEL or GE and will not be discussed in detail.

J.2.4.1. Phase 1: Determination of the Appropriate Carryover and Carryunder.

There are currently three options for the SEPD that directly affect the determination of the carryover (XCO) and carryunder (XCU). These are the conditions TRAC attempts to create at the end of the time step if the SEPD component is functioning. There are several conditions, discussed in [Section J.2.4.2.](#) that will cause the SEPD component to cease to separate.

J.2.4.1.1. Ideal Separator. If the user selects the ideal separator, the values of XCO and XCU are considered constants independent of any current flow conditions in the separator. These constant values are read in at the input stage of the calculation and do not change.

J.2.4.1.2. Mechanistic Separator. By selecting this option, the user instructs TRAC to calculate the target values of XCO and XCU based on the current conditions in the separator. To do this, TRAC uses the subroutines SSEPOR and SEPDX, provided by GE ([Ref. J-15.](#)), that predict the performance of a GE separator given the current flow conditions. The TRAC versions of these routines are identical to those provided by GE and used in TRAC-BWR and will not be discussed here. We again point out that the use of this option to model non-GE separators is not recommended. For separators that differ only slightly from a GE separator, however, the user can input geometric parameters to instruct TRAC as to those differences. The default values of the geometric input are listed in [Table J-2.](#)

J.2.4.1.3. User-Prescribed Separator. In many cases, the user knows that the separator being modeled is not ideal, nor is it a standard GE design. In those cases the user-prescribed separator should be used. This option, however, requires that the user know the performance characteristics of the separator. Given a particular set of conditions in the separator, TRAC can then refer to the user-supplied performance curves and determine target values of XCO and XCU . It is up to the user to correctly characterize the separator performance, and this typically takes the form of

$$XCO = XCO(\dot{m}_{total}, \text{input quality}) \text{ and}$$

$$XCU = XCU(\dot{m}_{total}, \text{input quality}) .$$

J.2.4.2. Phase 2: Determination of the Separator Exit Flows and Qualities.

Subroutine SEPDI, written by INEEL, calculates target values of the velocities, the derivatives of the velocities with respect to the pressure, and void fractions exiting the SEPD JCELL that would be necessary to yield the desired values of XCO and XCU . As discussed in [Section J.2.2.](#), a quasi-steady-state assumption is made that may not be strictly true for the rapidly varying conditions of a transient simulation.

If the mechanistic option of the SEPD component is used, the subroutine SEPDI also calculates the FRICs (additive-loss coefficients) required to produce a pressure drop that the GE subroutine SSEPOR determines is appropriate for the current conditions. These FRICs will be used in the next phase of the calculations. If any other option is used, it is up to the user to input the correct FRICs to suit the geometry.

J.2.4.3. Phase 3: Circumvention of the Normal Solution Method to Achieve Those Flows and Qualities. Subroutine SEPDI, which is called from subroutine SEPD2 (TEE2 in TRAC-M/F77), calculates the target values for XCO , XCU , convected void fractions, etc., and stores them in the common block SEPCB. This subroutine also calculates values for the velocities and the derivatives of the velocities with respect to the pressure, which are stored in the A array, the main data-storage array. The numerical routines, TF1DS1 and TF1DS, use these values, rather than ones they would otherwise calculate, during the remaining numerical calculations. If the mechanistic option of the SEPD is being used, the FRICs calculated in SEPDI are also used in TF1DS1 and TF1DS.

J.2.5. Weighting, Magnitude Limits, and Averaging
Currently, the only limit placed on the operation of the SEPD is that the FRICs calculated with the mechanistic option may not increase or decrease by more than a factor of 2 from one time step to the next.

J.2.6. Variations in Application of Correlation—Special Cases
Subroutine SEPDI, called from SEPD2 (TEE2 in TRAC-M/F77), begins by deciding whether the SEPD should in fact perform any separation. This is done by examining such things as the direction, magnitude, and quality of the incoming flows. If the quality of the input stream is too low ($\alpha < 0.05$), the component does not attempt to cause a separation. Likewise, if either the vapor or total mass-flow rate of the incoming stream is negative, or if the composition of the incoming stream is essentially all vapor ($1.0 - \dot{m}_{\text{vapor}}/\dot{m}_{\text{total}} < 10^{-6}$) then the SEPD component is deactivated. It becomes reactivated only when these conditions are no longer true.

J.2.7. Assessment
No assessment was performed on the separator model at LANL. Extensive assessment was performed at INEEL and is described in Ref. J-13.

J.2.8. Effects of Applying Correlation Outside of Database
The database for the mechanistic model is made up of GE separators. While we do not know the precise effects of using the mechanistic model for non-GE separators, we caution against the use of this option in these cases. As the SEPD is used, more experience will be gained. It is possible that more definitive statements could be made in the future.

J.2.9. Scaling Considerations
Any discussion of scaling for the SEPD component is dependent on the option invoked for the component. These options were outlined in Sections J.2.4.1.1, through J.2.4.1.3.

The ideal separator is an extremely simple-minded option that would rarely be appropriate, except in situations where nearly constant conditions at the separator were expected to be maintained throughout the simulation. Even then, the numbers input for the values of XCO and XCU would presumably be based on some experience with the sort of separator being simulated. In such cases then, the effect of scaling is presumed to be taken into account by the user.

The mechanistic option of the SEPD component relies on the model developed by GE that predicts the carryover and carryunder of the separator as a function of the local conditions. This GE model was developed specifically for their hardware design and as a result is most appropriate for full-scale applications.

The user-prescribed separator option requires that the user input performance curves for the SEPD component. It is therefore assumed that the user has appropriate information for the particular separator being modeled, and all effects of scale have been considered by the user.

J.2.10. Summary and Conclusions

The SEPD component of TRAC is an implementation of the model from the TRAC-BWR code, where it is also called the SEPD. As the component was not developed at LANL, we cannot certify the correctness of the method used. The primary development was performed at INEEL and at GE.

The component provides a means for the separation of an incoming two-phase mixture of liquid water and its vapor into one stream that is liquid enriched and one stream that is vapor enriched. Several options are available for determining the performance of the SEPD to enable a close simulation of the physical separator being modeled. The most sophisticated option, known as the mechanistic option, was developed by GE to model the separators that they manufacture. For modeling GE separators, this is the appropriate option. Options also exist to permit the users to input their own performance curves for the separator or to model the separator as one whose performance is independent of current local conditions. In both of these later cases, it is up to the user to provide realistic input.

The models assume that a quasi-steady state exists in the separator, and therefore models will likely break down for sufficiently rapid transients in the SEPD component.

J.3. FILLS

The FILL component replaces the motion equations where it is joined to a 1D component with a boundary condition specifying velocities or mass flows. Since the motion equations are eliminated at that mesh-cell interface, interfacial drag and wall friction terms are not required and nonphysical pressure gradients can develop with no effect on flow because the FILL cell input-specified pressure is used only to evaluate cell properties. Generally a FILL's behavior is well defined by input conditions, but there are peculiarities that must be watched.

The most obvious problem can occur when a user specifies a mass-flow rate from the system into the FILL. In that case, the FILL component determines liquid and vapor velocities into the FILL at the beginning of the time step, based on the old-time macroscopic densities from the adjacent component mesh cell. The end-of-the-step mass-flow rate is determined by the product of these velocities with the new-time macroscopic densities. If the fluid density in the cell adjacent to the FILL changes significantly over the duration of a time step, the actual mass flowing out of the FILL will be different from the expected value.

The user should exercise caution when setting up FILLS with properties that depend on nearby system parameters (especially pressure) through signal-variable (or control block based on signal-variable input) independent-variable evaluated tables. The tables are always evaluated with old-time variables, and situations can occur that are numerically unstable. These instabilities can be controlled by setting TWTOLD to a value greater than zero but less than one. This institutes time-step weighting with the result from the current evaluation of the table multiplied by $1 - \text{TWTOLD}$ and added to TWTOLD times the result from the preceding time step. In some cases, controlling the maximum rate of change of the velocity (or mass flow) with the input variable RFMX will also prevent problems from developing. Because of the infinite variety of functional dependencies that can be created with FILL tables, no set guidelines have been developed for the use of TWTOLD and RFMX. Before using them in a large system-model application, a user should carefully test such FILL-table functional forms that have independent or dependent parameters that vary rapidly.

J.4. BREAKS

A BREAK is often considered a pressure boundary condition, but it can affect the flow through more than just the pressure-gradient term in the motion equations. Lack of caution in selecting the fluid void fraction and fluid temperatures and pressure that determine the fluid microscopic densities in the BREAK mesh cell can result in unexpected gravitational pressure heads and a poor prediction of the flow regime used to compute the interfacial drag. When flow is in from a BREAK, the inflow momentum flux has been assumed to be zero in order to provide a numerically stable solution. The TRAC user can account for an inflow momentum flux by input specifying a static rather than dynamic pressure boundary condition (see [Section 2.1.7.2.1](#)). This requires knowing when the transient inflow from the BREAK occurs and then defining the pressure in its static rather than dynamic form. This can be done using control blocks to define the static or dynamic pressure boundary condition based on the flow direction of the BREAK junction velocity.

J.5. PLENUMs

A PLENUM is a simple generalization of 1D flow modeling to a single-cell component with an arbitrary number of junction connections to other 1D components. Having one mesh cell results in complete mixing of fluid from all junction connections as it flows through the PLENUM cell volume. A unique feature of a PLENUM component is that

the junction connections on a side convect momentum across the PLENUM component cell to the junction connections on the other side. There also can be junction connections on other sides that model the PLENUM cell center as having an infinitely large flow area and thus a zero momentum flux to/from the junction. Momentum can be convected across a fluid volume in all three directions with a 3D VESSEL component, but complete mixing as in a PLENUM component will not occur because two or more mesh cells in each direction are required to convect momentum across a VESSEL component between source connections. At the present time, heat structures cannot be coupled to a PLENUM hydro cell because of the different database form of a PLENUM component. The stratified-flow regime is not evaluated in the PLENUM cell. However, stratified flow is evaluated for the interfacial drag term at the PLENUM junctions.

J.6. Turbine (TURB) Component

Note: **Turbine (TURB) Component.** The base code for both TRAC-M/F77 and TRAC-M/F90 has a component to model turbines (the TURB component). In the TURB component energy is removed from the fluid as a result of work performed by the turbine. The TURB component received minimal support over its years in TRAC-P, and its general use was not, and is not, recommended. (It should be used with caution.) The TRAC-P TURB was brought over directly to TRAC-M. Currently, the TURB component is only available in TRAC-M/F77, and the same caution applies. The TRAC-P TURB has been removed entirely from TRAC-M/F90. An improved turbine modeling capability is to be included in a future TRAC-M/F90 version. Section J.6. describes the current (obsolete) TURB component.

In the turbine, energy is removed from the fluid as a result of work performed by the turbine. The model calculates the momentum and energy losses and the angular velocity as a function of the fluid-flow rate, fluid properties, and the turbine nozzle and blade geometry. These momentum and energy losses are subsequently subtracted from the TRAC momentum and energy equations, respectively.

Each TURB module represents an individual turbine stage. Thus the number of TURB components required in the TRAC model is equal to the total number of individual stages associated with the turbines rather than the number of turbines. However, the code internally recognizes the "family" of stages belonging to a particular turbine, which facilitates turbine-governing and overall turbine-performance calculations that depend on individual stage performances. All the data that are common among the family (i.e., data pertaining to the turbine rather than individual stages, such as the generator efficiency or the generator power demand) are entered under stage 1, and the other stages of the family access these data from stage 1. Modeling an individual stage with a TURB component allows the user complete flexibility for modeling extraction, regeneration, reheating, etc., using the other TRAC components.

In both the impulse and reaction turbines, the thermodynamics of the internal processes is basically the same. In the impulse turbine, the expansion of steam takes place in the nozzles or in fixed blades, and the kinetic energy of steam is converted into mechanical work at the blades. In the reaction turbine, the steam is allowed to expand in moving blades also, thereby transferring the reaction energy into mechanical work. Details of this process are discussed in many standard text books on thermodynamics. Salisbury (Ref. J-16.), in particular, discusses the subject comprehensively. The steam expansion process through a turbine stage is shown in Fig. J-12. Line AB represents isentropic expansion from pressure p_A to pressure p_B . Because of friction and other irreversible losses, the actual state of steam after expansion is B' and not B . Thus the steady-state energy equation gives

$$h_A + \frac{V_A^2}{2} = h_B + \frac{V_B^2}{2} = h_{B'} + \frac{V_{B'}^2}{2} + \frac{P}{\dot{m}} \quad , \quad (J-17)$$

where h is the fluid enthalpy, V is the fluid velocity, \dot{m} is the mass-flow rate, and P is the stage work per second. The velocity V_B is the ideal nozzle velocity, such that $P_{\text{ideal}} = \dot{m}V_B^2 / 2$, and is easily obtained using Eq. (J-17) along an isentrope.

The stage power, P , can therefore be calculated as

$$P = \eta_{\text{stg}} P_{\text{ideal}} = \eta_{\text{stg}} \dot{m} \frac{V_B^2}{2} \quad , \quad (J-18)$$

where η_{stg} is the stage efficiency.

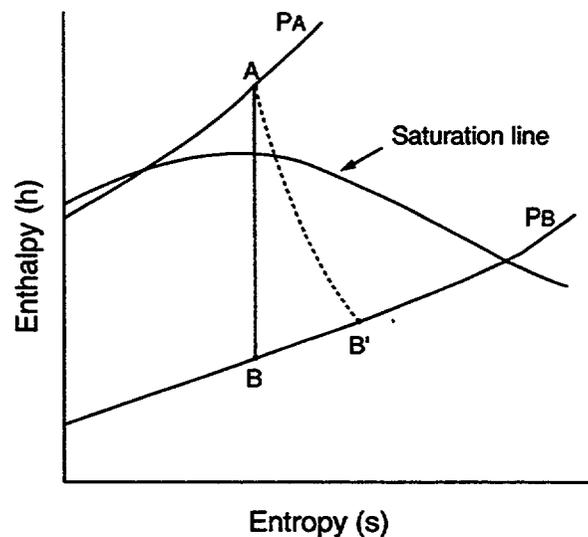


Fig. J-12. Steam expansion through a turbine stage.

The final state B' can be easily computed for a known η_{stg} under steady-state conditions using Eqs. (J-17), (J-18), the continuity equation, and the equation of state. This is the approach followed by Farman and Motloch (Ref. J-17.) in the RETRAN code. However, a preferable approach is to make use of the transient, two-phase TRAC hydrodynamic equations with appropriate sink terms in the momentum and energy equations. The momentum and energy terms corresponding to the turbine power are extracted as shown in a typical noding diagram in Fig. J-13. The energy is always extracted from cell number 2, whereas the momentum is extracted either from the second or the third cell edge depending upon the flow direction. A minimum of three cells are required to model a TURB component. It is clear from Eq. (J-17) that the velocity at the turbine exit would have been V_B had no work been done by the turbine. Owing to the turbine work, the exit velocity is $V_{B'}$ instead of V_B . Thus the momentum loss in the turbine stage is

$$M = \dot{m}(V_B - V_{B'}) = A_j \rho_2 V_j (V_B - V_{B'}) , \quad (J-19)$$

where A is the area. The subscript j (j may be 2 or 3, depending on the flow direction) refers to the momentum-sink cell edge. The density ρ_2 is from cell 2. Also, velocity V_B is the same as V_j , the actual cell-edge velocity calculated by the code after it accounts for the momentum loss. Therefore, writing Eq. (J-19) in terms of pressure loss, Δp , gives

$$\Delta p = \rho_2 V_j (V_B - V_{B'}) . \quad (J-20)$$

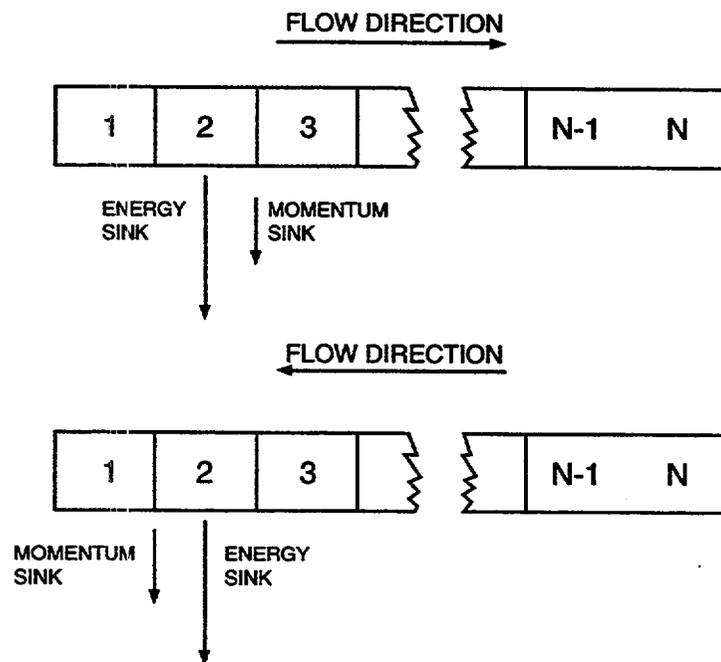


Fig. J-13. Turbine stage noding diagram depicting momentum and energy sink locations.

Experience has shown that translating the above pressure loss into an equivalent FRIC (FRIC is generally an input parameter to accommodate irreversible losses and can be related to an effective form loss K_{eff}) results in much improved code stability than subtracting Δp from the momentum equation directly (with its associated derivative $\partial \Delta p / \partial V_j$, of course). Therefore, writing Δp in terms of the additive-loss coefficient FRIC gives

$$\begin{aligned} \Delta p &= \text{FRIC} \frac{\rho_2 V_j^2 0.5(\Delta X_j + \Delta X_{j-1})}{d_j} \\ &= \frac{K_{\text{eff}}}{2} \rho_2 V_j^2 \quad , \end{aligned}$$

where d_j is the momentum source cell-edge diameter. Comparison of this equation with Eq. (J-20) leads to

$$\text{FRIC} = \frac{V_B - V_j}{V_j} \frac{d_j}{0.5(\Delta X_j + \Delta X_{j-1})} \quad . \quad (\text{J-21})$$

Because the above value of FRIC is applied at either cell edge 2 or cell edge 3 (depending upon flow direction), the user input additive FRIC values at these cell edges will be overwritten. Therefore FRIC(2) and FRIC(3) should always be input as 0.0.

The energy loss that should be taken out of the total energy equation is given by Eq. (J-18). However, TRAC uses the thermal energy equation, which is obtained after combining the total energy equation with the momentum equation. Some simple algebraic substitutions lead to the following expression for the sink term for the thermal energy equation:

$$P_{\text{enr}} = P - \frac{V_B + V_j}{2} M = P - A_j \frac{V_B + V_j}{2} \Delta p \quad , \quad (\text{J-22})$$

where P_{enr} is the amount to be extracted from the TRAC energy equation. The second term on the right-hand side of Eq. (J-22) appears as a result of combining the momentum equation with the energy equation.

Therefore, Eqs. (J-21) and (J-22) can be used to account for momentum and energy losses as a result of the power produced by a turbine stage provided the stage efficiency is known.

The stage efficiency is given by

$$\eta_{\text{stg}} = \eta_{\text{nb}} - \phi_m - \phi_{\text{rem}} \quad , \quad (\text{J-23})$$

where η_{nb} is the nozzle-bucket efficiency to be obtained from the velocity diagram, ϕ_m is the moisture loss, and ϕ_{rem} are the remaining miscellaneous stage losses whose

magnitudes are not easily definable. Term ϕ_{rem} includes nozzle-end loss, windage loss, diaphragm-packing loss, etc. The moisture loss is included as a separate term because it is the most significant loss and its functional formulation is known. These losses are described in detail by Salisbury in Ref. J-16. Salisbury recommends the following expression for the moisture loss:

$$\phi_m = 1.15(1 - x_t) \quad , \quad (J-24)$$

where x is the thermodynamic quality of steam.

The nozzle-bucket efficiency is calculated as

$$\begin{aligned} \eta_{nb} &= \frac{\text{energy transfer to the blades}}{\text{available energy at the nozzle}} \quad (J-25) \\ &= \frac{W \dot{m} \sum_{i=1}^{2n} V_{a_i}}{\dot{m} \frac{V_B^2}{2}} = \frac{2W}{V_B^2} \sum_{i=1}^{2n} V_{a_i} \end{aligned}$$

where W is the moving blade velocity, V_a is the axial component of velocity relative to the moving blades, and n is the number of rows of moving blades. A vector diagram for two rows of moving blades is presented in Fig. J-14, which can be extended to an arbitrary number of rows by our recursively appending a stationary blade row and a moving blade row in pairs to Fig. J-14. There is no limit on the number of rows allowed in a TURB module. If r is the fraction of stage energy transferred by reaction, the velocity vectors can be written as follows:

$$\begin{aligned} V_1 &= C_n V_B (1-r)^{1/2} \\ V_2 &= (V_1^2 + W^2 - 2V_1 W \cos \theta_1)^{1/2} \\ V_3 &= \left[(C_b V_2)^2 + r \frac{C_r^2}{2n-1} V_B^2 \right]^{1/2} \\ V_4 &= (V_3^2 + W^2 - 2V_3 W \cos \theta_2)^{1/2} \\ V_5 &= \left[(C_b V_4)^2 + r \frac{C_r^2}{2n-1} V_B^2 \right]^{1/2} \\ V_6 &= (V_5^2 + W^2 - 2V_5 W \cos \theta_3)^{1/2} \\ V_7 &= \left[(C_b V_6)^2 + r \frac{C_r^2}{2n-1} V_B^2 \right]^{1/2} \\ V_{a_1} &= V_1 \cos \theta_1 - W \\ V_{a_2} &= V_3 \cos \theta_2 \\ V_{a_3} &= V_5 \cos \theta_3 - W \\ V_{a_4} &= V_7 \cos \theta_4 \quad , \end{aligned} \quad (J-26)$$

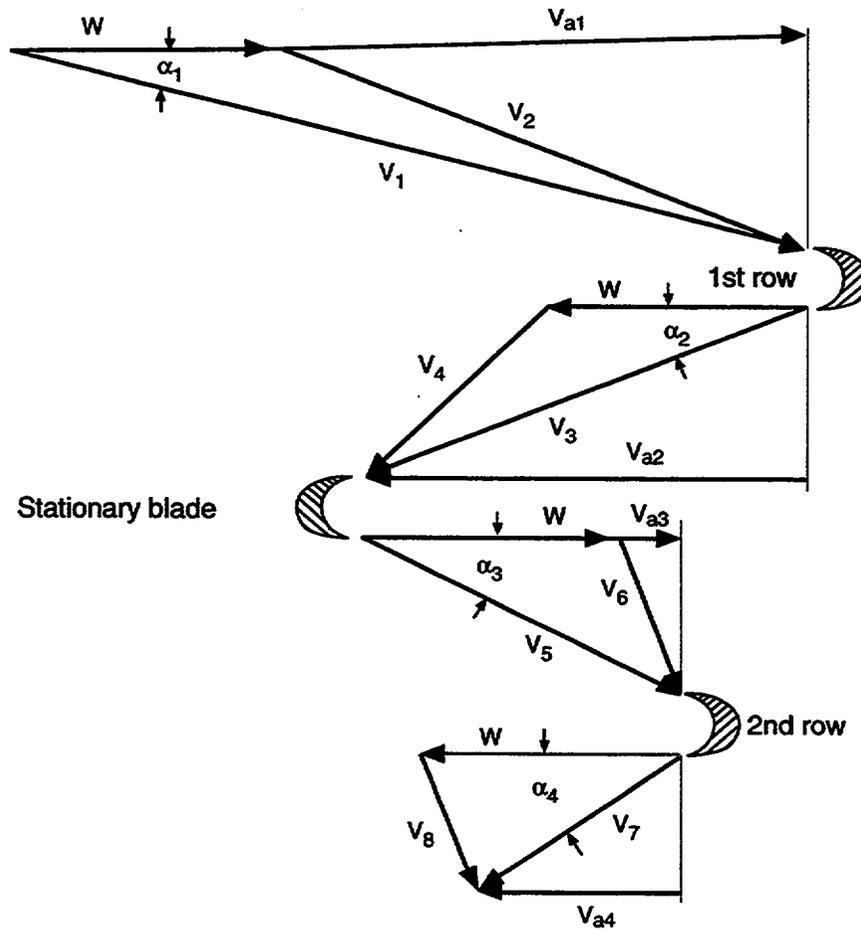


Fig. J-14. Velocity diagram for two rows of moving blades.

where C_n is the nozzle velocity coefficient, C_b is the bucket velocity coefficient, and C_r^2 is the fraction of total reaction energy actually delivered in all rows of blades. It is assumed that equal fractions of reaction energy are delivered in each blade row.

Although the remaining losses, ϕ_{rem} , required in Eq. (J-23) are not easily definable at off-design conditions, their magnitude is generally known at design conditions from the test data. In TRAC, the user is allowed an option of specifying the stage efficiency, η_{stg} , at design conditions. If η_{stg} is input, the code uses Eq. (J-23) to calculate ϕ_{rem} at design conditions. This value of ϕ_{rem} is then assumed constant throughout the transient. If the user does not input η_{stg} at design conditions, ϕ_{rem} is assumed to be zero.

Thus, to summarize, a TURB module performs the following sequence of calculations:

1. With the upstream and downstream pressures and velocities from the previous time step, the ideal nozzle velocity, V_B , is calculated and we assume steam is an ideal gas.
2. With the bucket velocity, W , from the previous time step, η_{nb} , is calculated from Eqs. (J-25) and (J-26).
3. The stage efficiency, η_{stg} , is then calculated from Eq. (J-23).
4. Stage power, P , is calculated from Eq. (J-18) with \dot{m} from the previous time step.
5. An equivalent FRIC for the momentum equation is calculated from Eq. (J-21) with V_j from the previous time step.
6. The amount to be extracted from the energy equation, P_{err} , is then calculated from Eq. (J-22).

So far we have discussed how each stage is handled by a TURB module, which represents an individual stage. In general, a turbine has more than one stage, and the governing is accomplished for the turbine as a whole and not for individual stages because only the total load on the turbine is known. Also, the angular speed among the "family" of stages is the same because of direct coupling of the stages. Thus additional calculations that are common among the family of stages are required. These calculations are done after all the stages are processed, and these common data are stored under the first stage of the family.

Total turbine power is calculated as

$$P_{total} = \sum_{i=1}^m P_i \quad , \quad (J-27)$$

where m is the total number of stages belonging to a turbine, i.e., the number of stages of the same family. However, the electrical generator power input is less than P_{total} because of the friction at the bearings of the turbine-generator shaft. Therefore, the power input to the generator, $P_{gen,input}$, is

$$P_{gen,input} = \sum_{i=1}^m P_i - \omega T_f \quad , \quad (J-28)$$

where ω is the turbine-generator angular speed and T_f is the frictional torque given by

$$T_f = K_1 + K_2\omega + K_3\omega^2 \quad , \quad (J-29)$$

where K_1 , K_2 , and K_3 are the user-input coefficients. The generator output, $P_{gen,output}$, is therefore

$$P_{\text{gen,output}} = \eta_{\text{gen}} \left[\sum_{i=1}^m P_i - \omega T_f \right] , \quad (\text{J-30})$$

where η_{gen} is the generator efficiency, which is treated as constant. At design conditions, this output must be equal to the generator power demand. However, because of uncertainties in stage power calculations, friction torque, and generator efficiency, this equality cannot be satisfied exactly. Thus a correction efficiency, η_{corr} , is defined, such that

$$\eta_{\text{corr}} = \frac{\text{generator power demand at design conditions}}{\left\{ \eta_{\text{gen}} \left[\sum_{i=1}^m P_i - \omega T_f \right] \right\}_{\text{design conditions}}}$$

This η_{corr} is calculated during initialization and is used as a constant multiplier during the entire transient. Thus Eq. (J-30) becomes

$$P_{\text{gen,output}} = \eta_{\text{corr}} \eta_{\text{gen}} \left[\sum_{i=1}^m P_i - \omega T_f \right] . \quad (\text{J-31})$$

At equilibrium conditions, this output should be equal to the generator power demand, P_{demand} , which is input by the user in tabular form. If $P_{\text{gen,output}} > P_{\text{demand}}$ the steam flow through the turbine should be reduced and vice versa. For example, Fig. J-15 shows two typical turbines (a two-stage throttle-governed and a three-stage bypass-governed) without any extractions to illustrate the use of special valve options to control steam flow. As the power demand increases, more steam has to be supplied to the turbine, and in case of a bypass-governed turbine, some additional steam has to be supplied to lower stages. However, as the steam flows through the turbines increase, the amount of steam bypassed must decrease. The turbine bypass valve therefore can be modeled using the appropriate VALVE option. It might, however, be preferable to operate the turbine bypass valve using a controller in order to maintain a constant pressure in the steam generator.

The turbine speed is maintained constant as long as the plant is connected with the external load, i.e., $P_{\text{demand}} \neq 0.0$, to maintain constant grid frequency. To meet this power demand at the desired grid frequency, the steam supply to the turbine is adjusted as discussed above. However, the turbine is allowed to coast down at the receipt of a turbine-trip signal (which is generated by tripping off the generator power demand). The turbine speed in this case is determined by the angular momentum balance

$$I \frac{d\omega}{dt} = T_{\text{turb}} - T_f - T_{\text{loss}} - T_{\text{demand}} , \quad (\text{J-32})$$

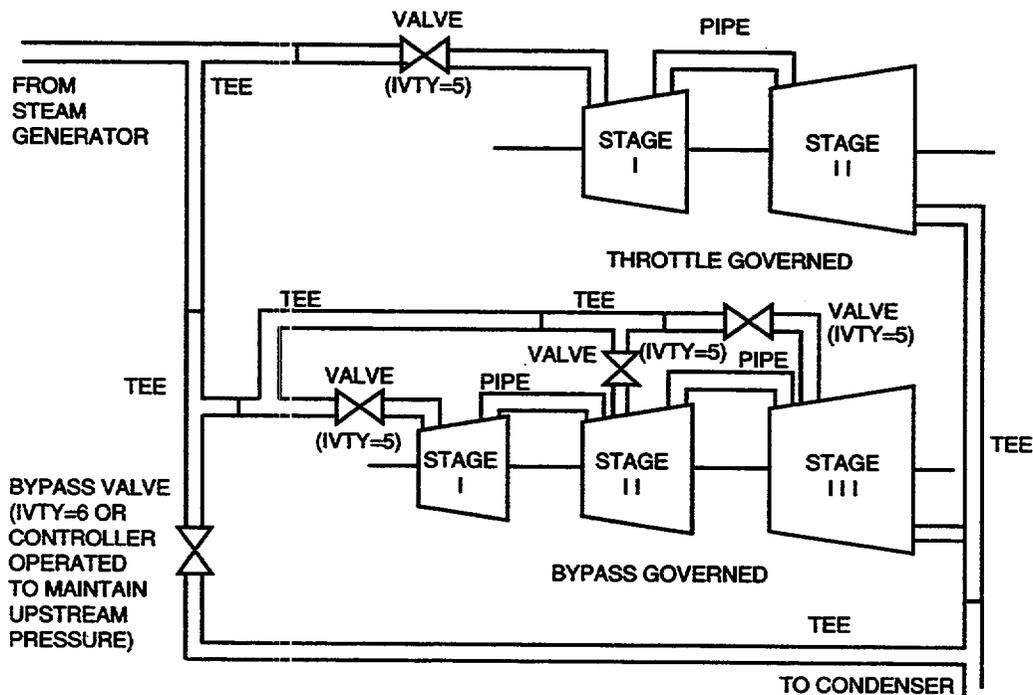


Fig. J-15. A typical noding diagram showing the combined use of turbine and VALVE components.

where I is the turbine-generator assembly moment of inertia, T_{turb} is the total turbine torque output from all stages, T_{loss} is the loss in torque as a result of generator losses, and T_{demand} is the torque due to external load. However, because the turbine is allowed to coast down only when the external load is zero, T_{demand} is 0.0. Also, because the generator is not producing any power, $T_{\text{loss}} = 0.0$. Thus Eq. (J-32) simplifies to

$$I \frac{d\omega}{dt} = \frac{1}{\omega} \sum_{i=1}^m P_i - T_f \quad . \quad (\text{J-33})$$

The TURB data are processed in a manner similar to the other 1D components. Subroutine RTURB reads data from the input file, and subroutine RETURB reads data from the restart file. Initialization of any remaining variables not provided by the input or restart files is performed in subroutine ITURB.

For the TURB component, subroutine TURBI calls TRBPOW, which calculates turbine stage efficiency and power. Two additional subroutines, TRBPRE and TRBPST, are called during the prepass and postpass stages, respectively, to perform calculations that are common for a family of different stages related to the same turbine.

J.7. Accumulator Modeling with the PIPE Component

Note: Accumulator Modeling. Early versions of TRAC-P had a specialized ACCUM component to model PWR accumulators. The special ACCUM models were later also included in the PIPE component (which also has built-in wall heat conduction that is not available in the ACCUM). The ACCUM component was removed from TRAC-P before TRAC-M development started.

TRAC provides two user-selected options in the PIPE component for modeling PWR accumulators. The PIPE component with input parameter IACC set to 1 selects the normal PIPE component (IACC = 0), plus an interface sharpener and additional output variables to monitor accumulator behavior. Setting IACC to 2 selects the features of IACC = 1, plus application of a liquid separator model at the bottom of the PIPE. Figure J-16. shows the typical nodalization of an accumulator with respect to gravity, i.e., a vertical stack of cells with cell 1 at the top and cell NCELLS at the bottom (NCELLS is the number of cells in the PIPE component).

The code sets the interfacial drag to zero at each internal interface j ($1 < j < \text{NCELLS} + 1$) in the PIPE component with IACC set to 1 or 2 to enhance the phase separation in the component. Additionally, the IACC = 2 option of the PIPE component will reset the additive friction factor (input array FRIC) at the NCELLS + 1 interface (outlet) as follows to prevent the gas phase from escaping from the component. The manipulation of the FRIC(NCELLS + 1) value begins in the subroutine that reads the input data; subroutine RPIPE sets the FRIC value at the last interface to $-1.123456 \times 10^{+29}$. Subroutine StbVel1D (FEMOM in TRAC-M/F77) interprets this FRIC value as a flag to force the FRIC applied in the liquid momentum equation to zero and the FRIC applied in the gas momentum equation to $+1.123456 \times 10^{+29}$.

The equivalent Darcy loss coefficient $K_{\text{NCELLS}+1}$ in the gas phase corresponding to the FRIC value at the last interface is dependent on the input geometry and is given by

$$K_{\text{NCELLS}+1} = \text{FRIC}_{\text{NCELLS}+1} \frac{\Delta X_{\text{NCELLS}} + \Delta X_{\text{NCELLS}+1}}{D_{\text{H,NCELLS}+1}}, \quad (\text{J-34})$$

where $D_{\text{H,NCELLS}+1}$ is the hydraulic diameter at the last interface, ΔX_{NCELLS} is the length of the last cell of the PIPE component, and $\Delta X_{\text{NCELLS}+1}$ is the length of the cell in the adjacent component across the NCELLS + 1 interface. This adjustment, which results in a very high additive loss on the gas phase at the exit of the component together with the zero additive loss on the liquid phase, essentially requires the gas velocity to be zero while the liquid flows subject only to the irreversible pressure drop caused by the interfacial drag.

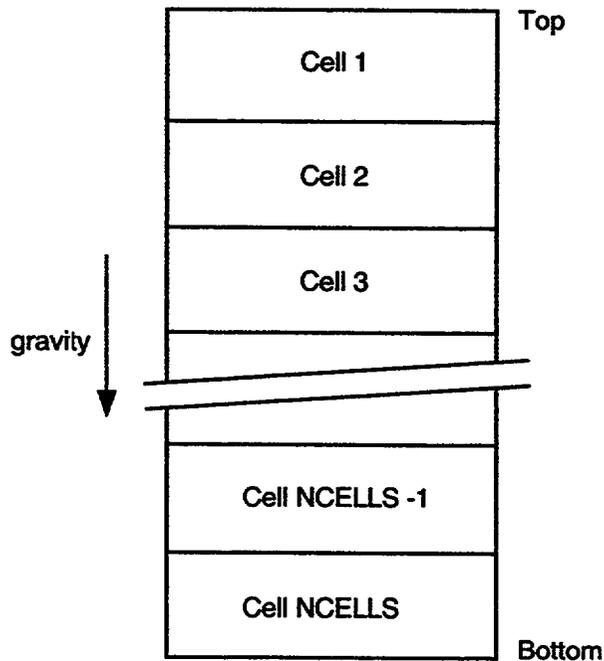


Fig. J-16. Typical component nodalization for the accumulator and pressurizer.

J.8. Pressurizer Component

The code defines the PRIZER component to represent the pressurizer. This component provides a special function during steady-state calculations (input parameter $STDYST > 0$ on Main-Data Card 4) in that the component simulates a BREAK component to set the system pressure and to permit the system fluid to swell or contract in response to temperature changes without requiring the user to model make-up and let-down systems. Also during the steady-state calculation, the code calculates the effects of the gravity head and thermal nonequilibrium in the fluid in the PRIZER component to prevent small secondary transients at the beginning of the transient calculation.

In addition to these very specialized functions during the steady-state calculations, the PRIZER component provides a representation of pressurizer heaters and sprays by manipulating the energy deposited in or extracted from the liquid in the PRIZER. The code assumes that the PRIZER consists of a vertical stack of cells, with cell 1 at the top and cell NCELLS at the bottom (see Fig. J-16.), and calculates a collapsed liquid level. The total liquid inventory in the PRIZER is

$$vol_{liq} = \sum_{i=1}^{NCELLS} (1 - \alpha_i) vol_i, \quad (J-35)$$

where vol_i is the volume of cell i and α_i is the associated void fraction. The collapsed liquid level z is

$$z = \sum_{j=1}^{k-1} \Delta X_{\text{NCELLS}+1-j} + \Delta X_{\text{NCELLS}+1-k} \frac{\text{vol}_{\text{liq}} - \sum_{j=1}^{k-1} \text{vol}_{\text{NCELLS}+1-j}}{\text{vol}_{\text{NCELLS}+1-k}} , \quad (\text{J-36})$$

where k is defined such that

$$\text{vol}_{\text{NCELLS}+1-k} > \text{vol}_{\text{liq}} - \sum_{j=1}^{k-1} \text{vol}_{\text{NCELLS}+1-j}$$

and ΔX_i is the length of cell i . Eq. (J-36) represents the standard method used in the code to calculate the collapsed liquid level.

If the collapsed liquid level $z \leq \text{ZHTR}$, an input parameter, no accounting of the heaters or sprays is made. For the case of $z > \text{ZHTR}$, the code calculates the total power P_{in} to be added to or extracted from the liquid as follows:

$$P'_{\text{in}} = P_{\text{HEAT}} \frac{p_{\text{set}} - p_1}{\text{DPMAX}} , \text{ and} \quad (\text{J-37})$$

$$\begin{aligned} P_{\text{in}} &= P'_{\text{in}} \quad \text{if} \quad P'_{\text{in}} < P_{\text{HEAT}} \\ &= P_{\text{HEAT}} \frac{P'_{\text{in}}}{|P'_{\text{in}}|} \quad \text{if} \quad P'_{\text{in}} \geq P_{\text{HEAT}} . \end{aligned} \quad (\text{J-38})$$

In the above equations, P_{HEAT} is the user-defined maximum effective power of the heaters and the sprays, p_{set} is the user-input pressure to which the system is controlled, DPMAX is the user-input pressure offset at which the heaters or sprays achieve their maximum effectiveness, and p_1 is the calculated pressure in the first (top) cell of the PRIZER.

The P_{in} is distributed among the cells based on the liquid in each cell by the following algorithm:

$$P_f = P_{\text{in}} \frac{\rho_{\text{li}} \text{vol}_i (1 - \alpha_i)}{\text{LMT}} , \quad (\text{J-39})$$

where LMT is the total liquid mass in the PRIZER defined by

$$\text{LMT} = \sum_{j=1}^{\text{NCELLS}} \rho_{\text{li}} \text{vol}_j (1 - \alpha_j) . \quad (\text{J-40})$$

Additionally, for each cell i , ρ_l is the microscopic liquid density, α is the void fraction, vol is the cell volume, and P_f is the power deposited in (extracted from) the liquid phase.

It is worth noting that if there is no liquid in the PRIZER, LMT is zero and the code does not perform the calculation to distribute the heat load among the individual cells. For this case, all of the P_f are zero.

