Idaho National Engineering and Environmental Laboratory

INEEL/EXT-2000-00489

April 2000

Modifications made to the MELCOR Code for Analyzing Lithium Fires in Fusion Reactors

B. J. Merrill

BECHTEL BWXT IDAHO, LLC

INEEL/EXT-2000-00489

Modifications made to the MELCOR Code for Analyzing Lithium Fires in Fusion Reactors

Brad J. Merrill

Published April 2000

Idaho National Engineering and Environmental Laboratory
Fusion Safety Program
Idaho Falls, Idaho 83415-3860

Prepared for the
U.S. Department of Energy
Office of Fusion Energy Science
Under DOE Idaho Operations Office
Contract DE-AC07-99ID13727

ABSTRACT

This report documents initial modifications made to the MELCOR code that allows MELCOR to predict the consequences of lithium spill accidents for evolving fusion reactor designs. These modifications include thermodynamic and transport properties of lithium, and physical models for predicting the rate of reaction of and energy production from the lithium-air reaction. A benchmarking study was performed with this new MELCOR capability. Two lithium-air reaction tests conducted at the Hanford Engineering Development Laboratory (HEDL) were selected for this benchmark study. Excellent agreement was achieved between MELCOR predictions and measured data. Recommendations for modeling lithium fires with MELCOR and for future work in this area concluded this report.

SUMMARY

Liquid lithium metal has long been considered as a potential coolant and breeder for magnetic and inertial deuterium-tritium (DT) fusion reactors. Lithium is a material that captures neutrons produced by the reactions of fusion plasmas to produce tritium fuel and convert the neutron kinetic energy into heat. Liquid lithium also exhibits excellent heat transfer properties as a coolant. However, lithium has one very potentially serious safety concern that must be accommodated in reactor design, that is that lithium is very chemically reactive with air, water, and concrete. Because these reactions are exothermic, the energy liberated by these reactions can elevate the temperatures of activated reactor structures and thereby release radioactive material from these structures by mechanisms such as surface oxidation. This radioactive material, plus the byproducts of the lithium reactions, will form aerosol particles that when combined with the convective air patterns produced by the lithium fire could result in the release of radioactive material to the environment during accident conditions.

With these safety concerns in mind, initial modifications to the MELCOR code have been completed that allows MELCOR to predict the consequences of lithium spill accidents for evolving fusion reactor designs. These modifications include the thermodynamic and transport properties of lithium, and physical models for predicting the rate of reaction of and energy production from the lithium-air reaction. Two lithium-air reaction tests conducted at the Hanford Engineering Development Laboratory (HEDL) were selected to benchmark this new capability in MELCOR. These tests were conducted in the 850 m^3 containment vessel of the Containment Systems Test Facility (CSTF). Lithium pool temperature, vessel atmosphere, vessel wall, and oxygen and nitrogen consumption were measured. These experiments were designated as a deep pool spill test, where 43 kg of liquid lithium at a temperature of 600 °C was transferred into a reaction pan, resulting in a pool diameter to depth ratio of 16, and a shallow pool spill test, where 100 kg of liquid lithium at a temperature of 500 °C was transferred into a reaction pan, resulting in a pool diameter to depth ratio of 0.9. Excellent agreement was achieved between MELCOR predictions and measured data. Recommendations for modeling lithium fires with MELCOR and for future work in this area concluded this report.

ACRONYMS

A_s	Surface area
$c_{v,p}$	Specific heat capacity at constant volume, pressure
CSTF	Containment Systems Test Facility
CV	Containment vessel
°C	Degrees Celsius
D	Mass diffusion coefficient
DT	Deuterium-tritium
EOS	Equation of State
FCPI	Forced convection pool interface
g	Gravitational constant
Gr	Grashof number
HEDL	Hanford Engineering Development Laboratory
h_a	Atmosphere convective coefficient
h_f	Heat of fusion
h_v	Vapor enthalpy
h_I	Ice enthalpy
h_R	Heat of reaction
k	Thermal conductivity
kg	Kilograms
K	Degrees Kelvin
K_f	Film mass flow rate
K_m	Mass transfer coefficient
L	Heat structure characteristic length
Li_2O	Lithium oxide
Li_3N	Lithium nitride
m	Meters
$M_{A,B}$	Molecular weights of gas A and B
MJ	MegaJoules
NCPI	Natural convection pool interface
NCRH	Natural convection reduced heat of reaction
Nu	Nusselt number
p	Pressure
Pr	Prandlt number
q_{cond}	Conduction heat flux
q_{conv}	Convection heat flux
Q	Heat flow
s	Entropy
Sc	Schmidt number
Sh	Sherwood number
T	Temperature
u	Internal energy
V_φ	Phase velocity
W	Watts
X_φ	Phase characteristic heat transfer length

SYMBOLS

β	Thermal expansion coefficient
δ	Film or ice thickness
ΔT	Temperature difference
G	Mass flux
μ	Viscosity
μm	Micron
Ω_D	Collision integral
f	Phase designation
?	Density
$\sigma_{A,B}$	Collision diameter of gas A and B
σ_B	Stefan-Boltzmann constant
θ	Angle of inclination from horizontal

CONTENTS

FIGURES

Modifications made to the MELCOR Code for Analyzing Lithium Fires in Fusion Reactors

1. INTRODUCTION

Liquid lithium metal has long been considered as a potential coolant and breeding material for magnetic and inertial deuterium-tritium (DT) fusion reactors. Lithium is a material that captures neutrons produced by the reactions of fusion plasmas to produce tritium and convert the neutron kinetic energy into heat. Liquid lithium also exhibits excellent heat transfer properties as a coolant. However, lithium has one very potentially serious safety concern that must be accommodated in a reactor design, that is that lithium is very chemically reactive with air, water, and concrete. Because these reactions are exothermic, the energy liberated by these reactions can elevate the temperatures of activated reactor structures and thereby release radioactive material from these structures by mechanisms such as surface oxidation. This radioactive material, plus the byproducts of the lithium reactions, will form aerosol particles that when combined with the convective air patterns produced by the lithium fire could result in the release of radioactive material to the environment during accident conditions. With these safety concerns in mind, initial modifications to the MELCOR code [1] have been completed that allows MELCOR to predict the consequences of lithium spill accidents for evolving fusion reactor designs. This report details these MELCOR modifications.

The following sections of this report describe the modifications required to include the thermodynamic and transport properties of lithium in MELCOR, the physical models adopted for predicting the rate of reaction of and energy production from the lithium-air reaction, a benchmarking study performed with this new MELCOR capability, and conclusions and modeling recommendations derived from the work presented in this report.

2. LITHIUM THERMODYNAMIC PROPERITES

Modifications have been made to the MELCOR code to allow lithium as the working fluid rather than water. These modifications deal with three specific areas: the Equation of State (EOS), transport properties, and heat structure condensate film growth. The following sub-sections describe the equations used to accomplish these modifications of the MELCOR code.

2.1 Equation of State

The MELCOR code EOS for water is based on the polynomials that can be found in Appendix A of Reference [2]. These polynomials relate pressure, specific internal energy, specific entropy and heat capacity to temperature and density, and are express in terms of the Helmholtz free energy (?) as follows:

$$? = ?_o(T) + R\,T \left[\ln ? + ?\,Q(?, t) \right] \qquad (1)$$

where

$$?_o = \sum_{i=1}^{6} C_i / t^{i-1} + C_7 \ln T + C_8 \, (\ln T) / t \qquad (2)$$

and

$$Q = (t - t_c) \sum_{j=1}^{7} (t - t_{aj})^{j-2} \left[\sum_{i=1}^{8} A_{ij} \, (? - ?_{aj})^{i-1} + e^{-E\,?} \sum_{i=9}^{10} A_{ij} ?^{i-9} \right] \qquad (3)$$

In Equations 1 through 3, the temperature (T) is in degrees Kelvin, the density (?) is in g/cm^3, t denotes 1000/T.[a] Given these basic polynomials, the pressure, as an example, can be determined as follows:

$$p(?, T) = ?^2 \left(\frac{\partial ?}{\partial ?} \right)_t = ?\,R\,T \left[1 + ?\,Q + ?^2 \left(\frac{\partial Q}{\partial ?} \right)_t \right] \qquad (4)$$

Additional thermodynamic properties required by MELCOR are also derived from thermodynamic relationships involving Helmholtz free energy and Equations 1 through 3, such as fluid internal energy (J/kg), enthalpy (J/kg), entropy (J/kg-K), specific heat (J/kg-K), and derivatives of pressure (Equation 4) with respect to temperature and density. Properties derived from these equations accurately represents either the liquid or vapor phase of water from near zero pressure to 100 MPa, and from 273.15 K to 1573 K. Single phase water properties are calculated in MELCOR by subroutine H2O1PH. For a two-phase mixture, the above polynomials are used to develop saturated liquid and vapor thermodynamic property arrays for the two-phase steam dome of a P-v-T diagram. The properties of these arrays are at equal temperature increments along the saturation line between the triple point and the critical point of water. These arrays are interpolated by MELCOR to obtain saturated properties based on the fluid temperature and to generate two-phase properties based on the mixture quality calculated from the fluid density. Two-phase water properties are calculated in MELCOR by subroutine H2O2PH. Water transport properties of thermal conductivity and viscosity for saturated liquid and vapor are determined from lookup tables contained in MELCOR subroutine MPDFVL. Water surface tension is obtained from a polynomial and used in the surface boiling heat transfer subroutine HSBOIL. Thermal expansion coefficient for water is defined by block data assignments in MELCOR. Binary diffusion coefficient for water vapor in a gas mixture is defined in subroutine MPDIF. The purpose of the modifications describe in this section is to replace these water properties with those of lithium. The approach taken was to modify MELCOR so that it could access the Fusion Safety fluid property database.

---

[a] For information regarding values for the constant terms of these equations refer to Reference 2

The Fusion Safety fluid property database was originally developed for the ATHENA code [3,4,5]. This database contains the thermodynamic and transport properties for thirteen fluids. The modifications discussed in this section allowed MELCOR to access nine of these fluids, those fluids that are of interest to fusion. These fluids are water, hydrogen, lithium, potassium, helium, nitrogen, sodium, NaK, and lithium-lead.[a] This database is an extension of the RELAP5 [6] environmental library. The RELAP5 library contains a water properties table and interpolation subroutines that compute thermodynamic properties from the water properties table based on specific fluid properties as inputs. For example, subroutine STH2X4 returns the properties of pressure, internal energy, enthalpy, coefficient of thermal expansion, specific heat capacity, isothermal compressibility, entropy, stream quality, and saturation pressure for a single phase fluid state (saturated liquid and vapor values for a two-phase state) given water temperature and specific volume as inputs. This database also contains transport property subroutines called THCOND and VISCOS that contain equations for liquid and vapor thermal conductivity and viscosity, and a subroutine called SURFTN that contains equations for saturated liquid surface tension. In addition to the above mentioned subroutines of the fluid property database, there are initialization routines that read in the fluid thermodynamic properties table, and triple point and critical point properties.

Given this background, five major modifications were made to the MELCOR code to change the working fluid of MELCOR from water to lithium. First, the water equations contained in subroutine H2O1PH where removed and replaced with a call to a new subroutine call FLUIDEOS. This subroutine accessed the fluid property database by calling subroutine STUPX4, a multi-fluids version of STH2X4. Subroutine FLUIDEOS then computes and returns the thermodynamic property information to subroutine H2O1PH as required by the MELCOR code. Second, a subroutine called INITFLD was written that on startup of a MELCOR problem calls the fluids database initialization subroutine ICMPNX and initializes properties required by H2O2PH. Subroutine INITFLD searches the local user file directory for the thermodynamic properties table. The file in which these properties reside has an acronym for a filename that contains a 'tpf' preceding the fluid chemical name, for example the file for lithium is tpfli.[b] The user must supply this file before MELCOR will execute. This search follows the order of the nine fluids listed above. The first file found by INITFLD becomes the working fluid of choice, which choice INITFLD notifies the user of by writing the fluid name to the user's computer monitor. After the fluid has been identified INITFLD calls subroutine ICMPNX to read the thermodynamic properties table. INITFLD then develops the saturation properties required by subroutine H2O2PH by calling subroutine FLUIDEOS, and finally INITFLD redefines the fluid molecular weight, and Lennard-Jones parameters used by MPDIF. Third, subroutine MPPRP1 was modified to call subroutines THCOND and VISCOS when fluid transport properties are required, and subroutine HSBOIL was changed to call subroutine SURFTN for fluid surface tension. Fourth, the subroutine

---

[a] The order of these name reflect the fluid number assumed by MELCOR, that is fluid one is water, two is hydrogen, etc.

[b] The other file names are tpfh2o for water, tpfh2 for hydrogen, tpfk for potassium, tpfhe for helium, tpfn2 for nitrogen, tpfna for sodium, tpfnak for NaK, and tpflipb for lithium-lead.

FLCOK that predicts choked flow in MELCOR flow paths was modified to treat fluids other than water. Subroutine FLCOK contains water specific polynomials that compute choked flow conditions given upstream fluid stagnation pressure and enthalpy [7]. These polynomials are analytic fits to the predictions from the Henry-Fauske correlation for sub-cooled water flow, and the Moody correlation for saturated water flow. For fluids other than water, subroutine FLCOK has been modified to call subroutine GCSUB. This subroutine is from the RELAP5 code, and it computes the critical mass flow for a liquid based on an iterative solution of the Henry-Fauske correlation. Finally, fluid triple point temperature and critical point temperature, pressure and density was propagated thoughout the MELCOR code where needed by the use of the common block EOSPARM, and printing to output files was changed to indicate the adopted fluid.

2.2 Extrapolation of EOS Below the Triple Point

If during a calculation the fluid temperature drops below the triple point temperature (T_{TP}) a strategy must be adopted to allow MELCOR to continue executing properly. The strategy adopted in this multi-fluids version of MELCOR is similar to that used for water previously [8] but with a simplifying assumption regarding fluid specific heat capacity. The adopted strategy for defining the fluid thermodynamic properties below T_{TP} relies on the assumptions that the:

1) fluid vapor is an ideal gas,

2) liquid density and derivatives of pressure with temperature are independent of temperature,

3) heat of fusion can be accounted for by an effective heat capacity, and

4) heat capacities of the ice/vapor are constant below T_{TP}.

With these assumptions, the pressure is determined from:

$$p(?,T) = p(?,T_{TP}) + \left(\frac{\partial p}{\partial T}(?,T_{TP}) \right)_? \left[T - T_{TP} \right] \qquad (5)$$

where the pressure [p(?,T_{TP})] and partial derivative of pressure with temperature ([?p/?T]$_?$) at T_{TP} are determined from the fluid EOS discussed in the previous section.