The sign of P_{in} , and hence of P_f , indicates whether heat is added or subtracted from the liquid. If the calculated pressure p_1 is above the control pressure p_{set} , the sprays are assumed to be on and heat is removed from the liquid phase. For this case, the model is not mechanistic for four reasons: (1) the reduction in pressure relies on condensation of vapor on the surface of the liquid pool as opposed to the spray itself, (2) removing energy from the liquid results in artificially subcooling the entire liquid pool, (3) there is a net energy extraction from the PRIZER that does not actually occur, and (4) the mass flows associated with the sprays do not occur in the PRIZER.

However, the case when $p_1 < p_{set}$ requires that the heaters be on. This case is reasonably mechanistic in that the heat is added to the liquid. However, the distribution of the energy is not necessarily correct nor does the 1D modeling with a single liquid velocity permit calculating the convective currents within the liquid pool set up by the heat addition. Subroutine PRZR1X performs the calculations described above. When looking at the coding, one should keep in mind that the code keeps a running sum of all of the sources of heat to each cell for each time step.

J.9. Valve

The VALVE module is used to model various types of valves associated with LWRs. The valve action is modeled by controlling the flow area and the form loss at one cell interface of a 1D component. The valve action may not be located at a valve component junction unless that junction is connected to a BREAK.

Two methods are provided for specifying the valve flow area. The flow area FA can be computed directly from a flow-area fraction (FAVLVE) according to

$$FA = \text{FAVLVE} \cdot \text{AVLVE} \quad , \quad (\text{J-41})$$

where AVLVE is the input value for the fully open valve flow area. Alternatively, the flow area may be computed from the relative position (XPOS) of the valve stem, which assumes a guillotine-type cut of circular cross section. The relative valve-stem position of $\text{XPOS} = 1$ corresponds to a fully open valve with flow area AVLVE.

Because the hydraulic diameter is used in the wall friction calculation, the fully open valve hydraulic diameter (HVLVE user input) is not changed during a given calculation. $\text{HD}(\text{ivps})$ is set equal to HVLVE during initialization and is held constant after the

initialization phase. The form loss for flow through the valve is adjusted according to the valve flow area based on

$$K = K_0 + 0.5(1 - b) + (1 - b)^2 , \quad (J-42)$$

where K_0 is the fully open valve form loss and b is the ratio of the valve flow area to the pipe flow area.

This formula is recommended in the Crane manual (Ref. J-18.) for gate and ball valves. This formula will overestimate the form loss for partially closed globe and angle valves. For globe and angle valves, K_0 tends to be large and therefore the total error during opening or closing a globe or angle valve is relatively small. K_0 is also restricted to be greater than 0.03. K_0 of zero has been observed to be numerically unstable and the Crane manual indicates that K_0 should range from $3f$ to $400f$ or (assuming $f = 0.01$) 0.03 to 4.0, depending upon the valve design.

The FA fraction or valve-stem position is entered as a constant or a tabular function defined by a table. Use of a table may be trip-initiated according to the control option selected. To increase the flexibility to model various types of valves, two valve tables may be input for trip-controlled valves. The first table is used when the trip set status is $ON_{forward}$ and the second table is used when the trip set status is $ON_{reverse}$. The independent variable for the table is a modeled-system parameter defined by a signal variable or a control block.

Many different types of valves can be modeled because of the flexibility available to choose the independent variable for the valve-action table and to implement table evaluation under trip control. Simple valves that either open or close on a trip may be modeled using an OFF-ON- or ON-OFF-type trip and a table that has relative time (since trip initiation) as the independent variable to obtain the desired rate of opening or closing. Valve leakage can be simulated by restricting the table minimum FA fraction or valve-stem position to a value greater than zero. Simple valves can be used to model pipe breaks or the opening of rupture disks.

A simple check valve can be modeled by using a valve table with the appropriate pressure gradient as its independent variable. Alternatively, a check valve can be modeled as a trip-controlled valve with the pressure gradient used as the trip signal and the valve table used to control the rate of valve movement.

A steam-flow control valve [or power-operated relief valve (PORV)] can be modeled using an $ON_{reverse}$ -OFF- $ON_{forward}$ trip to control it with the start closing pressure, end closing pressure, end opening pressure, and start opening pressure as the respective trip set points. The rate of opening ($ON_{forward}$ state) can be defined by the first valve table and the rate of closing ($ON_{reverse}$ state) by the second valve table. The rate of opening and closing will be the same only if the first valve table is entered.

A PORV can be modeled by using a table with pressure as the independent variable and a step-like function for the flow-area fraction or valve-stem position. In this case, it is important that the step function not be too steep or the valve flow area may oscillate because of the coupling between the flow through the valve and the pressure variable. A bank of PORVs can be modeled with a single valve component in the same manner by using a multistep function to simulate the multiple pressure set points corresponding to the various valves.

This calculation of flow area assumes a guillotine cut of a circular cross section and is based on standard mensuration formulas. Figure J-17. shows the assumed geometry. When the valve is fully open, the flow area corresponds to the full circle. The stem position is assumed to be normalized such that a zero position is fully closed and a position of 1 is fully open. The calculation of the area and the corresponding hydraulic diameter is normalized similarly.

Subroutine FAXPOS performs the calculation of the normalized flow area. If the relative valve-stem position XPOS is known, the code can calculate the normalized flow area. For a circle with a radius of 1, one can define an H such that

$$\begin{aligned} H &= 1 - h \\ &= \min(1.0, |1.0 - 2 \cdot XPOS|) \end{aligned} \quad (J-43)$$

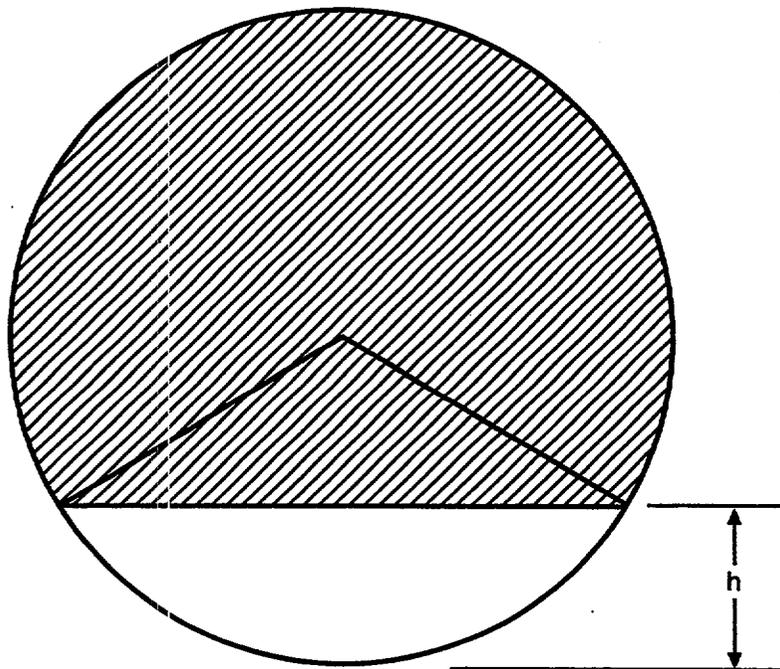


Fig. J-17. Assumed geometry for VALVE flow-area calculation.

where h is the dimension shown in Fig. J-17. Then half of the angle subtended by the chord is given by

$$\theta = \cos^{-1}H \quad , \quad (J-44)$$

and the resulting normalized flow area is

$$\begin{aligned} FA &= \frac{1}{\pi} \left(\theta - H\sqrt{1-H^2} \right) \\ &= 3.1830988 \times 10^{-1} \left(\theta - H\sqrt{1-H^2} \right) \quad . \end{aligned} \quad (J-45)$$

Because of symmetry, the total normalized flow area is

$$A_{\text{norm}} = FA \quad \text{if} \quad XPOS \leq 0.5 \quad (J-46)$$

$$= 1.0 - FA \quad \text{if} \quad XPOS > 0.5 \quad . \quad (J-47)$$

For the case in which the VALVE flow area is specified, the code solves Eqs. (J-44) through (J-48) iteratively for the angle θ and then for the stem position XPOS. The procedure is limited to 20 iterations and attempts to converge the solution for θ to less than 1.0×10^{-3} .

Now the angle θ is permitted to vary from 0 to π radians, such that 0 represents a fully closed valve and π a fully open valve. Therefore, the flow area A of the VALVE is

$$\begin{aligned} A &= R^2 \left[\theta + \sin \theta \cdot \cos \theta \cdot \frac{XPOS-0.5}{|XPOS-0.5|} \right] \\ &= 0.25 \left[\theta + \sin \theta \cdot \cos \theta \cdot \frac{XPOS-0.5}{|XPOS-0.5|} \right] \quad . \end{aligned} \quad (J-48)$$

Subroutine VLVEX applies the user-specified nominal full-open area AVLVE to the normalized value calculated in subroutine FAXPOS to calculate the actual flow area A :

$$A = AVLVE \cdot A_{\text{norm}} \quad . \quad (J-49)$$

REFERENCES

- J-1. "RELAP4/MOD5—A Computer Program for Transient Thermal-Hydraulic Analysis of Nuclear Reactors and Related Systems," Vol. 1, Aerojet Nuclear Company report ANCR-NUREG-1335 (September 1976).

- J-2. Victor L. Streeter and E. Benjamin Wylie, *Hydraulic Transients* (McGraw-Hill Book Company, New York, 1957).
- J-3. A.J. Stepanoff, *Centrifugal and Axial Flow Pumps*, 2nd ed. (John Wiley & Sons, Inc., New York, 1957).
- J-4. P. W. Runstadler, "Review and Analysis of State-of-the-Art of Multiphase Pump Technology," Electric Power Research Institute report EPRI/NP-159 (February 1976).
- J-5. O. Furuya, "Development of an Analytic Model to Determine Pump Performance Under Two-Phase Flow Conditions," Electric Power Research Institute report EPRI/NP-3519 (May 1984).
- J-6. D.J. Olson, "Experiment Data Report for Single- and Two-Phase Steady-State Tests of the 1-1/2-Loop Mod-1 Semiscale System Pump," Aerojet Nuclear Company report ANCR-1150 (May 1974).
- J-7. G. G. Loomis, "Intact Loop Pump Performance During the Semiscale MOD-1 Isothermal Test Series," Aerojet Nuclear Company report ANCR-1240 (October 1975).
- J-8. D. J. Olson, "Single- and Two-Phase Performance Characteristics of the MOD-1 Semiscale Pump Under Steady-State and Transient Fluid Conditions," Aerojet Nuclear Company report ANCR-1165 (October 1974).
- J-9. Douglas L. Reeder, "LOFT System and Test Description (5.5-Ft. Nuclear Core 1 LOCES)," EG&G Idaho, Inc. report TREE-1208 (NUREG/CR-0247) (July 1978).
- J-10. U.S. Nuclear Regulatory Commission memo, L. E. Phillips to U.S. Standard Problem Participants, Subject: Additional Information for Prediction of LOFT L2-3 (March 19, 1979).
- J-11. EG&G Idaho, Inc, letter, N. C. Kaufman to R. E. Tiller, Subject: Transmittal of Change 1 to NUREG/CR-0247 (Kau-243-80) (October 21, 1980).
- J-12. M. M. Giles, "Updated TRAC-BWR Completion Report: Generalized Leak Path and Separator Component Models," EG&G Idaho report SAAMD-83-029 (November 1983).
- J-13. W. L. Weaver, "TRAC-BWR Completion report: Implicit Separator/Dryer Model," EG&G Idaho report SE-RST-85-007 (August 1985).
- J-14. "TRAC-BD1 Manual: Extensions to TRAC-BD1/MOD1 (Draft)," Idaho National Engineering Laboratory report EGG-2417 (NUREG/CR-4391) (August 1985).

- J-15. Y. K. Cheung, V. Parameswaran, and J. C. Shaug, "BWR Refill-Reflood Program, Task 4.7 - Model Development, TRAC-BWR Component Models," General Electric Company report (NUREG/CR-2574) (April 1983).
- J-16. Kenneth Salisbury, *Steam Turbines and Their Cycles* (Robert E. Krieger Publishing Company, Huntington, New York, 1974).
- J-17. R. F. Farman and C. G. Motloch, "RETRAN Turbine and Condensing Heat Transfer Models," in "Proceedings of the First International RETRAN Conference" (Seattle, Washington, September 22-24, 1980), Electric Power Research Institute report EPRI WS-80-150.
- J-18. "Flow of Fluids," Crane Company Technical Paper 409 (May 1942).

APPENDIX K

ADDITIONAL MASS-FIELD CLOSURE

TRAC includes additional modeling capability to track a noncondensable gas in the gas field and a solute in the liquid field. We have incorporated the additional fields fairly simply by assuming that in a given cell, the noncondensable gas moves in a homogeneous mixture with the water vapor and that the solute is homogeneously mixed with the liquid. The advantage of these assumptions is that we need add only two additional mass-conservation equations: one for the noncondensable gas [Eq. (2-15)] and one for the liquid solute [Eq. (2-17)]. The addition of these two fields requires supplemental relationships to provide the necessary constraints and coupling to the rest of the fluid field equations. These relationships are described in Appendix K, Sections K.1, and K.2. The following nomenclature applies to Appendix K.

NOMENCLATURE

A :	flow area
c_p :	specific heat at constant pressure
c_v :	specific heat at constant volume
e :	internal energy
$H_{a1} - H_{a3}$:	coefficients of polynomial fit for air viscosity
$H_{b1} - H_{b3}$:	coefficients of polynomial fit for air viscosity
$H_{c1} - H_{c3}$:	coefficients of polynomial fit for hydrogen viscosity
$H_{d1} - H_{d3}$:	coefficients of polynomial fit for helium viscosity
k :	thermal conductivity
k_0 :	curve fit constant for thermal conductivity
m :	solute concentration
m_{solute} :	solute mass
M :	molecular weight
n :	curve fit exponent for thermal conductivity
P :	pressure
R :	universal gas constant
T :	temperature
V :	velocity
α :	void fraction
Δt :	time-step size
μ :	dynamic viscosity
ρ :	density

Subscripts

<i>a</i> :	refers to noncondensable gas in general
air:	air
<i>g</i> :	gas
helium:	helium
hydrogen:	hydrogen
<i>ℓ</i> :	refer to liquid

K.1. Noncondensable Gas

The inclusion of the noncondensable gas is more complex than that of the liquid solute because of the significant interactions that the noncondensable can have on interfacial and wall condensation. We have assumed that the noncondensable gas is homogeneously mixed with the water vapor so that we need only a single momentum equation for the gas field. Similarly, we assume that the noncondensable gas is in thermal equilibrium with the water vapor so that we need only a single energy equation for the gas field.

We have previously discussed noncondensable-gas effects in Appendices E, G, H, and I as they affect interfacial heat and mass transfer, wall heat transfer, and critical flow. This current discussion describes the correlations required to provide the necessary thermal and transport properties.

Note that TRAC also has an input option controlled by variable IEOS that can turn off phase change. In this case the noncondensable-mass equation is not used, but the user still selects among the noncondensable properties described in this appendix. The IEOS option has not been maintained and its use currently is not recommended.

K.1.1. Code Models

The code treats the noncondensable gas as an ideal gas, and the user can select the gas to be either air, hydrogen, or helium.¹ The current structure of the code permits only a single-species noncondensable gas throughout the complete code calculation. Other than the input option of air, hydrogen, or helium, the code does not permit the user to change the noncondensable gas through input to nitrogen, for example, to model the normal gas blanket in the accumulators of a PWR (for this case, the option to use air is a good approximation for nitrogen).

Equations (2-14) through (2-16) express the mixture relations necessary to describe the noncondensable-gas/water-vapor mixture density, internal energy, and pressure. From thermodynamics, we can define the noncondensable internal energy e_a in terms of the

1. More accurate properties also are available for helium. These optional properties for helium are not discussed here; see Ref. K-12 for complete details. Currently, these properties are only available for light water models (H₂O).

specific heat at constant volume c_v and the gas temperature T_g . The code describes the noncondensable viscosity μ_a with second-order polynomial fits to air, hydrogen, and helium data. The viscosity and specific heat are not strong functions of pressure (Ref. K-1.), and the correlations used for these two quantities do not include a pressure dependence.

The code calculates the thermodynamic and transport properties with the following correlations:

Internal Energy (Ref. K-2., pp. 21-22, Eqs. 1.29-1.32, for air and hydrogen; Ref. K-3., p. 247, for helium)

$$e_a = c_{va} T_g \quad (\text{K-1})$$

and

$$c_{va} = c_{pa} - \frac{R}{M_a} \quad , \quad (\text{K-2})$$

where c_{pa} is the specific heat at constant pressure, R is the universal gas constant, and M_a is the mass of one mole (molecular weight with units of kg) of the noncondensable gas. These two equations yield the following relations:

$$\left. \frac{\partial e_a}{\partial T_g} \right|_{P_a} = c_{va} \quad (\text{K-3})$$

and

$$\left. \frac{\partial e_a}{\partial P_a} \right|_{T_g} = 0.0 \quad . \quad (\text{K-4})$$

Density (Ref. K-1.)

$$\rho_a = \frac{P_a}{R_a T_g} \quad , \quad (\text{K-5})$$

$$\left. \frac{\partial \rho_a}{\partial P_a} \right|_{T_g} = \frac{1}{R_a T_g} \quad , \quad (\text{K-6})$$

and

$$\left. \frac{\partial \rho_a}{\partial T_g} \right|_P = -R_a \rho_a \left. \frac{\partial \rho_a}{\partial P_a} \right|_{T_g}, \quad (\text{K-7})$$

where ρ_a is the noncondensable microscopic density, P_a is the noncondensable partial pressure, and R_a is the noncondensable-gas constant defined as R/M_a .

Viscosity

Air (Ref. K-1.)

For $T_g \leq 502.15\text{K}$,

$$\mu_a = H_{a1} + H_{a2}(T_g - 273.15) + H_{a3}(T_g - 273.15)^2; \quad (\text{K-8})$$

For $T_g > 502.15\text{K}$,

$$\mu_a = H_{b1} + H_{b2}(T_g - 273.15) + H_{b3}(T_g - 273.15)^2; \quad (\text{K-9})$$

Hydrogen (Ref. K-1.)

$$\mu_a = H_{c1} + H_{c2}T_g + H_{c3}T_g^2; \quad (\text{K-10})$$

Helium (Ref. K-4.)

$$\mu_a = H_{d1} + H_{d2}T_g + H_{d3}T_g^2; \quad (\text{K-11})$$

where the H_s are polynomial coefficients to be defined later.

Thermal Conductivity

$$k_a = k_0 T_g^n. \quad (\text{K-12})$$

We use the subroutines THERMO and FPROP for thermodynamic and transport property calculations. SETEOS sets constants used in THERMO. IPROP calls THERMO and FPROP.

K.1.2. Range of Data Over Which Correlations Were Developed and Tested

We use the ideal-gas law for air, hydrogen, and helium density correlations. Because the ideal-gas law accurately predicts gas behavior for low pressure and high temperature, and TRAC usually deals with this range of pressures and temperatures, we consider the ideal-gas correlation to provide an adequate approximation for the gas densities.

Although the specific heat at constant pressure c_p is temperature dependent, we use a constant value for it. This assumption introduces errors that were deemed to be inconsequential to most transients of interest in PWRs. The thermal conductivity k is a strong function of temperature and a weak function of pressure. Consequently, it is computed using an exponential approximation with respect to temperature only. This assumption should introduce inconsequential errors to most transients of interest in PWRs.

The range of data over which the viscosity correlation was probably developed for air and hydrogen was $273.15 \text{ K} \leq T \leq 1073.15 \text{ K}$ (Ref. K-3.). TRAC will use this correlation, however, for $273.15 \text{ K} \leq T_g \leq 3000 \text{ K}$ and $1.0 \text{ Pa} \leq P \leq 45.0 \text{ MPa}$. The reference cited has no data for air or hydrogen viscosities with temperatures greater than 1073.15 K. However, Holman (Ref. K-1.) has viscosity of air to 2773.15 K and of hydrogen to 1173.15 K; we used these data in the assessment of the correlation. The range of data over which the helium viscosity correlation was developed is $255 \text{ K} \leq T_g \leq 800 \text{ K}$. TRAC uses this correlation for $273.15 \text{ K} \leq T_g \leq 3000 \text{ K}$ and $1.0 \text{ Pa} \leq P \leq 45.0 \text{ MPa}$. The reference cited has no data for helium viscosities at temperatures greater than 800 K.

K.1.3. Constants

We have defined in TRAC the following constants for use in Eqs. (K-1) through (K-12):

$$\begin{aligned}
 M_{\text{air}} &= 28.96461 \text{ kg} \cdot \text{mole}^{-1}; \\
 M_{\text{hydrogen}} &= 2.01594 \text{ kg} \cdot \text{mole}^{-1}; \\
 M_{\text{helium}} &= 4.00260 \text{ kg} \cdot \text{mole}^{-1}; \\
 c_{p,\text{air}} &= 1004.832 \text{ J} \cdot (\text{kg} \cdot \text{K})^{-1}; \\
 c_{p,\text{hydrogen}} &= 14533.2 \text{ J} \cdot (\text{kg} \cdot \text{K})^{-1}; \\
 c_{p,\text{helium}} &= 5234.0 \text{ J} \cdot (\text{kg} \cdot \text{K})^{-1}; \\
 H_{a1} &= 1.707623 \times 10^{-5} \text{ N} \cdot \text{s} \cdot \text{m}^{-2}; \\
 H_{a2} &= 5.927 \times 10^{-8} \text{ N} \cdot \text{s} \cdot (\text{m}^2 \cdot \text{K})^{-1}; \\
 H_{a3} &= -8.14 \times 10^{-11} \text{ N} \cdot \text{s} \cdot (\text{m}^{-2} \cdot \text{K}^{-2}); \\
 H_{b1} &= 1.735 \times 10^{-5} \text{ N} \cdot \text{s} \cdot \text{m}^{-2}; \\
 H_{b2} &= 4.193 \times 10^{-8} \text{ N} \cdot \text{s} \cdot (\text{m}^2 \cdot \text{K})^{-1}; \\
 H_{b3} &= -1.09 \times 10^{-11} \text{ N} \cdot \text{s} \cdot (\text{m}^{-2} \cdot \text{K}^{-2}); \\
 H_{c1} &= 4.175 \times 10^{-6} \text{ N} \cdot \text{s} \cdot \text{m}^{-2}; \\
 H_{c2} &= 1.588 \times 10^{-8} \text{ N} \cdot \text{s} \cdot (\text{m}^2 \cdot \text{K})^{-1}; \\
 H_{c3} &= 7.6705 \times 10^{-13} \text{ N} \cdot \text{s} \cdot (\text{m}^{-2} \cdot \text{K}^{-2}); \\
 H_{d1} &= 5.9642 \times 10^{-6} \text{ N} \cdot \text{s} \cdot \text{m}^{-2}; \\
 H_{d2} &= 5.2047 \times 10^{-8} \text{ N} \cdot \text{s} \cdot (\text{m}^2 \cdot \text{K})^{-1}; \\
 H_{d3} &= -1.5345 \times 10^{-11} \text{ N} \cdot \text{s} \cdot (\text{m}^{-2} \cdot \text{K}^{-2});
 \end{aligned}$$

$$\begin{aligned}
k_{0,\text{air}} &= 2.0910 \times 10^{-4} \text{ W}\cdot(\text{m}\cdot\text{K}^{n+1})^{-1}; \\
k_{0,\text{hydrogen}} &= 1.6355 \times 10^{-3} \text{ W}\cdot(\text{m}\cdot\text{K}^{n+1})^{-1}; \\
k_{0,\text{helium}} &= 3.366 \times 10^{-3} \text{ W}\cdot(\text{m}\cdot\text{K}^{n+1})^{-1}; \\
n_{\text{air}} &= 0.8460; \\
n_{\text{hydrogen}} &= 0.8213; \\
n_{\text{helium}} &= 0.6680; \text{ and} \\
R &= 6.022169 \times 10^{26} \times 1.380622 \times 10^{-23} \text{ J}\cdot(\text{mole}\cdot\text{K})^{-1} \\
&= 8314.339 \text{ J}\cdot(\text{mole}\cdot\text{K})^{-1}.
\end{aligned}$$

The code specifies the thermodynamic parameters in the above list in subroutine SETEOS. Function VISCV sets the coefficients for the dynamic viscosity μ_a , as well as for the temperature switch in the expression for air.

K.1.4. Assessment of the Correlation as Applied in TRAC

The equations used for calculating the viscosity of air and hydrogen were obtained by fitting a quadratic polynomial to data from Ref. K-3, with a least-squares method. The error associated with these fits is no greater than 3.34% for air with $502.15 \text{ K} < T_g \leq 1073.15 \text{ K}$, 2.91% for air with $273.15 \text{ K} \leq T_g \leq 502.15 \text{ K}$, and 5.23% for hydrogen. Reference K-3 gives values for viscosity only for temperatures up to 1073.15 K, but TRAC uses the polynomial fit for temperatures up to 2973.15 K. Tables K-1 and K-2 compare the results of Eqs. (K-8) and (K-9) with the data in Ref. K-1. The value for viscosity that TRAC calculates is over 25% low for air at 2500.00 K (as compared to data from Ref. K-1; see Table K-2). One should keep in mind that other things will happen in the code before the temperature gets this high, so that FPROP will probably never have to calculate viscosity with such a high temperature. Data for the viscosity of hydrogen at values greater than 1073.15 K were not available. The error associated with the viscosity of helium over the range $273.15 \text{ K} \leq T_g \leq 1090.15 \text{ K}$ is no greater than 5.6%. Data for this assessment were taken from Ref. K-4. Viscosity data for helium at temperatures greater than 1090.15 K were not available.

Figures K-1 through K-10 compare the specific heat at constant volume from TRAC with data at various temperatures and pressures for both air and hydrogen. The data are from Ref. K-4; we have converted them to specific heat at constant volume through the use of Eq. (K-2). The figures clearly show that the assumed constant specific heat is valid only at low pressures and temperatures; as the temperature or pressure or both increase, the errors associated with the assumption become quite large. Even Eq. (K-2) breaks down as the gas deviates from an ideal gas. We could improve the accuracy of the calculations with a noncondensable gas by representing the specific heats as functions of pressure and temperature. Figures K-11 through K-15 compare the TRAC specific heat at constant volume for helium (c_{helium}) with data from Ref. K-5. This reference contains no data for helium for temperatures greater than 590 K. The TRAC values for c_{helium} vary from the data by approximately 1.2%.

TABLE K-1.
Viscosity Polynomial Assessment for Air
T ≤ 502.15 K

Temperature (K)	Viscosity (Ref. K-1.) (N · s · m ⁻²)	Viscosity (TRAC) (N · s · m ⁻²)	% Error
273.15	1.70900E-05	1.70762E-05	-0.0805734
283.15	1.75900E-05	1.76608E-05	+0.402445
293.15	1.80800E-05	1.82291E-05	+0.824502
303.15	1.85600E-05	1.87811E-05	+1.19111
313.15	1.90400E-05	1.93168E-05	+1.45373
323.15	1.95100E-05	1.98362E-05	+1.67212
333.15	1.99700E-05	2.03394E-05	+1.84972
343.15	2.04300E-05	2.08263E-05	+1.93965
353.15	2.08800E-05	2.12969E-05	+1.99650
363.15	2.13200E-05	2.17512E-05	+2.02247
373.15	2.17500E-05	2.21892E-05	+2.01945
383.15	2.21800E-05	2.26110E-05	+1.94315
393.15	2.26000E-05	2.30165E-05	+1.84279
403.15	2.30200E-05	2.34057E-05	+1.67537
413.15	2.34400E-05	2.37786E-05	+1.44450
423.15	2.38500E-05	2.41352E-05	+1.19593
433.15	2.46500E-05	2.44756E-05	-0.707546
443.15	2.50500E-05	2.47997E-05	-0.999321
453.15	2.54400E-05	2.51075E-05	-1.30711
463.15	2.58200E-05	2.53990E-05	-1.63056
473.15	2.62000E-05	2.56742E-05	-2.00676
483.15	2.65800E-05	2.59332E-05	-2.43345
493.15	2.69600E-05	2.61759E-05	-2.90849

TABLE K-2.
Viscosity Polynomial Assessment for Air
T > 502.15 K

Temperature (K)	Viscosity (Ref. K-1.) (N · s · m ⁻²)	Viscosity (TRAC) (N · s · m ⁻²)	% Error
503.15	2.73300E-05	2.64173E-05	-3.33959
513.15	2.77000E-05	2.67854E-05	-3.30195
550.00	2.84800E-05	2.81229E-05	-1.25393
600.00	3.01800E-05	2.98904E-05	-0.959697
650.00	3.17700E-05	3.16033E-05	-0.524561
700.00	3.33200E-05	3.32618E-05	-0.174578
750.00	3.48100E-05	3.48658E-05	+0.160339
800.00	3.62500E-05	3.64153E-05	+0.455993
850.00	3.76500E-05	3.79103E-05	+0.691317
900.00	3.89900E-05	3.93508E-05	+0.925274
950.00	4.02300E-05	4.07367E-05	+1.25963
1000.00	4.15200E-05	4.20682E-05	+1.32040
1100.00	4.44000E-05	4.45677E-05	+0.377699
1200.00	4.69000E-05	4.68492E-05	-0.108389
1300.00	4.93000E-05	4.89126E-05	-0.785735
1400.00	5.17000E-05	5.07581E-05	-1.82186
1500.00	5.40000E-05	5.23856E-05	-2.98969
1600.00	5.63000E-05	5.37950E-05	-4.44932
1700.00	5.85000E-05	5.49865E-05	-6.00598
1800.00	6.07000E-05	5.59600E-05	-7.80895
1900.00	6.29000E-05	5.67154E-05	-9.83238
2000.00	6.50000E-05	5.72529E-05	-11.9186
2100.00	6.72000E-05	5.75724E-05	-14.3268
2200.00	6.93000E-05	5.76738E-05	-16.7766
2300.00	7.14000E-05	5.75573E-05	-19.3875
2400.00	7.35000E-05	5.72228E-05	-22.1459
2500.00	7.57000E-05	5.66702E-05	-25.1384

As mentioned before, the ideal-gas law, used for predicting air, hydrogen, and helium densities, approximates the densities for the gases well (generally less than 5% error for these gases) for high temperatures and low pressures. However, the limits inside THERMO and FPROP allow the pressures to get very high ($1.0 \text{ Pa} \leq P \leq 45.0 \text{ MPa}$), and errors may be as high as 20% for air, 26% for hydrogen, and 15% for helium (Figs. K-16 through K-30.) when the pressure reaches its upper limit and the temperature is low (data for these calculations are from Ref. K-6. for the density of air, from Ref. K-7. for the density of hydrogen, and from Ref. K-3. for the density of helium). Reference K-5. has no data for helium for temperatures greater than 590 K.

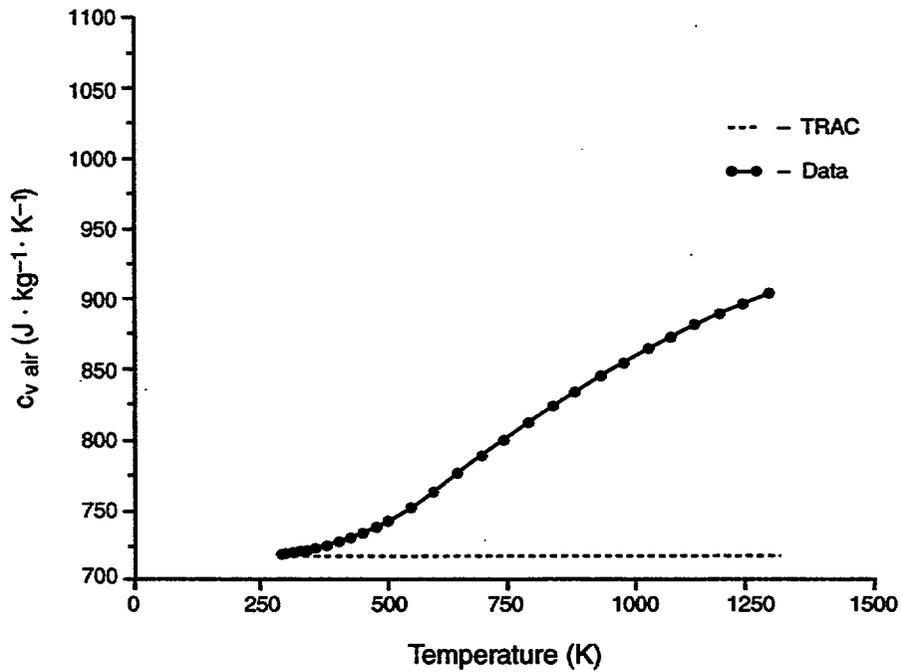


Fig. K-1. Specific heat vs. temperature for air at 100 kPa.

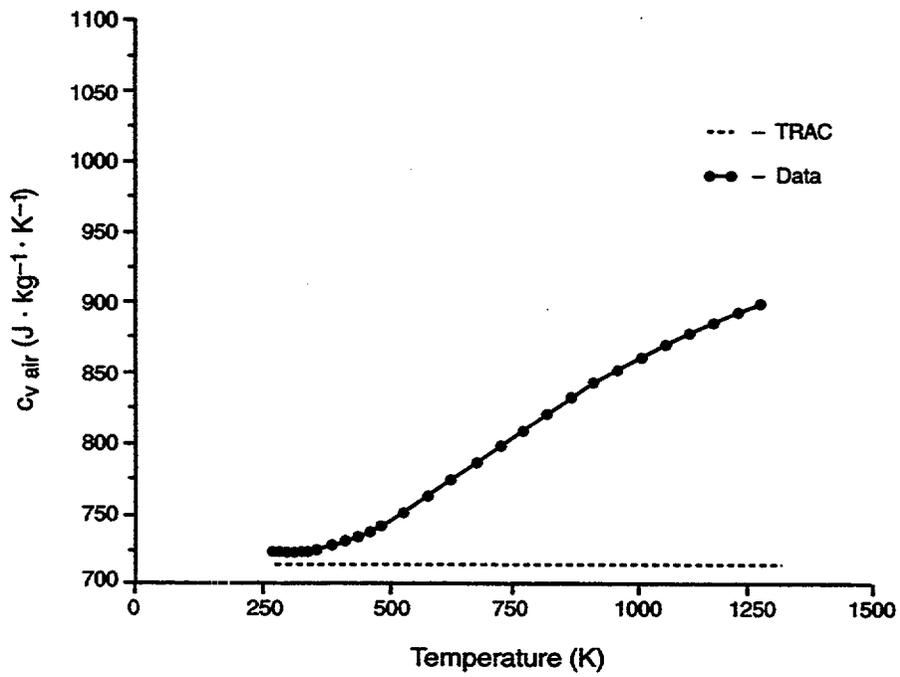


Fig. K-2. Specific heat vs. temperature for air at 400 kPa.

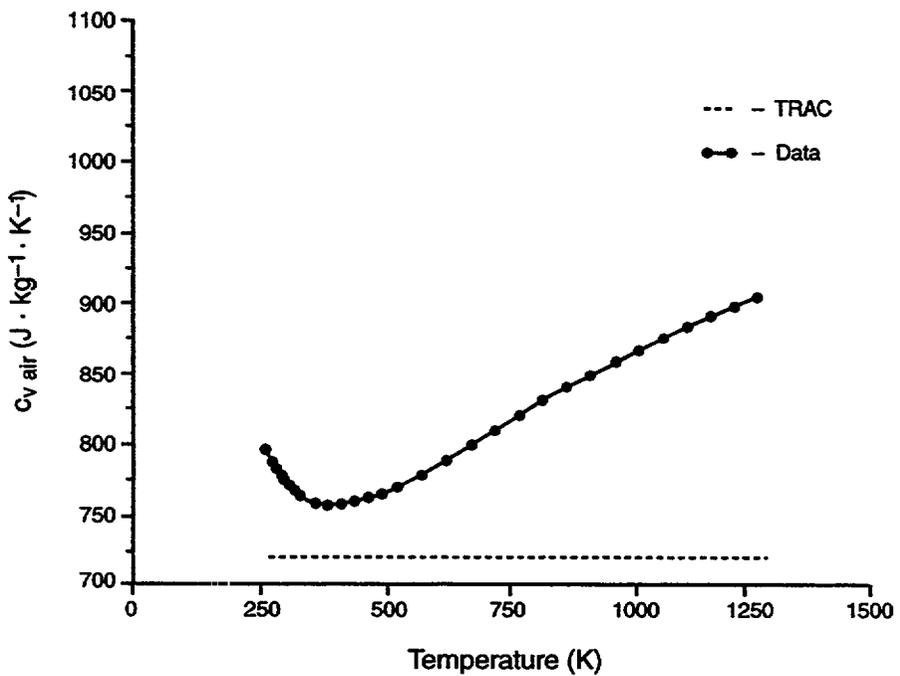


Fig. K-3. Specific heat vs. temperature for air at 4 MPa.

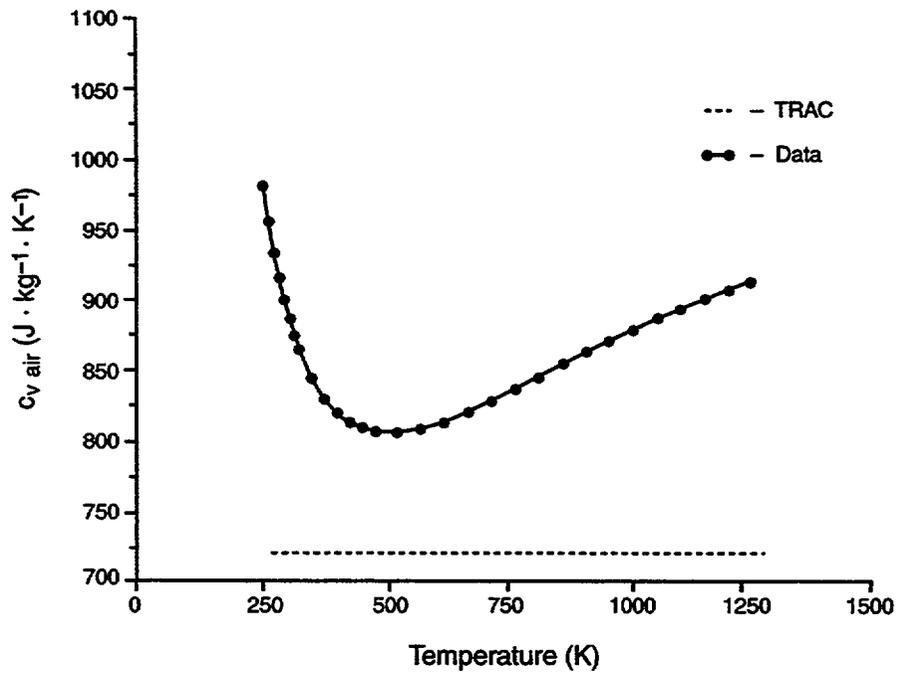


Fig. K-4. Specific heat vs. temperature for air at 15 MPa.

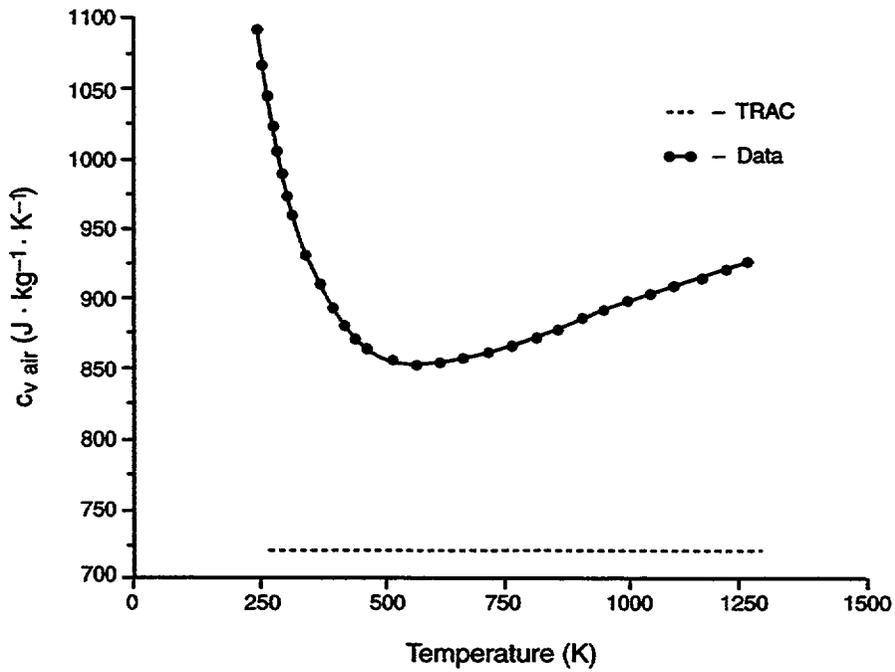


Fig. K-5. Specific heat vs. temperature for air at 40 MPa.