The specific internal energy below T_{TP} is found by integrating the following equation [9]:

$$du = c_v\, dT + \left[T \left(\frac{\partial p}{\partial T} \right) - P \right] dv \qquad (6)$$

Assuming that the liquid heat of fusion (h_f) can be treated as an effective specific heat capacity defined as the heat of fusion (J/kg) divided by an arbitrary temperature change ? (= 1K), the “ice” internal energy for T_{TP}-? < T < T_{TP} can be calculated from,

$$u_l(?,T) = u_l(?,T_{TP}) + \frac{h_f}{?}\left[T - T_{TP} \right] \qquad (7)$$

If the additional assumption is made that the ice specific heat capacity below this temperature is constant and equal to that of the liquid at T_{TP}, then ice specific internal energy for $T < T_{TP}$-? becomes:

$$u_l(?,T) = u_l(?,T_{TP}-?) + c_{vl}\Big|_{TP}\,[T - T_{TP} + ?] \tag{8}$$

For a constant vapor heat capacity defined below T_{TP}, the vapor specific internal energy becomes:

$$u_v(?,T) = u_v(?,T_{TP}) + c_{vv}\Big|_{TP}\,[T - T_{TP}] \tag{9}$$

The specific entropy below T_{TP} is found by integrating by the following equation [10]:

$$ds = c_v\frac{dT}{T} + \left(\frac{\partial p}{\partial T}\right)_v dv \tag{10}$$

With the above specific heat capacity assumptions, the result is the following ice specific entropy below T_{TP}:

$$\begin{aligned} s_l(?,T) &= s_l(?,T_{TP}) + \frac{h_f}{?}\ln\left(\frac{T}{T_{TP}}\right) & ;T_{TP} - ? < T < T_{TP} \\ s_l(?,T) &= s_l(?,T_{TP}-?) + c_{vl}\Big|_{TP}\ln\left(\frac{T}{T_{TP}-?}\right) & ;T < T_{TP} - ? \end{aligned} \tag{11}$$

And finally, the vapor phase the specific entropy becomes:

$$s_v(?,T) = s_v(?,T_{TP}) + c_{vv}\Big|_{TP}\ln\left(\frac{T}{T_{TP}}\right) \quad ;\ T < T_{TP} \tag{12}$$

An arbitrary fluid viscosity of 10^4 kg/m-s is assigned to the 'liquid' state below T_{TP}. This assumption simulates the formation of ice from the liquid phase by creating a highly viscous fluid that should remain nearly stationary after formation. The thermal conductivity of a fluid below the T_{TP} is maintained at a constant value that equals that of the ice at this temperature. The vapor viscosity and thermal conductivity are also set equal to those at T_{TP}. These extrapolations are meant only as approximations, and may require refinements in the future.

2.3 Films on Cold Structures

Condensation of vapor will occur on walls that are below the saturation temperature of the vapor. The condensed vapor will transfer energy to the wall, and form a film that will decrease the heat flow to the wall by acting as a thermal resistance. Version 1.8.2 of the MELCOR code has a heat structure film model that accounts for the thermal resistance, energy transfer, and stored energy associated with a film. It also has a crude conservation of mass model for the film. This model assumes that the condensed vapor will cause the film to grow until the film reaches a maximum value (default of 0.15 mm), after which thickness any additional condensate will drain directly into the enclosure pool. This film will remain on the heat structure until the heat structure temperature exceeds the vapor saturation temperature, causing the film to evaporate.

This model was modified in two ways. The first modification adds the solution of a mass balance equation to predict the thickness of films on non-horizontal heat structures. This equation is as follows:

$$\rho_f L \frac{\partial d}{\partial t} = L\,\Gamma_c - K_f \qquad (13)$$

where

ρ_f = film density (kg/m^3)
L = heat structure characteristic length (m)
δ = film thickness (m)
Γ_c= condensation mass flux (kg/m^2)
K_f = film mass flow rate per unit width

The adopted film mass flow rate relationship is that derived by Nusselt for laminar films [11], and is defined as follows:

$$K_f = \left[\frac{\rho_f(\rho_f - \rho_v)g \sin \theta d^3}{3\mu_f}\right] \qquad (14)$$

where

ρ_v = vapor density (kg/m^3)
g = gravitational constant (m/s^2)
θ = angle of heat structure inclination from horizontal
μ = film viscosity (kg/m-s)

Equations 13 and 14 are solved in finite difference form to give the time-dependent film thickness, which film thickness can exceed the user-specified maximum value. For horizontal heat structures (i.e., floors and ceilings) the standard MELCOR film model is used.

If the heat structure surface is colder than T_{TP}, the film will freeze to the heat structure. To account for this possibility, the film model switches from the solution of a mass balance equation to the solution of an energy balance equation written for the structure surface to determine the rate of film growth. This energy balance equation assumes that the net heat flux at the surface results in either liquid freezing or ice melting. This equation can be written as:

$$\rho\, h_f \frac{\partial d}{\partial t} = q_{cond} - q_{conv} - G_c\,(h_v - h_i) \qquad (15)$$

where

ρ = ice density (kg/m^3)
h_f = heat of fusion (J/kg)
d = ice thickness (m)
q_{cond} = heat flux conducted through ice (W/m^2)
q_{conv} = heat flux convected to ice surface by atmosphere (W/m^2)
h_v = vapor enthalpy (J/kg)
h_i = ice enthalpy (J/kg)

The conducted and convected heat fluxes are defined as follows:

$$q_{conv} = h_a \, (T_a - T_i) \tag{16}$$

$$q_{cond} = \frac{k}{d} \, (T_i - T_s) \tag{17}$$

where

k = thermal conductivity of ice (W/m^2-K)
T_i = ice surface temperature (K)
T_s = structure surface temperature (K)
h_a = convective coefficient from atmosphere (W/m^2-K)
T_a = temperature of atmosphere (K)

The ice surface temperature for this model is the equilibrium temperature found by setting the left-hand side of Equation 15 to zero. If this temperature is above T_{TP} then it is set equal to T_{TP}. If it is below T_{TP} by 1 K, then Equation 13 is used with the film mass flow rate set to zero. Equations 15 through 17 are solved at all surfaces that are below the T_{TP}, regardless of heat structure orientation. However, the rate of growth of the ice layer is limited to the time dependent flow of mass condensing on a given surface, that is the product of the heat structure surface area times the condensation mass flux.

A FORTRAN listing of the more extensively modified MELCOR subroutines regarding the changes describe in Section 2 is contained in Appendix A.

3. LITHIUM-AIR REACTION MODEL

This section presents the chemical kinetics equations proposed for modeling lithium-air reactions with MELCOR. In addition, this section reviews the MELCOR pool-atmosphere interface mass and heat transport model of a MELCOR non-equilibrium control volume. Parallels are drawn between this model and the adopted lithium-air reaction model.

3.1 Lithium Reaction Equations

Lithium reacts with both major constituents of air. Lithium reacts with oxygen to form lithium oxide and with nitrogen to form lithium nitride. The chemical molar reaction equations that described these reactions are as follows:

$$4\,Li + O_2 \rightarrow 2\,Li_2O \tag{18}$$

and

$$6\,Li + N_2 \rightarrow 2\,Li_3N \tag{19}$$

The lithium-air reactions are exothermic, and the energy liberated is temperature dependent. Equations that describe the heat of reaction as a function of temperature were developed from the HSC code [12] for these modifications. The heat of reaction (J/gmole) for lithium oxide and lithium nitride are:

$$h_{R\,Li2O} = \; 4187 \text{ x } (143.05 + 2.284 \text{ x } 10^{-3}\,T - 3.376 \text{ x } 10^{-6}\,T^2) \tag{20}$$

$$h_{R\,Li3N} = 4187 \text{ x } (39.26 + 2.832 \text{ x } 10^{-3}\ T - 1.034 \text{ x } 10^{-6}\ T^2) \qquad (21)$$

for T < 180 °C, and

$$h_{R\,Li2O} = 4187 \text{ x } (144.60 + 2.172 \text{ x } 10^{-3}\ T - 3.679 \text{ x } 10^{-6}\ T^2 + 4.385 \text{ x } 10^{-10}\ T^3) \qquad (22)$$

$$h_{R\,Li3N} = 4187 \text{ x } (41.19 + 6.315 \text{ x } 10^{-3}\ T - 1.148 \text{ x } 10^{-5}\ T^2 + 8.405 \text{ x } 10^{-10}\ T^3) \qquad (23)$$

for 180 °C < T < 1600 °C.

The above equations describe the chemical composition and energy release from these lithium reactions, but they do not describe the rate at which the reaction proceeds. The reaction rate model adopted for this model is the same model adopted for the LINT code [15]. That is, the reaction rate proceeds as fast as nitrogen or oxygen can diffuse to the lithium pool surface through a gas boundary layer that exists above this reacting surface. Because convective mass and heat transfer through surface boundary layers can be described by analogous boundary layer conservation equations [13], it has been determined that the surface boundary layer mass transfer coefficient can be related to heat transfer coefficient though the following relationship [14]:

$$Sh = Nu \left(\frac{Sc}{Pr} \right)^{1/3} \qquad (24)$$

where,

Sh = Sherwood number = $\frac{K_{mi} L}{D_i}$

Nu = Nusselt number = $\frac{h_a L}{k}$

Sc = Schmidt number = $\frac{m}{r\ D}$

Pr = Prandlt number = $\frac{m\ c_p}{k}$

and where

K_{mi} = specie mass transfer coefficient (m/s)
L = surface characteristic length (m)
D_i = specie mass diffusion coefficient (m^2/s)
h_a = atmosphere heat transfer coefficient (W/m^2-K)
k = atmosphere thermal conductivity (W/m-K)
μ = atmosphere viscosity (kg/m^2-s)
ρ = atmosphere density (kg/m^3)
c_p = atmosphere specific heat (J/kg-K).

If the assumption is made, as is the case in the LINT code [15], that the appropriate atmosphere heat transfer coefficient is that obtained from the following correlation for turbulent natural convection on a horizontal surface:

$$\mathrm{Nu} = 0.13\,(\mathrm{Gr}\ \mathrm{Pr})^{1/3} \qquad (25)$$

where

Gr = Grashof number = $g\ b\ \Delta T\ L^3 \left(\frac{r}{m}\right)^2$
β = atmosphere thermal expansion coefficient (K^{-1})
ΔT = atmosphere to pool temperature difference (K)

the result is the following relationship for the mass transfer coefficient:

$$K_{m_i} = 0.13\left(\frac{D_i}{L}\right)(\mathrm{Gr}\ \mathrm{Sc})^{1/3} \qquad (26)$$

Given this mass transfer coefficient, the mass flux (kg/m^2-s) of oxygen diffusing to the lithium surface, for example, is

$$\Gamma_{O2} = K_{m_{O2}}\, r_{O2} \qquad (27)$$

Because we are assuming that the rate of oxygen consumption equals the rate of oxygen arrival, that is an infinite reaction rate at the pool surface, the rate of oxygen consumption (kg/s) equals the product of Equation 27 times the pool surface area. This mass consumption has been implemented in MELCOR as a mass sink in the atmosphere conservation of mass equation. The energy liberated (W) by this mass consumption is given by

$$Q_{O2} = \Gamma_{O2}\ A_S\ h_{R_{Li2O}} / (2\ M_{O2}) \qquad (28)$$

where the factor of 2 represents the molar ratio of oxygen in lithium oxide and M_{O2} is the molecular weight of oxygen. Similar relationships can be derived for the consumption of nitrogen by the lithium. Heating of the lithium pool, added to MELCOR as a liquid bulk heating rate (W), equals the sum of the energy liberated by the oxygen and nitrogen reactions defined as follows:

$$Q_{POOL} = Q_{O2} + Q_{N2} \qquad (29)$$

The rate (kg/s) at which lithium is consumed is added to MELCOR as a mass sink in the pool conservation of mass equation and is defined as follows:

$$\dot{m}_{Li} = -\ A_S\ M_{Li}\ (\ 4\ \Gamma_{O2} / M_{O2} + 6\ \Gamma_{N2} / M_{N2}\ ) \qquad (30)$$

The rates at which Li_2O and Li_3N form are determined in a similar manner. A fraction of this material will be mobilized as aerosols, with the remainder will stay in the pool. To simulate this aerosol production behavior, the user must activate the aerosol transport option in MELCOR through specifying the appropriate MELCOR user input. This new model will subsequently place a constant fraction, a value determined from experiments as discussed in 4.1, of the Li_2O and Li_3N mass produced into the smallest aerosol bin size specified by the user input. This model assumes that MELCOR aerosol class 13 is Li_2O and class 15 is Li_3N, unless the user designates two new aerosol classes that have the names Li_2O and Li_3N. This model adds the remaining Li_2O and Li_3N produced to the aerosol mass residing in the control volume pool. If the aerosol transport capability is not invoked, this model will not conserve the resulting Li_2O and Li_3N masses.

Material properties for the above equations are obtained by MELCOR from the Fusion Safety material property database. The binary gas diffusion coefficients (m^2/s) are determined from a correlation proposed by Wilke and Lee [16]. This correlation is as follows:

$$D_{AB} = 1.01325 \times 10^{-2} \left(3.03 - 0.98/M_{AB}^{1/2}\right) T^{3/2} / \left(p\, M_{AB}^{1/2}\, \sigma_{AB}^{2}\, \Omega_D\right) \qquad (32)$$

where,

M_{AB} = 2 [(1/M_A) + (1/M_B)]$^{-1}$
M_A, M_B = molecular weights of gas A and B (g/mol)
p = pressure (Pa)
σ_{AB} = [σ_A + σ_B]/2
σ_A , σ_B = collision diameter of gas A and B (Å)
Ω_D = collision integral.

For a gas mixture, MELCOR calculates and effective binary diffusion coefficient, D_{AM}, by a method proposed by Wilke [17], which is as follows:

$$\frac{1 - y_A}{D_{AM}} = \sum_{j=1, \neq A}^{M} \frac{y_j}{D_{Aj}} \qquad (33)$$

where,

y_j = is the mole fraction of gas j
D_{Aj} = binary diffusion coefficient of gas A through gas j (m^2/s).

As defined, the predicted binary diffusion coefficient is for oxygen diffusing through a gas mixture of nitrogen and lithium vapor.

The MELCOR code will activate this new lithium-air reaction capability if:

1) lithium and oxygen or nitrogen co-exist in the same MELCOR control volume,
2) the MELCOR computational volume has been identified as a non-equilibrium control volume.

The second item is less obvious than the first. However, if the computational volume is an equilibrium volume then there is no temperature difference between the atmosphere and the pool, the Nusselt number is zero, and by default there will be no mass transfer predicted by this model. Aside from these points, the lithium-air reaction model will activate automatically without additional user input.

It should be mentioned that this model departs from the LINT lithium-air pool surface reaction model in three ways. First, the heat of reaction in LINT does not vary with temperature. Second, the boundary layer gas in LINT is assumed to consist entirely of argon. Third, LINT accounts for lithium-water vapor reaction. However, these are minor differences because:

1) the majority of the heat generated by the lithium-air reaction comes from the oxygen reaction that varies by less than 5% for the temperature range of 180 C < T < 1600 C,

2) the difference between oxygen binary diffusion coefficients in argon, nitrogen, or air is less than 20 %, and

3) the water vapor content of ambient air is minimal.

The majority of the equations for this new MELCOR capability reside in a subroutine named LIFIRE. A FORTRAN listing of this new subroutine is contained in Appendix B.