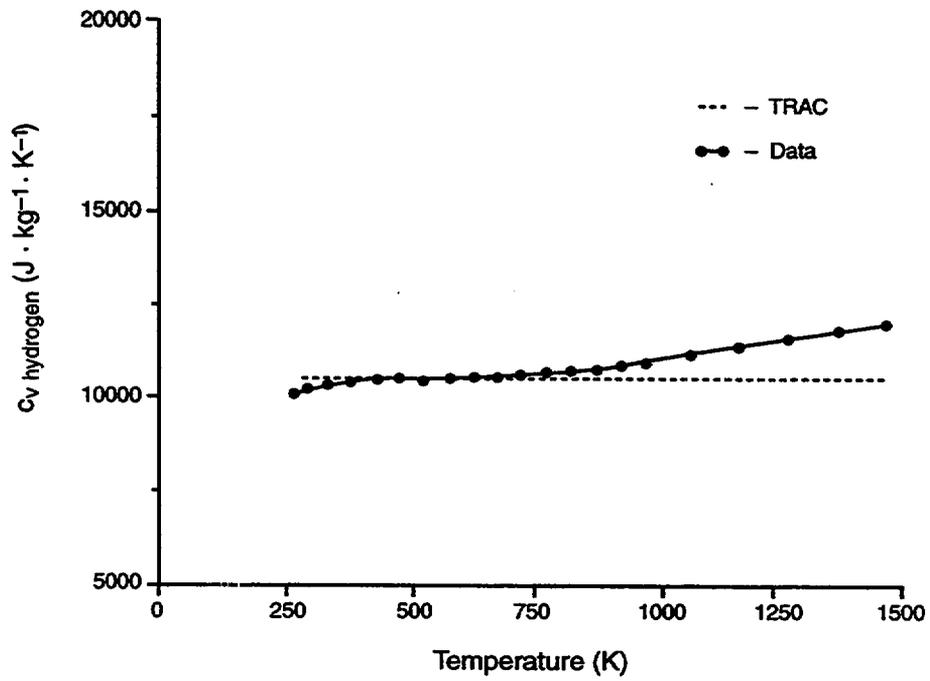


Fig. K-6. Specific heat vs. temperature for hydrogen at 100 kPa.

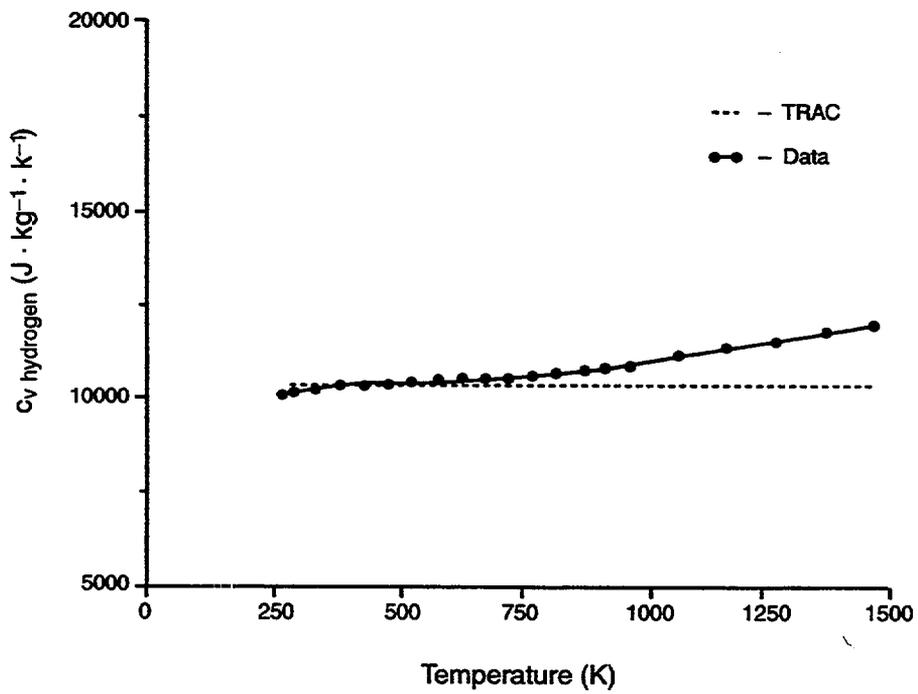


Fig. K-7. Specific heat vs. temperature for hydrogen at 400 kPa.

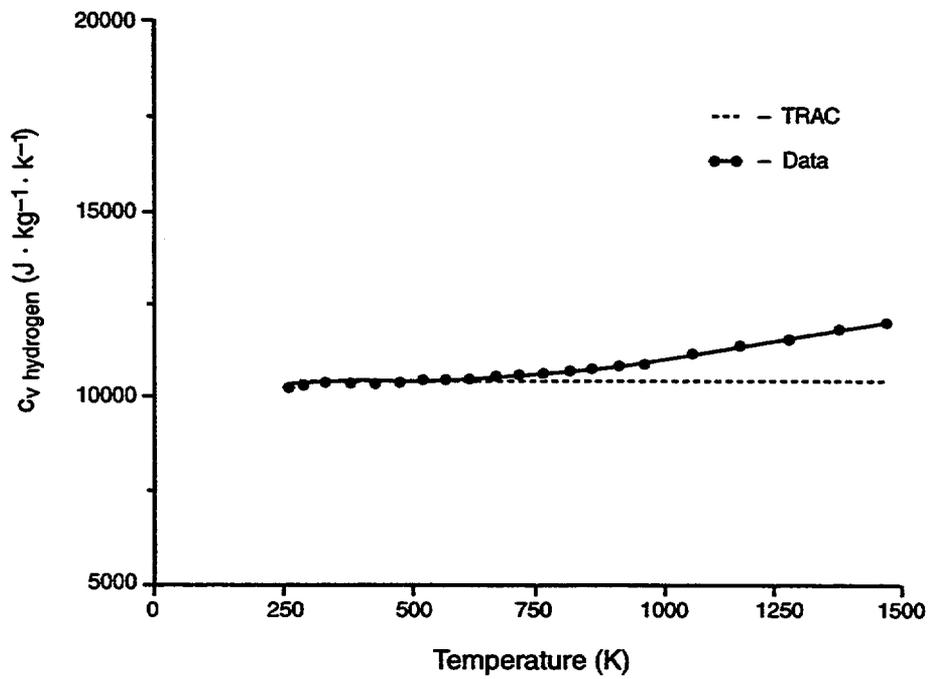


Fig. K-8. Specific heat vs. temperature for hydrogen at 4 MPa.

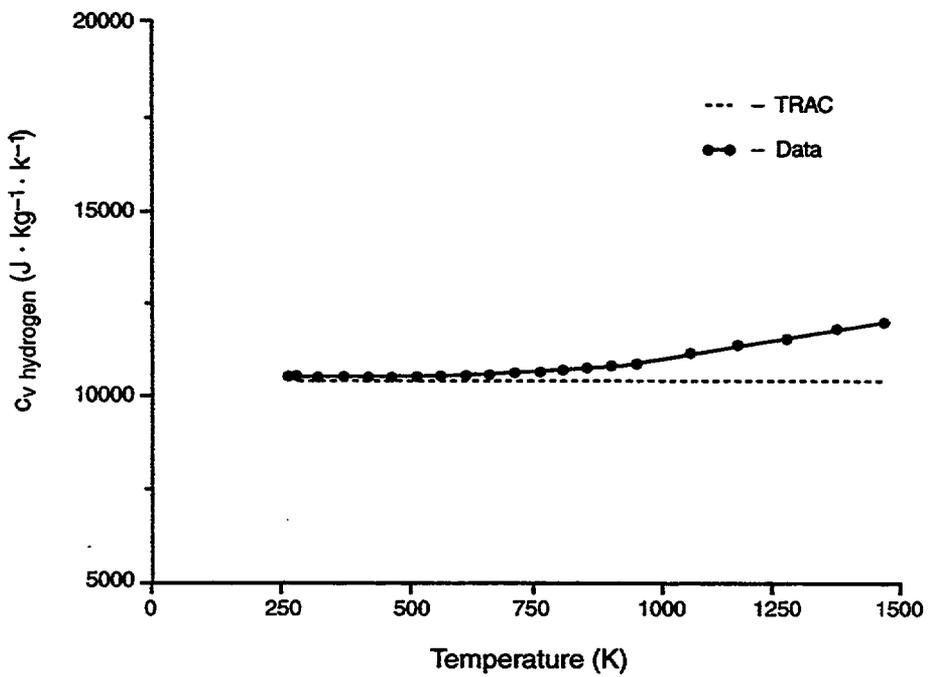


Fig. K-9. Specific heat vs. temperature for hydrogen at 15 MPa.

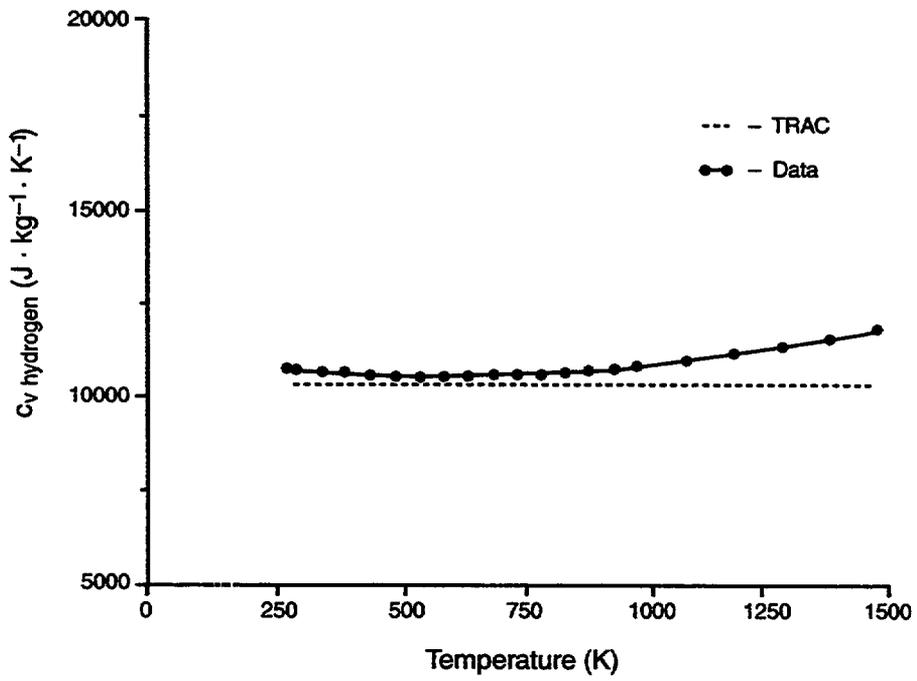


Fig. K-10. Specific heat vs. temperature for hydrogen at 40 MPa.

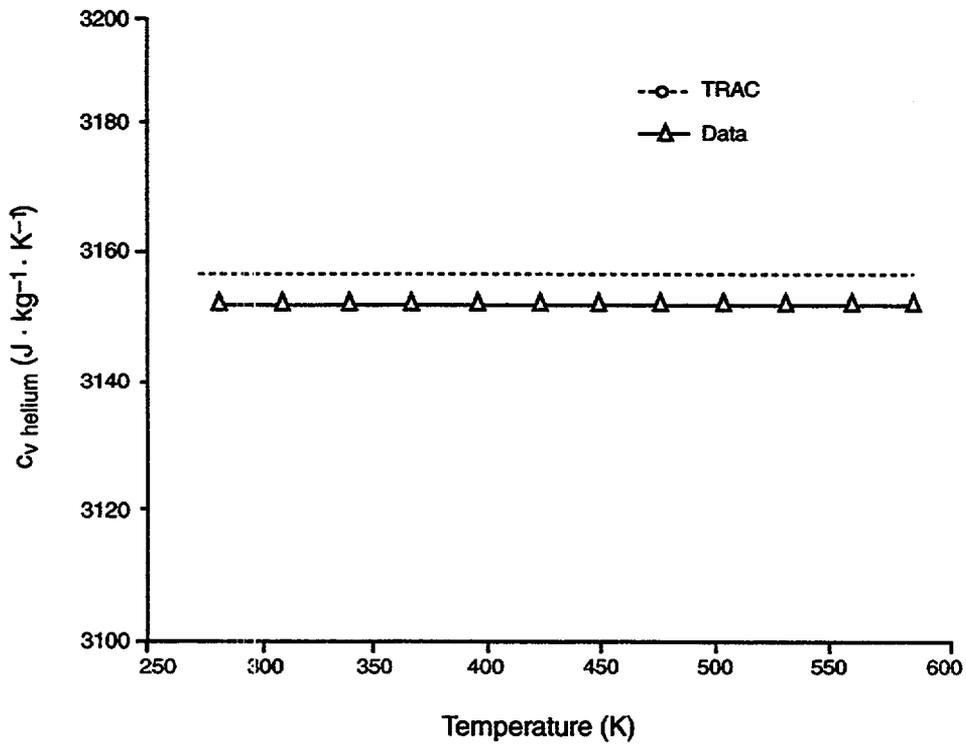


Fig. K-11. Specific heat vs. temperature for helium at 100 kPa.

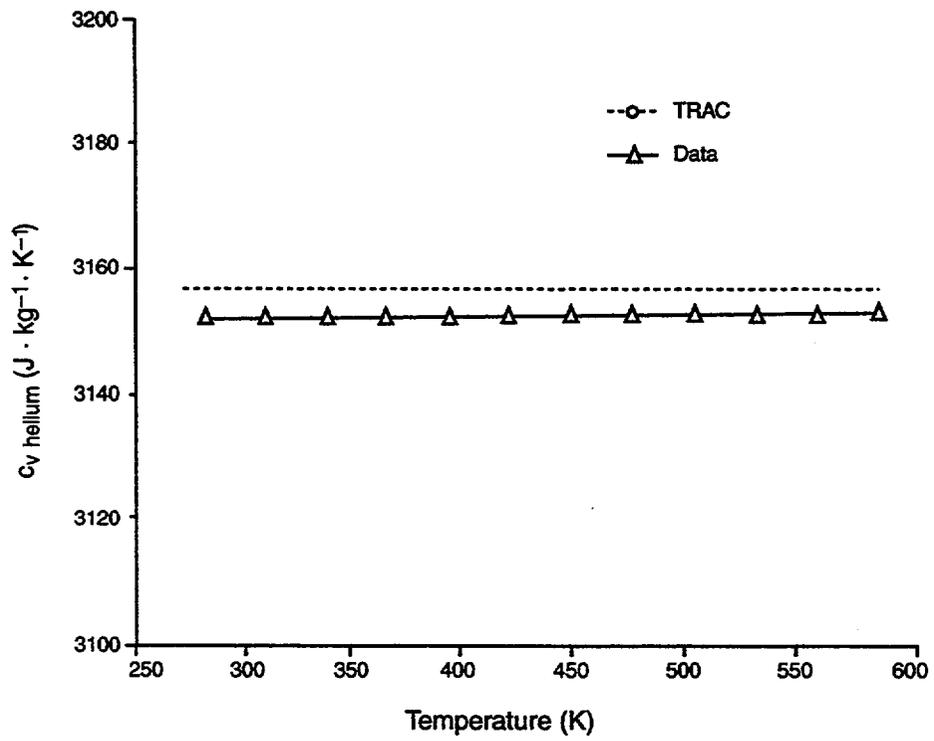


Fig. K-12. Specific heat vs. temperature for helium at 400 kPa.

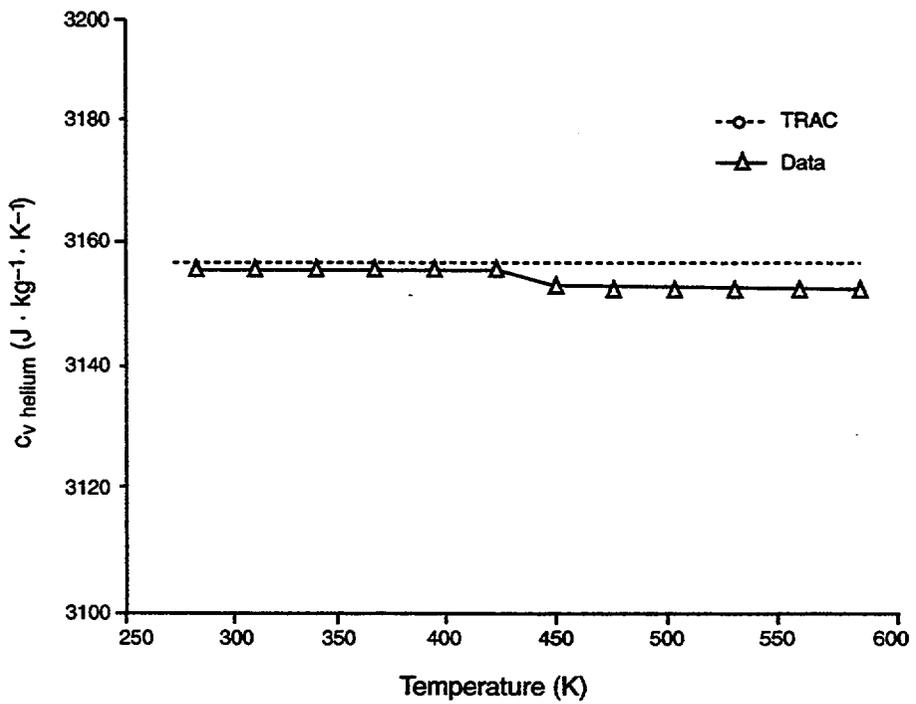


Fig. K-13. Specific heat vs. temperature for helium at 4 MPa.

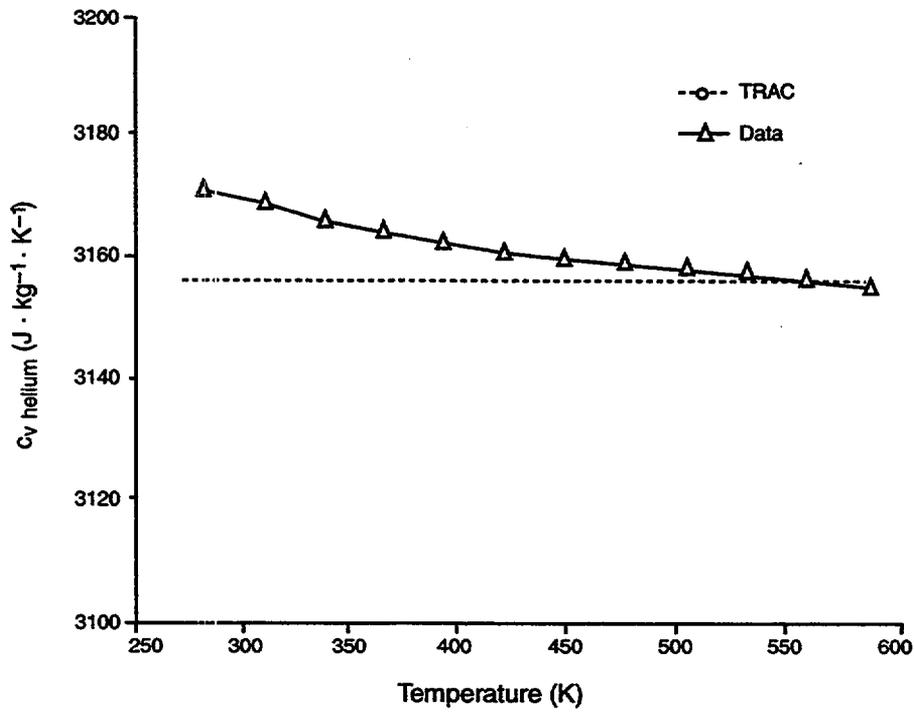


Fig. K-14. Specific heat vs. temperature for helium at 15 MPa.

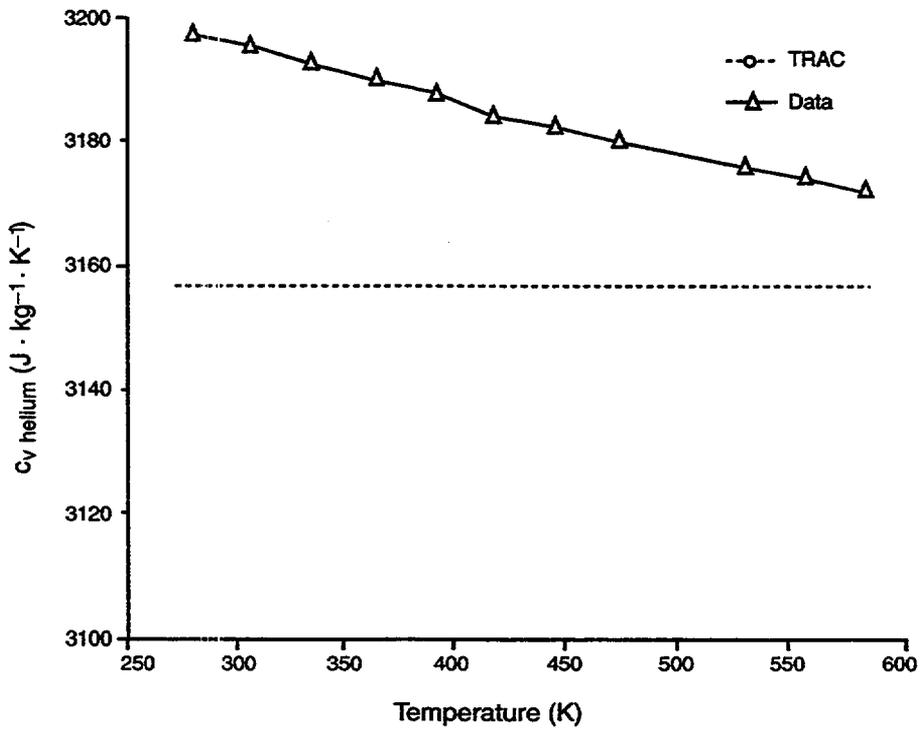


Fig. K-15. Specific heat vs. temperature for helium at 40 MPa.

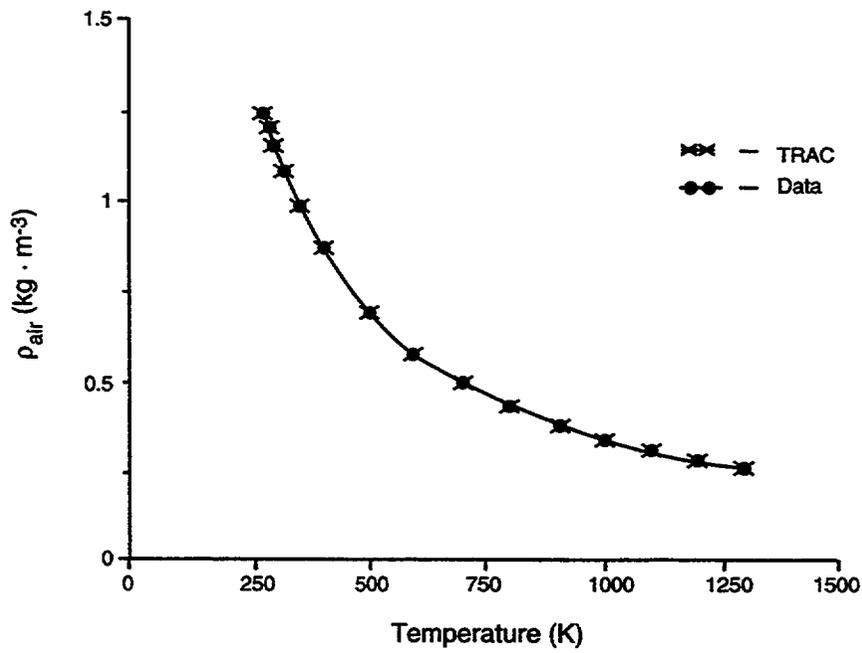


Fig. K-16. Density vs. temperature for air at 100 kPa.

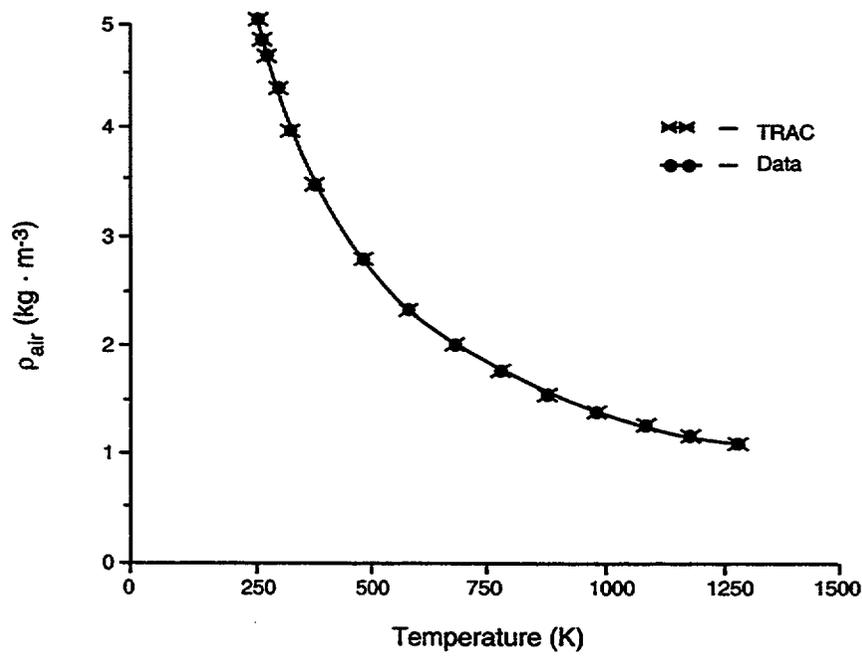


Fig. K-17. Density vs. temperature for air at 400 kPa.

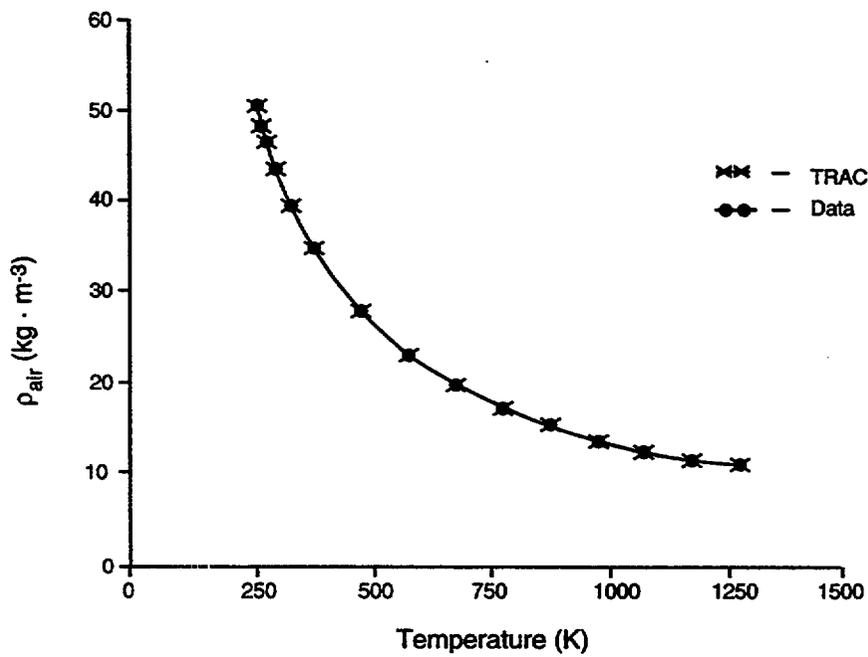


Fig. K-18. Density vs. temperature for air at 4 MPa.

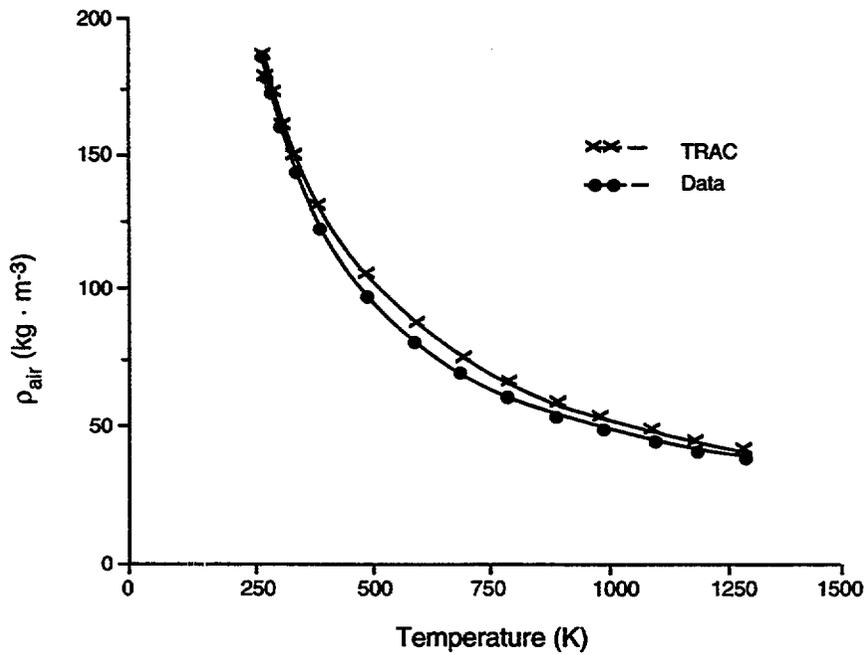


Fig. K-19. Density vs. temperature for air at 15 MPa.

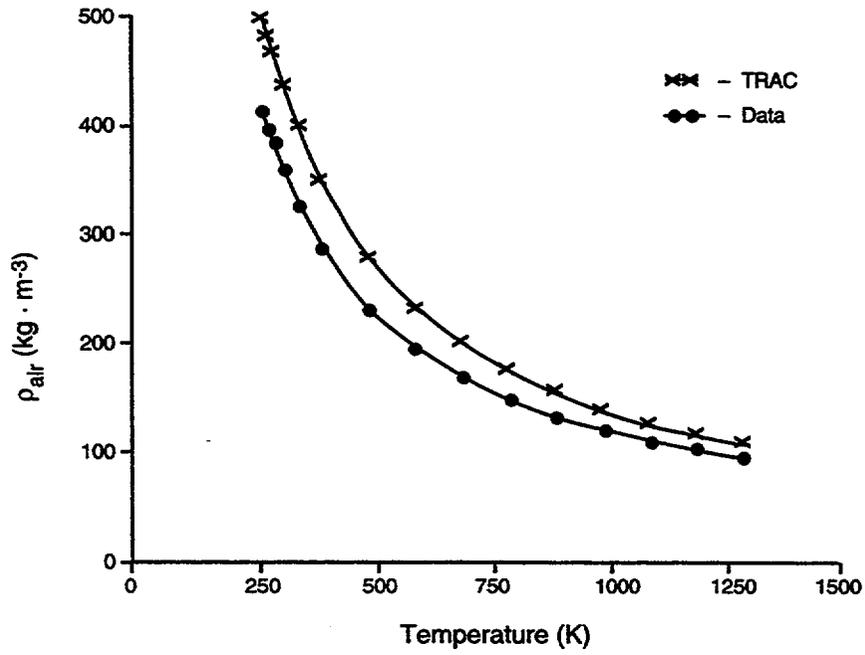


Fig. K-20. Density vs. temperature for air at 40 MPa.

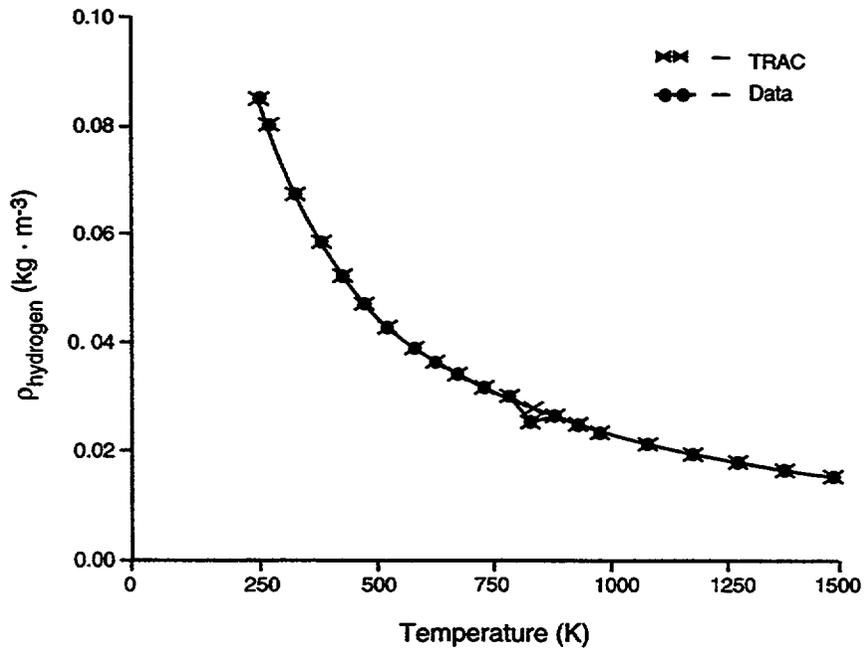


Fig. K-21. Density vs. temperature for hydrogen at 100 kPa.

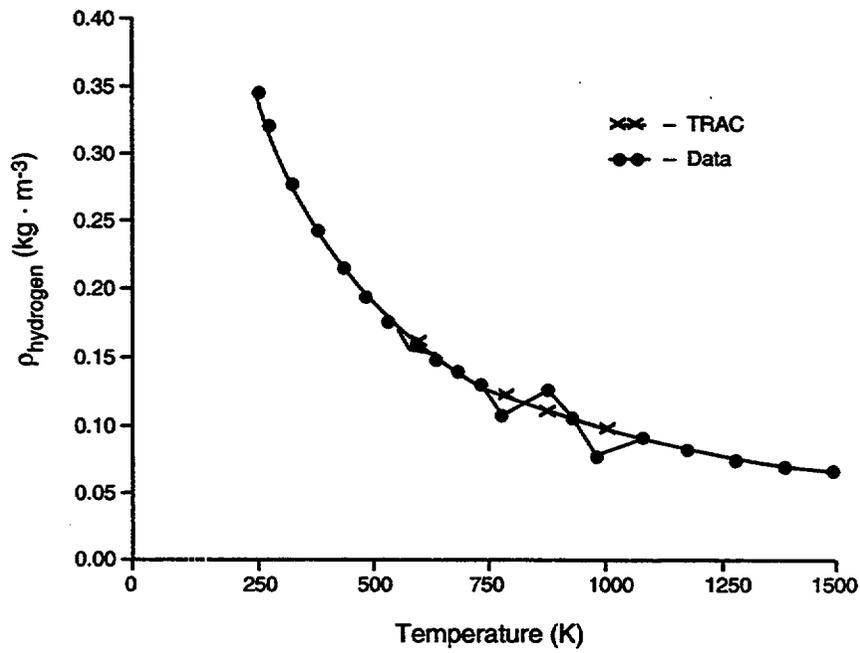


Fig. K-22. Density vs. temperature for hydrogen at 400 kPa.

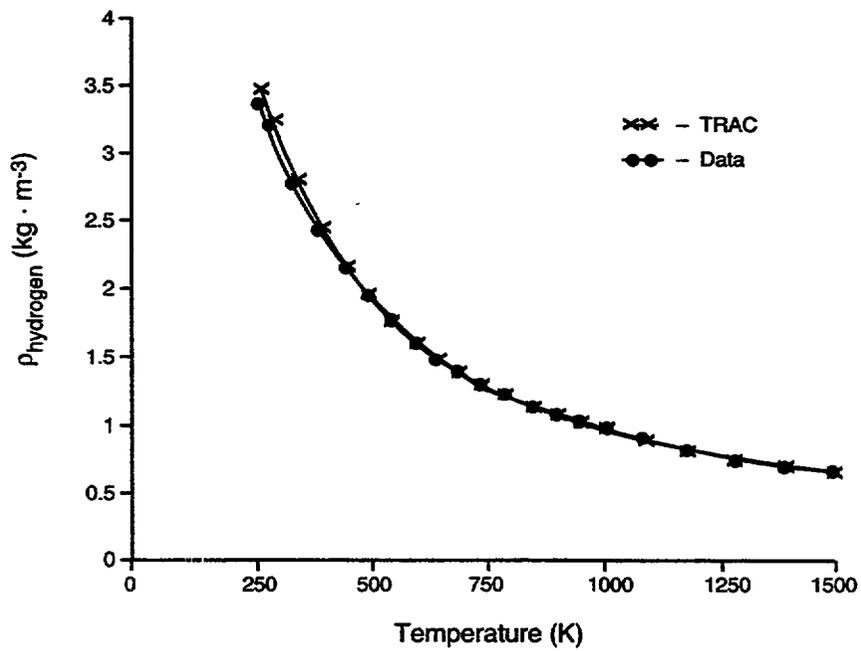


Fig. K-23. Density vs. temperature for hydrogen at 4 MPa.

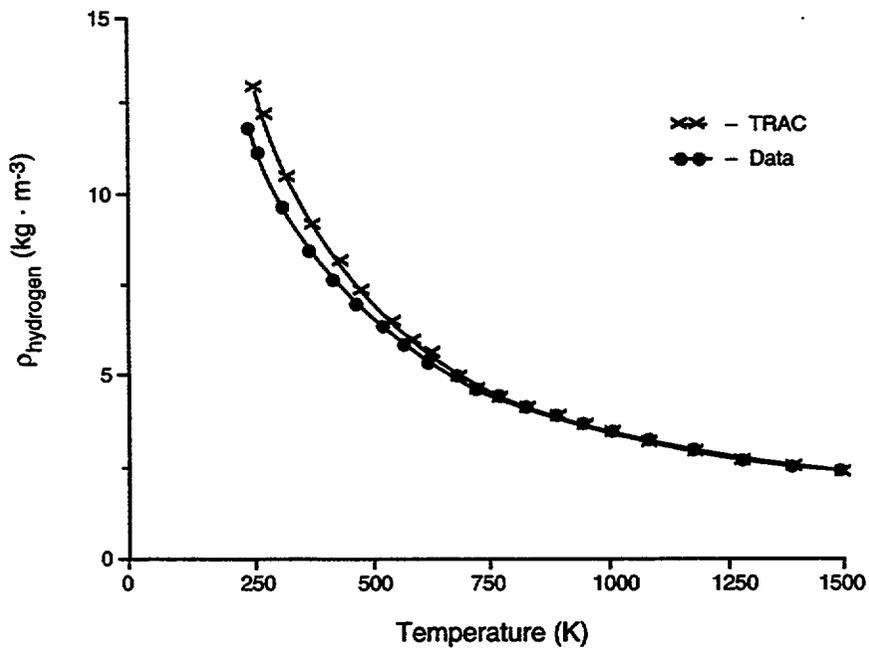


Fig. K-24. Density vs. temperature for hydrogen at 15 MPa.

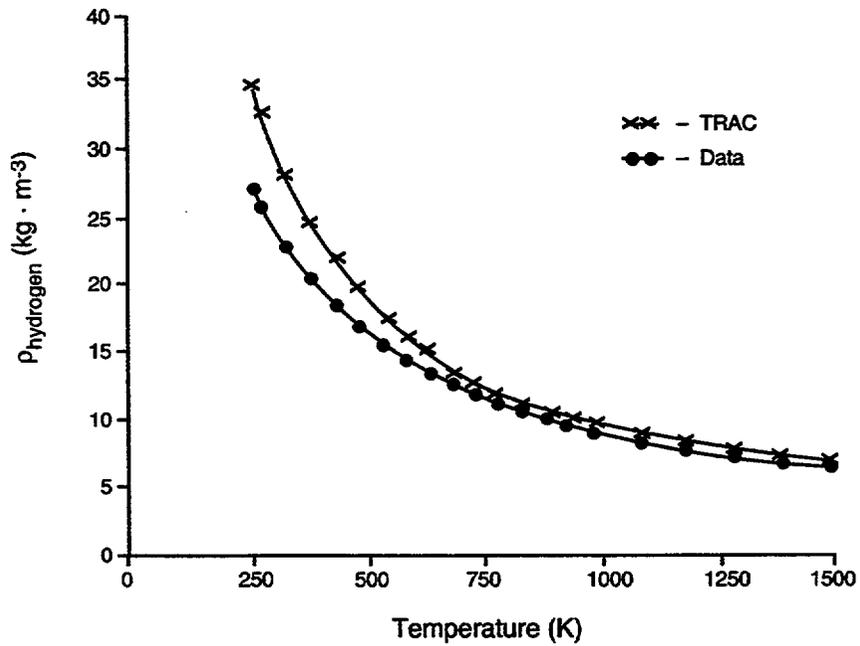


Fig. K-25. Density vs. temperature for hydrogen at 40 MPa.