3.2 MELCOR Inter-phase Heat and Mass Transfer Model

This section has been added to illustrate the similarities between the MELCOR heat and mass transport model for non-equilibrium volumes and the lithium-air reaction model described in the previous section, and to form as a basis for future reference during the benchmarking discussion of Section 4. In MELCOR [7], heat flows between the pool and atmosphere as calculated by:

$$Q_{\varphi S} = A_s \left[h_\varphi \left(T_\varphi - T_I \right) + \sigma_B \left(T_\varphi^4 - T_I^4 \right) \right] \qquad (34)$$

where,

$Q_{\varphi S}$ = heat flow from phase φ to the pool-atmosphere interface (W)
T_φ = pool or atmosphere phase temperature (K)
T_I = pool-atmosphere interface temperature (K)
σ_B = Stefan-Boltzmann radiation constant (W/m^2-K^4)
A_s = pool surface area (m^2)

The pool-atmosphere interface temperature is found by equating the heat from each phase to this interface. Because this heat transfer depends on the interface temperature, an iterative procedure is used to obtain the interface temperature. The heat transfer coefficient per phase is the maximum value of that predicted from forced convection, natural convection, or conduction. For forced convection at the pool-atmosphere interface, the following correlation is used:

$$\mathrm{Nu}_{\mathrm{forced},\, j} = 0.02\ r_j\ \mathrm{c}_{\mathrm{p}j}\ \mathrm{V}_j\ \frac{\mathrm{X}_j}{\mathrm{k}_j} \tag{35}$$

where,

$\mathrm{Nu}_{\mathrm{forced},\, \varphi}$ = phase (pool or atmosphere) forced convection Nusselt number
ρ_φ = phase density (kg/m^3)
$c_{p\varphi}$ = phase specific heat (J/kg-K)
V_φ = phase velocity (m/s)
k_φ = phase thermal conductivity (W/m-K)
X_φ = phase characteristic heat transfer length (m).

The characteristic length is the minimum of the diameter of the pool and the height of the pool or atmosphere volume. For natural convection, both laminar and turbulent convection are considered. For the atmosphere, these correlations are:

$$\mathrm{Nu}_{\mathrm{laminar,\ A}} = 0.54\left(\mathrm{Gr\ Pr}\right)^{1/4} \tag{36}$$

for laminar natural convection, and

$$\mathrm{Nu}_{\mathrm{turbulent,A}} = 0.14\left(\mathrm{Gr\ Pr}\right)^{1/3} \tag{37}$$

for turbulent natural convection. The heat transfer coefficient equals the maximum of the above correlations times the thermal conductivity divided by the heat structure characteristic length for a given phase.

The mass transfer coefficient of vapor at the pool surface is calculated with Equation 24, based on the maximum Nusselt number from Equations 35, 36, and 37. In MELCOR this mass transfer represents the mass of lithium leaving the pool due to evaporation or the mass of lithium leaving the atmosphere due to condensation. The similarity with the lithium-air reaction model of Section 3.1 is that both models use Equation 24. However, the lithium-air reaction model only considers turbulent natural convection for the transport of oxygen or nitrogen through the gas boundary layer, while the MELCOR model considers forced, and laminar or turbulent natural convection for transport of lithium. This point will be addressed again during the benchmarking discussed that follows.

4. BENCHMARK STUDY

This section contains a description of two lithium spill tests conducted at the Handford Engineering Development Laboratory (HEDL) in the Containment Systems Test Facility (CSTF). An experimental facility description and the results of two lithium spill tests are present in Section 4.1. The MELCOR model developed to simulate the

CSTF is detailed in Section 4.2. Comparisons of the lithium spill experimental data with MELCOR predictions are given in Sections 4.3 and 4.4.

4.1 HEDL Lithium Spill Tests

A series of lithium reaction test were conducted at the HEDL in the 1980s for the Office of Fusion Energy under the supervision of the Fusion Safety Program at the Idaho National Engineering Laboratory [18]. These experiments we conducted in the CSTF. This facility is essentially a 850 m^3 containment vessel (CV) that weighted approximately $7.7x10^4$ kg, had a diameter of 7.6 m, a height of 20.3 m, and an internal surface area of 520 m^2. A schematic of this facility appears in Figure 1. Several of the tests conducted in this facility were reported in detail by Reference [18]. Two lithium-air spill tests (LA-4 and LA-5) from this reference were selected to benchmark the lithium-air reaction model capability of MELCOR.

During Tests LA-4 and LA-5, liquid lithium was introduced into a reaction pan located at the bottom of the CV. Schematic drawings of the reaction pans for these tests appear in Figures 2 and 3. As can be seen, the cylindrical reaction pan is situated in the center of three concentrically stacked stainless steel (SS316) cylindrical catch pans. Between the first and second catch pan and between the third catch pan and the reaction pan resides a 6.4 cm thick layer of firebrick. The firebrick acts as a thermal insulator. The reaction pan for Test LA-4 has a diameter of 40.6 cm and a height of 61.0 cm, and the outer surface of this reaction pan is wrapped with a 7.6 cm thick layer of KAOWOOL insulation. The reaction pan for Test LA-5 has a diameter of 160 cm and a height of 25.4 cm. The outer surface of this reaction pan is bare, but a lid has been added to this pan to terminate the lithium-air reaction during the test.

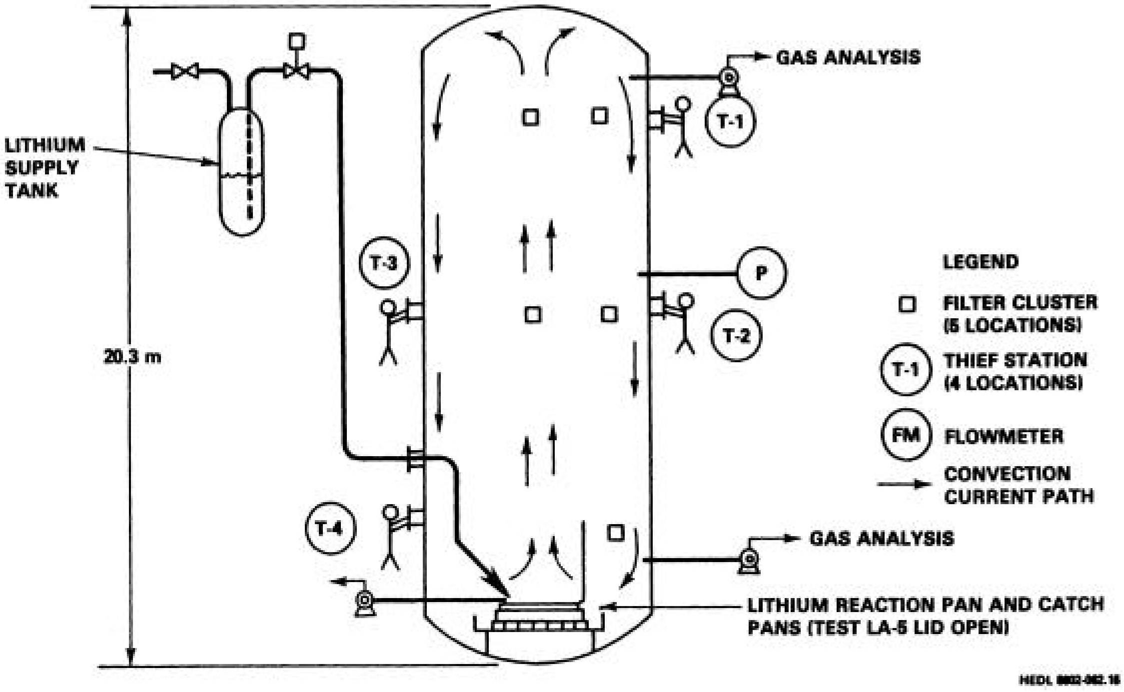

Figure 1. CSTF lithium pool-air reaction test schematic.