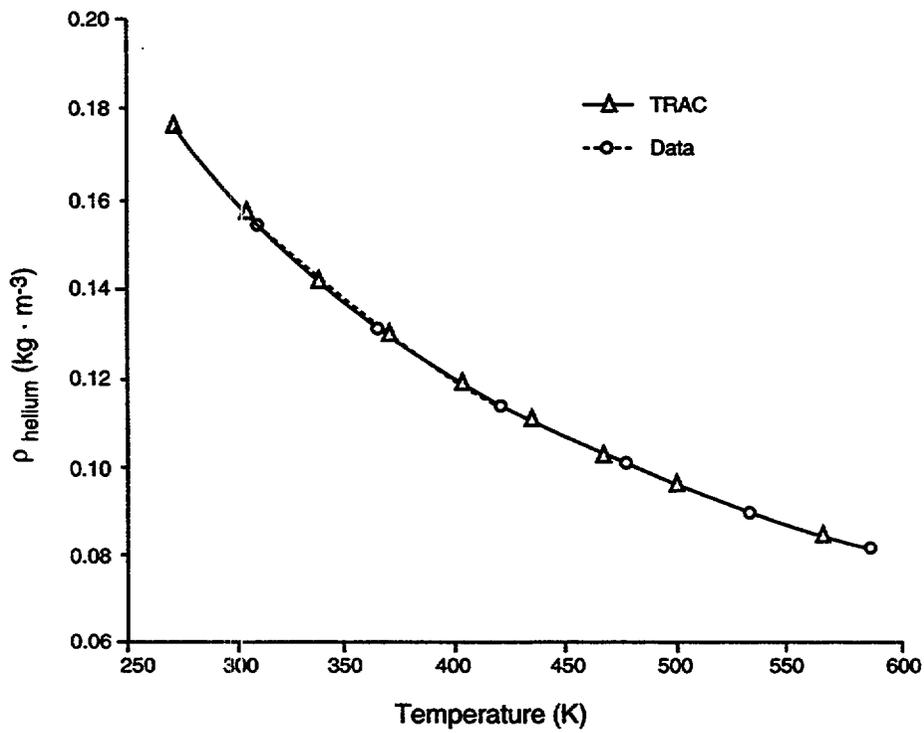


Fig. K-26. Density vs. temperature for helium at 100 kPa.

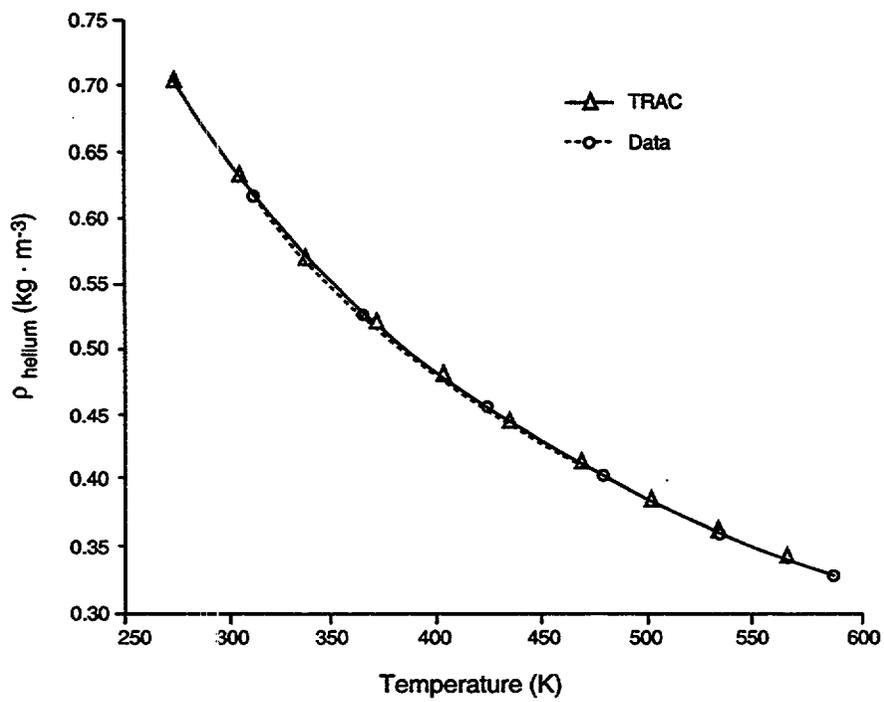


Fig. K-27. Density vs. temperature for helium at 400 kPa.

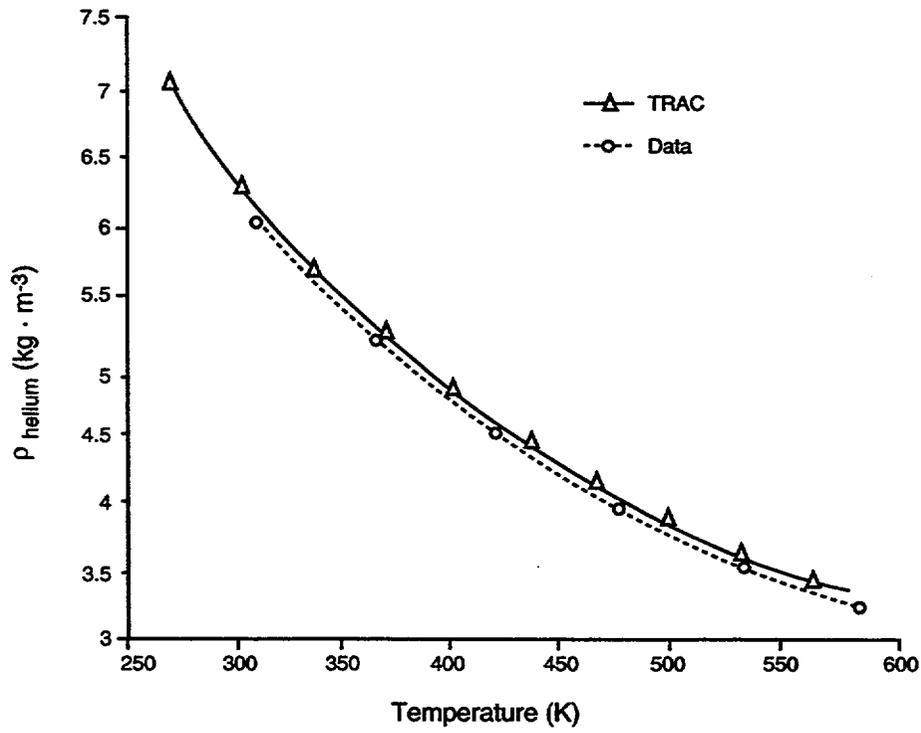


Fig. K-28. Density vs. temperature for helium at 4 MPa.

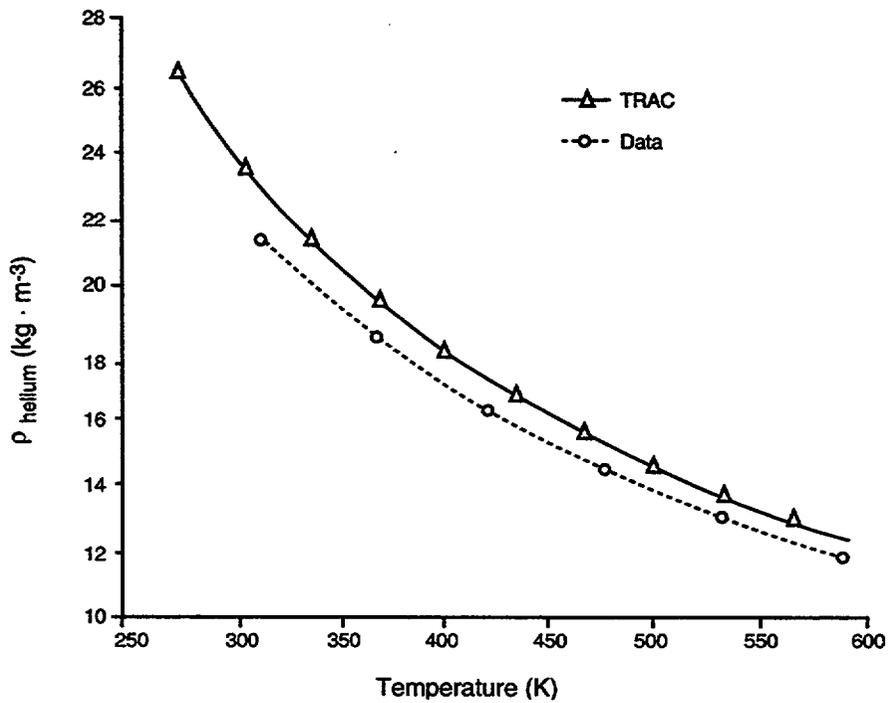
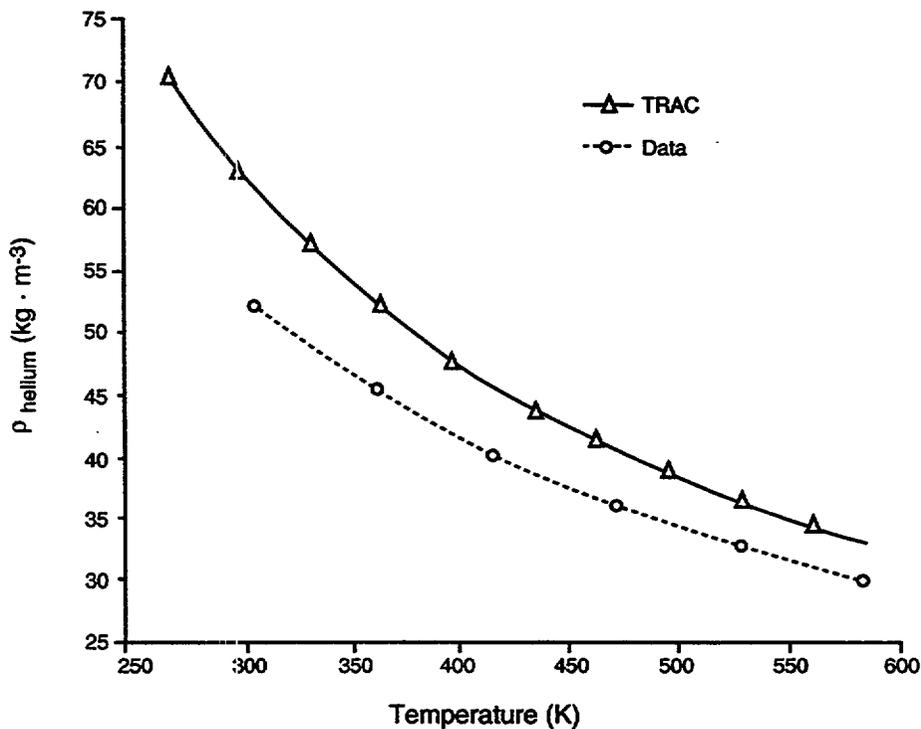


Fig. K-29. Density vs. temperature for helium at 15 Mpa.



P

Fig. K-30. Density vs. temperature for helium at 40 MPa.

We use an exponential approximation with respect to temperature for the thermal conductivity. For air (Fig. K-31.), the percentage error is no more than 5% for temperatures less than 900 K. Similarly, the error for the hydrogen value at 100 kPa is no higher than 5% for temperatures below 860 K (Figs. K-32. through K-36.). The data for the thermal conductivity of air and hydrogen are taken from Ref. K-4. Data for helium thermal conductivity come from Ref. K-8. The error for helium is less than 5% for temperatures below 850 K (Fig. K-37.). The temperature range of these noncondensable gases of greater than 5% accuracy is suitable for most transients of interest in PWRs.

K.1.5. Conclusions

The code uses a constant for the specific heat of the gases when these values are really temperature dependent. If higher precision were desired, we could incorporate a polynomial fit or tables to specify these values. We feel, however, that a constant approximation for the specific heat yields satisfactory results for most transients of interest.

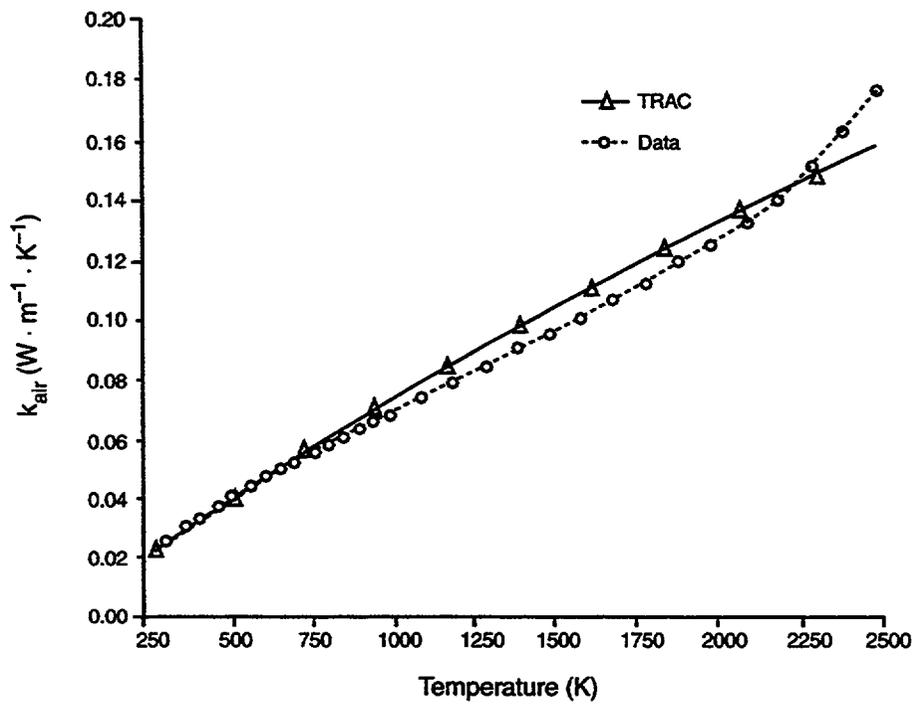


Fig. K-31. Thermal conductivity vs. temperature for air at 100 kPa.

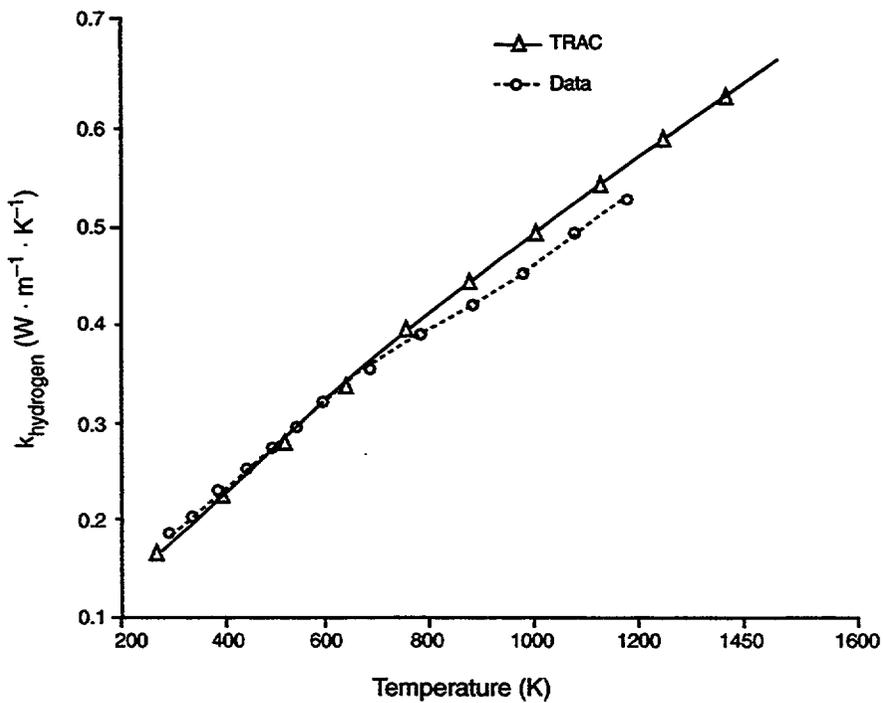


Fig. K-32. Thermal conductivity vs. temperature for hydrogen at 100 kPa.

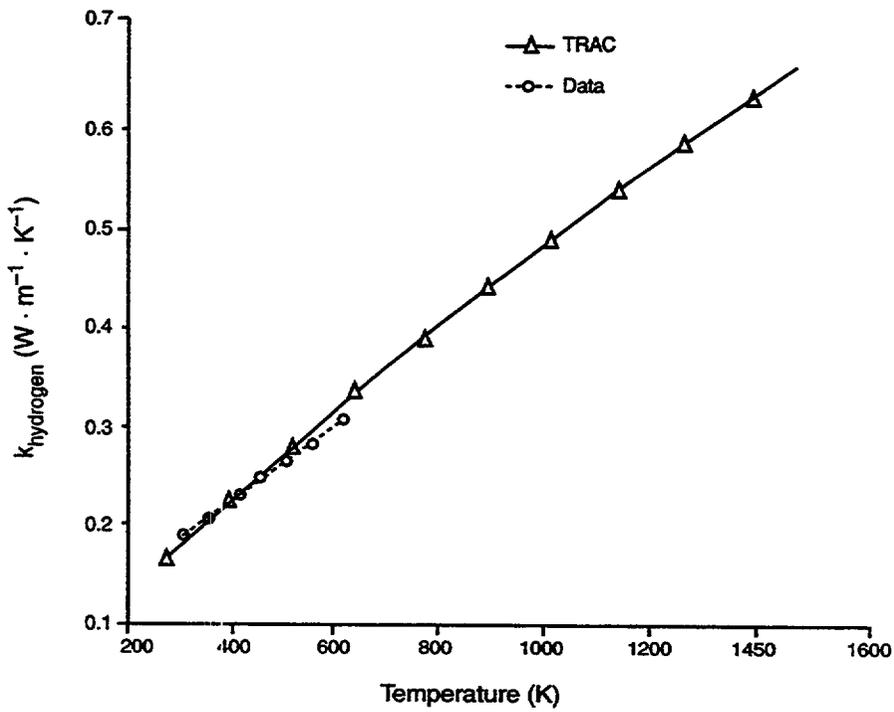


Fig. K-33. Thermal conductivity vs. temperature for hydrogen at 1 MPa.

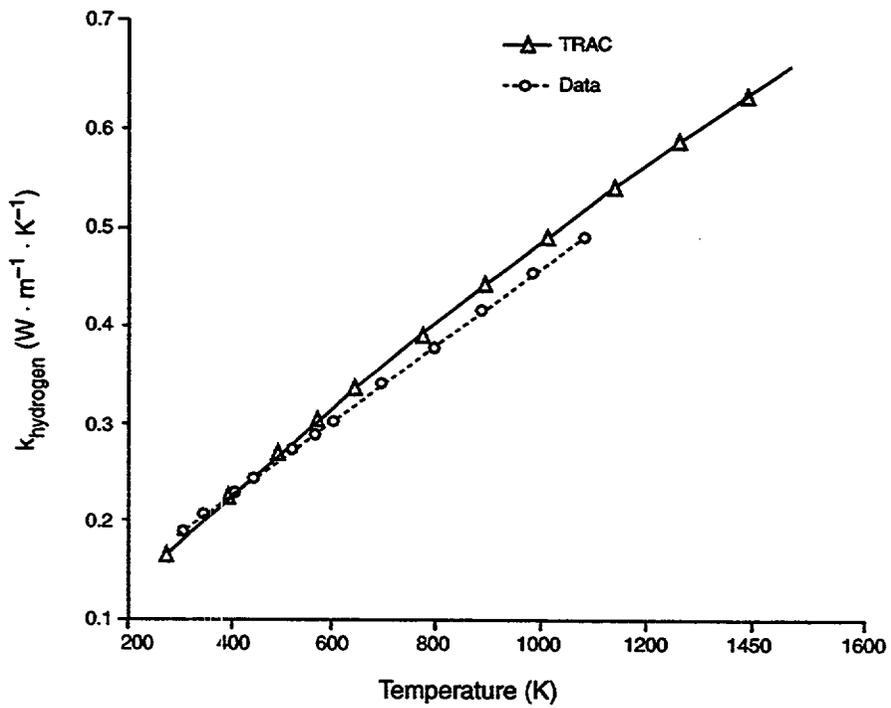


Fig. K-34. Thermal conductivity vs. temperature for hydrogen at 4 MPa.

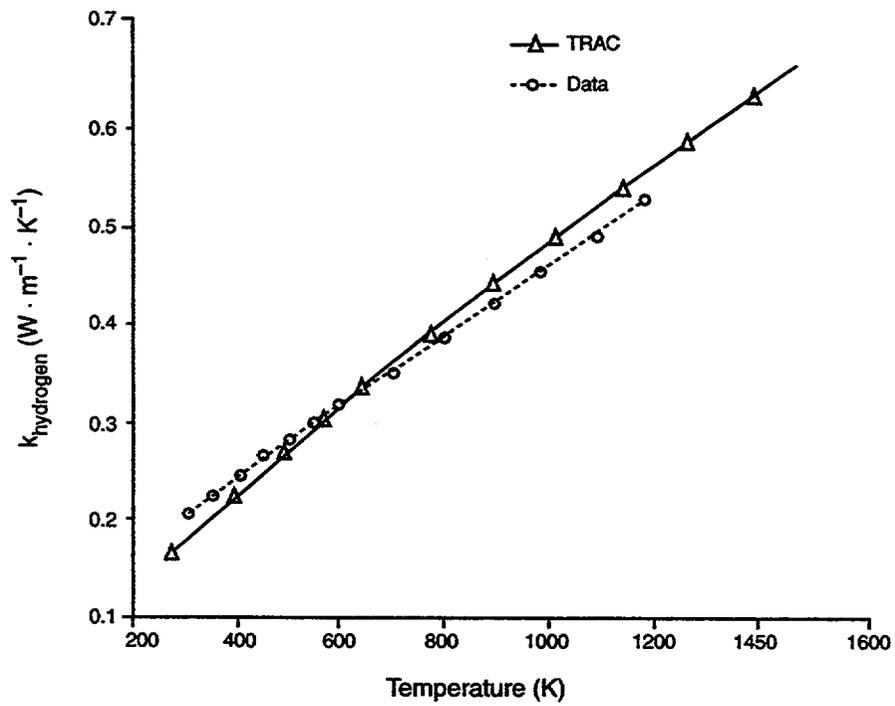


Fig. K-35. Thermal conductivity vs. temperature for hydrogen at 15 MPa.

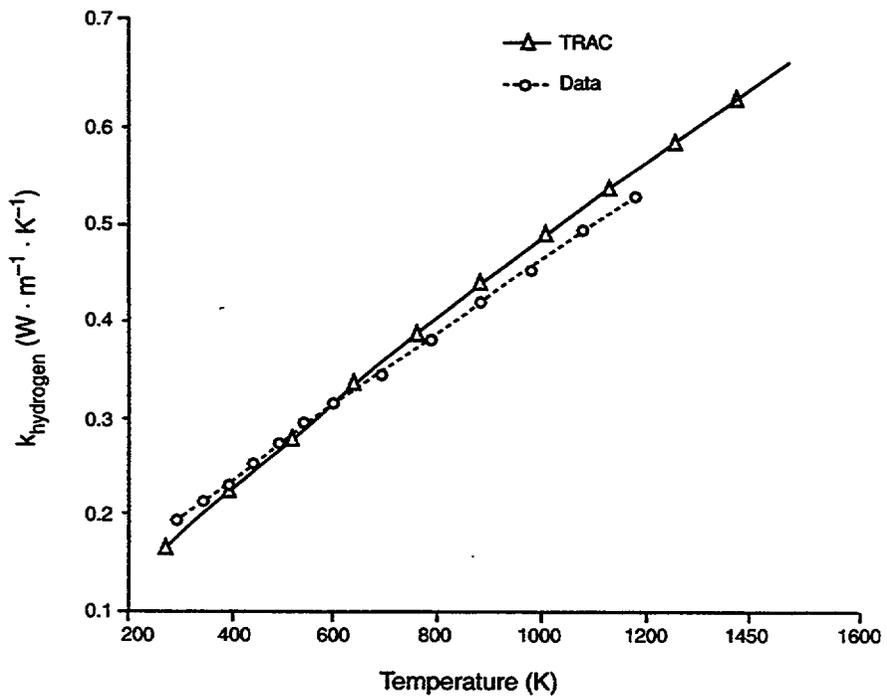


Fig. K-36. Thermal conductivity vs. temperature for hydrogen at 40 MPa.

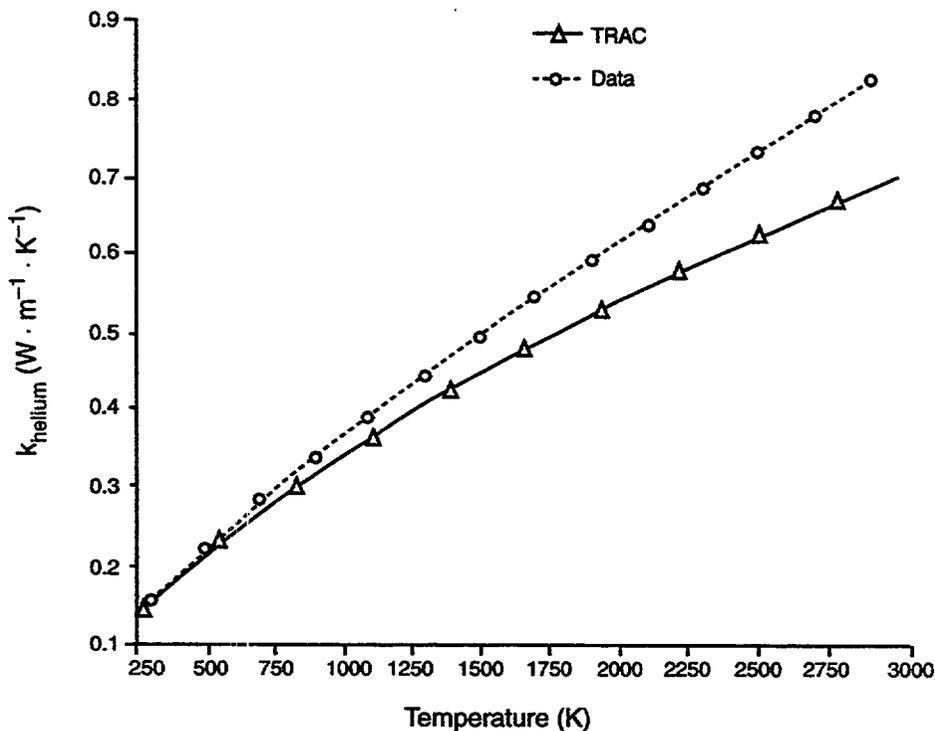


Fig. K-37. Thermal conductivity vs. temperature for helium at constant pressure.

K.2. Liquid Solute

TRAC has an extra mass-continuity equation for tracking a liquid solute (orthoboric acid or a user-defined solute) through all hydro components [see Eqs. (2-17) and (2-42) through (2-44)]. If the liquid solute is orthoboric acid, its concentration in the reactor-core region can be used to determine its neutronic reactivity-feedback effect on the point-reactor kinetics solution for the reactor-core power. The liquid solute has no other effect on the liquid-field processes and properties. The equations solved in TRAC require a correlation for the maximum solubility of the solute material in liquid water.

Because there is no hydraulic feedback effect from the liquid solute on the fluid dynamic solution, the solute mass provides the capability to investigate numerical-diffusion effects and the movement and subsequent distribution of injected coolant.

K.2.1. Model Description

The solubility model for the solute is based on a simple linear correlation between the solute concentration and the liquid temperature. This model is appropriate for dilute solutions of orthoboric acid. The solubility model was developed based on information in Ref. K-9. (p. B-84).

TRAC simulates orthoboric acid with its liquid solute model because of boron's useful design characteristics. Boron has a thermal-neutron-capture cross section of 755 barns that can be used to introduce negative neutronic reactivity to control the reactor-core power. Its reference concentration molality of 0.32 allows appropriate amounts to be dissolved in the liquid-water coolant. Orthoboric acid is also physically and chemically very stable over its range of application (Ref. K-10, p. 218). The default solute in TRAC is orthoboric acid, which can be used in the reactivity-feedback calculation. If the user wishes to use a solute other than orthoboric acid, he can do so by inputting new solubility specifications (see NAMELIST variable ISOLCN and the solubility-parameters card following the NAMELIST-variable input).

TRAC calculates the solute concentration in subroutine BKSSTB for 1D components, in subroutine BKSPLN for the plenum component, and in subroutines FF3D and STBME3 for the 3D vessel component. The evaluation in each of these subroutines has two steps. The first step calculates the solute density and the second step calculates the amount of plated-out solute as well as the final solute concentration in each of the component's mesh cells for the time step.

TRAC calculates the solute density with Eq. (2-17). This equation calculates the solute density directly, with no regard for the solubility of orthoboric acid or for the possible occurrence of plating out. At this point, TRAC assumes infinite solubility of the material already in solution. In doing this calculation for any given mesh cell, TRAC considers each interface with an adjoining mesh cell. The amount of solute mass entering through a mesh-cell interface is defined by

$$m_{\text{solute}} = V_{\ell} \cdot \rho_{\ell} \cdot m \cdot \Delta t \cdot (1 - \alpha) \cdot A \quad , \quad (\text{K-13})$$

where ρ_{ℓ} is the donor-cell liquid density, m is the ratio of donor-cell solute mass to liquid mass, and A is the flow area between the mesh cells.

The convected solute mass is evaluated implicitly by its stabilizer mass equation. In 3D vessel components when NSTAB = 0, the SETS3D stabilizer equations are not evaluated and the convected solute mass is evaluated explicitly by a basic form of its 3D mass equation.

The second set of calculations computes the amount of plated-out solute and obtains the final liquid-solute-concentration ratio. If the cell has voided completely during the time step, all solute in the cell must have either convected out or plated out, and the liquid solute concentration is zero. If the cell has not voided completely (that is, if the cell still contains some liquid coolant), TRAC calls function CONCF, which determines the maximum solubility of orthoboric acid as a function of temperature. This solubility value is known at 303 K and 373 K (Ref. K-9, p. B-84). For temperatures between these two values, TRAC determines the solubility by linear interpolation. For temperatures less than 303 K, TRAC uses the value at 303 K; for temperatures greater than 373 K, TRAC uses the value at 373 K. The maximum possible solute density is the water density times the maximum solubility determined by CONCF. If the new solute density exceeds

this maximum, the code assumes the excess has plated out, and the solute density becomes the maximum value. If the new solute density is less than the maximum, the code assumes that plated-out solute, if it exists in the cell, reenters solution at an infinite rate to the extent that either all the plated-out solute dissolves or the solution becomes saturated. TRAC uses Eqs. (2-43) and (2-44) to calculate the final solute density and the amount plated out. TRAC then finds the new concentration ratio by dividing the new solute density by the coolant density. Plated-out solute remains in the cell until it redissolves.

If a solute other than orthoboric acid is defined, the user must input new maximum and minimum temperatures, as well as solubility values for both temperatures. TRAC calculates the maximum solubility using linear interpolation.

The boron concentration in orthoboric acid is one of four parameters that affect the neutronic reactivity of the reactor core. TRAC calculates their core-region averaged values in subroutine RFDBK. (For a more complete description of reactivity feedback, see Appendix M, Section M.2.5.) When calculating reactivity feedback from boron, TRAC takes into account all forms of boron in the reactor core,

1. dissolved orthoboric acid in the liquid coolant,
2. orthoboric acid plated on core structure,
3. borosilicate glass in burnable-poison pins, and
4. boron oxide in control-rod-cluster pins.

The solubility model for orthoboric acid affects items 1 and 2.

TRAC makes the following assumptions when implementing the solute-tracking feature of the code:

1. The presence of solute in the coolant does not affect any of the liquid properties of the coolant, nor does it affect interfacial or wall heat transfer in any way (this assumption makes the solute concentration a good marker with which the fluid from a particular source can be followed).
2. While the code does allow for plated-out solute, it does not take into account the effects of such solute buildup except through neutronic reactivity feedback when the solute is orthoboric acid. Plated-out solute on structure surfaces could affect wall heat transfer and surface friction (Ref. K-10.).
3. Plating out occurs instantaneously if the concentration exceeds the maximum solubility, such as when liquid coolant boils or flashes. Similarly, the plated-out material dissolves at an infinite rate. No allowance is made for a supersaturated orthoboric solution.
4. The code assumes that the solubility is a function of temperature and is not pressure dependent.

5. The maximum-solubility temperature dependence is assumed to be linear and defined over a limited temperature range.

Reference K-9. (p. B-84) indicates that the maximum solubility of orthoboric acid at 303 K is 6.35 g/100 cm³ of water; at 373 K, 27.6 g/100 cm³ of water. We interpreted this to mean respective maximum solute concentrations of 0.0635 and 0.276 kg solute/kg water at 303 K and 373 K; this interpretation is consistent with the data in Ref. K-11. (p. 149).

K.2.2. Assessment

TRAC will use the above model to calculate the solubility and concentration of orthoboric acid for temperatures from 273.15 K, the triple-point temperature, to 647.3 K, the critical-point temperature of water. Figure K-38. shows a comparison of the TRAC approximation for solubility with the data from Ref. K-11. (pp. 149-150); the data in Ref. K-10. (p. 221) are similar. The linear approximation for the solubility of boric acid compares well with the data in the temperature range bounded by the two temperatures. On the low end of the temperature scale where we approximate the solubility at 273 K by the solubility at 303, there is a 134% error. Reference K-11. indicates that the solubility is 1 kg/kg water at 444 K, where there is a -72% error associated with the TRAC approximation. The solubility curve based on Ref. K-11. becomes very steep as the temperature approaches 444 K.

If the solubility limits are exceeded, we believe that the typical result would be the creation of a supersaturated solution and/or a suspended precipitate that would not be plated out on the structure surface.

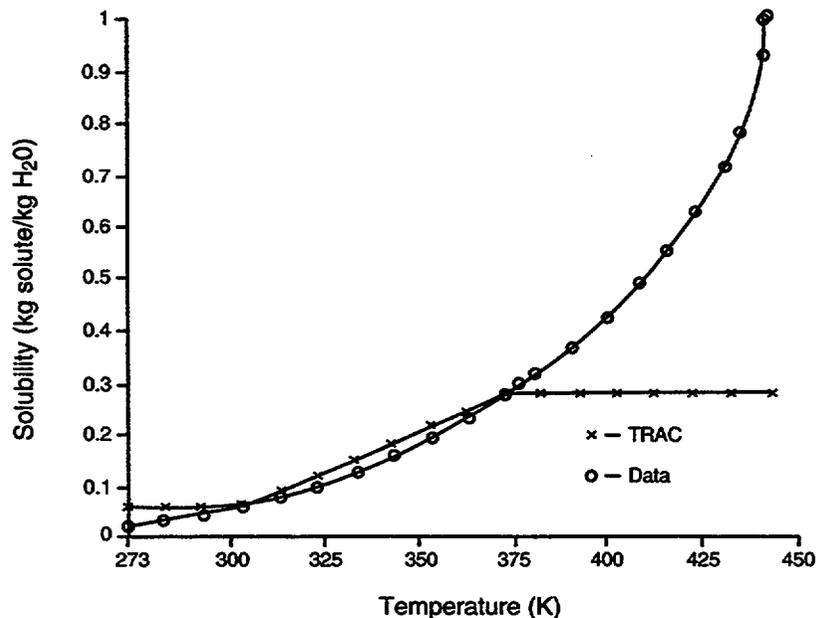


Fig. K-38. Orthoboric-acid solubility vs. liquid temperature.

K.2.3. Summary and Conclusions

We are now aware of data that would permit the construction of a solubility curve in TRAC that is valid over a much wider range of temperatures, although there is still a question of how to treat solubility in the metastable range above 442 K. We believe that such a model should be incorporated. For now, the user can input a linearly dependent maximum-solubility curve fit to data over the temperature range of interest if the default maximum-solubility curve is inappropriate. We do not believe that it is important to incorporate rates into the dissolution/plating processes because most transients in PWRs will not approach the solubility limits.

REFERENCES

- K-1. J. P. Holman, *Heat Transfer* (McGraw-Hill Book Co., Inc., New York, 1981).
- K-2. Frank M. White, *Fluid Mechanics* (McGraw-Hill Book Co., Inc., New York, 1979).
- K-3. Kuzman Raznjevic, *Handbook of Thermodynamic Tables and Charts* (McGraw-Hill Book Co., Inc., New York, 1976).
- K-4. J. M. Sicilian and R. P. Harper, *Heavy-Water Properties for the Transient Reactor Analysis Code*, Flow Science, Inc. report FSI-85-Q6-1 (1985).
- K-5. *Physical and Thermodynamic Properties of Helium* (Whittaker Controls, Los Angeles, 1957).
- K-6. Nissim Marshall, *Gas Encyclopaedia* (Elsevier Scientific Publishing Company, Amsterdam, 1976).
- K-7. N. B. Vargaftik, *Tables on the Thermophysical Properties of Liquids and Gases, in Normal and Dissociated States*, 2nd ed. (John Wiley and Sons, New York, 1975).
- K-8. *CRC Handbook of Chemistry and Physics*, 63rd ed. (The CRC Press, Inc., Boca Raton, Florida, 1982).
- K-9. *CRC Handbook of Chemistry and Physics*, 62nd ed. (The CRC Press, Inc., Boca Raton, Florida, 1981).
- K-10. Paul Cohen, *Water Coolant Technology of Power Reactors* (Gordon and Breach Science Publishers, New York, 1969).
- K-11. Leopold Gmelin, *Gmelins Handbuch der Anorganischen Chemie*, Supplement to Volume 1, System Number 13 (Boron) (Verlag Chemie, GmbH., Weinheim/Bergstrasse, 1954).

K-12. E. A. Haytcher and J. C. Lin, "Modification to Helium Properties in the TRAC-PF1/MOD2 Code," Los Alamos National Laboratory report LA-N7-1 (December 1992).

APPENDIX L

ADDITIONAL FUEL ROD MODELS

The nomenclature applicable to this appendix follows.

NOMENCLATURE

F :	view factor
$h_{contact}$:	heat-transfer coefficient for fuel-cladding interfacial contact
h_{gap} :	gap heat-transfer coefficient
h_{gas} :	heat-transfer coefficient for gap-gas conductance
$h_{radiation}$:	heat-transfer coefficient for fuel-cladding thermal radiation
k_{gas} :	gap-gas thermal conductivity
m'_{Zr} :	mass per unit length of zirconium
q'''_{mw} :	heat source from metal-water reaction
r :	radius
R_c, R_i :	cladding inner radius
R_f :	solid-fuel radius (also total fuel radius)
R^*_f :	fuel radius, including cracked fuel
R_o :	cladding outer radius
$R_{f,new}$:	solid-fuel radius after thermal expansion
$R_{i,new}$:	cladding inner radius after thermal expansion
t :	time
T :	temperature
T_{ci} :	cladding inside temperature
T_{cf} :	fuel centerline temperature
T_{co} :	cladding outside temperature
T_{REF} :	reference temperature (298.0 K)
T_{surf} :	fuel surface temperature
u :	radial displacement
α :	linear thermal-expansion coefficient
Δr_{gap} :	fuel-cladding radial gas gap
Δt :	time step
ΔT :	difference between the actual temperature and T_{REF}
δ :	cracked fuel thickness
δ_0 :	initial undeformed radial thickness of the crack fuel

- δ_r : factor that includes the mean surface roughness of the fuel and the cladding plus the temperature jump distances
- ε : emissivity
- ν : Poisson's ratio
- ρ : density
- ρ_{Zr} : density of zirconium metal
- ρ_{ZrO_2} : density of zirconium oxide
- σ : Stefan-Boltzmann constant
- τ : total oxygen consumed

Subscripts

- c : cladding
- f : fuel

L.1. Fuel-Cladding Gap Conductance

Subroutine GAPHT calculates the gap heat-transfer coefficient (HTC), h_{gap} , as a function of three components: gap-gas conductance, fuel-cladding interfacial contact, and fuel-cladding thermal radiation. The superposition of these three components gives

$$h_{gap} = h_{gas} + h_{contact} + h_{radiation} \quad (L-1)$$

where

$$h_{gas} = k_{gas} / (\Delta r_{gap} + \delta_r) \quad (L-2)$$

$$h_{radiation} = \frac{\sigma F (T_f^4 - T_c^4)}{(T_f - T_c)} \quad (L-3)$$

and

$$F = \frac{1}{\frac{1}{\varepsilon_f} + \frac{R_f}{R_c} \left(\frac{1}{\varepsilon_c} - 1 \right)} \quad (L-4)$$

where k_{gas} is the gap-gas thermal conductivity, T_f and T_c are the temperatures at the fuel outer surface and cladding inner surface at radii R_f and R_c , respectively, ε_f and ε_c are the corresponding emissivities, and σ is the Stefan-Boltzmann constant. A value of 4.4×10^{-6} m is used for δ_r , which includes the mean surface roughness of the fuel and the cladding plus the temperature jump distances (Refs. L-1 and L-2). The contact

conductance, $h_{contact}$ is zero in the present code. (In the following discussion on thermal expansion, the radii R_f and R_c are called R_f^* and R_r , respectively.)

The fuel-cladding radial gas gap, Δr_{gap} , is found by using the uncoupled, quasi-static approximation (Ref. L-3). In this approximation the mechanical coupling term in the energy equation and the inertial term in the mechanical force balance are omitted. By neglecting these terms, we assume that the fuel-cladding strains minimally affect the temperature distribution and that displacements are instantaneous. Figure L-1. shows the fuel-cladding gap system modeled in three regions: solid fuel, cracked fuel, and cladding. Gap changes are found by calculating the radial displacement of each region caused by thermal expansion.

The calculations for the deformation of a hollow or solid circular cylindrical body of outer radius b , inner radius a (where $a = 0$ for a solid cylinder), and height h are given in Ref. L-3. for the case where the ratio h/b is large compared to unity. Other assumptions are made that the cylindrical surfaces are free of forces, axial displacement is allowed, and the temperature distribution is a function of the radial distance r only. Because the uncoupled, quasi-static approximation is used, the temperature distributions are assumed to be known from the energy balance. The radial displacement u is given by

$$u(r) = \frac{\alpha}{r(1-\nu)} \left[(1+\nu) \int_a^r \Delta T r \, dr + \frac{(1-3\nu)r^2 + a^2(1+\nu)}{b^2 - a^2} \int_a^b \Delta T r \, dr \right] \quad , \quad (L-5)$$

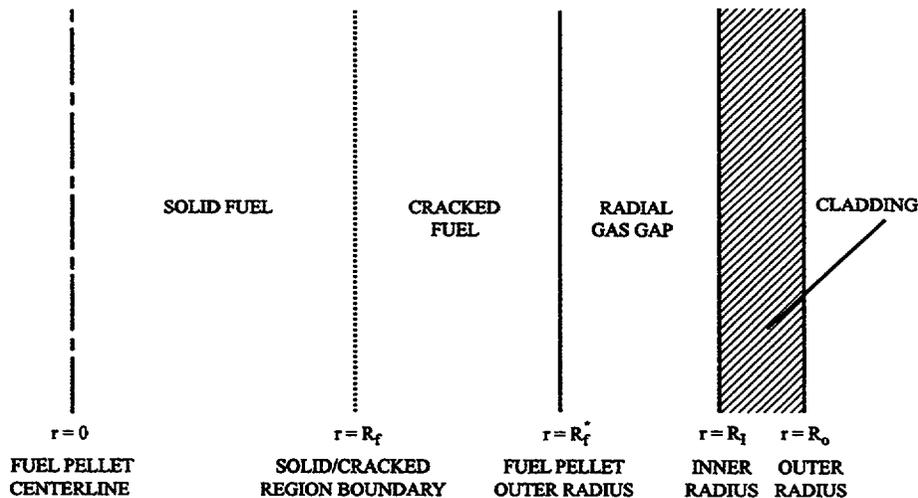


Fig. L-1. Fuel-rod geometry.

where ν is Poisson's ratio, α is the linear thermal-expansion coefficient, and ΔT is $(T - T_{REF})$. For purposes of this calculation, T_{REF} is set to 298.0 K. Equation (L-5) is used to calculate the radial displacement of the cladding inner radius and the solid-fuel radius, $r = R_I$ and $r = R_f$, respectively. If radial symmetry is assumed (so that $\nu = 0$), the results become

$$u(R_I) = \frac{2\alpha_c R_I}{R_o^2 - R_I^2} \int_{R_I}^{R_o} \Delta T_c r \, dr \quad (L-6)$$

and

$$u(R_f) = \frac{2\alpha_f R_f}{R_f^2} \int_0^{R_f} \Delta T_f r \, dr. \quad (L-7)$$

The cladding inner radius and the solid-fuel radius after thermal expansion are

$$R_{I, new} = R_I + u(R_I) \quad (L-8)$$

and

$$R_{f, new} = R_f + u(R_f). \quad (L-9)$$

The following equation is used for the cracked-fuel thickness,

$$\delta = \delta_o \left[1 + \frac{\alpha_f R_f^*}{R_f^* - R_f} \int_{R_f}^{R_f^*} \Delta T_f \, dr \right], \quad (L-10)$$

where δ_o is the initial, undeformed radial thickness of the cracked fuel,

$$\delta_o = R_f^* - R_f. \quad (L-11)$$

A parabolic radial temperature distribution is assumed across the fuel pellet,

$$T_f = T_{cl} + (T_{surf} - T_{cl})(r^2/R_f^2), \quad (L-12)$$

where T_{cl} is the fuel centerline temperature and T_{surf} is the fuel surface temperature. A linear temperature profile is assumed across the cladding,

$$T_c = (T_{co} - T_{cl})(r - R_I)/(R_o - R_I) + T_{cl}, \quad (L-13)$$

where T_{co} and T_{cl} are the cladding outside and inside temperatures, respectively.

The radial gap width after thermal expansion is

$$\text{gap width} = R_{I, new} - (R_{f, new} + \delta) \quad (L-14)$$

or

$$\begin{aligned} \text{gap width} = & (R_I - R_f) + \frac{2\alpha_c R_I}{R_o^2 - R_I^2} \int_{R_I}^{R_o} \Delta T_c r dr - \frac{2\alpha_f}{R_f} \int_0^{R_f} \Delta T_f r dr \\ & - \delta_o \left[1 + \frac{\alpha_f}{R_f^* - R_f} \int_{R_f}^{R_f^*} \Delta T_f dr \right]. \end{aligned} \quad (L-15)$$

Subroutine DELTAR evaluates the fuel-cladding radial spacing.

L.2. Metal-Water Reaction

When Zircaloy reaches a sufficiently high temperature in a steam environment, an exothermic reaction may occur that influences the peak cladding temperatures. This exothermic reaction, called the zirconium-steam reaction, is given by



With a sufficient steam supply, the following reaction-rate equation (Refs. L-4 and L-5) is assumed valid:

$$\tau \frac{d\tau}{dt} = \eta_1 \exp\left(-\frac{\eta_2}{T}\right), \quad (L-17)$$

where

τ = total oxygen consumed (kg/m^2), $\eta_1 = 16.8 \text{ kg}^2/\text{m}^4\text{s}$, and $\eta_2 = 2.007 \times 10^4 \text{ K}$.

The kinetic parameter τ is converted to an effective zirconium-oxide layer thickness according to

$$1.5(R_o - r) = \tau / (0.26 \rho_{\text{ZrO}_2}), \quad (L-18)$$

where r = reacting surface radius (m), R_o = cladding outer radius (m), and ρ_{ZrO_2} = density of zirconium oxide (kg/m³).

Equation (L-18) is based on a reacted-material volume expansion of 50% in the radial direction. This assumption leads to $\rho_{ZrO_2} = 0.90\rho_{Zr}$. Equation (L-18) allows Eq. (L-17) to be rewritten as

$$\tau \frac{d\tau}{dt} = -\eta_3(R_o - r) \frac{dr}{dt},$$

where

$$\eta_3 = (0.351\rho_{Zr})^2.$$

The method outlined in Ref. L-4 is used to calculate the zirconium-oxide penetration depth and associated heat source. The mass per unit length of zirconium (m'_{Zr}) consumed by the reaction in one time step is

$$(m'_{Zr}) = \pi\rho_{Zr}[(r^n)^2 - (r^{n+1})^2]. \quad (L-19)$$

Equation (L-17) is used to calculate r^{n+1} , yielding

$$r^{n+1} = R_o - [(R_o - r^n)^2 + 2(\eta_1/\eta_3)\Delta t \exp(-\eta_2/T)]^{1/2}. \quad (L-20)$$

If a single-region cladding is assumed, the heat source (q'''_{mw}) added to the conduction equation is

$$q'''_{mw} = 6.45 \times 10^6 m'_{Zr} [\Delta t (R_o^2 - R_I^2) \pi]^{-1}, \quad (L-21)$$

where R_I is the inner cladding radius and 6.45×10^6 J/kg corresponds to the energy released per kilogram of oxidized zirconium.

The metal-water reaction is calculated only at locations that correspond to hydrodynamic-cell boundaries. The effect of the metal-water reaction is included in the intermediate nodes (fine mesh) by linear interpolation of the effect along the length of the cell. The error produced by this method of accounting for the metal-water reaction is small except in a hydrodynamic cell where quenching is occurring. At the bottom of such a cell the rod temperature is low, and the metal-water reaction has no effect. At the top of the cell the metal-water reaction is calculated correctly. When TRAC interpolates in the cell the effect of the metal-water reaction is underestimated. If the rod temperatures exceed 1273 K (the onset of the metal-water reaction), the user should

watch for this situation; if the peak cladding-temperature location is in the same hydrodynamic cell as the quench front, the temperature will be under predicted.

REFERENCES

- L-1. L. S. Tong and J. Weisman, *Thermal Analysis of Pressurized Water Reactors*, 2nd ed. (American Nuclear Society, La Grange Park, Illinois, 1979).
- L-2. P. E. MacDonald and J. Weisman, "Effect of Pellet Cracking on Light Water Reactor Fuel Temperatures," *Nuclear Technology* 31, 357-366 (1976).
- L-3. B. A. Boley and J. H. Weiner, *Theory of Thermal Stresses* (John Wiley and Sons, Inc., New York, 1960).
- L-4. J. V. Cathcart, "Quarterly Progress Report on the Zirconium Metal-Water Oxidation Kinetics Program," Oak Ridge National Laboratory report ORNL/NUREG/TM-41 (August 1976).
- L-5. "MATPRO-Version 11: A Handbook of Materials Properties for Use in the Analysis of Light Water Reactor Fuel Rod Behavior," Idaho National Engineering Laboratory report TREE-1280 (NUREG/CR-0497) (February 1979).

APPENDIX M

REACTOR-CORE POWER MODEL

The following nomenclature applies to this appendix. On various occasions, mnemonics are used that are equivalent to the code's FORTRAN variables instead of the symbols. These mnemonics are also listed in the nomenclature.

NOMENCLATURE

A_i :	horizontal cross-sectional area of fuel-rod node i
a_1, a_2 :	functions of time defined in <u>Eq. (M-27)</u>
a_p, b_i :	linear interpolation parameters for power history defined in <u>Eq. (M-12)</u>
BCR :	control-rod-cluster-pin solute concentration ($\text{kg} \cdot \text{m}^{-3}$)
$BCR0, BCR1$:	polynomial coefficients for BCR defined in <u>Eq. (M-16)</u>
B_m :	core-averaged solute concentration ($\text{kg} \cdot \text{m}^{-3}$)
BPP :	equivalent burnable-poison-pin solute concentration ($\text{kg} \cdot \text{m}^{-3}$)
$BPP0, BPP1$:	polynomial coefficients for BPP defined in <u>Eq. (M-15)</u>
B_r :	ratio of solute mass to liquid-coolant mass (ppm)
C_i :	power of delayed-neutron precursor concentration in group i (W)
$CPOWER$:	relative power density of an average rod
D, D^* :	functions of a_1 and a_2 (Eqs. <u>M-30</u> , <u>M-31</u> , <u>M-39</u> , and <u>M-40</u>)
ED_j :	MeV of decay energy per fission per second for decay-heat group j ($\text{MeV} \cdot \text{fission}^{-1} \cdot \text{s}^{-1}$)
E_j :	effective energy fraction of decay-heat group j
H_j :	energy of decay-heat precursor concentration group j (W · s)
$IDROD$:	input array that provides location of supplemental rod
I :	number of delayed-neutron groups or total number of radial nodes in a fuel rod
J :	number of decay-heat groups or total number of average calculational rods in the core
K :	number of input-specified coarse-mesh boundaries axially
k :	neutron multiplication constant
k_{ex} :	excess reactivity expressed in terms of the neutron multiplication constant ($k - 1$)
t :	a constant
m_i :	fuel or coolant mass if x is T_f or T_c in <u>Eq. (M-14)</u> ; 1 if x is α , B_m or B_r in <u>Eq. (M-14)</u>
m :	arbitrary index

NZPWTB:	normalized power shape
NRDX:	number of fuel rods
P :	thermal power from fission (W)
P_i :	power in mesh cell i
P_1, P_2, P_1^*, P_2^* :	polynomial coefficients for P
P_{eff} :	effective thermal power
POWEXP:	neutron flux distribution parameter
P_{tot} :	average core power
Q_k :	MeV per fission for a given isotope k
$q(t)$:	function of time defined in <u>Eq. (M-27)</u>
q''' :	power density
q'''_{supl} :	supplemental rod power-density distribution
R :	neutronic reactivity, including both programmed reactivity and feedback reactivity
r :	radial
R_i :	polynomial coefficient for R
RDPWR:	relative radial-power density
RPKF:	power-peaking factors
S :	thermal power from an external source of neutrons (W)
S_f :	scale factor to normalize the relative-power distributions
T_c :	core-averaged coolant temperature
T_f :	core-averaged fuel temperature
t :	time (s)
V_i :	volume of mesh cell i
x :	dummy variable
ZPOWR:	relative power density
ZPWTB:	the relative axial-power density in a given rod
α :	core-averaged gas volume fraction
β :	total fraction of delayed neutrons
β_i :	fraction of delayed neutrons in group i
Δk :	change in reactor multiplication factor
Δt :	time-step size
Δz :	elevation distance
Λ :	effective prompt-neutron lifetime (s)
λ_i :	decay constant for delayed-neutron precursors in group i (s^{-1})
λ_j^H :	decay constant for decay-heat group j

ρ :	density
τ :	dummy integration variable
θ :	azimuthal
ε :	error in estimating P
$\varepsilon_1, \varepsilon_2$:	error bounds in estimating P

Subscripts

<i>fdbk</i> :	feedback
<i>i</i> :	delayed-neutron group or cell number or dummy index for fuel-rod node
<i>j</i> :	decay-heat group or dummy index for cell identification
<i>k</i> :	dummy isotope index or dummy index for core level
<i>prog</i> :	programmed

Superscripts

cor:	corrected
est:	estimated
$n - 1, n, n + 1$:	old-, current-, and new-time values

M.1. Partitioning of the Core Power into the Heat-Conduction Mesh

The user must supply as a part of the input for the heat-structure components several arrays that enable the code to partition the heat generation into the finite-difference mesh cells that describe the fuel rods (or fuel-rod simulators). These arrays include *RDPWR*, the relative radial-power density at the radial node positions in the rod, [Fig. M-1.](#); *CPOWR*, the relative power density in each (r, θ) mesh cell in the horizontal plane, [Fig. M-2.\(A\)](#); *ZPWTB*, the relative axial-power density in a given rod, [Fig. M-3.](#); and *RPKF*, the fuel-rod power-peaking factors (relative to the average fuel rod in the horizontal mesh cell) for the additional rods, [Fig. M-2.\(B\)](#). The code assumes that all calculational rods have the same *RDPWR* and *ZPWTB* arrays and that the *CPOWR* and *RPKF* arrays do not vary in the axial direction.

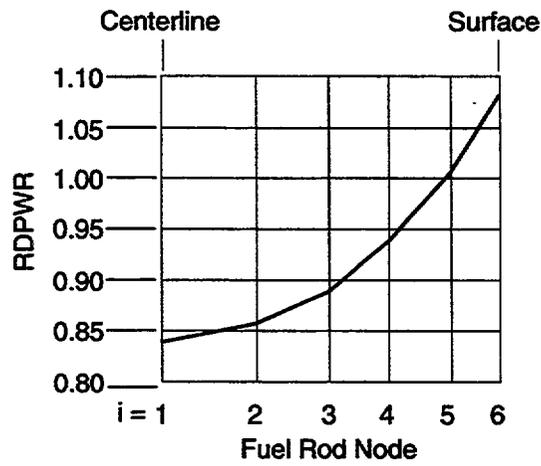
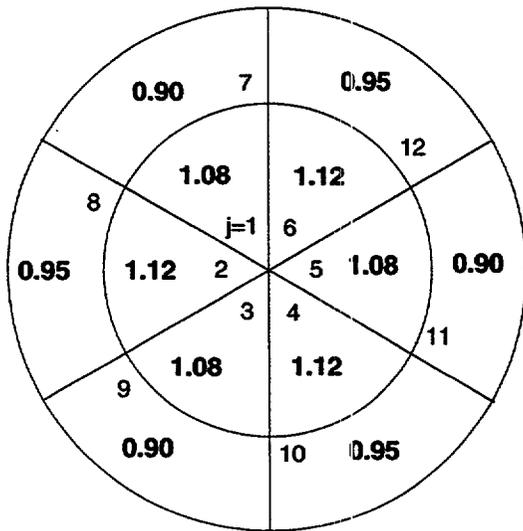


Fig. M-1. Relative radial-power density at radial node positions in the fuel rod.

(A) CPOWER: Relative power density in each core mesh cell for hydro/coupled rod elements



(B) RPKF: Relative power density for supplemental non-hydro/coupled rod elements

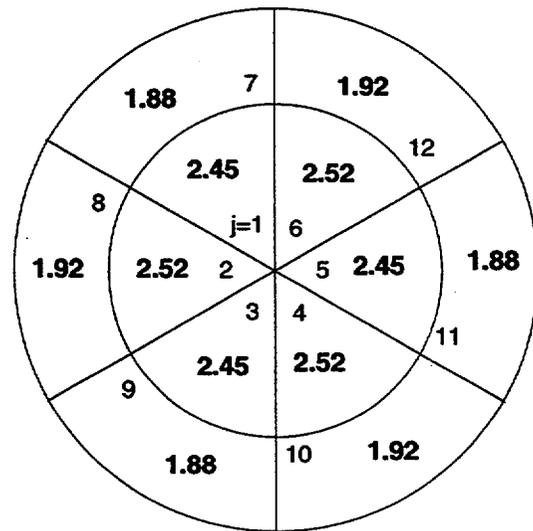


Fig. M-2. Relative power density in each (r, θ) mesh cell in the horizontal plane.

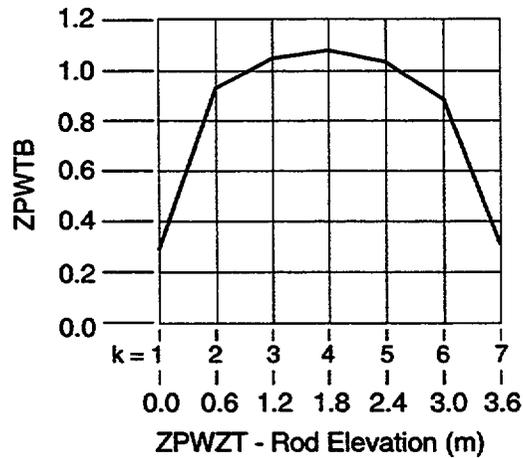


Fig. M-3. Relative axial-power density in a fuel rod.

There are a number of input options available for specifying the axial-power shape as discussed in Section 4.3 of the TRAC-M/F90 User's Manual (Ref. M-5.). The code also permits the user to specify the axial-power shape as a function of an independent variable. This independent variable can be either a signal-variable parameter or a control-block output signal. The purpose of this general definition of the axial-power shape is to permit the code user to vary the axial-power shape during the calculation if desired. An example of this variation is to vary the axial-power shape to account for the changes in power distribution resulting from control-rod insertion, as shown in Fig. M-4.

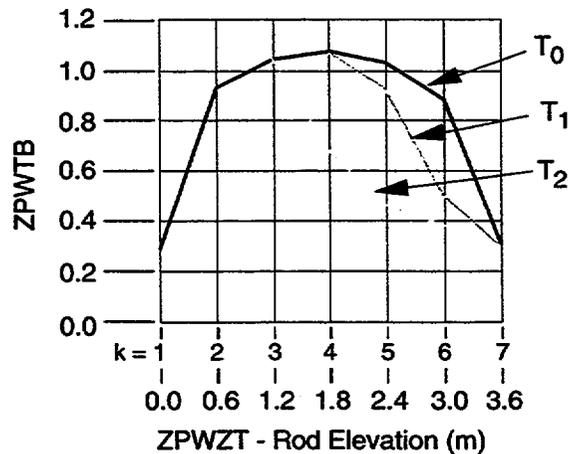


Fig. M-4. Example of multiple entries of axial-power shape to model control-rod insertion as a function of time.

The input-processing subroutine RROD2 normalizes each individual axial-power shape by the following formula:

$$NZPWTB_k = ZPWTB_k \left(\frac{\sum_{i=1}^K \Delta z_i}{\sum_{i=1}^K ZPWTB_i \Delta z_i} \right), \quad (M-1)$$

where $ZPWTB$ is the input power shape and $NZPWTB$ is the resulting normalized power shape. The summations are over the axial height of the core and Δz_i is the elevation distance over which the i^{th} shape element applies. (In general, Δz_i is the distance between the centers of adjacent levels; however, at the bottom, it is half the height of the lowest level inside the core; at the top, half the height of the top level in the core.) $ZPOWR$, the relative power density, is determined by interpolation from the $NZPWTB$ array.

The code computes the actual solution for the fuel-rod temperatures with a call to subroutine CORE3. Subroutine CORE3 controls the thermal analysis of all calculational rods in the core and calls subroutine FROD for each rod. Subroutine FROD calculates the temperature distribution for an individual rod with a call to subroutine RODHT. The partitioning of the core power is set up initially in subroutine IROD, with partitioning carried out in subroutine RODHT.

The power density q''' in fuel-rod node i in cell j in core level k is

$$q'''_{ijk} = S_f P_{tot} RDPWR_i CPOWR_j ZPOWR_k, \quad (M-2)$$

where P_{tot} is the average core power over the time step, $RDPWR_i$ is the relative power density in fuel node i , $CPOWR_j$ is the relative power of the average rod in cell j , $ZPOWR_k$ is the relative power density at core elevation k , and S_f is the scale factor that normalizes the three input relative-power distributions to a core volume-averaged value of unity. Subroutine CORE3 sets P_{tot} to be

$$P_{tot} = \frac{P^n + P^{n+1}}{2}, \quad (M-3)$$

where P is the total core power, and the superscripts n and $n + 1$ indicate the beginning and end of time step, respectively. The following expression defines the scale factor S_f .

$$S_f = 1 / \left[\sum_{i=1}^I \sum_{j=1}^J \sum_{k=1}^K (A_i \cdot RDPWR_i \cdot NRDX_j \cdot CPOWR_j \cdot ZPOWR_k \cdot \Delta z_k) \right], \quad (M-4)$$

where A_i is the horizontal cross-sectional area of fuel-rod node i , and $NRDX_j$ is the number of fuel rods in cell j for which the average fuel-rod condition is to be evaluated.

The summations in Eq. (M-4) are over all nodes in all rods: I is the number of radial nodes in a rod (NODES in the input), J is the number of average calculational rods in the core (from the input, ICRR · NTSX), and K is the number of input-specified coarse-mesh boundaries axially (corresponds to the number of interfaces between levels in the core, ICRU – ICRL + 1).

For the supplemental rod l in cell j (recall that the location of the supplemental rod is specified in input array IDROD), the power density distribution q'''_{supl} is

$$q'''_{supl} = RPFK_j \cdot q'''_{ijk} \text{ provided the value of } IDROD_l = j, \quad (M-5)$$

where $RPFK_j$ is the specified peaking factor of the supplemental rod relative to the average rod in the same cell.

The input defines all of the necessary parameters in Eqs. (M-2) through (M-5) for partitioning P_{tot} throughout the 2D heat-conduction mesh for each calculational rod in the core. For the most part, the variable names used above correspond to those used in the input description; the exception is A_i , which is calculated from input radial-node radii RADRD.

Because the axial-power shape is normalized in the input-processing subroutines, the factor does not need to be shown explicitly in the expression for S_f . Also, S_f contains the grouping $A_i \cdot \Delta z_k$, which is the volume of the (i, k) mesh cell in the rod; this volume times $NRDX_j$ provides the total volume of actual rods in mesh cell (i, k) with the same power density. Therefore, S_f not only normalizes to the total power throughout the core but also converts the power into a volumetric heat-generation rate for the calculational rod.

M.2. Power Evaluation and Reactor Kinetics

The code determines the core power in subroutine RKIN by either of two methods. In the first method, the user specifies power to be a constant or to be defined by an RPWTB power table supplied through input. The table is a function of a system signal-variable parameter or a control-block output-signal parameter. Values between entries in the table are determined by linear interpolation. Power determination can be trip-controlled by evaluating the power table when the controlling trip is ON and by holding the power constant when the trip is OFF. In the second method, the user determines power from the solution of the point-reactor-kinetics equations. These equations specify the time behavior of the core power level with neutronic reactivity (R) [the sum of programmed reactivity (R_{prog}) and feedback reactivity (R_{fwbk})] as the driving parameter. The user inputs programmed reactivity (defining reactivity effects not accounted for by feedback reactivity, such as fuel reactivity and control-rod movement) with the same forms that define power in the first method (as a constant or by a RPWTB table). Subroutine RFDBK evaluates feedback reactivity based on changes in the core-averaged fuel temperature, coolant temperature, gas volume fraction, and solute concentration.

M.2.1. Point-Reactor Kinetics

The point-reactor-kinetics equations (Refs. M-1. and M-2.) are a coupled set of $(I + 1)$ first-order differential equations defining the total fission power P and the delayed-neutron precursor concentrations C_i as a function of time. These equations are given by

$$\frac{dP}{dt} = \frac{R - \beta}{\Lambda} P + \sum_{i=1}^I \lambda_i C_i + \frac{S}{\Lambda(1 - R)} \quad (\text{M-6})$$

and

$$\frac{dC_i}{dt} = -\lambda_i C_i + \frac{\beta_i P}{\Lambda} \quad \text{for } i = 1, 2, \dots, I, \quad (\text{M-7})$$

where

- P = thermal power (W) that results from fission occurring at time t ;
- t = time (s);
- R = neutronic reactivity, (including both programmed reactivity and feedback reactivity) $= (k - 1)/k = R_{prog} + R_{f\bar{a}bki}$;
- k = neutron multiplication constant $= 1 + \Delta k_{prog} + \Delta k_{f\bar{a}bki}$;
- R_{prog} = programmed reactivity $= \Delta k_{prog}/k$;
- $R_{f\bar{a}bki}$ = feedback reactivity $= \Delta k_{f\bar{a}bki}/k$;
- β = total fraction of delayed neutrons $= \sum_{i=1}^I \beta_i$;
- β_i = fraction of delayed neutrons in group i ;
- Λ = effective prompt-neutron lifetime (s);
- λ_i = decay constant for the delayed-neutron precursors in group i (s^{-1});
- C_i = power of the delayed-neutron precursor concentration in group i (W);
- I = number of delayed-neutron groups; and
- S = thermal power (W) from an external source of neutrons in the reactor cores that are producing fission.

The TRAC program uses the Kaganove method (Ref. M-3.) outlined in Section M.2.6. to solve these equations.

The code applies the thermal-power solution to the decay-heat equations

$$\frac{dH_j}{dt} = -\lambda_j^H H_j + E_j P \quad \text{for } j = 1, 2, \dots, J, \quad (\text{M-8})$$

where

- P = solution of Eqs. (M-6) and (M-7),
- H_j = energy of the decay-heat precursor concentration in group j ($W \cdot s$),
- λ_j^H = decay constant for decay-heat group j ,
- E_j = effective energy fraction of decay-heat group j , and
- J = number of decay-heat groups.

After solving each j^{th} equation represented by Eq. (M-8) for the decay-heat group concentration H_j , the code calculates the total thermal power generated in the reactor-core fuel at time t . The total power comprises nuclear fission, fission-product decay, and external sources as shown by

$$P_{eff} = \left(1 - \sum_{j=1}^J E_j \right) P + \sum_{j=1}^J \lambda_j^H H_j + S . \quad (M-9)$$

The code requires the number of delayed-neutron groups, I ; the delayed-neutron parameters, λ_i and β_i ; the number of decay-heat groups, J ; the decay-heat parameters, λ_j^H and E_j ; and either the total fission-power history $P(t)$ for $-t \leq 0$ or the initial delayed-neutron precursor concentrations $C_i(0)$ and decay-heat concentrations, $H_j(0)$.

M.2.2. Default Data for the Delayed-Neutron Groups

If $I \leq 0$ is input (input parameter NDGX), TRAC internally sets I to 6 and defines λ_i and β_i with the values in Table M-1.

TABLE M-1.
Delayed-Neutron Constants

Group i	Decay Constant λ_i (s^{-1})	Neutron Fraction β_i
1	3.87	0.000169
2	1.40	0.000832
3	0.311	0.00264
4	0.115	0.00122
5	0.0317	0.00138
6	0.0127	0.000247

If $I \leq 0$ and no total fission-power history is input, the code assumes that an initial steady-state condition exists to initialize $C_i(0)$ internally. The code sets the time derivative in Eq. (M-7) to zero and calculates the initial values of the C_i from the following:

$$C_i(0) = \frac{\beta_i}{\lambda_i \Lambda} P(0) \text{ for } i = 1, 2, \dots, I, \quad (\text{M-10})$$

where $P(0)$ is the initial power specified through input.

M.2.3. Default Data for the Decay-Heat Groups

If $J \leq 0$ is input, TRAC internally sets J to 69 and defines λ_j^H and ED_j with the values in Table M-2. These decay-heat parameters were obtained from the 1979 ANS decay-heat standard given in Ref. M-4. If the default decay-heat groups are used or if the user inputs 69 or 71 decay-heat groups, the fraction of fission power due to each of the three isotopes, U^{235} , Pu^{239} , and U^{238} , must be input. In addition, in order to convert the MeVs of decay energy per fission to a fraction of the total fission energy, the MeVs per fission for each of the three isotopes, U^{235} , Pu^{239} , and U^{238} , must be input. ED_j in Table M-2 is converted to E_j using the following equation:

$$E_j = ED_j (\lambda_j^H Q_k),$$

where Q_k is the MeV/fission for isotope k .

If $J \leq 0$ and no total fission-power history is input, the code assumes that an initial steady-state condition exists to initialize $H_j(0)$ internally. The code sets the time derivative in Eq. (M-8) to zero and calculates the initial values of the H_j from the following:

$$H_j(0) = \frac{E_j}{\lambda_j^H} P(0) \text{ for } j = 1, 2, \dots, J, \quad (\text{M-11})$$

where $P(0)$ is the initial steady-state power specified through input.

M.2.4. Fission Power History

If the total fission-power history $P(-t)$ is input, TRAC evaluates $C_i(0)$ and $H_j(0)$ from Eqs. (M-7) and (M-8). The $P(-t)$ consists of tabular data pairs (t_l, P_l) for $l = 1, 2, \dots, L$, where $-t_l + 1 < -t < -t_l$. The code assumes that the total fission power varies linearly between data pairs in the power-history table; that is,

$$P(t) = a_l + b_l t \text{ over the time interval } -t_{l+1} \leq t \leq -t_l. \quad (\text{M-12})$$

TABLE M-2.
Decay-Heat Constants

Group	Decay Constant λ_j^H (s ⁻¹)	MeV of decay energy per fission per second ED_j (MeV/fission-s)
for isotope U²³⁵		
1	22.138	6.5057×10^{-01}
2	0.51587	5.1264×10^{-01}
3	0.19594	2.4384×10^{-01}
4	0.10314	1.3850×10^{-01}
5	0.033656	5.5440×10^{-02}
6	0.011681	2.2225×10^{-02}
7	0.0035870	3.3088×10^{-03}
8	0.0013930	9.3015×10^{-04}
9	0.00062630	8.0943×10^{-04}
10	0.00018906	1.9567×10^{-04}
11	5.4988×10^{-05}	3.2535×10^{-05}
12	2.0958×10^{-05}	7.5595×10^{-06}
13	1.0010×10^{-05}	2.5232×10^{-06}
14	2.5438×10^{-06}	4.9948×10^{-07}
15	6.6361×10^{-07}	1.8531×10^{-07}
16	1.2290×10^{-07}	2.6608×10^{-08}
17	2.7213×10^{-08}	2.2398×10^{-09}
18	4.3714×10^{-09}	8.1641×10^{-12}
19	7.5780×10^{-10}	8.7797×10^{-11}
20	2.4786×10^{-10}	2.5131×10^{-14}
21	2.2384×10^{-13}	3.2176×10^{-16}
22	2.4600×10^{-14}	4.5038×10^{-17}
23	1.5699×10^{-14}	7.4791×10^{-17}
for isotope Pu²³⁹		
24	10.02	2.083×10^{-01}
25	0.6433	3.853×10^{-01}
26	0.2186	2.213×10^{-01}
27	0.1004	9.460×10^{-02}
28	0.03728	3.531×10^{-02}

TABLE M-2. (cont)
Decay-Heat Constants

Group	Decay Constant λ_j^H (s⁻¹)	MeV of decay energy per fission per second ED_j (MeV/fission-s)
29	0.01435	2.292×10^{-02}
30	0.004549	3.946×10^{-03}
31	0.001328	1.317×10^{-03}
32	0.0005356	7.052×10^{-04}
33	0.0001730	1.432×10^{-04}
34	4.881×10^{-05}	1.765×10^{-05}
35	2.006×10^{-05}	7.347×10^{-06}
36	8.319×10^{-06}	1.747×10^{-06}
37	2.358×10^{-06}	5.481×10^{-07}
38	6.450×10^{-07}	1.671×10^{-07}
39	1.278×10^{-07}	2.112×10^{-08}
40	2.466×10^{-08}	2.996×10^{-09}
41	9.378×10^{-09}	5.107×10^{-11}
42	7.450×10^{-10}	5.703×10^{-11}
43	2.426×10^{-10}	4.138×10^{-14}
44	2.210×10^{-13}	1.088×10^{-15}
45	2.640×10^{-14}	2.454×10^{-17}
46	1.380×10^{-14}	7.557×10^{-17}
for isotope U²³⁸		
47	3.2881	1.2311
48	0.93805	1.1486
49	0.37073	7.0701×10^{-01}
50	0.11118	2.5209×10^{-01}
51	0.036143	7.1870×10^{-02}
52	0.013272	2.8291×10^{-02}
53	0.0050133	6.8382×10^{-03}
54	0.0013655	1.2322×10^{-03}
55	0.00055158	6.8409×10^{-04}
56	0.00017873	1.6975×10^{-04}
57	4.9032×10^{-05}	2.4182×10^{-05}

TABLE M-2. (cont)
Decay-Heat Constants

Group	Decay Constant λ_j^H (s ⁻¹)	MeV of decay energy per fission per second ED_j (MeV/fission-s)
58	1.7058×10^{-05}	6.6356×10^{-06}
59	7.0465×10^{-06}	1.0075×10^{-06}
60	2.3190×10^{-06}	4.9894×10^{-07}
61	6.4480×10^{-07}	1.6352×10^{-07}
62	1.2649×10^{-07}	2.3355×10^{-08}
63	2.5548×10^{-08}	2.8094×10^{-09}
64	8.4782×10^{-09}	3.6236×10^{-11}
65	7.5130×10^{-10}	6.4577×10^{-11}
66	2.4188×10^{-10}	4.4963×10^{-14}
67	2.2739×10^{-13}	3.6654×10^{-16}
68	9.0536×10^{-14}	5.6293×10^{-17}
69	5.6293×10^{-15}	7.1602×10^{-17}

Substituting Eq. (M-12) into Eq. (M-7) and integrating the resulting equation analytically from $t = -t_L$ [where $C_i(-t_L) = 0$ is assumed] to $t = 0$ gives

$$C_i(0) = \tag{M-13}$$

$$\frac{\beta_i}{\lambda_i \Lambda} \sum_{l=1}^{L-1} \left\{ \left(a_l - \frac{b_l}{\lambda_l} \right) (1 - \exp[-\lambda_i(t_{l+1} - t_l)]) + b_l(t_{l+1} \exp[-\lambda_i(t_{l+1} - t_l)] - t_l) \right\} \exp[-\lambda_i t_l].$$

One obtains a similar expression for $H_j(0)$ by doing the same for Eq. (M-8). The expression for $H_j(0)$ is Eq. (M-13) with replacement of C_i by H_j , λ_i by λ_j^H , β_i by E_j , and Λ by 1. TRAC evaluates these two expressions for $C_i(0)$ and $H_j(0)$ when a fission-power-history table is input.

M.2.5. Reactivity Feedback

Subroutine RFDBK evaluates reactivity feedback. The reactivity-feedback model is based on the assumption that only changes in the core-averaged fuel temperature (T_f), coolant temperature (T_c), gas volume fraction (α), and solute mass (B_m , solute concentration in the coolant volume, or B_r , ratio of solute mass to liquid-coolant mass multiplied by 1×10^6) affect the neutron-multiplication reactivity of the reactor core. The code determines core-averaged values by applying a weighting factor defined as the fuel or coolant density times volume times power to the temperatures and solute ratio and a

weighting factor defined as the volume times power to the gas volume fraction and solute concentration. These factors approximate the product of the local adjoint flux, neutron flux, and volume. The (power)² is the neutron flux times the (fission cross section)². The neutron flux times the (fission cross section)² is approximating the adjoint flux. The adjoint flux is commonly called the importance function as applied in perturbation theory. The (fission cross section)² makes the neutron flux perform like the adjoint flux. This weighting-factor product, i.e., adjoint flux times neutron flux times volume, is from perturbation theory where it is used to spatially weight the change in reaction-rate cross sections to estimate their reactivity change. TRAC approximates this weighting-factor product by the product of the affecting-material mass (density times volume) and the fission-power density distribution raised to a user-specified power *POWEXP*.

The averaging process over the reactor core for the general reactivity-feedback parameter *x* is the following:

$$\bar{x} = \frac{\sum_i x_i \rho_i V_i P_i^{POWEXP}}{\sum_i \rho_i V_i P_i^{POWEXP}}, \quad (M-14)$$

where P_i is the power and V_i is the volume of mesh cell i and the assumptions are over all cells in the reactor core. If x is T_f , T_c , or B_r , then ρ_i is the density of the fuel or coolant in cell i ; if x is α or B_m , then ρ_i is set to 1.0 for all i .

For *POWEXP* = 1.0, the fission reaction-rate cross-section distribution approximates the adjoint-flux distribution. For *POWEXP* = 2.0, when the fission reaction-rate cross section is spatially constant, the neutron flux approximates the adjoint flux (Galerkin approximation). The appropriate value to use for *POWEXP* will depend on the fuel-loading distribution in the reactor core. Keep in mind, however, that two levels of approximation are being made. First, it is assumed that the affecting-material density times volume times the fission-power density distribution raised to the *POWEXP* power is a good approximation for the perturbation-theory product of the adjoint flux, neutron flux, and volume. Second, it is assumed that the theoretical basis for the use of that weighting factor for reaction-rate cross sections can be applied to the reactivity-feedback parameters that TRAC averages over the reactor core. The TRAC-M/F90 User's Manual (Ref. M-5.) suggests that *POWEXP* be set to 2.0 for the case where the core enrichment is radially constant. A *POWEXP*=1 value may be more appropriate for the case where the radial-power distribution is constant across the reactor core. The user has the option of replacing any of the three spatial shapes (fuel-rod radial, core horizontal plane, and core axial) in the fission-power density distribution with a different shape to weight the spatial averaging of reactivity-feedback parameters over the reactor core. These spatial-shape replacements are specified through input. The axial-power distribution can be input as a function of some user-defined independent variable. For example, if that independent variable is the programmed reactivity of the control rods, the effect of the control rods on the axial-power shape can be accounted for when the control rods are

inserted or withdrawn. An example of this is given in Section 4.3 of the TRAC-M/F90 User's Manual (Ref. M-5.).

The user defines a reactivity coefficient for each of the reactivity-feedback independent-variable parameters, $x = T_f, T_c, \alpha,$ and B_m or B_r , by choosing one of the reactivity-coefficient forms in Table M-3. Each reactivity coefficient is defined through input by a table of reactivity-coefficient values that are dependent on all four reactivity-feedback parameters. Each parameter has one or more values specified in defining the table. Defining one value for a parameter corresponds to the reactivity coefficient having no dependence on that reactivity-feedback parameter. Defining two or more values for a parameter gives the reactivity-coefficient dependence on one or more of the reactivity-feedback parameters. Linear interpolation is used to evaluate the reactivity coefficient for values of each parameter between its table values. Multidimensional linear-surface interpolation is performed when the reactivity coefficient is dependent on two or more of the reactivity-feedback parameters. An example of defining input data for the reactivity feedback model is given in Appendix I of the TRAC-M/F90 User's Manual (Ref. M-5.).