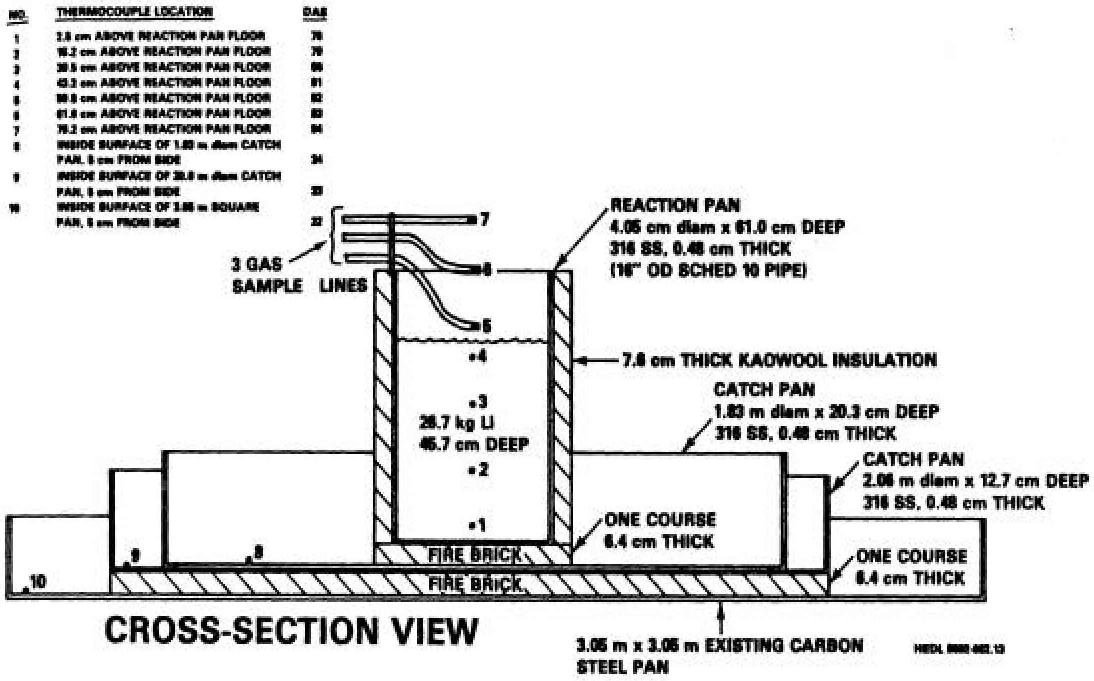

Figure 2. Reaction pan and catch pan schematic of Test LA-4.

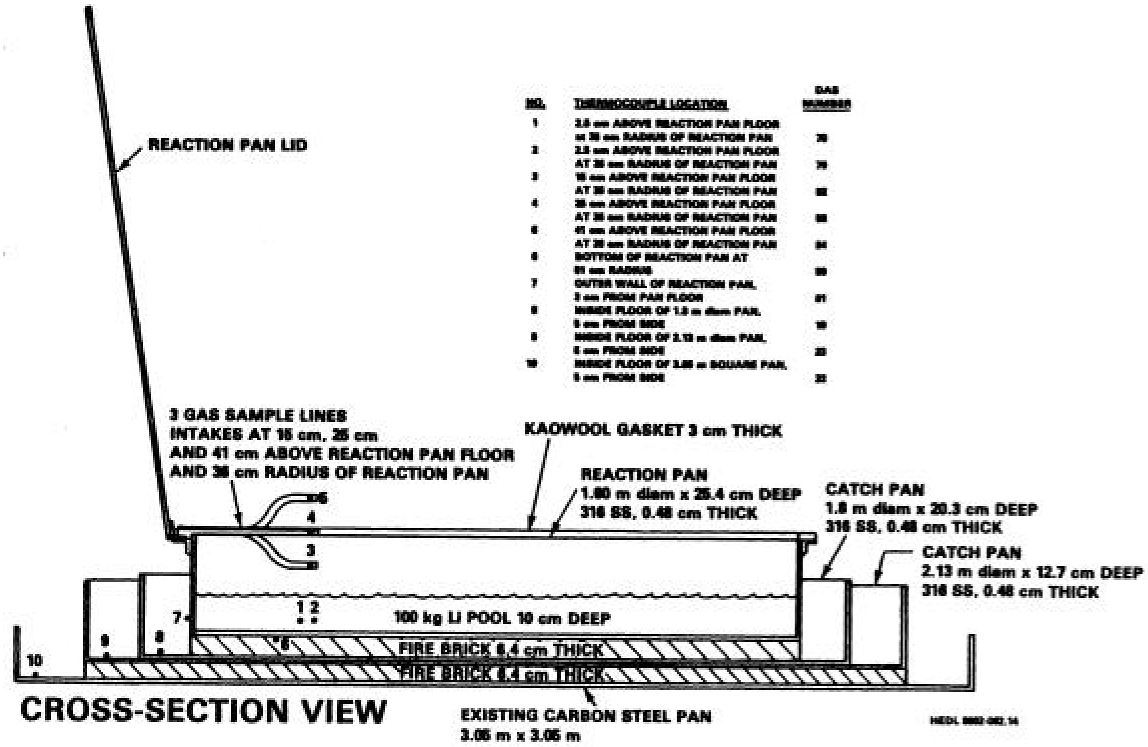

Figure 3. Reaction pan and catch pan schematic of Test LA-5.

For Test LA-4, 43 kg of liquid lithium at a temperature of 600 °C was transferred in 115 seconds into the reaction pan that had been preheated to about 200 °C. This resulted in a lithium pool height of 45.7 cm. For Test LA-5, 100 kg of liquid lithium at a temperature of 500 °C was transferred into an unheated reaction pan. The resulting lithium pool height for Test LA-5 was 10 cm. Given the fact that the initial pool diameter to height ratio (D/h) for Test LA-4 was approximately 0.89 and that for Test LA-5 was approximately 16, Test LA-4 was designated as a deep pool test and Test LA-5 designated as a shallow pool test.

These tests were well instrumented. There were 49 thermocouples located inside and outside of the CV. These thermocouples measure lithium pool, reaction pan, CV atmosphere, and CV wall temperatures. The arrangement of the pool and pan thermocouples can be seen in Figures 2 and 3. In addition to these thermocouples, the CV gas pressure and composition (at five locations) were continuously monitored during these tests, and five sampling stations existed for post-test aerosol analysis.

Some of the results obtained from Tests LA-4 and LA-5 are reproduced in Figures 4 through 8. Figure 4 gives the measure pool temperatures for LA-4. Four temperatures are plotted, with each temperature existing at an increasing height in the pool. As can be seen, there is very little temperature difference within the 45.7 cm deep pool. The temperature continually rises from the initial temperature of 600 °C to a value of 1070 °C by 3200 seconds. Beyond this time these temperatures undergo a rapid rise to near 1200 °C. This second rise is caused by a melt through of the reaction pan, and the release of the lithium pool into the catch pan. The pool temperatures for LA-5 are given in Figure 5. The measured temperatures show a very gradually rise beyond 410 °C over the first 1000 seconds of experiment. As reported in Reference [18]: “During the initial 900 seconds, localized flames were observed as small, “popcorn kernel”-like fires on the surface of the pool. Lithium appeared to wick up on these “kernels,” burn and slowly grow in diameter to cover the pool surface.” After 1000 seconds, the pool temperature continually rises to reach maximums of about 1100 °C by 3900 seconds, at which time the reaction pan lid was closed to terminate the reaction experiment.

Figures 6 and 7 contain average atmosphere temperature, average wall temperature, and pressure of the CV for Tests LA-4 and LA-5, respectively. The vessel atmosphere and wall temperatures were obtained by averaging the temperature measurements from nine atmosphere thermocouples and the measurements of seven vessel wall thermocouples, respectively, all located at different locations (r,θ,z). Reference [18] did not discuss the logic behind the thermocouple selection process for these average temperature measurements. The average temperatures of Test LA-4 change very little prior to the failure of the reaction pan at 3200 seconds. A larger average temperature change can be noted for Test LA-5.

Figure 8 contains the oxygen and nitrogen consumption for Tests LA-4 and LA-5. The values plotted were calculated from the ideal gas law based on the measured gas composition, temperature, and pressure. The maximum rate of nitrogen consumption was

0.27 gmole/m^2-s for Test LA-4 and 0.20 gmole/m^2-s for Test LA-5. When comparing this result with measurements of pool temperature, it can be seen that the nitrogen reaction in LA-4 begins to drop at about 1030 °C and nearly terminates by a temperature of 1060 °C. In test LA-5 this transition begins at a lower pool temperature, about 930 °C. The maximum rate of consumption for oxygen was 0.33 gmole/m^2-s for Test LA-4 and 0.17 gmole/m^2-s for Test LA-5. Note the rather dramatic drop in oxygen consumption rate between these tests. This will be discussed in more detail in Section 4.4.