Solute in the derivative and solute dependence of a reactivity coefficient is defined in terms of the solute concentration in the coolant channel, B_m , in units of kg/m^3 , a macroscopic density, or the solute mass to liquid-coolant mass ratio, B_r , in units of parts solute per million parts liquid coolant (ppm). The solute reactivity coefficient is based on the change in the amount of solute without changing T_c and α when the coefficient is defined as a derivative of B_r . The reactivity-coefficient derivative of B_m is affected by changes in T_c and α . Consequently, having the reactivity coefficient as a derivative of B_r is better if data are available. The solute dependence of a reactivity coefficient is important when the reactivity coefficient is sensitive to a shift in the neutron-energy spectrum. Through the proportionality to the amount of neutron capture in solute, the B_m - or B_r -dependence characterizes this shift. Cross-section/neutronics codes are needed to determine this dependence separate from TRAC.

TABLE M-3.
Reactivity-Coefficient Forms

Form Number	Reactivity-Coefficient Form
0	$\frac{\partial k}{\partial x}$
1	$\frac{1}{k} \frac{\partial k}{\partial x} \equiv \frac{\partial R}{\partial x}$
2	$x \frac{\partial k}{\partial x}$
3	$\frac{x \partial k}{k \partial x} \equiv x \frac{\partial R}{\partial x}$

All forms of solute in the reactor core are included in the evaluation of B_m or B_r for solute dependence of the reactivity coefficients. Examples include dissolved boric acid in the liquid coolant, boric acid plated on core structure, borosilicate glass in burnable-poison pins, and boron oxide in control-rod-cluster pins. The first two forms are based on boric-acid convection into the reactor core throughout the modeled system and on plate-out of boric acid in the reactor core as a result of coolant dryout. For the latter two forms, the TRAC user defines the burnable-poison-pin and control-rod-cluster-pin forms through input with what would be their equivalent solute concentrations in the coolant channel. BBP or BCR define the equivalent concentration in the coolant. A first-order-polynomial function of the core-averaged coolant temperature T_c , the coefficients of which are input parameters, describes the equivalent burnable-poison-pin solute concentration as

$$BPP = BPP0 + BPP1 \cdot T_c . \quad (M-15)$$

Whereas the physical amount of solute in the burnable-poison pins is constant during the transient, its effective concentration for neutron capture increases with coolant temperature because of reduced spatial self-shielding of solute to neutrons in those pins. The amount of control-rod-cluster-pin solute in the reactor core depends on the amount of control-rod insertion into the core. Its effective concentration is input as a first-order-polynomial function of programmed reactivity R_{prog} as follows:

$$BCR = BCR0 + BCR1 \cdot R_{prog} . \quad (M-16)$$

Both BPP and BCR in the previous two equations have units of macroscopic solute density ($\text{kg} \cdot \text{m}^{-3}$) in the coolant-channel volume. Programmed reactivity is assumed to be proportional to the amount of effective solute-mass change caused by control-rod movement in the core. Initially, when the reactor core is at steady-state conditions, $BCR = BCR0$.

Only the dissolved boric acid and that plated on solid structures in the reactor core are included in the evaluation of B_m or B_r for the solute reactivity-coefficient derivative parameter because the amount of solute in burnable-poison pins is constant during the transient that is being evaluated. Control-rod movement changes the amount of solute, but the reactivity change associated with this movement is accounted for in the user-defined programmed reactivity R_{prog} . There is no reactivity feedback from control-rod movement because it is already accounted for in R_{prog} .

After all four reactivity coefficients are evaluated by linear interpolation in their four dimensionally dependent tables, feedback reactivity is evaluated in terms of its change in the neutron multiplication constant over the last time step, $\Delta t^{n-1} = t^n - t^{n-1}$,

$$k^n R_{fdbk}^n = \sum_{j=1}^n (\Delta k_{fdbk})^j \quad (M-17)$$

where

$$(\Delta k_{fdbk})^j = \sum_{i=1}^4 \frac{1}{2} \left[\left(\frac{\partial k}{\partial x_i} \right)^j + \left(\frac{\partial k}{\partial x_i} \right)^{j-1} \right] \times [x_i^j - x_i^{j-1}] ,$$

x_1 is T_f , x_2 is T_c , x_3 is α , and x_4 is B_m or B_r . The superscripts n and $n-1$ indicate that the parameter is evaluated at the beginning of the current time step and at the beginning of the previous time step, respectively. There are four reactivity-coefficient-defining forms in Table M-3. Forms 1, 2, and 3 as input are converted to form 0 by TRAC for use in Eq. (M-17). TRAC obtains the value of $(\partial k / \partial x_i)^n$ in Eq. (M-17) by multiplying the reactivity coefficients defined by reactivity-coefficient forms 1, 2, and 3 (Table M-3.) by k^{n-1} , $1/x_i^n$, and k^{n-1}/x_i^n , respectively. Using k^{n-1} rather than k^n (which is not yet known) to convert reactivity-coefficient forms 1 and 3 to $\partial k / \partial x_i$ is an approximation. The values of $\partial k / \partial x_i$ and x_i at time t^n (the start of the present time step) are evaluated as described above; their values at time t^{n-1} are those saved from their evaluation at the start of the previous time step.

The change in x_i over the last time step (with the other x_i parameters held constant in defining the reactivity coefficient) is defined in the last factor of Eq. (M-17). The term B_m or B_r includes only dissolved boric acid in the liquid coolant and boric acid plated on the core structure. There is no change in the B_r -equivalent solute concentration in burnable-poison pins when T_f , T_c , and α are not considered to vary.

To determine reactivity feedback during the current time step $\Delta t^n = t^{n+1} - t^n$, TRAC would need to know the end-of-time-step values of T_f , T_c , α , B_m or B_r . To evaluate that reactor-core state requires knowing the current core state and the total energy-generation rate, $P_{eff}(t)$, over the time step Δt^n . The neutronic reactivity $R(t)$ defined by the reactivity-feedback contribution, $R_{fdbk}(t)$ must be known to determine the value of $P_{eff}(t)$ from the solution of the point-reactor-kinetics equations. Thus, the evaluation of reactivity feedback during the current time step requires first knowing the time-step solution.

The TRAC code handles this difficulty of needing to know the reactivity feedback in order to evaluate it in time step Δt^n by assuming the Δk feedback-reactivity rate is zero. The Δk programmed-reactivity rate is assumed to be the same as in the previous time step, Δt^{n-1} ,

$$(\Delta k_{prog}^{est})^{n+1} = \Delta t^n \left\{ \frac{\Delta k_{prog}^n}{\Delta t^{n-1}} \right\} . \quad (M-18)$$

We estimate $(k_{prog})^n$ because the input-defined independent variable needed to evaluate its tabular definition at t^{n+1} is a function of the reactor-core state that is known at time t^n . We have tested a similar approximation for $(\Delta k_{prog}^{est})^n$ in TRAC, but near steady state it caused the sign of $(\Delta k_{fdbk}^{est})^n$ to change each time step. It is better to assume that $(\Delta k_{fdbk}^{est})^n$ is zero than to estimate its value based on its value in the previous time step. Feedback effects for each time step are more tightly coupled to $(\Delta k_{fdbk}^{est})^n$ than to $(\Delta k_{prog}^{est})^n$, which causes the former estimate to be numerically unstable in some situations. Thus, estimates for the neutron-multiplication constant and neutronic reactivity for the reactor state at the end of the present time step are

$$(k^{est})^{n+1} = k^n + (\Delta k_{prog}^{est})^{n+1} \quad (M-19)$$

and

$$(R^{est})^{n+1} = \frac{[(k^{est})^{n+1} - 1]}{(k^{est})^{n+1}} \quad (M-20)$$

After each Δt^{n+1} time-step solution, TRAC compares the actual programmed reactivity with its estimated value. The code corrects any discrepancy by applying

$$(\Delta k^{cor})^n = [(\Delta k_{prog})^n - (\Delta k_{prog}^{est})^n] \cdot \min\left(\frac{\Delta t^{n-1}}{\Delta t^n}, 2\right) \quad (M-21)$$

during the next time step, Δt^n . To prevent a time-step reactivity correction from becoming very large by applying it over a time step $\Delta t^n \ll \Delta t^{n-1}$, TRAC constrains $(\Delta k^{cor})^n$ in Eq. (M-21) to be no more than twice the Δt^{n-1} time-step value. Including this correction in Δk from the previous time step in the neutron-multiplication constant end-of-time-step estimate at t^{n+1} gives Eq. (M-19) in the modified form,

$$(k^{est})^{n+1} = k^n + (\Delta k_{prog}^{est})^{n+1} + (\Delta k^{cor})^n, \quad (M-22)$$

where

$$k^n = k^{n-1} + (\Delta k_{prog})^n + (\Delta k_{fdbk})^n.$$

In this procedure, we assume that $(\Delta k^{cor})^n$ and $(\Delta k_{fdbk})^{n-1}$ are small enough not to require an iterative evaluation of the present time-step solution and that $(\Delta k^{cor})^n$ is large enough to be included in the next time-step solution rather than be neglected. The code applies a similar estimate and correction procedure to the end-of-time-step power for the case in which the total reactor-core power is specified directly by a table.

M.2.6. Solution of the Point-Reactor Kinetics

The code solves the point-reactor-kinetics equations [Eqs. (M-6) through (M-8)] by the Kaganove method (Ref. M-3). Its derivation in Ref. M-3 approximates the time dependence of P and $k_{\text{ex}} = k - 1 = R / (1 - R)$ over each integration time step by second-order polynomials and assumes $\Lambda(t) = \ell / [1 + k_{\text{ex}}(t)]$ where ℓ is a constant. For TRAC, it is more appropriate to approximate the time dependence of P by a second-order polynomial, R by a first-order polynomial, and Λ by a constant because TRAC linearly extrapolates its estimate of $R(t)$ over the fluid-dynamics time step to be evaluated and because the weak time dependence of Λ generally is unknown. These modified assumptions, when applied to the point-reactor-kinetics equations, simplify the form that these equations take when analytically integrated over the neutronics integration time step Δt . The derivation of those analytically integrated equations follows. TRAC then evaluates them for $\Delta t^n / \Delta t$ integration time steps during the Δt^n fluid-dynamics time step.

Currently in TRAC, the neutronics solution is done before the hydraulic and heat-conduction solutions. Otherwise, the linear extrapolation estimates on the previous page would not have to be done if the hydraulic and heat-conduction solutions were done first. The heat-conduction solution would be more implicit if it were done after the neutronics solution. A preferred order would be to do the hydraulics solution first, the neutronics solution next, and then the heat-conduction solution.

We assume that

$$P(t) = P(0) + P_1 t + P_2 t^2 \quad , \quad (\text{M-23})$$

$$R(t) = R(0) + R_1 t \quad , \quad (\text{M-24})$$

and

$$\Lambda, \lambda_i, \beta_i, \lambda_j^H, \text{ and } E_j \text{ are constant for } 0 \leq t \leq \Delta t,$$

where

$$P(0), R(0), \text{ and } R(1) = [(R^{\text{est}})^n - R(0)] / \Delta t^n \text{ are known values.}$$

Solving Eq. (M-7) for $C_i(t)$ in terms of a functional of the fission power $P(t)$ gives

$$C_i(t) = C_i(0) \exp [-\lambda_i t] + \frac{\beta_i}{\Lambda} \int_0^t \exp [-\lambda_i(t - \tau)] P(\tau) d\tau \quad . \quad (\text{M-25})$$

Substituting Eq. (M-25) into Eq. (M-6) and integrating the resulting equation term-by-term gives

$$P(t) = P(0) + \int_0^t \frac{R(\tau)P(\tau)}{\Lambda} d\tau - \sum_{i=1}^I \frac{\beta_i}{\Lambda} \int_0^t \exp[-\lambda_i(t-\tau)] P(\tau) d\tau + \sum_{i=1}^I C_i(0) [1 - \exp(-\lambda_i t)] + \frac{S}{\Lambda} \left[1 - R(0)t - R_1 \frac{t^2}{2} \right]. \quad (\text{M-26})$$

Substituting Eq. (M-23) for $P(\tau)$ and Eq. (M-24) for $R(\tau)$ in Eq. (M-26), evaluating the integrals, and rearranging the resulting equation in terms of the P_1 and P_2 unknowns gives

$$a_1(t)P_1 + a_2(t)P_2 = q(t), \quad (\text{M-27})$$

where

$$a_1(t) = t\Lambda - t^2 \left[\frac{1}{2}R(0) + \frac{1}{3}R_1 t \right] + \sum_{i=1}^I \beta_i t^2 I_{i1}(t),$$

$$a_2(t) = t^2\Lambda - t^3 \left[\frac{1}{3}R(0) + \frac{1}{4}R_1 t \right] + \sum_{i=1}^I \beta_i t^3 I_{i2}(t),$$

$$q(t) = S + t \left[R(0) + \frac{1}{2}R_1 t \right] [P(0) - S] + \sum_{i=1}^I [\Lambda \lambda_i C_i(0) - \beta_i P(0)] t I_{i0}(t),$$

and

$$I_{im}(t) = \frac{\int_0^t \exp[-\lambda_i(t-\tau)] \tau^m d\tau}{t^{m+1}}, \quad \text{for } m = 0, 1, 2.$$

The P_1 and P_2 polynomial coefficients are evaluated by requiring Eq. (M-27) to be satisfied for $t = \Delta t$ (at the end of the integration time step) and $t = \Delta t/2$ (at the midpoint of the integration time step). Solving the two equations

$$a_1(\Delta t)P_1 + a_2(\Delta t)P_2 = q(\Delta t) \quad (\text{M-28})$$

and

$$a_1\left(\frac{\Delta t}{2}\right)P_1 + a_2\left(\frac{\Delta t}{2}\right)P_2 = q\left(\frac{\Delta t}{2}\right) \quad (\text{M-29})$$

for P_1 and P_2 gives

$$P_1 = \frac{\left[q(\Delta t)a_2\left(\frac{\Delta t}{2}\right) - q\left(\frac{\Delta t}{2}\right)a_2(\Delta t) \right]}{D} \quad (\text{M-30})$$

and

$$P_2 = \frac{\left[q\left(\frac{\Delta t}{2}\right)a_1(\Delta t) - q(\Delta t)a_1\left(\frac{\Delta t}{2}\right) \right]}{D}, \quad (\text{M-31})$$

where

$$D = a_1(\Delta t)a_2\left(\frac{\Delta t}{2}\right) - a_1\left(\frac{\Delta t}{2}\right)a_2(\Delta t).$$

The total fission power at the end of the integration time step $t = \Delta t$ from Eq. (M-23) is

$$P(\Delta t) = P(0) + P_1\Delta t + P_2\Delta t^2. \quad (\text{M-32})$$

If we know $P(t)$ for $0 \leq t \leq \Delta t$, we can evaluate $C_i(\Delta t)$ by substituting Eq. (M-23) into Eq. (M-25) and using

$$\Delta t I_{i0}(\Delta t) = \frac{1 - \exp(-\lambda_i \Delta t)}{\lambda_i}$$

by analytically integrating its definition in Eq. (M-27) as follows:

$$C_i(\Delta t) = C_i(0)[1 - \lambda_i \Delta t I_{i0}(\Delta t)] + \frac{\beta_i}{\Lambda} [P(0) \Delta t I_{i0}(\Delta t) + P_1 \Delta t^2 I_{i1}(\Delta t) + P_2 \Delta t^3 I_{i2}(\Delta t)] \text{ for } i = 1, 2, \dots, I. \quad (\text{M-33})$$

The code evaluates the decay-heat equations in the same way to give

$$H_j(\Delta t) = H_j(0)[1 - \lambda_j^H \Delta t I_{j_0}(\Delta t)] + E_j [P(0) \Delta t I_{j_0}(\Delta t) + P_1 \Delta t^2 I_{j_1}(\Delta t) + P_2 \Delta t^3 I_{j_2}(\Delta t)] \text{ for } j = 1, 2, \dots, J, \quad (\text{M-34})$$

where

$$\Delta t^{m+1} I_{jm}(\Delta t) = \int_0^{\Delta t} \exp[-\lambda_j^H (\Delta t - \tau)] \tau^m d\tau .$$

The code evaluates $\Delta t^{m+1} I_{im}(\Delta t)$ with the following recursion relation:

$$\Delta t^{m+1} I_{im}(\Delta t) = \frac{\Delta t^m - m \Delta t^m I_{im-1}(\Delta t)}{\lambda_i} , \quad \text{for } m = 1, 2 \quad (\text{M-35})$$

once

$$\Delta t I_{i_0}(\Delta t) \frac{1 - \exp(-\lambda_i \Delta t)}{\lambda_i}$$

is first evaluated. For $\lambda_i \Delta t \ll 1$, Eq. (M-35) results in the loss of several least-significant digits of accuracy with each application of the recursion formula. Thus, for $\lambda_i \Delta t < 1$, TRAC first evaluates $\Delta t^3 I_{i_2}(\Delta t)$ using the Maclaurin expansion,

$$\Delta t^{m+1} I_{im}(\Delta t) = \Delta t^{m+1} m! \left[\frac{1}{(m+1)!} - \frac{\lambda_i \Delta t}{(m+2)!} + \frac{(\lambda_i \Delta t)^2}{(m+3)!} - \dots \right] \quad (\text{M-36})$$

for $m = 2$, and then evaluates the reciprocal of the Eq. (M-35),

$$\Delta t^m I_{im-1}(\Delta t) = \frac{\Delta t^m - \lambda_i \Delta t^{m+1} I_{im}(\Delta t)}{m} \text{ for } m = 2, 1. \quad (\text{M-37})$$

The accuracy of the second-order polynomial approximation for $P(t)$ can be increased by decreasing the integration time-step size Δt . We would like Δt to be as large as possible, however, while we maintain a desired level of accuracy in approximating $P(t)$. To achieve this, the following procedure for automatically adjusting the reactor-kinetics time step in TRAC is used. In the same manner that Eqs. (M-28) and (M-29) were defined, Eq. (M-27) also could be required to be satisfied at $t = \Delta t / 4$ by

$$a_1 \left(\frac{\Delta t}{4} \right) P_1 + a_2 \left(\frac{\Delta t}{4} \right) P_2 = q \left(\frac{\Delta t}{4} \right). \quad (\text{M-38})$$

The above equation together with Eq. (M-29), which is evaluated at $t = \Delta t / 2$, are solved over the time range $0 \leq t \leq \Delta t / 2$ for $P_1 = P_1^*$ and $P_2 = P_2^*$ to give

$$P_1^* = \frac{q \left(\frac{\Delta t}{2} \right) a_2 \left(\frac{\Delta t}{4} \right) - q \left(\frac{\Delta t}{4} \right) a_2 \left(\frac{\Delta t}{2} \right)}{D^*} \quad (\text{M-39})$$

and

$$P_2^* = \frac{q \left(\frac{\Delta t}{4} \right) a_1 \left(\frac{\Delta t}{2} \right) - q \left(\frac{\Delta t}{2} \right) a_1 \left(\frac{\Delta t}{4} \right)}{D^*}, \quad (\text{M-40})$$

where

$$D^* = a_1 \left(\frac{\Delta t}{2} \right) a_2 \left(\frac{\Delta t}{4} \right) - a_1 \left(\frac{\Delta t}{4} \right) a_2 \left(\frac{\Delta t}{2} \right).$$

This solution over half the integration time step provides a more accurate value for $P(t)$ at $t = \Delta t / 2$,

$$P^* \left(\frac{\Delta t}{2} \right) = P(0) + P_1^* \left(\frac{\Delta t}{2} \right) + P_2^* \left(\frac{\Delta t}{2} \right)^2, \quad (\text{M-41})$$

than the solution over the full integration time step at $t = \Delta t / 2$,

$$P \left(\frac{\Delta t}{2} \right) = P(0) + P_1 \left(\frac{\Delta t}{2} \right) + P_2 \left(\frac{\Delta t}{2} \right)^2. \quad (\text{M-42})$$

The code uses the difference between these two values as a measure of the error in the solution over the full integration time step Δt

$$\varepsilon(\Delta t) = \left| \frac{P \left(\frac{\Delta t}{2} \right) - P^* \left(\frac{\Delta t}{2} \right)}{P^* \left(\frac{\Delta t}{2} \right)} \right|. \quad (\text{M-43})$$

It is desirable to increase Δt if the above ε is too small and to decrease Δt if ε is too large. Toward this end, the code defines two error bounds, ε_1 and ε_2 , to provide the desired convergence such that,

1. if $\varepsilon_1 < \varepsilon(\Delta t) < \varepsilon_2$, the code maintains the present value of reactor-kinetics time step Δt ;
2. if $\varepsilon_1 < \varepsilon_2 \leq \varepsilon(\Delta t)$, the code halves the value of Δt because the error in $P(\Delta t)$ is too large; and
3. if $\varepsilon(\Delta t) \leq \varepsilon_1 < \varepsilon_2$, the code doubles the value of Δt because the error in $P(\Delta t)$ is too small.

When Δt is halved, the integration time step is reevaluated. When the third criterion is satisfied, the number of remaining integration time steps to be evaluated over the fluid-dynamics time step must be an even number for the integration time step to be doubled (and the number of remaining integration time steps to be halved). TRAC is programmed to use $\varepsilon_1 = 10^{-6}$ and $\varepsilon_2 = 10^{-4}$. A numerical study showed these values to yield a maximum fractional error less than 10^{-4} and an average fractional error less than 10^{-5} in the fission power solution. The point-reactor-kinetics solution required < 1% of TRAC execution time.

M.3. Conclusions Regarding the Reactor-Core Power Model

The core-power model in TRAC, with all the options and features, has proved to be sufficiently general for describing the core assemblies in most of the reactor-safety-related experiments, particularly ones that are integral in nature, and in reactor plants, provided that the power shape only changes in time in the axial direction. For this to be true, the axial-power shape must be specified as a function of some user-defined independent variable for the axial-power-shape table. The option to specify core power directly with a tabular prescription is generally sufficient for modeling most transients of interest where the power-level behavior is known.

The point-reactor kinetics with reactivity feedback has demonstrated in many cases of the TRAC developmental assessment problem set the ability to describe the transient power response of a reactor core. For example, in the case of anticipated transients without scram, the core-power behavior would not be known because of reactivity-feedback effects, and in this case, reactor kinetics is a necessary capability. In analyzing experiments from the LOFT facility (for example, the L6-1 test simulating a loss-of-steam-load transient) and transients from power plants (for example, the Zion PWR standard test problem), reactor kinetics has demonstrated a good predictive capability. For performing best-estimate analyses of large-break LOCAs, reactor kinetics has demonstrated the ability to predict the shutdown of core power by core voiding before the control-rod scram is effective.

The deficiencies of the core-power model are principally related to the fact that reactor kinetics is based on a point-reactor assumption, and the power distribution is defined by a superposition of fuel-rod-radial, core-axial, and (r, θ) -plane shapes. Axial-shape changes in the power distribution with a known dependence can be simulated, but

fuel-rod-radial and core (r, θ) -plane power-shape changes and power distributions that are not definable by superimposed fuel-rod-radial, core-axial, and (r, θ) -plane shapes cannot be modeled. Reactivity feedback is also based on the point-reactor assumption with little theoretical justification for the procedure used to core average the reactivity-feedback parameters. Asymmetric and localized reactivity-feedback effects and control-rod movements in the reactor core cannot be modeled accurately. However, within these modeling limitations, the core-power model is accurate and efficient in determining the time-dependent behavior of the reactor-core power. To further assess the point-kinetics model requires having good core neutronics data (power shape, reactivity coefficients, etc.) and use of neutronics codes to generate the point-kinetics input data.

REFERENCES

- M-1. "Reactor Physics Constants," Argonne National Laboratory report ANL-5800, 2nd ed. (July 1963).
- M-2. A. Rodkowsky, Ed., "Naval Reactors Physics Handbook, Vol. 1: Selected Basic Techniques," Library of Congress Number 65-60008 (1964).
- M-3. J. J. Kaganove, "Numerical Solution of the One-Group Space-Independent Reactor Kinetics Equations for Neutron Density Given the Excess Reactivity," Argonne National Laboratory report ANL-6132 (1960).
- M-4. "American National Standard for Decay Heat Power in Light Water Reactors," American Nuclear Society publication ANSI/ANS-5.1 (1979).
- M-5. R. G. Steinke, V. Martinez, N. M. Schnurr, J. W. Spore, and J. V. Valdez, "TRAC-M Fortran 90 (Version 3.0) User's Manual," Los Alamos National Laboratory document LA-UR-00-835 (February 2000).

APPENDIX N

CONTROL PROCEDURE

In this appendix, the description of the control procedure is directly related to the Fortran coding. Consequently, very few equations and symbols are used. The variables are usually described using mnemonics, quite often through a generalized format. Almost all the variables are locally applicable within this appendix with little (if any) reference to other sections. We believe that the description provided for these variables within the text is satisfactory and no additional nomenclature is necessary at the beginning of the section.

This appendix first discusses the general forms of component-action tables and rate-factor tables. Following this discussion are detailed descriptions of TRAC's signal-variable, control-block, trip, and control-parameter evaluation logic. Finally, TRAC's special logic for steady-state evaluation is presented.

N.1. Component-Action Table

Modeling a component's adjustable-hardware action is done by means of a component-action table specified through the component input data. The table defines the component hardware action as a tabular function of an independent variable defined by physical-system parameters. These parameters monitored from the PWR physical-system model are called signal variables in TRAC. They can be applied as the component-action table's independent variable directly or after having been operated on by function operators called control blocks. In this section, the independent variable that is discussed is the independent variable of the component-action table.

Determining a hardware action by evaluating its component-action table is based on a component-action-type option parameter. Its Fortran variable name has the general form DXXTY where letters XXX are to be replaced by one to three letters characterizing the component action. Those letters for each of the actions in Table N-1 (a replica of Table 2-1.) are shown in the far right column. Through the option parameter, the user specifies if the action is to be held constant, evaluated at all times from the component-action table, or evaluated from the component-action table only when an assigned controlling trip switch is ON and held constant when the trip is OFF. If the component-action table is to be evaluated under trip control, a trip ID number with general-form variable name IXXTR is specified as part of the component input data. The actual definition of the trip with the ID number label referenced here is specified in the trip input-data section located elsewhere in the input-data file. When DXXTY defines no trip control for evaluating the component-action table, IXXTR should be defined by zero or by a blank field on input.

TABLE N-1.
Adjustable Component Hardware
Actions by the Control Procedure

Actions	Components	Variable -Name Letters
Pressure boundary condition and fluid state	BREAK	B
Velocity or mass-flow boundary condition and fluid state	FILL	F
Reactor-core programmed reactivity or power	HEAT STRUCTURE	RPW
Reactor-core axial-power shape	HEAT STRUCTURE	ZPW
Energy deposition in the coolant	PIPE, TEE, TURBINE	POW or PW
Energy generation in the wall	PIPE, PUMP, TEE, VALVE	QP3 or QP
Pump rotational speed	PUMP	PMP or OMG
Turbine power demand	TURBINE	TRB or POP
Valve flow-area fraction or relative stem position	VALVE	V

The independent variable is defined by physical-system parameters in the form of a signal variable or a control-block output variable. The variable to be used is defined by its ID number with the general Fortran variable name IXXXSV. Positive ID numbers define signal variables; negative ID numbers define control-block output variables. Signal variables and control blocks are described in Sections N.3. and N.4.

The number of entry pairs (independent-variable value, component-action value) of tabular input data defining the component-action table is specified by the general Fortran variable name NXXXTB. A plus or minus sign is applied to its value to define the form of the independent variable. When NXXXTB is positive, the independent variable is the IXXXSV parameter. When NXXXTB is negative, the independent variable is the change in the value of the IXXXSV variable during the time the table is evaluated. For this latter form, the initial value of the component action corresponds to the component-action value in the table with an independent-variable value of zero. Internal to TRAC, the change in the IXXXSV variable value is incorporated into the table's abscissa-coordinate values during each time step the table is evaluated by translation of all the abscissa-coordinate (independent-variable) values in the table. As a result, the last interpolated component-action value from the table will correspond to the independent

variable having a value of zero after translation. The user does not need to be concerned with this internal translation of the table's abscissa-coordinate values by TRAC. The only time its effect can be seen by the user is when the component-action table is output on restart. The user only needs to be concerned with initially defining a $NXXXTB < 0$ table with the initial component-action value corresponding to the abscissa-coordinate value of zero.

When a $NXXXTB < 0$ table is to be evaluated under trip control, the change in the $IXXXSV$ variable value over each time step is further multiplied by the value of the controlling trip ON/OFF set status. Fortran variable $ISET$ defines the trip set status. OFF has the value 0, and the two forms of ON, $ON_{forward}$ and $ON_{reverse}$ have values +1 and -1, respectively. Because the component-action table under trip control is only evaluated when the trip is ON, $ISET$ has the value +1 or -1 when the change in the $IXXXSV$ variable value is multiplied by it to evaluate the independent variable. Hence, the trip set-status value affects the direction of interpolated movement in the table. The forward and reverse subscripts for ON correspond to the direction of the interpolated movement action.

The effect the values for $DXXXTR$, $IXXXSV$, and $NXXXTB$ has on defining the independent variable are summarized in Table N-2. Evaluation of the component-action table is trip controlled when $DXXXTR \neq 0$. No component-action table is defined (the component action is held constant) when $IXXXSV = 0$ is input. When $NXXXTB < 0$ is input with $IXXXSV \neq 0$, a table is defined, but it has no tabular data. This is a special case where the component action is defined directly by its independent-variable form, that is, a signal variable ($IXXXSV > 0$) or a control-block output variable ($IXXXSV < 0$).

TABLE N-2.
Defined Forms of the Component-Action
Table's Independent Variable

Component-Action Table's Defining Variables			Component-Action Table's Independent-Variable Form^a
DXXXTR	IXXXSV	NXXXTB	
All values	>0	≥ 0	A
0	>0	<0	DA
$\neq 0$	>0	<0	DA*ISET
All values	<0	≥ 0	B
0	<0	<0	DB
\neq	<0	<0	DB*ISET

a. A is a signal variable; B is a control-block output variable.

In the above definition, the direction of interpolation in the component-action table can be trip controlled when $I\text{XXXTR} \neq 0$ and $N\text{XXXTB} < 0$. The trip set-status value (± 1) is applied to the magnitude of the movement defined by the change in the signal variable or control-block output variable. In most situations, this form is sufficient for modeling the component-action's independent variable. There are a few situations, however, where the user would like further control over the magnitude of the interpolated movement. The user can do this either with a connected series of control-block operators defining the independent variable or by applying a rate factor to the independent-variable forms in Table N-2. Operating on a parameter signal (signal-variable value) with a connected series of control blocks and defining the independent-variable form with the last control-block output variable are extended applications of one of the last three forms in Table N-2. Applying a rate factor to any one of the six forms is a direct adjustment to the rate of interpolated movement when evaluating the component-action table.

N.2. Rate-Factor Table

A rate factor can be applied to the component-action table's independent variable by entering a rate-factor table of tabular data. Specifying a rate-factor table requires that its component-action table be defined ($I\text{XXXSV} \neq 0$) and that the user decide to apply a rate factor to the independent-variable form in Table N-2.

Fortran variables $N\text{XXXSV}$ and $N\text{XXXRF}$ define a rate-factor table in the same way that $I\text{XXXSV}$ and $N\text{XXXTB}$, respectively, define the component-action table. In addition to the six independent-variable forms in Table N-2, entering $I\text{XXXTR} \neq 0$, $N\text{XXXSV} = 0$, and $N\text{XXXRF} \neq 0$ defines the independent-variable form for the rate-factor table to be the difference between the trip-signal value and the trip set-point value that turns the trip OFF. This additional form provides a convenient means for adjusting the rate of interpolated movement in the component-action table. The further the trip signal departs from its desired value, specified by its set-point value that turns the trip OFF, the larger the rate factor and the faster the component action can be adjusted in order to return the trip signal to its desired value. Section N.5, describing how trips are defined, clarifies the explanation for the above form.

When a rate-factor table is defined, the procedure for evaluating the component-action value is as follows. First, TRAC evaluates the rate-factor table's independent-variable value (defined by $I\text{XXXTR}$, $N\text{XXXSV}$, and $N\text{XXXRF}$). The value is then used to linearly interpolate in the rate-factor table (when $N\text{XXXRF} \neq 0$) or define directly (when $N\text{XXXRF} = 0$) the rate-factor value. Next, the component-action table's independent-variable value (defined by $I\text{XXXTR}$, $I\text{XXXSV}$, and $N\text{XXXTB}$) is evaluated and multiplied by the rate factor. This product value is used to interpolate linearly in the component-action table (when $N\text{XXXTB} \neq 0$) or define directly (when $N\text{XXXTB} = 0$) the component-action value. The component hardware action in the modeled PWR system is then defined with this value. This procedure is evaluated at the beginning of each time step with the current state of the system parameters to evaluate the independent-variable forms for both tables. Not defining a rate-factor table ($N\text{XXXSV} = 0$ and $N\text{XXXRF} = 0$) are

input) reduces the above procedure to evaluating the component-action table's independent-variable value and then using that value to interpolate linearly in the component-action table or define directly the component-action value.

The component-action table and rate-factor table defining variables `DXXXTR`, `DXXXSV`, `NXXXTB`, `NXXXSV`, and `NXXXRF` are specified in that order on one input-data card. There is such a card for each type of component action in the component input data unless the `DXXXTY` option specifies that the component action is to be held constant. Then neither component-action tables nor defining variables are input. There is also a second input-data card defining the component action with the following real-valued general-form variables: `XXXIN`, `XXXOFF`, `RXXXMX`, and `XXXSCL`. Variable `XXXIN` defines the initial value for the component action. This value is used when the component action is to be held constant or when the user is evaluating the component action under trip control and the trip is initially OFF. For component actions that are trip controlled, variable `XXXOFF` defines the component-action value that is used when the trip is OFF after having been ON. Entering `XXXOFF = -1 × 1019` defaults TRAC to using the last component-action value evaluated when the trip was ON. Variable `RXXXMX` defines the maximum rate of change of the component-action value allowed by the user during the simulation. When the component action evaluated from the table varies faster than this maximum rate, the component action is redefined to vary at the maximum rate. Later when the rate of change of the table-evaluated component action falls below the maximum rate, the applied component-action value will be able to catch up to the value determined from the table. Variable `XXXSCL` is a scale factor to be applied during input to the component-action values in its input table.

The component-action table (with `|NXXXTB|*2` values) and its rate-factor table (with `|NXXXRF|*2` values) are input in the array data section of the component input. All component-action tables, with two exceptions, have one component-action value for each independent-variable value. Thus, there are two values for each of the `|NXXXTB|` entry pairs in their tabular data. The two exceptions are the axial-power-shape and the energy-generation-in-the-wall component-action tables. Corresponding to each independent-variable value is an axial-power shape with a relative-power value at each axial interface in the reactor-core model or a QPPP-factor shape with a QPPP-factor value for each mesh cell in the component model.

This section has discussed application of the component-action table in its general form. The `XXX` letters in all the Fortran variable names defined will need to be replaced by the component action's one to three letters in [Table N-1](#), to give the Fortran variable names. A main point to recognize from this section is that there is one form to the numerical model for applying adjustable control to a component action, even though there are nine different component hardware actions that can be modeled. Understanding the general definition of control variables `DXXXTY`, `DXXXTR`, `DXXXSV`, `NXXXTB`, `NXXXSV`, `NXXXRF`, `XXXIN`, `XXXOFF`, `RXXXMX`, and `XXXSCL` gives the user the ability to model the dynamics of a myriad of possible control procedures encountered in PWR operation. The next four sections on signal variables, control blocks, trips, and ordered evaluation

of control parameters go into the details of the definition of these tools used to model the control procedure.

N.3. Signal Variables

Signal variables are PWR-modeled system parameters that the TRAC user defines for the control-procedure model to use to determine the adjustment of component hardware actions. Signal variables have their values determined by TRAC at the beginning of each time step and are based on the state of the modeled system. Their values are used in the control procedure to determine the magnitudes of adjustable component hardware actions, which are then applied over the time step. This procedure approximates the magnitudes of the signal variables and component actions as being constant over the time step with step changes at the beginning of each time step. In reality, they vary continuously. Using smaller time steps approximates this more accurately. With the stability-enhanced, two-step, hydrodynamic solution method in TRAC, large time steps are possible. When evaluating with large time steps, control adjustment will be numerically delayed in time by a fraction of a time step. Both the satisfied trip criteria for starting an action and the evaluated dependence that determines the action's magnitude encounter a fractional time-step lag. It is the accumulation of the effects of these time lags that can contribute a significant error when simulating a control procedure that the TRAC user needs to be concerned about.

A unique, positive identification number assigned to Fortran variable name IDSV is defined by the TRAC user for each signal variable. This number serves as the signal-variable label for referencing the use of its value within the control procedure. There are 88 PWR-modeled system parameters that are definable as signal variables. Those signal-variable parameters are listed in Table N-3. Each parameter is defined by a parameter number assigned to Fortran variable name ISVN when a signal variable is defined. Parameter number 0 is problem time defined by Fortran variable names ETIME (beginning-of-time-step problem time), DTO (previous time-step size), or DELT (present time-step size). Parameter numbers 1 through 17 are as yet undefined parameters that the user may program TRAC to determine. Any system parameter can have its value assigned to either of these parameter numbers by programming their value assignment in subroutine SVSET. The remaining system parameters in Table N-3 (except for 55 and 56) are defined for a specific component. Column 2 in Table N-3 shows which component types (1D for the 1D hydro components, 3D for the 3D vessel hydro components, and HS for the heat-structure component) that each signal-variable parameter can be defined for. Parameter numbers 20 through 40 and 65 through 101 require one or two cell/interface or node-cell/node-row numbers (cell_#/iface_# or ncell_#/nrow_#) to further delimit the location in the component where the parameter is defined. Parameter numbers 55 and 56 are trip parameters that, when defined as signal variables, can be used for other control purposes.