Based on post test examinations of the aerosols on CV walls and deposition trays, and aerosol collection stations that monitored suspended aerosol concentrations during these tests, Reference [18] was able to estimate that 10% of the lithium that reacted during Test LA-4 became an aerosol while 3.4% of the lithium during Test LA-5 became an aerosol. The initial size of the aerosol particles suspended in the atmosphere was reported as being between 0.1 μm and 1.0 μm. These particles grew from the process of agglomeration to obtain mass median diameters of about 6 μm with a geometric standard deviation of about 2.4 by 10^4 seconds. Whether these particles originated by homogeneous nucleation of vapor from the CV atmosphere or were ejected as particulate from the surface of the pool as a consequence of the energetic lithium-air reaction was not reported by Reference [18].

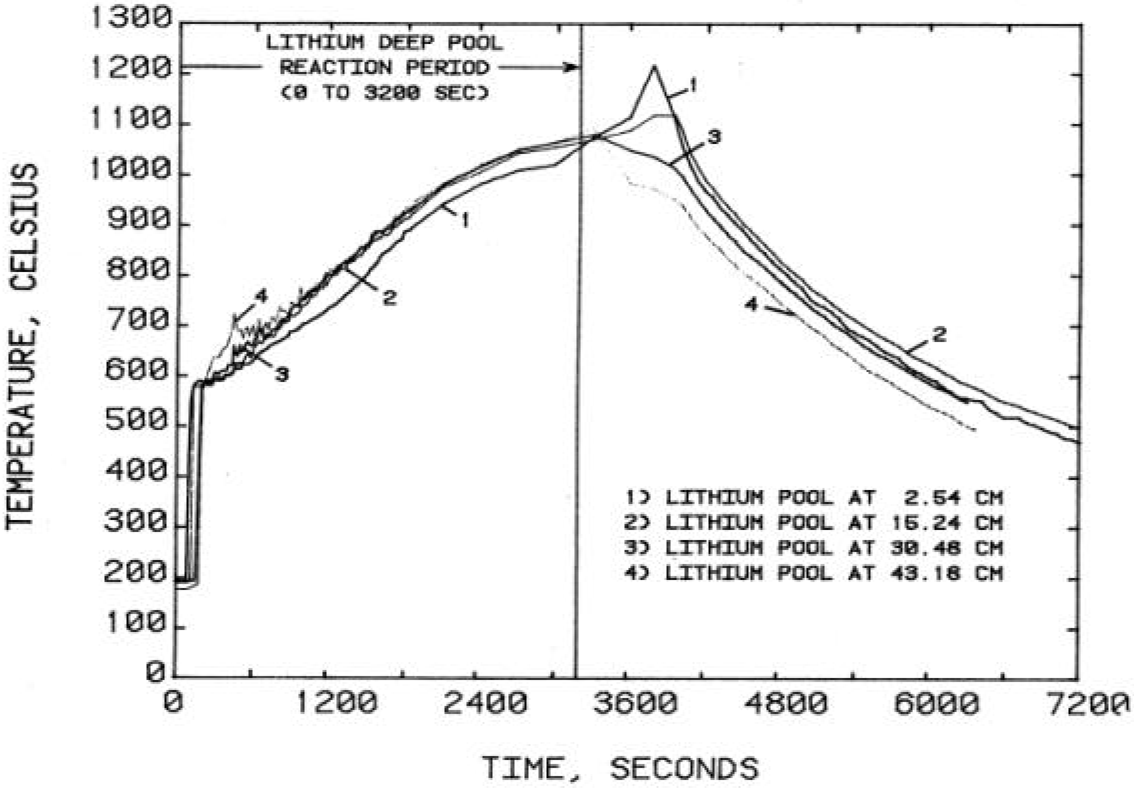

Figure 4. Lithium pool temperatures during Test LA-4.

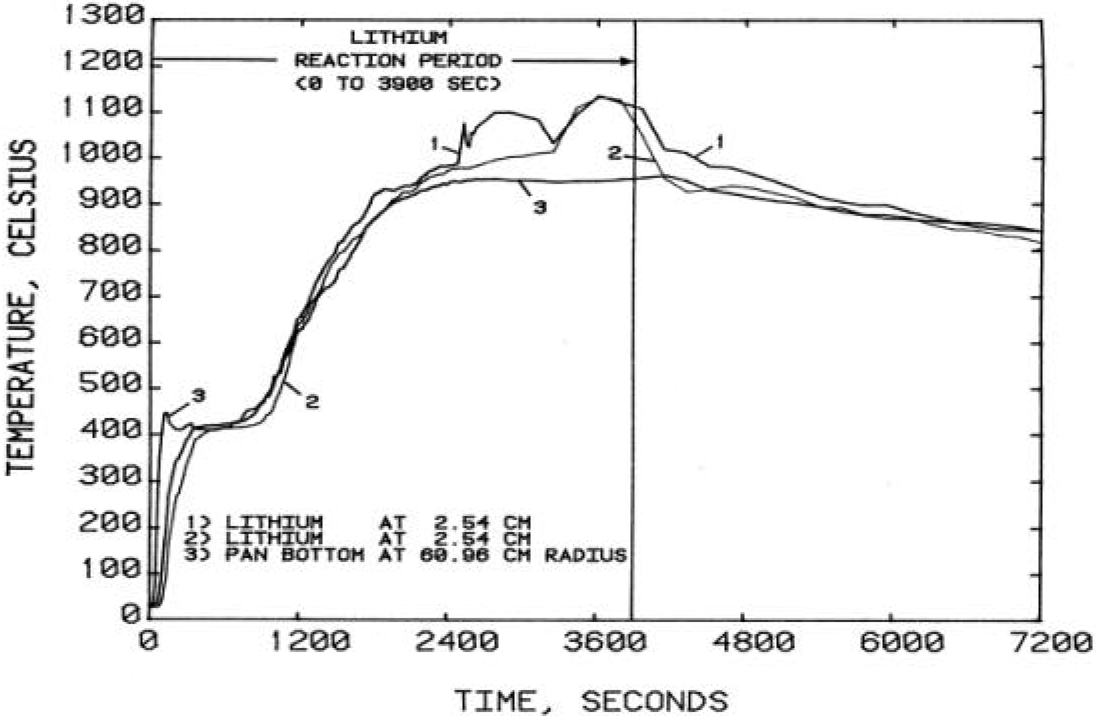

Figure 5. Lithium pool temperatures during Test LA-5.