**TABLE N-3.
Signal-Variable Parameters**

Signal-Variable Parameter No.	Defined For	Parameter Description	Input Location Numbers
0	Global	Problem time (s)	-1 (dto), 0(time), 1(dt)
1-17		Undefined	
18	HS	Reactor power (W, Btu/hr)	Component
19	HS	Reactor-power period (s)	Component
20	1D, 3D	Liquid level (m, ft)	Component, Cell1, Cell2
21	1D, 3D	Cell pressure (Pa, psia)	Component, Cell1, Cell2
22	1D, 3D	Cell gas temperature (K, °F)	Component, Cell ₁ , Cell ₂
23	1D, 3D	Cell liquid temperature (K, °F)	Component, Cell1, Cell ₂
24	ID	Slab inner-surface temperature (K, °F)	Component, Cell ₁ , Cell ₂
25	HS	Slab/rod surface temperature (K, °F)	Component, Nrow ₁ , Nrow ₂
26	HS	Slab/rod temperature (K, °F)	Component, Nrow ₁ , Nrow ₂
27	1D, 3D	Cell gas volume fraction	Component, Cell ₁ , Cell ₂
28	3D	Yθ-face gas mass flow (kg/s, lb _m /hr)	Component, Iface ₁ , Iface ₂
29	1D, 3D	Z-face gas mass flow (kg/s, lb _m /hr)	Component, Iface ₁ , Iface ₂
30	3D	XR-face gas mass flow (kg/s, lb _m /hr)	Component, Iface ₁ , Iface ₂
31	3D	Yθ-face liquid mass flow (kg/s, lb _m /hr)	Component, Iface ₁ , Iface ₂
32	1D, 3D	Z-face liquid mass flow (kg/s, lb _m /hr)	Component, Iface ₁ , Iface ₂
33	3D	XR-face liquid mass flow (kg/s, lb _m /hr)	Component, Iface ₁ , Iface ₂
34	3D	Yθ-face gas velocity (m/s, ft/s)	Component, Iface ₁ , Iface ₂
35	1D, 3D	Z-face gas velocity (m/s, ft/s)	Component, Iface ₁ , Iface ₂
36	3D	XR-face gas velocity (m/s, ft/s)	Component, Iface ₁ , Iface ₂
37	3D	Yθ-face liquid velocity (m/s, ft/s)	Component, Iface ₁ , Iface ₂

TABLE N-3. (cont)
Signal-Variable Parameters

Signal-Variable Parameter No.	Defined For	Parameter Description	Input Location Numbers
38	1D, 3D	Z-face liquid velocity (m/s, ft/s)	Component, Iface ₁ , Iface ₂
39	3D	XR-face liquid velocity (m/s, ft/s)	Component, Iface ₁ , Iface ₂
40	1D, 3D	Boron concentration (kg/m ³)	Component, Cell ₁ , Cell ₂
41	1D	Pump rotational speed (rad/s, rpm)	Component
42	1D	Valve flow-area fraction	Component
43	1D	Valve stem position	Component
44	HS	Neutron multiplication constant $k_{\text{eff}} = 1 + \Delta k_{\text{prog}} + \Delta k_{\text{fdbk}}$	Component
45	HS	Programmed Δk_{prog}	Component
46	HS	Feedback Δk_{fdbk}	Component
47	HS	Fuel-temperature feedback Δk_{Tf}	Component
48	HS	Coolant-temperature feedback Δk_{Tc}	Component
49	HS	Gas-volume-fraction feedback Δk_{α}	Component
50	HS	Solute concentration-ratio feedback $\Delta k_{\beta m}$	Component
51	HS	Core density-volume-power ^{powexp} – averaged fuel temperature (K, °F)	Component
52	HS	Core density-volume-power ^{powexp} – averaged coolant temperature (K, °F)	Component
53	HS	Core volume-power ^{powexp} – averaged gas volume fraction	Component
54	HS	Core density-volume-power ^{powexp} – averaged solute concentration ratio (kg _{sol} /kg _{liq} , lb _{m-sol} /lb _{m-liq}) or (ppm)	Component
55	Global	Trip-signal value	Trip ID
56	Global	Trip set-status value	Trip ID
57	HS	Prompt-fission power (W, Btu/hr)	Component
58	HS	Decay-heat power (W, Btu/hr)	Component

TABLE N-3. (cont)
Signal-Variable Parameters

Signal- Variable Parameter No.	Defined For	Parameter Description	Input Location Numbers
59	HS	Average slab/rod maximum surface temperature (K, °F)	Component
60	HS	Additional slab/rod maximum surface temperature (K, °F)	Component
61	1D	Pump head (m^2/s^2 , $lb_f\text{-ft}/lb_m$)	Component
62	1D	Pump torque ($Pa\text{-}m^3$, $lb_f\text{-ft}$)	Component
63	1D	Pump momentum source (Pa, psia)	Component
64	1D	Valve hydraulic diameter (m, ft)	Component
65	3D	Y θ -face hydraulic diameter (m, ft)	Component, Iface ₁ , Iface ₂
66	1D, 3D	Z-face hydraulic diameter (m, ft)	Component, Iface ₁ , Iface ₂
67	3D	XR-face hydraulic diameter (m, ft)	Component, Iface ₁ , Iface ₂
68	3D	Y θ -face mix. mass flow (kg/s, lb_m/s)	Component, Iface ₁ , Iface ₂
69	1D, 3D	Z-face mixture mass flow (kg/s, lb_m/s)	Component, Iface ₁ , Iface ₂
70	3D	XR-face mixture mass flow (kg/s, lb_m/s)	Component, Iface ₁ , Iface ₂
71	3D	Y θ -face mixture average velocity (m/s, ft/s)	Component, Iface ₁ , Iface ₂
72	1D, 3D	Z-face mixture average velocity (m/s, ft/s)	Component, Iface ₁ , Iface ₂
73	3D	XR-face mixture average velocity (m/s, ft/s)	Component, Iface ₁ , Iface ₂
74	1D, 3D	Cell gas density (kg/m^3 , lb_m/ft^3)	Component, Cell ₁ , Cell ₂
75	1D, 3D	Cell liquid density (kg/m^3 , lb_m/ft^3)	Component, Cell ₁ , Cell ₂
76	1D, 3D	Cell mixture density (kg/m^3 , lb_m/ft^3)	Component, Cell ₁ , Cell ₂
77	1D, 3D	Cell air density (kg/m^3 , lb_m/ft^3)	Component, Cell ₁ , Cell ₂
78	1D, 3D	Cell air mass (kg/m^3 , lb_m/ft^3)	Component, Cell ₁ , Cell ₂

TABLE N-3. (cont)
Signal-Variable Parameters

Signal-Variable Parameter No.	Defined For	Parameter Description	Input Location Numbers
79	1D, 3D	Cell air pressure (Pa, psia)	Component, Cell ₁ , Cell ₂
80	1D, 3D	Cell air internal energy (W-s/kg, Btu/lb _m)	Component, Cell ₁ , Cell ₂
81	1D, 3D	Cell gas internal energy (W-s/kg, Btu/lb _m)	Component, Cell ₁ , Cell ₂
82	1D, 3D	Cell liquid internal energy (W-s/kg, Btu/lb _m)	Component, Cell ₁ , Cell ₂
83	1D, 3D	Cell saturation temperature based on vapor saturation pressure (K, °F)	Component, Cell ₁ , Cell ₂
84	1D, 3D	Cell saturation temp. based on total pressure (K, °F)	Component, Cell ₁ , Cell ₂
85	1D, 3D	Cell gas specific heat (W-s/kg-K, Btu/lbm-°F)	Component, Cell ₁ , Cell ₂
86	1D, 3D	Cell liquid specific heat (W-s/kg-K, Btu/lbm-°F)	Component, Cell ₁ , Cell ₂
87	1D, 3D	Cell latent heat of vaporization (W-s/kg, Btu/lb _m)	Component, Cell ₁ , Cell ₂
88	1D, 3D	Total surface heat loss to the gas (W, Btu/hr)	Component, NCell ₁ , NCell ₂
89	1D, 3D	Total surface heat loss to the liquid (W, Btu/hr)	Component, NCell ₁ , NCell ₂
90	1D, 3D	Cell gas/liquid interfacial heat flow (W, Btu/hr)	Component, Cell ₁ , Cell ₂
91	HS	ROD/SLAB gas HTC (W/m ² -K, Btu/ft ² -°F-hr)	Component, NCell ₁ , NCell ₂
92	HS	ROD/SLAB liquid HTC (W/m ² -K, Btu/ft ² -°F-hr)	Component, Ncell ₁ , Ncell ₂
93	1D	Cell SLAB gas HTC (W/m ² -K, Btu/ft ² -°F-hr)	Component, Cell ₁ , Cell ₂
94	1D	Cell SLAB liquid HTC (W/m ² -K, Btu/ft ² -°F-hr)	Component, Cell ₁ , Cell ₂
95	1D, 3D	Cell interfacial area x gas HTC (W/K, Btu/hr-°F)	Component, Cell ₁ , Cell ₂

TABLE N-3. (cont)
Signal-Variable Parameters

Signal-Variable Parameter No.	Defined For	Parameter Description	Input Location Numbers
96	1D, 3D	Cell interfacial area x liquid HTC (W/K, Btu/hr-°F)	Component, Cell ₁ , Cell ₂
97	3D	Yθ-face interfacial drag C _i (kg/m ⁴ , lb _m /ft ⁴)	Component, Iface ₁ , Iface ₂
98	1D, 3D	Z-face interfacial drag C _i (kg/m ⁴ , lb _m /ft ⁴)	Component, Iface ₁ , Iface ₂
99	3D	XR-face interfacial drag C _i (kg/m ⁴ , lb _m /ft ⁴)	Component, Iface ₁ , Iface ₂
100	1D, 3D	Cell solid-solute concentration (kg/m ³ , lb _m /ft ³)	Component, Cell ₁ , Cell ₂
101	1D, 3D	Cell vapor generation rate (kg/m ³ -s, lb _m /ft ³ -hr)	Component, Cell ₁ , Cell ₂
102	1D, HS	Total inner surface heat loss (W, Btu/hr)	Component
103	1D, HS	Total outer surface heat loss (W, Btu/hr)	Component
104	1D, 3D	Cell mixture temperature (K, °F)	Component, Cell ₁ , Cell ₂

For mesh cells on the secondary side of a TEE component, Cell₁ and Cell₂ are defined as composites of the total number of mesh cells on the primary side, NCELL1, plus the desired mesh-cell number on the secondary side, N2, i.e.

$$\text{Cell}_{\#} = \text{NCELL1} + \text{N2 where } \# = 1, 2.$$

Similarly, for interfaces on the secondary side of a TEE component, Iface₁ and Iface₂ are defined as composites of the total number of interfaces on the primary side, NCELL1 + 1, plus the desired interface number on the secondary side, N2, i.e.,

$$\text{Iface}_{\#} = \text{NCELL1} + 1 + \text{N2 where } \# = 1, 2.$$

For the 3D VESSEL component, Cell₁ and Cell₂ (Iface₁ and Iface₂) are defined as composites of 1000 times the relative-cell number in the horizontal X-Y or R-θ plane plus the axial-level number; i.e.,

$$\text{Cell}_\# \text{ or } \text{Iface}_\# = 1000 \cdot (\text{horizontal-plane relative-cell number}) \\ + (\text{axial-level number}) \quad \text{where } \# = 1, 2.$$

When the signal-variable definition spans all cells from Cell_1 to Cell_2 (Iface_1 to Iface_2), a 2D array of mesh cells (interfaces) between the minimum and maximum relative-cell numbers and the minimum and maximum axial-level numbers are considered in the VESSEL component.

Note: TRAC-M/F90, post-Version 3.0. Versions of TRAC-M/F90 released after Version 3.0 will require that the signal variable cell number (input variable ICN1) for FILL, BREAK, and PLENUM components be input as 1. This is not a requirement for Version 3.0, TRAC-M/F77, or MOD2.

For heat-structure components, Ncell_1 and Ncell_2 (Nrow_1 and Nrow_2) are defined as composites of three location numbers: 1000000 times the node number i plus 1000 times the rod/slab number j plus the axial node-cell or node-row number k ; i.e.,

$$\text{Ncell}_\# \text{ or } \text{Nrow}_\# = 1000000 \cdot i + 1000 \cdot j + k \text{ where } \# = 1, 2.$$

Hydro node cell k lies between heat-transfer node rows k and $k + 1$. There are NCRZ node cells and NCRZ + 1 node rows. For signal-variable parameters that are defined on the heat-structure surface, i should be 1 for the inner surface or the value of NODES for the outer surface. When i is not 1 or NODES for surface-defined parameters, TRAC internally sets $i = 1$ for slabs and $i = \text{NODES}$ for rods. Signal-variable parameter numbers 88 and 89 define the total surface heat loss to the vapor or liquid from all NRODS rods/slabs so the TRAC user needs to define j to be 0 for these two signal-variable parameters. When the heat-structure signal-variable definition spans all node cells (node rows) from Ncell_1 to Ncell_2 (Nrow_1 to Nrow_2), a 1D (for the ISVN = 88 and 89 parameters), 2D (for all surface-defined parameters), or 3D array of locations between the minimum and maximum values of i , j , and k are considered in the heat structure.

Using these composite definitions for tee secondary-side mesh cells (interfaces), vessel mesh cells (interfaces), and heat-structure node cells (node rows) enables the TRAC user to define them with a single signal-variable definition form that normally would apply only for single-sided 1D components.

Signal-variable parameter numbers 28 through 39, 65 through 73, and 97 through 99 define mass flow, velocity, hydraulic diameter, and interfacial drag coefficient and have their Iface_1 and Iface_2 numbers associated with mesh-cell interfaces. A 1D component's mesh-cell interface i lies between mesh cells $i-1$ and i . For the 3D VESSEL component, the outer-azimuthal, the outer-radial, and the upper-axial interfaces have the number of their mesh cell. If mass flow, velocity, hydraulic diameter, or interfacial drag coefficient are specified on X (or R) or Y (or θ) interfaces for 1D components, their definitions will internally default to the Z-interface definition.

There are six different forms for defining signal-variable parameters when they are location-dependent, that is, when they have |ISVN| parameter numbers 21 through 40 and 65 through 101. The signal variable is the

1. parameter value in mesh cell $|Cell_1|$ or $|Cell_2|$ when $Cell_2 = 0$ or $Cell_1 = 0$, respectively,
2. maximum value of the parameter in all mesh cells between mesh cells $|Cell_1|$ and $|Cell_2|$ when $Cell_1$ and $Cell_2$ are both positive,
3. minimum value of the parameter in all mesh cells between mesh cells $|Cell_1|$ and $|Cell_2|$ when $Cell_1$ and $Cell_2$ are both negative,
4. volume-averaged value of the parameter in all mesh cells between mesh cells $|Cell_1|$ and $|Cell_2|$ when $Cell_1$ and $Cell_2$ are of opposite signs,
5. difference between the parameter values in mesh cells $|Cell_1|$ and $|Cell_2|$ when the ISVN parameter number is prefixed with a minus sign, and
6. difference between the parameter values at the beginning of the present time step and at the beginning of the previous time step in mesh cell $|Cell_1|$ or $|Cell_2|$ when $Cell_2 = 0$ or $Cell_1 = 0$, respectively, and the ISVN parameter is prefixed with a minus sign.

The fifth and sixth forms can only have parameter numbers -101 through -65 and -40 through -21. The fifth form's signal variable is the parameter value in $|Cell_1|$ minus the parameter value in $|Cell_2|$. Mesh-cell numbers are defined by the absolute (positive) values of $Cell_1$ and $Cell_2$. Forms 3 and 4 above are the only ones that define $|Cell_1|$ and $|Cell_2|$ with negative values. For all six forms, it is the absolute value of $Cell_1$ and $Cell_2$ that is used to define the mesh-cell number.

Parameter number 20 is the liquid level between mesh cells $|Cell_1|$ or $|Cell_2|$. The liquid level

$$\sum_{i=|Cell_1|}^{I \pm 1} \Delta x_i + f \Delta x_I$$

is the geometric length of collapsed liquid coolant from the bottom interface of mesh cell $|Cell_1|$, where Δx_i is the length of mesh cell i . The collapsed liquid-vapor interface is at a distance $f \Delta x_I$ above the bottom interface of mesh cell I . The values of f and I are defined by requiring that the volume of liquid between mesh cells $|Cell_1|$ and $|Cell_2|$ be collapsed to fill the fluid volume of mesh cells $|Cell_1|$ to $I \pm 1$ (minus when $|Cell_1| < |Cell_2|$ and plus when $|Cell_1| > |Cell_2|$) and the fraction f of the volume of mesh cell I ; i.e., f is defined by

$$\sum_{i=|\text{Cell}_1|}^{|\text{Cell}_2|} (1 - \alpha_i)V_i = \sum_{i=|\text{Cell}_1|}^{i \pm 1} V_i + fV_i$$

where α_i and V_i are the void fraction and volume of mesh cell i , respectively. This collapsed liquid column can only be evaluated for mesh cells within a component and only for the primary side or secondary side of a TEE component. To combine the liquid heights in several connected components requires defining a liquid water-level signal variable for each component (two signal variables for both sides of a TEE component) and summing them with control blocks using appropriate weighting factors. The cosine of the angle that the component mesh cells are inclined from vertical needs to be multiplied by the liquid water level to give the liquid's vertical height.

N.4. Control Blocks

Control blocks are function operators that operate on input-parameter signals to produce an output-parameter signal. The output signal can then be used as an input parameter to other control blocks. In this way, a control procedure for a component action can be constructed by coupling control-block operators in series and in parallel. PWR-system parameters, defined by signal variables, provide the input-parameter signals for the initial control block(s) and possible subsequent control blocks in a control procedure. The control procedure ends with a final control block whose output-signal value is used to define a component-action table's independent variable of the action directly when the table has no tabular data.

Table N-4. lists the 66 control-block function operations that are available in TRAC. The user selects the desired function operator for a control block and assigns its evaluation for later use in evaluating the control block. The remaining information in the table is the control-block function name and its defined mathematical operation. XOUT is the control-block output value, which can be referenced with the control-block ID number by other control blocks for their input ICB# or by component-action tables for defining their independent variable. The control-block ID number is specified by the user through input with Fortran variable name IDCB and is required to be a negative integer greater than -10 000. Thus, for a control block, defining ICB# with a positive ID number inputs a signal-variable value to the control block; a negative ID number inputs a control-block output-signal value.

In addition to the five possible X#, L#, and C# control-block input parameters in Table N-4., all control blocks require from input a constant gain factor, G, and constant minimum and maximum limit values, XMIN and XMAX, which constrain the output value of XOUT. XMIN and XMAX enable the user to model the nonlinear saturation of a control signal by electronic hardware. The gain factor can be used to model the amplification or attenuation effect on the signal from such hardware. Variables G, XMIN, XMAX, and C# are input with Fortran variable names CBGAIN, CBXMIN, CBXMAX, and CBCON#, respectively.

**TABLE N-4.
Control-Block Function Operations**

No.	Control-Block Name	Control-Block Mathematical Operational ^a	Block ^{b,c} Input			Block ^c Const.	
			1	2	3	1	2
1	Absolute value	$XOUT = G*ABS(X1).$	X1				
2	Arccosine	$XOUT = G*ACOS(X1),$ XOUT in radians.	X1				
3	Add	$XOUT = G*(X1 + X2).$	X1	X2			
4	Integerizer	$XOUT = G*FLOAT(IFIX(X1)).$	X1				
5	Logical "and"	$XOUT = G$ $IF((L1.EQ.1.0)$ $.AND.(L2.EQ.1.0));$ $= 0.0$ otherwise.	L1	L2			
6	Arcsine	$XOUT = G*ASIN(X1),$ XOUT in radians.	X1				
7	Arctangent	$XOUT = G*ATAN(X1)$ XOUT in radians.	X1				
8	Arctangent 2	$XOUT = G*ATAN(X1,X2),$ XOUT in radians.	X1	X2			
9	Constant	$XOUT = G*C1.$				C1	
10	Cosine	$XOUT = G*COS(X1),$ X1 in radians.	X1				
11	Dead band dead zone or dead space	$XOUT = G*(X1 - C2)$ $IF(X1.GT.C2);$ $= G*(X1 - C1)$ $IF(X1.LT.C1);$ OR $= 0.0$ otherwise	X1			C1	C2
12	Derivative	$XOUT = G*(dX1/dt).$	X1	(X2)			
13	Double integrator with XOUT limited	$XOUT = G*[∫X1dt + X2]dt$ + XOUT, X1 and (X2) are reset to 0.0 if XOUT is against a limit and the sign of X1 does not change. X2 = 0.0 and XOUT = 0.0 initially. X2 is $∫X1dt$ summed over all previous time. C1 = 0.0 or 1.0 is the polynomial order for approximating the time dependence of X1 and its integral.	X1	(X2)	(X3)	C1	(C2)
14	Divide	$XOUT = G*X1/X2$	X1	X2			

TABLE N-4. (cont)
Control-Block Function Operations

No.	Control-Block Name	Control-Block Mathematical Operational ^a	Block ^{b,c} Input			Block ^c Const.	
			1	2	3	1	2
15	Logical "exclusive or"	XOUT = G IF(L1 + L2.EQ.1.0); = 0.0 otherwise.	L1	L2			
16	Logical "equivalent"	XOUT = G IF(L1.EQ.L2); = 0.0 otherwise.	L1	L2			
17	Exponential	XOUT = G*EXP(X1).	X1				
18	Logical "flip-flop"	XOUT = G or 0.0, valued flip-flop output that changes state whenever L1 changes state (only if L3 = 1.0).	L1	(L2)	L3		
19	Gate	XOUT = G*X1 IF(L2.EQ.1.0); = 0.0 IF(L2.EQ.0.0).	X1	X2			
20	Greater than or equal to	XOUT = G IF(X1.GE.X2); = 0.0 otherwise.	X1	(X2)			
21	Greater than	XOUT = G IF(X1.GT.X2) = 0.0 otherwise.	X1	X2			
22	Input switch	XOUT = G*X1 IF(L3.EQ.1.0); = G*X2 IF(L3.EQ.0.0).	X1	X2	L3		
23	Integrate	XOUT = G*∫X1dt + XOUT, XOUT = 0.0 initially. X1 is not reset to 0.0 when XOUT is against its limits. C1 = 0.0 or 1.0 is the polynomial order for approximating the time dependence of X1.	X1			C1	(C2)

TABLE N-4. (cont)
Control-Block Function Operations

No.	Control-Block Name	Control-Block Mathematical Operational ^a	Block ^{b,c} Input			Block ^c Const.	
			1	2	3	1	2
24	Integrate with mode control	$XOUT = 0.0$ IF((L2 + L3).EQ.0.0), reset mode; = XOUT IF((L2 + L3).EQ.1.0), hold mode; or = $G \int X1 dt + XOUT$ IF((L2 + L3).EQ.2.0), integrate mode. C1 = 0.0 or 1.0 is the polynomial order for approximating the time dependence of X1.	X1	L2	L3	C1	(C2)
25	Logical "inclusive or"	$XOUT = 0.0$ IF((L1 + L2).EQ.0.0); = G otherwise.		L1	L2		
26	First-order lag	$XOUT = XO$, where XO is the problem time-domain output from the Laplace transform operator $(G \cdot X1(s) + C1 \cdot XO(t=0)) / (C1 \cdot s + 1)$ operating on the X1(t) input function X1(s) in the Laplace transform s domain; that is, XO(t) is the solution of the first-order differential equation $C1 \cdot dX0 / dt + XO = G \cdot X1$ in time where $C1 \geq 0.0$ and G is a constant gain factor. XO is initialized to $G \cdot X1$ at t=0.	X1			C1	
27	Logic delay	$XOUT = 0.0$ IF((L1.EQ.0.0).OR. (TIME.GT.(C1 + C2))); = G IF((L1.EQ.1.0).AND. (TIME ≤ (C1 + C2))), where (C2) is the time when L1 switches from 0.0 to 1.0.	L1	(L2)		C1	(C2)

TABLE N-4. (cont)
Control-Block Function Operations

No.	Control-Block Name	Control-Block Mathematical Operational ^a	Block ^{b,c} Input			Block ^c Const.	
			1	2	3	1	2
28	Logic general-purpose counter	XOUT = 0.0 IF(L3.EQ.0.0), reset mode; = G times the number of times L1 has changed state since enabled (when L3 = 1.0), count mode.	L1	(L2)	L3		
29	Logic input switch	XOUT = G*L1 IF(L3.EQ.1.0); = G*L2 IF(L3.EQ.0.0).	L1	L2	L3		
30	Lead-lag transfer function'	XOUT = XO, where XO is the problem time-domain output from the Laplace transfer operator (G*X1(s)*(C1*s + 1) + C2*XO(t=0) - C1*G*X1(t=0))/(C2*s + 1) operating on X1(t) input function X1(s) in the Laplace transform s domain; that is, XO(t) is the solution of the first-order differential equation C2*dXO/dt + XO = G*C1*dX1/dt + X1 where C1 ≥ 0.0, C2 ≥ 0.0, and G is a constant gain factor. XO is initialized to G*X1 at t=0.	X1	(X2)	(X3)	C1	C2
31	Limited integrator	XOUT = G* ∫X1dt + XOUT, X1 is set to 0.0 if XOUT is against a limit and the sign of X1 does not change; XOUT = 0.0 initially. C = 0.0 or 1.0 is the polynomial order for approximating the time dependence of X1.	X1			C1	(C2)
32	Natural logarithm	XOUT = G*ALOG(X1).	X1				

TABLE N-4. (cont)
Control-Block Function Operations

No.	Control-Block Name	Control-Block Mathematical Operational ^a	Block ^{b,c} Input			Block ^c Const.	
			1	2	3	1	2
33	Less than or equal to	XOUT = G IF(X1 ≤ X2); = 0.0 otherwise	X1	X2			
34	Less than	XOUT = G IF(X1.LT.X2); = 0.0 otherwise	X1	X2			
35	Maximum of two signals	XOUT = G*AMAX1(X1,X2).	X1	X2			
36	Maximum during transient signals	XOUT = G*AMAX1(X1,XOUT).	X1				
37	Minimum of two	XOUT = G*AMIN1(X1,X2).	X1	X2			
38	Minimum during transient	XOUT = G*AMIN1(X1,XOUT).	X1				
39	Multiply	XOUT = G*X1*X2	X1	X2			
40	Logical "not and"	XOUT = 0.0 IF((L1 + L2).EQ.2.0); = G otherwise.	L1	L2			
41	Logical "not equal"	XOUT = G IF(L1.NE.L2); = 0.0 otherwise.	L1	L2			
42	Logical "not inclusive or"	XOUT = G IF((L1 + L2).EQ.0.0); = 0.0 otherwise.	L1	L2			
43	Logical "not" or negation	XOUT = G IF(L1.EQ.0.0); = 0.0 IF(L1.EQ.1.0).	L1				
44	Positive difference	XOUT = G*(X1 - X2) (IF(X1.GT.X2); = 0.0 otherwise.	X1	X2			
45	Quantizer	XOUT = G*FLOAT (IFIX(X1 + 0.5)) IF(X1.GE.0.0); = G*FLOAT (IFIX(X1 - 0.5)) IF(X1.LT.0.0).					
46	Ramp	XOUT = G*(TIME-C1) IF(TIME.GT.C1); = 0.0 otherwise.					

TABLE N-4. (cont)
Control-Block Function Operations

No.	Control-Block Name	Control-Block Mathematical Operational ^a	Block ^{b,c} Input			Block ^c Const.	
			1	2	3	1	2
47	Random number generator	XOUT = G*RANF where 0.0 ≤ RANF ≤ 1.0 IF(TIME.GE.C1); = 0.0 otherwise.				C1	
48	Sign function	XOUT = +G* X1 IF(X2.GE.0.0); = -G* X1 IF(X2.LT.0.0).	X1	X2			
49	Sine	XOUT = G*SIN(X1), X1 in radians.	X1				
50	Sign inversion	XOUT = -G*X1.	X1				
51	Second-order transfer function	XOUT = XO, where XO is the problem time-domain output from the Laplace transform operator (G*X1(s) + C1*XO(t=0) + C2*(s *XO(t=0) + dXO/dt(t=0)))/(C2*s *s + C1 *s + 1) operating on the X1(t) input function X1(s) in the Laplace transform s domain that is, XO(t) is the second-order differential equation C2* d(dXO/dt)/dt + C1* dXO/dt + XO = G*X1 where C1 ≥ 0.0, C2 ≥ 0.0, and G is a constant gain factor. XO is initialized to G*X1 at t=0.	X1	(X2)	(X3)	C1	C2
52	Square root	XOUT = G*SQRT(X1).	X1				
53	Step	XOUT = G IF(TIME.GE.C1); = 0.0 otherwise.				C1	
54	Subtract	XOUT = G*(X1 - X2).	X1	X2			
55	Tangent	XOUT = G*TAN(X1) X1 in radians.	X1				
56	Sum constant	XOUT = G*(X1 + C1).	X1			C1	
57	Sum three	XOUT = G*(X1 + X2 + X3).	X1	X2	X3		

TABLE N-4. (cont)
Control-Block Function Operations

No.	Control-Block Name	Control-Block Mathematical Operational ^a	Block ^{b,c} Input			Block ^c Const.	
			1	2	3	1	2
58	Variable limiter	$XOUT = G \cdot X2$ IF(X1.GT.X2), at upper limit; $= G \cdot X3$ IF(X1.LT.X3), at lower limit; or $= G \cdot X1$ otherwise between limits.	X1	X2	X3		
59	Weighted summer	$XOUT = G \cdot (C1 \cdot X1 + C2 \cdot X2)$.	X1	X2		C1	C2
60	Exponentiate	$XOUT = G \cdot (X1^{**}X2)$, where $X1 > 0$.	X1	X2			
61	Zero-order hold	$XOUT = G \cdot X1$ IF(L2.EQ.1.0); $= XOUT$ otherwise	X1	L2			
100	Time delay	$XOUT = G \cdot X1$ where X1 is evaluated at the time the control block is input IF(TIME \leq C1); $= G \cdot X2(TIME - C1)$ otherwise {X1(TIME - C1) is evaluated at time (TIME - C1)}. The variable is the number of storage table pairs for saving values of X1 over the last C1 seconds. X1 is stored at intervals of approximately $C1 / (n - 1)$ s; the control block will use linear interpolation to obtain the desired value of X1(TIME - C1).	X1	n	(X3)	C1	
101	Function of one independent variable	$XOUT = G \cdot f_n(X1)$. The variable n is the number of function table pairs of values to be input.	X1	n	(X3)	(C1)	

TABLE N-4. (cont)
Control-Block Function Operations

No.	Control-Block Name	Control-Block Mathematical Operational ^a	Block ^{b,c} Input			Block ^c Const.	
			1	2	3	1	2
102	Function of two or three independent variables	$XOUT = G \cdot f(X1, X2, X3)$. The variable n is the composite number $10000.0 \times N_1 + 100.0 \times N_2 + N_3$, where $N_1 =$ number of X1 values ($2 \leq N_1 \leq 99$), $N_2 =$ number of X2 values ($2 \leq N_2 \leq 99$), and $N_3 =$ number of X3 values ($2 \leq N_3 \leq 99$), Input zero for X3 and N_3 for a tabular function of two independent variables. Input the function table in the following order: the N_1 values of X1, the N_2 values of X2, the N_3 values of X3, and the $[N_1 \cdot N_2 \cdot \max(1, N_3)]$ function values.	X1	X2	X3	n	(C2)
200	PI Controller	$XOUT = A = A_0 + \Delta A$ in Fig. N-1, where X1 = ID number of F, X2 = ID number of F_d when F_d is an input parameter rather than a constant value; C1 = F_d a constant value; C2 = A_0 , the initial XOUT; $G = (\Delta A / \Delta F)$; $XMIN = A_{min}$; and $XMAX = A_{max}$. A third input-data card is required to specify $\Delta t_d > 0.0$ and $\tau \geq 0.0$. Refer to Fig. N-1 for a schematic diagram of the functional form of this control block.	X1	X2		C1	C2

TABLE N-4. (cont)
Control-Block Function Operations

No.	Control-Block Name	Control-Block Mathematical Operational ^a	Block ^{b,c} Input			Block ^c Const.	
			1	2	3	1	2
201	PID Controller	$XOUT = A = A_0 + \Delta A$ in Fig. N-2, where X1 = ID number of F; X2 = ID number of F_d when F_d is an input parameter rather than a constant value; C1 = F_d constant value; C2 = A_0 , the initial XOUT; G = $(\Delta A / \Delta F)$; XMIN = A_{min} ; and XMAX = A_{max} . A third input data-card is required to specify $\Delta t_d > 0.0$, $\tau > 0.0$, and $0.0 \leq W_i \leq 1.0$. Refer to Fig. N-2 for a schematic diagram of the functional form of this control block.	X1	X2	(X3)	C1	C1

- a. XOUT appearing on the right-hand side of the defining equation indicates its value from the previous evaluation. G is a constant gain factor applied to the XOUT operation result. XOUT appearing on the left-hand side of the defining equation is constrained by constant lower and upper limits XMIN and XMAX in defining the control-block output value.
- b. An "X" parameter indicates a continuous variable; an "L" parameter indicates a logical (or discrete) parameter having a value of 0.0 or 1.0 only.
- c. Variables enclosed in parentheses are not input variables, but are used internally by the control block for data storage.

By default, the output value of a control block is in SI units if the control block value has units. The control-block output can be defined to be in non-SI units. This requires setting some NAMELIST input/output option variables and also some additional control-block input that defines the control-block output units. This option is more fully discussed in Chapter 5 of the User's Manual.

Figure N-1. presents the general schematic of a control block. The number of input parameters on the left-hand side ranges from 0 to 3 depending on the control-block operation number. On the other hand, there is only one output parameter from a control block. Its value can vary continuously or stepwise depending on the type of control-block operation. To have the output value 0.0 or 1.0 when it is logical (ICBN = 5, 15, 16, 18, 20, 21, 25, 27, 28, 29, 33, 34, 40, 41, 42, and 43) requires that CBGAIN = 1.0 be input to the control block.

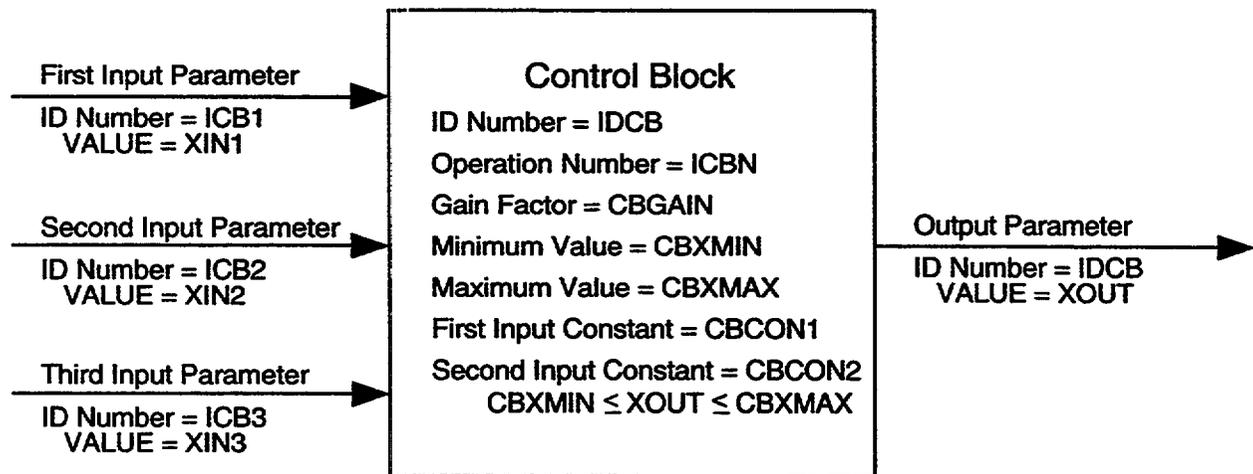


Fig. N-1. Control-block diagram.

TRAC evaluates control blocks by having driver subroutine CBSET call subroutines CONBLK for ICBN = 1 to 61, DELAY for ICBN = 100, LININT for ICBN = 101, and LINT4D for ICBN = 102. Subroutines CONBLK and DELAY are modified forms of subroutines with the same names written by Giles and Milton for the TRAC-BD1 computer program (Ref. N-1). Their control-block operations ICBN = 56 (defining problem time) and 57 (setting trip status), which are done by signal variable ISVN = 0 and trips in TRAC, have been replaced by operations that sum a constant to an input parameter and sum three input parameters. The rectangular-integration control blocks (ICBN = 13, 23, 24, and 31) have been further generalized to allow the user the option of selecting trapezoidal-rule integration instead.

Before evaluating a control block, the signal variables and control blocks that define its input parameters need to have their values accessed. When evaluating a control procedure with connected control blocks, the connecting arrows define the direction and order for evaluating the control blocks. Time lag produced by forward difference approximating an integrand (compared with evaluating the input parameters first using a backward-difference approximation) has an unstabilizing effect. When the error to be corrected in the signal is increasing, corrective action is numerically delayed by a forward-difference approximated integral controller. Eventually when enough corrective action has been applied to start decreasing the error signal, the forward-difference-approximated integral controller delays the removal of corrective action causing the error signal to overshoot its desired zero value. The inherent weakness of integral controllers to overshoot and produce an oscillatory response that can become unstable (Ref. N-2.) is enhanced by evaluating its integral with a forward-difference approximation. Using a backward-difference approximation has just the opposite effect. Its numerical difference error acts to reduce overshoot by the integral controller. This discussion only applies to the rectangular-integration option (which is zero-order

accurate). When trapezoidal-rule integration (which is first-order accurate) is used, the input parameters must be evaluated first for the control-block algorithm to evaluate correctly.

Consider the very useful proportional-plus-integral (PI) controller shown in Fig. N-2. The proportional-plus-integral-plus-derivative (PID) controller shown in Fig. N-3 is an enhancement of the PI controller that uses the derivative to control the rate of adjustment. Both controllers are control blocks that define a control procedure for a component action. The input PWR-system parameter F and output component action A could be a pressure to control a valve, a mass flow rate to control a pump, a fluid level to control a feedwater pump or valve, or a pressure to control a heater. The purpose of these controllers is to compare the system-parameter value with its desired value and to apply corrective action to return the system parameter to its desired value when it departs. These control blocks could be used to model an actual PI or PID electronic-circuit controller or to adjust a component action to maintain a quoted system-parameter value.

The PI controller shown in Fig. N-2 performs the following operations. In the lower-left corner of the figure, the sum-constant function subtracts from the PWR-system parameter F the desired value F_d to define the ΔF error signal. The next three functions estimate the ΔA change in the component action from its initial value, which should cause ΔF to go to zero as the calculation proceeds. The integrate function integrates ΔF from the initial time to the present. Its integral represents the accumulated error in F . The add function sums its ΔF (proportional) and $\int \Delta F dt$ (integral) input signals with appropriate $(\Delta A/\Delta F)_{est}$ and $(\Delta A/\Delta F)_{est}/\Delta t_d$ gain factors to output the PI controller's estimated change in the component action that is required from its initial value. $(\Delta A/\Delta F)_{est}/\Delta t_d \cdot \int \Delta F dt$ corresponds to the ΔA_{PI} value that has accumulated up to the current time; $(\Delta A/\Delta F)_{est} \cdot \Delta F$ corresponds to the desired instantaneous change in ΔA_{PI} at the current time that the PI controller estimates is needed to correct for the ΔF error still remaining. The first-order lag function accounts for the fact that the $(\Delta A/\Delta F)_{est} \cdot \Delta F$ instantaneous change in a component action may not be physically possible or numerically desirable. The maximum rate of change in ΔA is constrained by the CBCON1 time constant, τ , of the first-order lag function. Through the value of τ , the maximum adjustment rate of the hardware can be modeled. When an instantaneous change in ΔA_{PI} over a time step is desired, the first-order lag function evaluates a slower change in ΔA based on the first-order differential equation $\tau * d\Delta A / dt + \Delta A = \Delta A_{PI}$. The value of ΔA will lag the value of ΔA_{PI} in time by an amount based on the magnitude of τ . Finally, the sum-constant function in the lower-right corner of the figure sums the accumulated ΔA change in the component action since the start of the calculation to the component-action's initial value, A_o . For this controller, its control-block output signal A is applied directly to the component action by defining a component-action table with no entry pairs, NXXXTB = 0.

Proportional + Integral (PI) Controller

$$\Delta A_{PI} = (\Delta A / \Delta F)_{est} [\Delta F + (1/\Delta t_d) \int \Delta F dt]$$

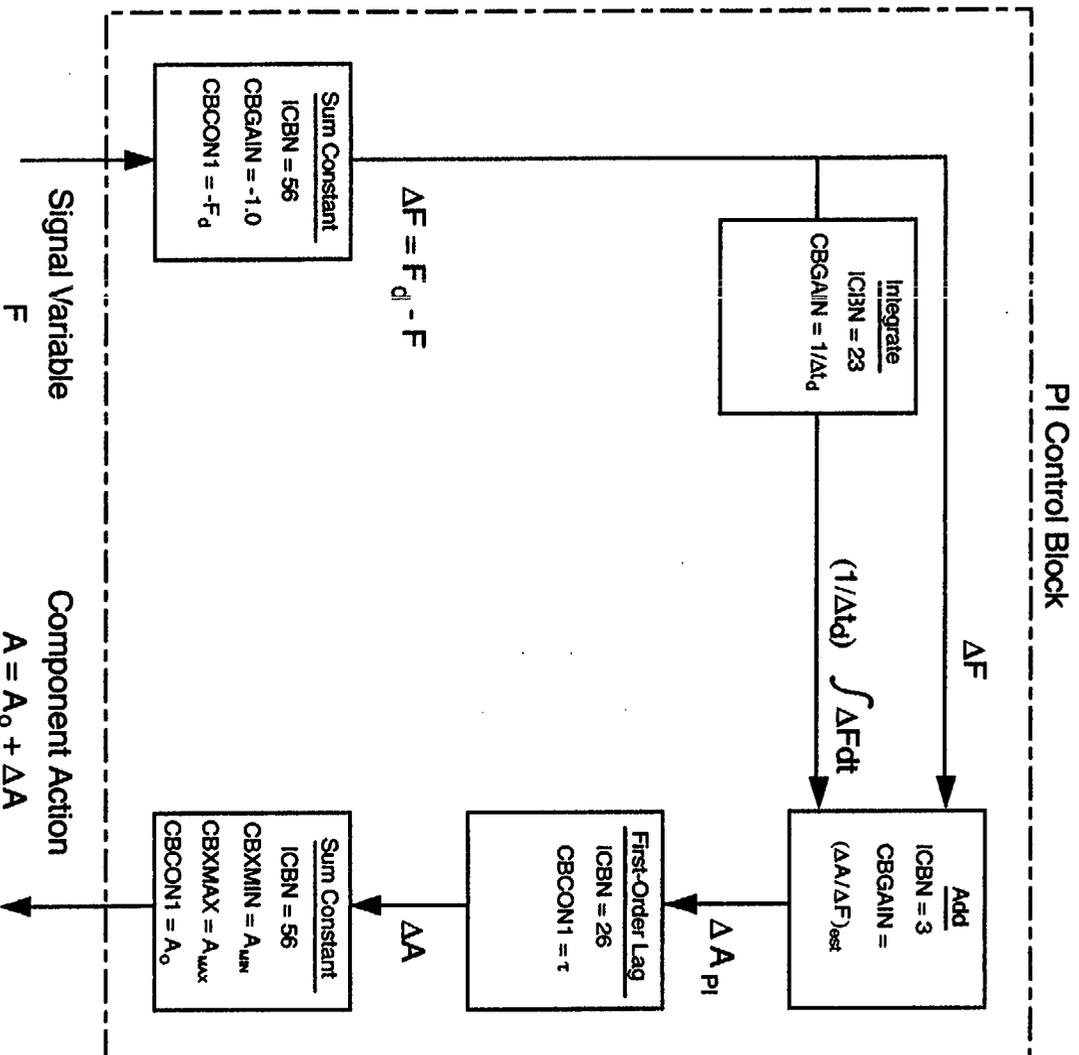


Fig. N-2. Proportional plus integral controller diagram.

Proportional + Integral + Derivative (PID) Controller

$$\Delta A_{PID} = (\Delta A / \Delta F)_{est} \left[\Delta F + (1 / \Delta t_d) \int \Delta F dt + \overline{\Delta t_d (d\Delta F / dt) + \Delta F} \right]$$

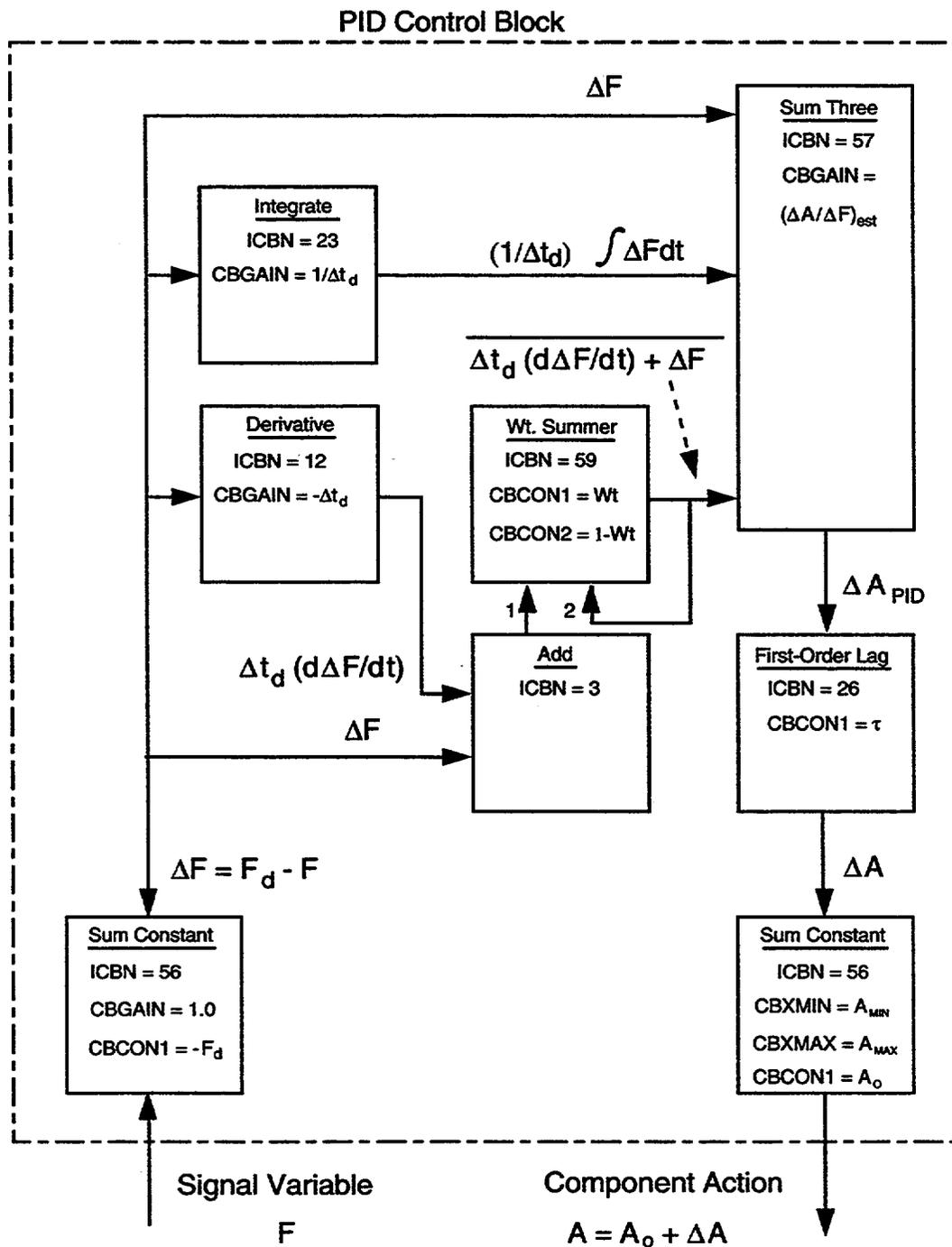


Fig. N-3. Proportional plus integral plus derivative controller diagram.

N.5. Trips

Trips are ON-OFF logical switches for controlling the evaluation of component actions. When the controlling trip for a component action is ON, the component action is evaluated from its component-action table. When the controlling trip is OFF, the component action is not evaluated; it is held constant at its initial value if its trip has never been ON or optionally at its last evaluated value when its trip was ON or at a user-defined input value for when its trip is OFF after being ON. Trips also can be used to implement special time-step data, generate restart-data dumps, and terminate problem execution at the time the trip is set ON.

Determining the ON or OFF set status of a trip is based on the value of its user-defined trip signal. Over the trip-signal value range, the user defines two or four set-point values that delimit two or three subranges labeled with ON or OFF set status. When the value of the trip signal lies within a given subrange, the trip is assigned the set-status label of that subrange. Table N-5. shows 5 of the 10 possible trip-signal-range types that define set status. The ON set status has two forms: $ON_{forward}$ and $ON_{reverse}$. Trip-controlled component action is evaluated for both. They differ only with regard to the sign of the numerical value 1 for the ON set status. The numerical values associated with the three set-status flags in Table N-5. are -1 for $ON_{reverse}$, 0 for OFF, and $+1$ for $ON_{forward}$.

Input variable ISRT defines the signal-range type for a trip. Table N-5. shows the signal-range types for $ISRT = 1, 2, 3, 4,$ and 5 . Assigning a negative sign to these ISRT values and changing $ON_{forward}$ to $ON_{reverse}$ and $ON_{reverse}$ to $ON_{forward}$ defines the other five types. The criteria to decide when the set status of a trip is to be changed are stated in the lower half of Table N-5. The present set status of the trip determines which criterion is to be used. The criteria and signal-range type definitions are based on the requirement that the set-point values satisfy $S_1 < S_2$ for $ISRT = \pm 1$ and ± 2 and $S_1 < S_2 < S_3 < S_4$ for $ISRT = \pm 3, \pm 4,$ and ± 5 .

For $ISRT = \pm 1$ and ± 2 , there are two set-status subranges. The trip signal needs to be equal to or greater than the S_2 set-point value to change from the left to the right subrange set status. To change from the right to the left subrange set status, the trip signal needs to be equal to or less than S_1 . When the trip-signal value lies between S_1 and S_2 , the trip has the set status of the left or right subrange from which its signal came.

For $ISRT = \pm 3, \pm 4,$ and ± 5 , there are three set-status subranges. Changing to either of the other two set-status labels is determined in much the same way as for a two-subrange signal-range type. The trip-signal value is compared with S_1 or S_2 to change to left or middle set status and compared with S_3 or S_4 to change to the middle or right set status. The trip-signal value crossing two subranges during a time step will result in the set-status criteria being satisfied twice even though the last criterion being satisfied is the only one edited by TRAC for the time step.

**TABLE N-5.
Trip Signal-Range Types**

ISRT Signal-Range Type Number	Trip Signal-Range Diagram with Subrange Set Status			
1	ON _{forward}	OFF		Trip Signal
	S ₁	S ₂	→	
2	OFF	ON _{forward}		Trip Signal
	S ₁	S ₂	→	
4	ON _{forward}	OFF	ON _{reverse}	Trip Signal
	S ₁	S ₂	S ₄	
5	OFF	ON _{forward}	OFF	Trip Signal
	S ₁	S ₂	S ₄	
	ON _{forward}	OFF	ON _{forward}	Trip Signal
	S ₁	S ₂	S ₃ S ₄	

Making ISRT negative changes ON_{forward} to ON_{reverse} and ON_{reverse} to ON_{forward}.

ISRT Signal-Range Type Number	Description of How the Trip Set Status is Changed
1	When ON _{forward} trip is set to OFF when the trip signal $\geq S_2$. When OFF, trip is set to ON _{forward} when the trip signal $\leq S_1$.
2	When OFF, trip is set to ON _{forward} when the trip signal $\geq S_2$. When ON _{forward} trip is set to OFF when the trip signal $\leq S_1$.
3	When ON _{forward} trip is set to OFF when the trip signal $\geq S_2$ and $< S_4$. When ON _{forward} trip is set to ON _{reverse} when the trip signal $\geq S_4$. When OFF, the trip is set to ON _{forward} when the trip signal $< S_1$. When OFF, the trip is set to ON _{reverse} when the trip signal $\geq S_4$. When ON _{reverse} trip is set to OFF when trip signal $\leq S_3$ and $> S_1$. When ON _{reverse} trip is set to ON _{forward} when trip signal $\leq S_1$.
4	When OFF _(left) trip is set to ON _{forward} when signal $\geq S_2$ and $< S_4$. When ON _{forward} trip is set to OFF when trip signal $\leq S_1$ or $\geq S_4$. When OFF _(right) trip is set to ON _{forward} when signal $\leq S_3$ and $> S_1$.
5	When ON _{forward(left)} trip is set to OFF when signal $\geq S_2$ and $< S_4$. When OFF, trip is set to ON _{forward} when trip signal $\leq S_1$ or $\geq S_4$. When ON _{forward(right)} trip is set to OFF when signal $\leq S_3$ and $> S_1$.

Defining a trip-signal range to have more than three set-status subranges can be done by joining two or more trips having the same trip signal and different set-point values. A trip-controlled trip (defined later in this section) is used to join them and to respond as if it were a four or more set-status subrange trip. Whereas trips with three set-status subranges are defined directly by $ISRT = \pm 3, \pm 4, \text{ and } \pm 5$, they could be constructed by joining two trips having two set-status subranges with a trip-controlled trip. Three set-status subrange trips have been defined directly, however, because they are a convenient form for regulating a component action under trip control and save the user from having to define three trips to achieve that form.

Signal-range type $ISRT = \pm 3$ is the form generally used to regulate a component action under trip control. The trip signal is the controlling parameter, and the component action is to be adjusted so that the trip signal lies within its desired value range from S_1 to S_4 . When the trip signal falls below S_1 , the left-subrange ON set status has the component action evaluated in a manner that causes the trip-signal value to increase. When the trip signal rises above S_4 , the right-subrange ON set status adjusts the component action in the opposite manner to cause the trip-signal value to decrease. For this application, the component-action table needs to have its independent-variable form defined by the change in its $IXXXSV$ variable value so that the $ISET$ set-status value is applied as a factor to the independent variable of Table N-2. This is done by defining the number of entry pairs in the component-action table to be negative, that is, $NXXXTB < 0$. As described above, the set-point values define the desired value range for the trip signal. The range can be as small or as large as the user feels is appropriate for constraining the trip-signal value based on trip control of the component action.

Each trip-signal set point is defined by a constant value with the possibility of a tabular function being applied to that value as a factor. The user inputs for each set point a constant set-point value to array variable $SETP(I)$, $I = (1,2)$ or $I = (1,4)$ [that satisfies $S_1 < S_2$ or $S_1 < S_2 < S_3 < S_4$], and an ID flag to array variable $IFSP(I)$ to indicate if a tabular-function factor is to be applied to the $SETP(I)$ value. No such factor is applied when $IFSP(I) = 0$; the set-point value is the constant $SETP(I)$. Entering $IFSP(I) \neq 0$ defines the ID number for a set-point factor table of tabular data to be input after the trip and trip-signal defining data. The set-point factor-table independent variable is defined by a signal variable or a control-block output variable. The variable (not the change in the variable value) defines the independent variable of the table. When a set-point factor is to be applied to a set point, its table is linearly interpolated for each time step on the basis of the abscissa-coordinate signal-variable or control-block output-variable value. The input $SETP(I)$ constant value is multiplied by the interpolated function value to determine the set-point value that is used. Set-point factor tables are useful when set points for regulating a component action vary on the basis of some system-parameter dependence. For example, the desired water level in the steam-generator secondary-side downcomer might depend on the reactor power level. A trip with varying water-level set-point values, dependent on the reactor power level, could be used to control the secondary-side feedwater pump or valve.

Associated with each set point is a delay time defined by input array variable DTSP(I), $I = (1,2)$ or $I = (1,4)$. The delay time provides a second criterion that needs to be satisfied before the trip set status is changed. The first criterion requires that the trip-signal value reach or cross the set-point value tested. Once that criterion is met, the second criterion then requires that the delay time of that set-point pass before changing the trip set status. With a delay time, the user is able to model the time necessary for the system hardware or the control-room operator to process the controlling signal and initiate hardware action. Delay time needs to be defined with a non-negative value (negative values are treated as zero). TRAC edits a message when each criterion is satisfied by a trip. For large delay times, it is possible for a number of first-criterion tests to be satisfied before their delay times pass. TRAC can handle up to five such delay-time pending set-status changes stacking up for a trip before the next one is forced to replace the last one in the stack. When it does, a warning message is printed. For a delay-time pending set-status change to stack up, the time its first criterion was met (based on linear interpolation of the trip-signal value to equal the set-point value) plus its delay time must be later than all pending times added earlier to the stack. When that time is equal to or less than one or more pending times in the stack, they are discarded from the stack by this new entry. Through this mechanism, the user can prevent temporary spikes in a trip signal from setting a trip ON and its component action being evaluated. The user does this by defining shorter delay times for the set points that turn a trip OFF than for the set points that turn a trip ON. The difference between these two delay times is the required time interval the trip-signal value needs to stay in the ON subrange to prevent a satisfied OFF criterion that follows from discarding its pending ON set-status change from the stack.

There are three forms for defining the trip signal: a signal parameter, a signal expression, or a trip set-status signal. Two input variables define the trip signal: ITST defines the trip-signal type number and IDSG defines the trip-signal ID number. A signal-parameter trip is defined by $ITST = 1$ and has a signal-parameter trip signal. That signal parameter is a signal variable with ID number IDSG when $IDSG > 0$ or a control-block output variable with ID number IDSG when $IDSG < 0$. A signal-expression trip is defined by $ITST = 2$ and has a signal-expression trip signal. The trip signal is an arithmetic expression operating on the values of one or more signal variables and control-block output variables. Its definition is input as trip-signal-expression input data with ID number IDSG. The form for that definition is described in the next paragraph. A trip-controlled trip (or trip set-status signal trip) is defined by $ITST = 3$ and has a trip set-status signal. This signal is the sum or product of the set-status values of two or more other trips. The ID numbers of those trips are input as trip-controlled trip-signal data with ID number IDSG. The set-status values of those trips are to be summed if $IDSG > 0$ or multiplied if $IDSG < 0$ to evaluate the trip-controlled trip signal.

Evaluating a signal-expression trip signal involves evaluating an arithmetic expression with 1 to 10 arithmetic subexpressions. Each subexpression has an arithmetic operator and two arguments. The arithmetic operators are addition, subtraction, multiplication, division, exponentiation, maximum value, minimum value, and absolute value. Argument values are defined by signal variables, control-block output variables, input

constants, or an earlier evaluated subexpression in the list of subexpressions. The value of the signal expression is the value of the last subexpression in its list. Through its arithmetic operators, a signal expression allows signal variables from different components of the modeled PWR system to be combined in defining the trip signal. Signal variables that define a subexpression argument can be flagged in the signal-expression definition for reevaluation only at the times the signal-expression trip is set ON. In this way, signal variables evaluated at different times as well as in different components can be combined in defining a trip signal.

The AND or OR logic of blocking or coincidence trips can be implemented with trip-controlled trips. Applying the AND logical operator defines a blocking trip that is ON only if all the trips assigned to it are ON. The trip-controlled trip that has this form multiplies ($IDSG < 0$) the set-status values of its assigned trips in defining the trip signal and uses an $ISRT = 2$ signal-range type with $0 < S_1 < S_2 < 1$. It turns out in this case that the trip-controlled trip signal has the same value (0 or 1) as its subrange set-status value. Applying the OR logical operator defines a coincidence trip that is ON only if one or more of the assigned trips are ON. Its corresponding trip-controlled trip adds ($IDSG > 0$) the set-status values of its assigned trips and uses the same signal-range type and set-point values as above. For a more general coincidence trip, we specify that a number N or more (where $1 \leq N \leq N_i$) of the N_i assigned trips need to be ON for the coincidence trip to be ON. Within AND or OR logic, this requires logical expressions having logical subexpressions. For example, if $N = 2$ of $N_i = 3$ assigned trips need to be ON for the coincidence trip to be ON, its logical expression would need to have the form: (Trip #1 AND Trip #2) OR (Trip #2 AND Trip #3) OR (Trip #3 AND Trip #1). The form of this logical expression will be different for different values of N or N_i . With trip-controlled trips, the required form is the sum of the set-status values of its assigned trips in all cases. The only difference is in defining the set points according to $(N - 1) < S_1 < S_2 < N$. In this case, the trip-controlled trip-signal value, which is the number of assigned trips that are ON, needs to be greater than S_2 (that is, the value N or greater) to meet the criterion for setting the trip-controlled trip ON.

The three trip-controlled trip examples above barely scratch the surface of the number of ways trip-controlled trips can be used to model AND/OR logical expressions. Switching from an $ISRT = 2$ to an $ISRT = 1$ signal-range type applies a NOT logical operator to the logical expression being modeled. Assigning a trip-controlled trip to the definition of another trip-controlled trip signal provides further modeling flexibility. The above discussion assumes that the ON set status of the assigned trips is $ON_{forward}$. Assigning trips with $ON_{reverse}$ set-status subranges to a trip-controlled trip signal provides another dimension for modeling possible logical expressions.

During input, trips are given ID numbers that assign them to

1. control component actions by defining their ID numbers to the general form variables $IDXXTR$ in the component input data,

2. define trip-controlled trip signals with their set-status values, and
3. define signal variables with the trip-signal or set-status value.

Trip ID numbers of magnitude 1 to 9999 are allowed. A negative ID number defines a trip that is to be evaluated during steady-state and transient calculations. A positive ID number trip, on the other hand, is evaluated only during transient calculations. During steady-state calculations, the set-status value specified by input-variable ISET and an initialized trip-signal value of 10^{20} define the positive ID trip constant state. A positive ID trip can be used as a blocking trip during a steady-state or transient calculation when the trip-evaluated set status is the opposite of its input ISET value.

N.6. Control-Parameter Evaluation

Signal variables, control blocks, and trips are assigned ID numbers when input by the user. Their ID numbers provide the means for referencing their values for use in the control procedure. Their ID numbers also define the order in which they are to be evaluated. Within each control-parameter type (signal variables, control blocks, and trips), parameters are evaluated in the order of increasing magnitude of their ID numbers. Thus, control blocks are ordered for evaluation with their ID numbers decreasing in negative value; trips are ordered with their ID numbers increasing in magnitude with either a plus or minus sign. This ordering is done by subroutine ORDER after the control-parameter data are input, and the control-parameter restart data with ID numbers different from those input are added on. For example, a control-block input with ID number -5 would replace a control block with the same ID number from the restart file. Subroutine ORDER would reorder the list of control blocks so that control-block ID = -5 would be placed before control blocks with IDs < -5 and after control blocks with IDs > -5. A trip input with ID number 11 would replace a trip with the same ID from the restart file but not a trip with ID number -11. For the case of two trips having ID numbers with the same magnitude, that is, 11 and -11, subroutine ORDER does not change their relative positions in the trip list. With trip ID = 11 input and trip ID = -11 added on to the trip list from the restart file, trip ID = -11 would follow after trip ID = 11 in the reordered list of trips. Subroutine ORDER is called once at the end of input and restart data processing to reorder the signal-variable, control-block, and trip lists. This saves having to search through ID numbers to determine their evaluation order when the control parameters are evaluated at the beginning of each time step.

Through ID numbers, the order for evaluating control parameters of a given type is defined. The order for evaluating control-parameter types is by default signal variables first, then control blocks, and finally trips. Since signal variables have almost all of their parameters defined by the modeled PWR system, it is appropriate to evaluate them first. The input parameters for control blocks are signal variables and control blocks so that, with proper ordering of control blocks, control blocks are evaluated after signal variables. Trip signals are defined by signal variables, control blocks, and the set status of other trips. With proper ordering of trips, evaluation of trip signals follows the evaluation of signal variables and control blocks.

In general, this approach is able to evaluate control-parameter values at the beginning of the time-step state of the PWR system. There are a few situations, however, where such is not the case. Defining a trip signal or set status to be a signal variable (ISVN = 55 or 56) gives its value at the beginning of the previous time step because signal variables are evaluated before trips. In this case, they should be evaluated after their trips. A control-block procedure with an implicit feedback loop (Fig. N-4, for example) is another exception. To evaluate X with the Add operator control block, $\ln(X)$ and thus X need to be known. By evaluating each of the two control blocks once, no matter which is evaluated first, the value of X used to evaluate $\ln(X)$ will be one time step behind the value that should be used. Accumulation of such errors over many time steps can cause the value of X to significantly lag its actual value in time. Instability can even result from that lag error. TRAC has not yet been programmed to recognize such implicit-loop structures to implement internally an iterative-loop evaluation able to determine the consistent-solution state.

A multipass control-parameter evaluation procedure has been programmed in TRAC to aid the user in addressing the above two situations. Within each evaluation pass, the user decides which signal variables, control blocks, and trips (in that order) are to be evaluated. Trip signal and set-status signal variables can then be evaluated on a second control-parameter evaluation pass after their defining trips were evaluated on the first pass. The implicit control-block loop can be reevaluated with as many passes as the user decides are needed for convergence. On the last loop evaluation pass, all control blocks whose input parameters are affected by the loop evaluation would then be evaluated. During each of the passes, the user specifies which control parameters, if any, are to be evaluated for each type. Those parameters that are evaluated span a range of ID numbers that the user specifies. All control parameters with ID numbers in that range are evaluated during that pass. An example of such a procedure follows. Assume the following control parameters are to be evaluated:

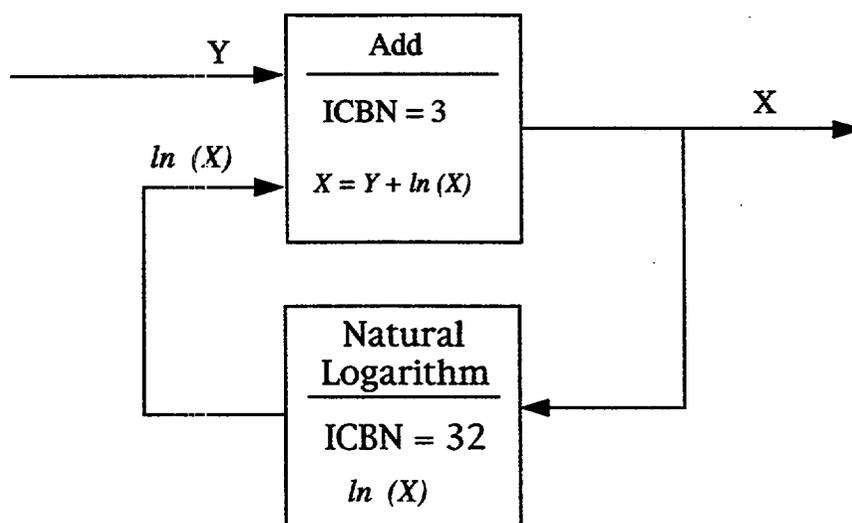


Fig. N-4. Implicit control-block loop example.

1. Signal variables with IDs 3 to 35 define system (component) parameters; signal variables with IDs 39 to 72 define the signal and set status of trips with IDs ± 7 to ± 18 as well as define other system parameters.
2. Control blocks with IDs -1 to -12 require input parameters from signal variables with IDs 3 to 17 and provide input parameters for control blocks in an implicit loop; control blocks with IDs -13 to -17 are in an implicit loop for which five iterative evaluations are felt to be sufficient to converge on a consistent solution state and control blocks with IDs -18 to -42 require input parameters from the implicit-loop control blocks and from signal variables.
3. Trips with ID ± 1 to ± 18 require signal variables with IDs 3 to 26 and control blocks with IDs -6 to -10 to define their signals; trips with IDs ± 19 to ± 30 require signal variables with IDs 18 to 72 and control blocks with IDs -5 to -40 to define their signals.

Evaluating these control parameters could be done with the five evaluation passes shown in Table N-6. After five passes (assuming convergence of the implicit-loop evaluation), all control parameters would have values consistent with each other and the beginning-of-time-step state of the system.

TABLE N-6.
Table ID Numbers of Control Parameters Evaluated During
each of Five Control-Parameter Evaluation Passes

Evaluation Pass Number	Signal Variables	Control Blocks	Trips
1	3 to 35	-1 to -17	± 1 to ± 18
2	39 to 75	-13 to -17	none
3	none	-13 to -17	none
4	none	-13 to -17	none
5	none	-13 to -42	± 19 to ± 30

N.7. Steady-State Evaluation

The TRAC steady-state evaluation capability provides a time-independent solution that may be of interest in its own right, or as an initial condition for a transient calculation. There are three types of steady-state calculations: generalized, constrained, and static check. A generalized steady-state (GSS) calculation evaluates the time-independent condition of an arbitrarily modeled system with variable system parameters (pump

rotational speed, valve flow area, energy sources, etc.) held constant at their input-specified steady-state values. A constrained steady-state (CSS) calculation is performed like a GSS calculation with the addition of user-selected controllers that adjust specific system parameters to achieve desired steady-state values for specific thermal-hydraulic parameters. A static-check steady-state calculation checks for the presence of unknown or erroneous momentum or energy sources in the modeled system by setting the rotational speed of all pumps and energy sources to zero. All coolant flow in the system should go to zero because of surface friction as the static-check calculation is evaluated.

N.7.1. Generalized Steady State

The same subroutines are called to evaluate the transient and steady-state calculation. For the GSS calculation, however, the following additional procedures are implemented.

1. Pressurizer components are modeled as constant pressure boundary conditions during steady-state calculations. Their coolant mass and energy inventory as well as pressure remain constant regardless of the coolant mass-flow conditions between the pressurizer and its adjoining components.
2. Pump components have their momentum sources averaged over time to prevent oscillations in the solution.
3. The point-reactor kinetics or input-specified power option is not evaluated for the powered heat-structure components. Instead, the nuclear core power during the beginning of the calculation is defined by the product of a constant power-to-flow ratio times the coolant mass flow through the reactor-core region. [The power-to-flow ratio is defined by the difference between liquid-coolant internal energies evaluated by subroutine THERMO, and is based on the input-specified liquid-coolant temperatures in the mesh cells above and below the NCRZ hydro cells coupled to the heat structure. Cells (levels) 1 and NCELLS (NASX) are used when the heat structure is coupled to these cells (levels).] Later in the calculation when the coolant mass flow through the reactor core is approaching its asymptotic value and satisfies the criterion, $((V_{L_i}^{n+1} - V_{L_i}^n) / [|V_{L_i}^{n+1}|(t^{n+1} - t^n)]) \leq \text{EPSPOW} = 0.1$ for mesh cells i along the lower interface of the reactor core, the power is set for the remainder of the calculation at its steady-state power level, RPOWRI. (In addition, $|V_{L_i}^{n+1}|$ must be $>10^{-4} \text{ m}\cdot\text{s}^{-1}$ for all such cells.) for the non-typical case where $(1-\alpha_i) \leq 10^{-4}$, analogous tests are made for the gas flow. For $0.0001 < \alpha_i < 0.9999$, both the liquid and gas flows must satisfy the flow criteria. These tests are not effective until at least 5 time steps have been completed.

The user can override this logic with NAMELIST variables IPOWR (an integer switch) and TPOWR (a problem time). The defaults are IPOWR = 0

and TPOWR = 10³⁰ seconds. With these defaults, the flow logic described above controls the steady-state reactor power. With IPOWR = 0, the user can choose any time, TPOWR, to activate the steady-state power, RPOWRI; the power will be set to RPOWRI at TPOWRI or that time determined by the flow logic, whichever occurs first. With IPOWR = -1, only TPOWR will determine the time. For both IPOWR = 0 and -1, the power-to-flow-ratio logic is in effect until RPOWRI is set on. Finally, IPOWR = 1 will set RPOWRI on at the start of the steady-state calculation.

4. The time-step sizes for the heat-transfer and fluid-flow calculations are not required to be equal during a steady-state calculation. The user may specify the ratio of these time-step sizes by the time-step data parameter RTWFP. This permits compensation for the difference between the natural time scales of the two processes. Caution should be taken so that the diffusion limit in the rod axial heat conduction is not exceeded because the time-step ratio is too large. Inputting namelist variable NRSLV = 1 and evaluating an implicit axial heat conduction calculation eliminates this concern.
5. During steady-state calculations, trips with positive identification numbers are not evaluated; only trips with negative identification numbers have their set status evaluated each time step. Even though conditions may exist that would cause a change in the set status of positive identification-number trips, their set status remains constant at its input-specified value.
6. Only trip-controlled component actions with the set status of their trip ON are evaluated during a steady-state calculation. All other component actions are held constant at their initial input value even though the input data for the component define a varying component action for the transient calculation.

Steady-state calculations are transient calculations with the above procedures that converge in time to a time-independent solution state of the thermal-hydraulic system. The steady-state calculation ends when the problem time reaches the TEND ending time of the last time-step data set from file TRACIN or when the following convergence criteria, which are tested every ten time steps, are all satisfied:

- for the mesh-cell-centered parameters, where x_i = coolant pressure, void fraction, liquid temperature, vapor temperature, and air pressure,

$$\max_i \left| \left(x_i^{n+1} - x_i^n \right) / \left[x_i^{n+1} \times \left(t^{n+1} - t^n \right) \right] \right| < \text{EPSS}$$

for all mesh cells i in all hydro components; and

- for the mesh-cell-interface velocities V_{m_i} , where $m = \ell$ for liquid coolant and $m = v$ for vapor coolant,

$$\max_i \left(\left| V_{m_i}^{n+1} - V_{m_i}^n \right| / \left[\max(1, |V_{m_i}^{n+1}|) \times (t^{n+1} - t^n) \right] \right) < \text{EPSS}$$

for all mesh-cell interfaces i in all velocity component directions in all hydro components. The value of EPSS is specified by the user through input (word 3 on Main-Data Card 5). The liquid parameters are tested only when $(1 - \alpha_i) > \text{MAXFLN} = 10^{-4}$, vapor parameters are tested only when $\alpha_i > \text{MAXFLN}$, and the void fraction is tested only when $\text{MAXFLN} < \alpha_i < (1 - \text{MAXFLN})$ and x_i^{n+1} in the denominator is $\text{AMAX1}(\alpha_i, 1 - \alpha_i)$ where α_i the void fraction in mesh cell i . After every 100 time steps or when a large edit is performed, the maximum value of each convergence criterion and its component mesh-cell location are edited to the terminal and message file.

If the TRAC steady-state calculation reaches the ending time of the last time-step data set and all the parameter convergence criteria have not been met simultaneously, the warning message "steady-state solution not converged" is edited to the terminal, message file, and output file before the calculation ends.

N.7.2. Constrained Steady State

The GSS calculation described in the previous section holds all variable system parameters (pump speed, valve closure, physical dimensions, etc.) constant at their initial input-specified values and solves for the system's time-independent fluid-state solution. This solution generally differs from measured steady-state fluid conditions in an actual system because the input-specified description of the physical system usually is not measured or known precisely and the thermal-hydraulic numerical models and phenomena parameters in TRAC are based on various areas of approximation. All this simulation error gets concentrated in the steady-state fluid solution. The TRAC user may prefer instead to allow the physical-system description to vary from its measured or "best-estimate" values so that some of the steady-state solution's thermal-hydraulic parameters are evaluated to have their measured values. For example, given the measured pump-impeller rotational speed and coolant mass flow, the TRAC user may prefer to allow the impeller speed to vary from its measured speed in order to calculate the measured coolant mass flow through the pump. The TRAC user may prefer to calculate the measured upstream coolant pressure or coolant mass flow through a valve by varying the valve closure state from its measured or best-estimate value. What the TRAC user needs is the ability to constrain selected thermal-hydraulic parameters to their measured or specified values because he believes these values to be accurate and important to the phenomena of interest. This can be done by varying other thermal-hydraulic and physical-system parameters, which are less well known or less important, from their measured or estimated values. By selectively biasing steady-state simulation error to less important system parameters and physical-hardware conditions, the TRAC user can achieve a more accurate steady-state solution and initial condition for analysis of transient system behavior.

A CSS calculation provides the TRAC user with this capability. The user selects and specifies through input proportional plus integral (PI) controllers to adjust specific system parameters in a manner that drives and then constrains selected thermal-hydraulic parameters to desired steady-state values. The TRAC user selects from four types of controllers in Table N-7, that are defined internally in TRAC.

Among those four types are many possible combinations of adjusted and monitored parameters. For each of those combinations are different component types in a modeled system to which they may be applied. Several dozen controllers may be selected for a full-plant model steady-state calculation. For example, a full-plant simulation may have as many as a dozen pumps modeled on the primary and secondary sides of several coolant loops to which the TRAC user may decide to apply a type 1 or 3 controller to each or some portion of these pumps.

The specification of controllers through user-defined input is a simple process. First, the TRAC user chooses the hydro or heat-structure components to which a controller will be applied. In this process, the user selects the adjusted parameter component action that will be varied to constrain the monitored parameter to a desired value. Then, the user defines the desired value of the monitored parameter by the input-specified value for the monitored parameter at its location in the component data. The only other required input for each selected controller is the allowable range over which the adjusted parameter can be varied. This range is defined by a minimum and a maximum value for the adjusted parameter. In this way, the TRAC user constrains the adjusted parameter so that it is not varied outside of an acceptable value range. When the adjusted parameter is constrained to its minimum or maximum value, the controller will no longer be able to constrain the monitored parameter to its desired value.

TABLE N-7.
CSS Calculation Controller Types

Controller Type Number	Adjusted Parameter Allowed to Vary	Monitored Parameter Constrained to Desired Value
1	Pump rotational speed	Coolant mass flow through pump
2	Valve flow area	Valve upstream pressure or coolant mass flow through valve.
3	Pump rotational speed, valve flow area, or fill coolant mass flow	Coolant mass flow through pump, valve, or fill is constrained to equal coolant mass flow at a location elsewhere in modeled system
5	Heat-structure heat-transfer area, thermal conductivity, or coupled hydraulic channel pressure	Single-phase coolant temperature or two-phase gas volume fraction

TRAC uses control-block function 200 to internally define and evaluate the PI controller that is used by each controller type. The only exception is the direct definition of the FILL component's coolant mass flow such that it equals the coolant mass flow monitored elsewhere in the system for a type 3 controller. All PI controller parameters have representative values defined internally in TRAC so that the user does not have to specify appropriate values. The tau (τ) parameter for the first-order lag function of the PI controller is set to 2 s. (For a type 5 controller, tau is constrained to be no less than 5 times the current or previous time-step size.) The Δt_d convergence time constant is set to 4 s for controller types 1 and 2, and to 10 s for controller types 3 and 5. The most sensitive parameter that affects controller adjustment is its gain factor, $(\Delta A/\Delta F)_{est}$. Seven different power-plant models, including some with multiple variations, apply all combinations of controller types on both the primary and secondary sides. These models were tested (Refs. N-3. and N-4.) to determine the applicability of the following gain factors programmed in TRAC:

ROMEGA/(RRHO*RFLOW) for a pump in type 1 and 3 controllers where ROMEGA, RRHO, and RFLOW are pump-characteristic input data,

$-10^5 \leq [-\max(10^{-2}, 1-f_A) / \max(10^4, \Delta P_{o_valve})] \leq -10^{-7}$ for a valve with monitored upstream pressure in a type 2 controller,

$5 \times 10^{-4} \leq [\max(10^{-2}, f_A) / \max(5, \dot{m}_{o_valve})] \leq -10^{-2}$ for a valve with monitored coolant mass flow in type 2 and 3 controllers, and

various values for type 5 controllers, which can depend on the adjusted and monitored parameters and the heat-structure wall thickness and nodalization. (Details are given in Ref. N-4.)

Each controller application has a unique value for its optimum gain factor. Fortunately, the integral property of PI controllers makes them robust and stable enough to converge their adjustment with a gain factor that differs by as much as an order of magnitude from its optimum value. The TRAC user can help to provide an appropriate gain factor for valve adjustment by input defining representative conditions for pressure drop or coolant mass flow and valve closure at the valve interface. If the usual input is provided, the valve controller's gain-factor value is constrained by its above definition to an acceptable value.

The adjustment of each controller indirectly feeds back through the thermal-hydraulic solution on the entire system and the other controllers. Direct coupling can occur in one situation, however. A type 2 controller that is adjusting a valve to achieve a desired upstream pressure is directly affected by a type 5 controller that is adjusting the pressure in the same loop. When this interaction occurs, the heat-structure pressure-adjustment factor is applied to the desired pressure upstream of the valve as well as to the pressure of all BREAK components with a fluid-path connection to the heat structure.

N.7.3. Static-Check Steady State

A static-check steady-state calculation checks for the presence of unknown or extraneous momentum and energy sources in the modeled system. TRAC internally turns off the momentum source in all pumps by setting their impeller rotational speed to zero. The energy sources defined for all components also are set to zero. If the gravity vectors that are defined at all mesh-cell interfaces through input are consistent and the modeled system is connected properly, all fluid motion should go to zero eventually and the temperature of all the fluid and structure with heat-transfer paths should equilibrate during the static-check calculation. These results can be used as a positive test to determine that no unknown or extraneous momentum sources exist in the modeled system. The time required for this calculation can be reduced significantly by starting with no fluid motion in the system and with the same fluid and structural temperatures in a mesh cell.

REFERENCES

- N-1. M. M. Giles and J. D. Milton, "TRAC-BD1 Control System Model," TRAC-BWR Completion report WR-CD-81-056 (November 1981).
- N-2. H. L. Harrison and J. G. Bollinger, *Introduction to Automatic Control*, (International Textbook Company, Scranton, Pennsylvania, 1964), p. 144.
- N-3. R. G. Steinke, "Constrained Steady-State Calculation in TRAC-PF1," Los Alamos National Laboratory document LA-UR-89-2123 (July 1989).
- N-4. R. G. Steinke, "Constrained Steady-State Type-5 Controller in TRAC-PF1/MOD2," Los Alamos National Laboratory document LA-UR-96-0996 (March 1996).

BIBLIOGRAPHIC DATA SHEET

(See instructions on the reverse)

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

NUREG/CR-6724

2. TITLE AND SUBTITLE

TRAC-M/FORTRAN 90
(Version 3.0)
Theory Manual

3. DATE REPORT PUBLISHED

MONTH	YEAR
July	2001

4. FIN OR GRANT NUMBER
W6245

5. AUTHOR(S)

J.W. Spore, J.S. Elson, S.J. Jolly-Woodruff, T.D. Knight, J.-C. Lin,
R.A. Nelson, K.O. Pasamehmetoglu, LANL

J.H. Mahaffy, C. Murray, PSU

6. TYPE OF REPORT

Technical

7. PERIOD COVERED *(Inclusive Dates)*

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

Los Alamos National Laboratory
Los Alamos, NM 87545

Pennsylvania State University
University Park, PA 16802

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

Division of Systems Analysis and Regulatory Effectiveness
Office of Nuclear Regulatory Research
U.S. Nuclear Regulatory Commission
Washington, DC 20555-0001

10. SUPPLEMENTARY NOTES

F. Odar, NRC Project Manager

11. ABSTRACT *(200 words or less)*

The Transient Reactor Analysis Code (TRAC) was developed to provide advanced best-estimate predictions of postulated accidents in light-water reactors. The TRAC-P program has provided this capability for pressurized water reactors and for many thermal-hydraulic test facilities for approximately 20 years. However, the maintenance and portability of TRAC-P had become cumbersome because of the historical nature of the code and the inconsistent use of standardized FORTRAN. Thus, the Modernized TRAC (TRAC-M) was developed by recoding the TRAC-P algorithms to take advantage of the advanced features available in the FORTRAN 90 programming language while conserving the computational models available in the original code. The Theory Manual is one of four describing various features of TRAC-M/F90. The Theory Manual provides a detailed description of the field equations, solution procedure, numerics, and closure correlations and models of the code. This report also provides a basis for each model, correlation, or solution strategy through references to original literature and/or a description of the development process. Additionally, it lists the assumptions made in the implementation, including the definitions of required parameters not normally calculated by the code, and it describes other details of the implementation.

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

TRAC
TRAC-M
FORTRAN 90
thermal hydraulics
pressurized water reactor

13. AVAILABILITY STATEMENT

unlimited

14. SECURITY CLASSIFICATION

(This Page)

unclassified

(This Report)

unclassified

15. NUMBER OF PAGES

16. PRICE



Federal Recycling Program