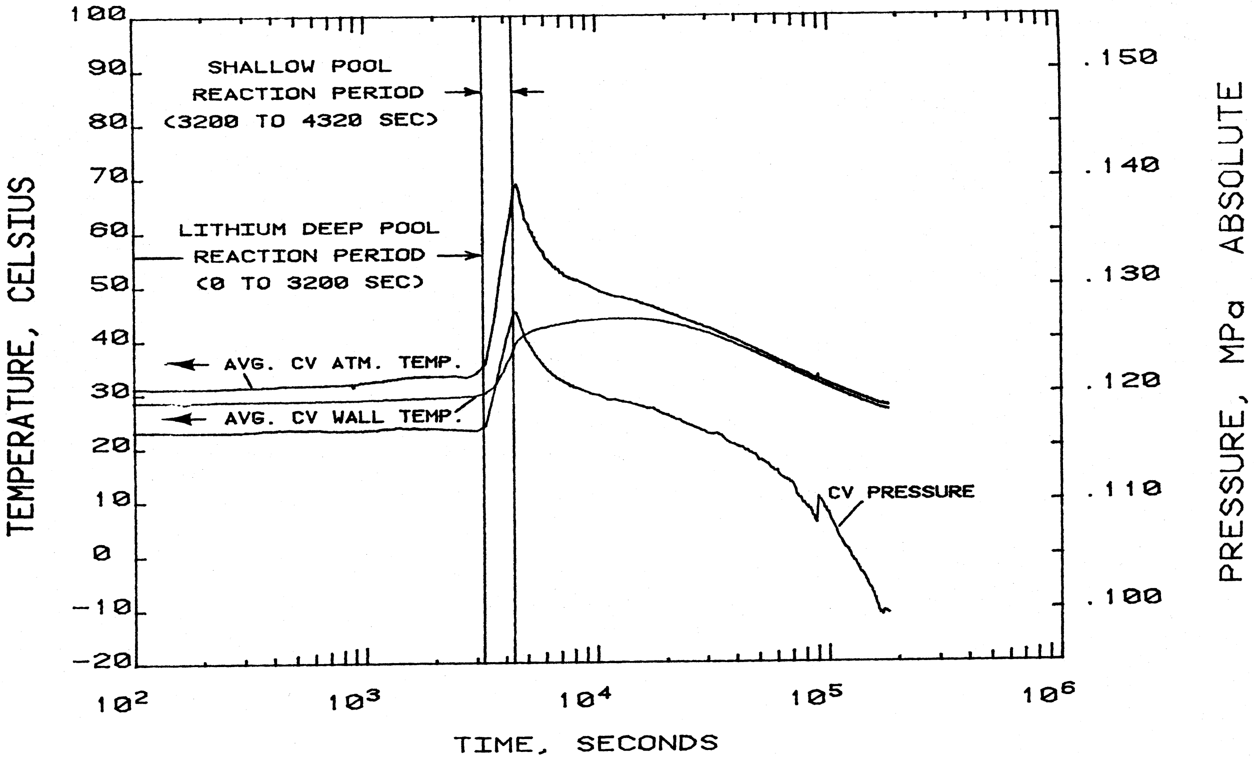

Figure 6. Containment vessel pressure, average atmosphere temperature and average vessel wall temperature during Test LA-4.

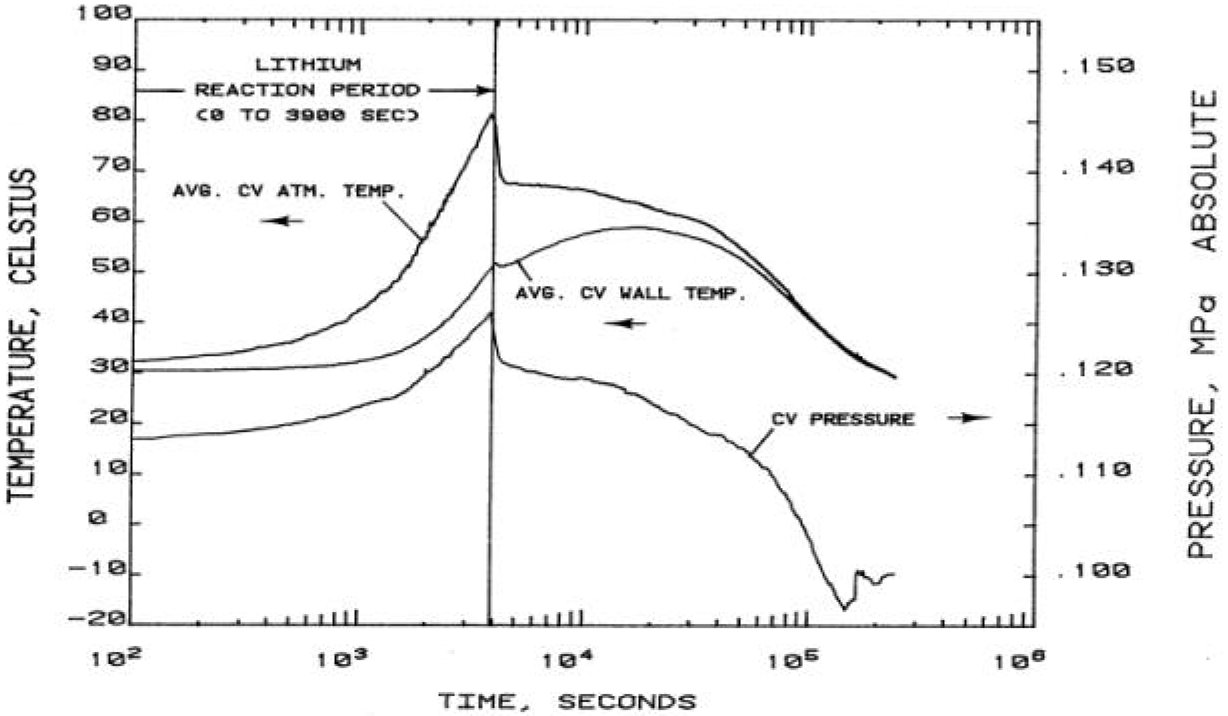

Figure 7. Containment vessel pressure, average atmosphere temperature and average vessel wall temperature during Test LA-5.

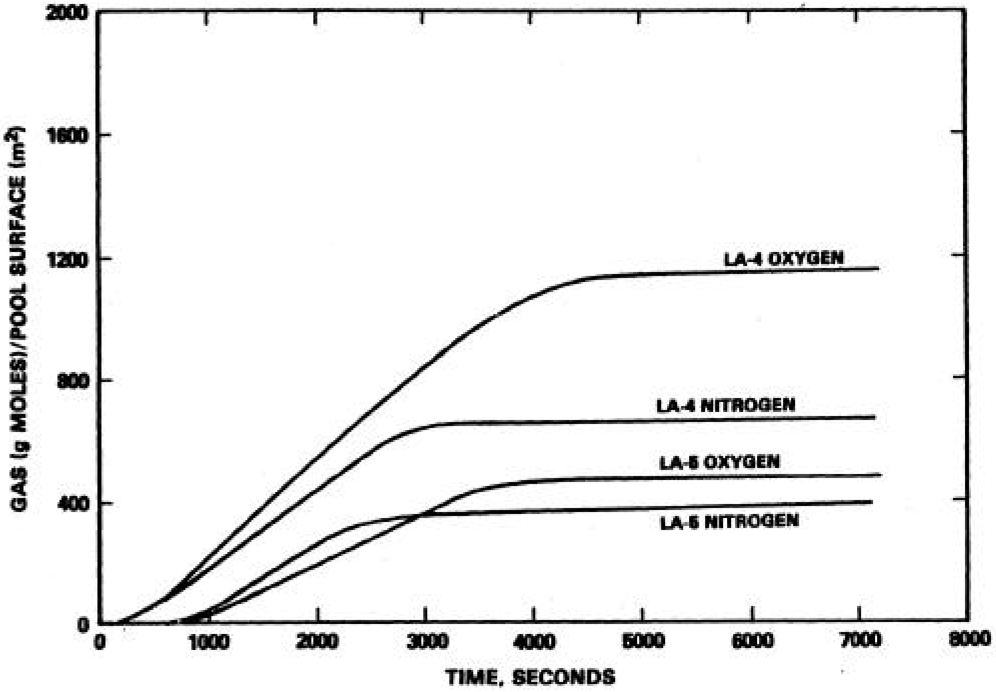

Figure 8. Nitrogen and oxygen consumption during Tests LA-4 and LA-5.

4.2 MELCOR Model of the HEDL CSTF

Figure 9 contains a schematic of the MELCOR HEDL CSTF model developed for the benchmarking studies. As can be seen, this model contains 11 MELCOR control volumes, 15 MELCOR flow paths, and 13 MELCOR heat structures representing the CV of the CSTF. The model divides the CV into six sections in the axial direction, and two in the radial direction. The cylindrical portion of the CV was modeled by four axial zones, and the top and bottom head volumes were modeled by the two remaining axial zones. This nodalization allows the air heated by the lithium pool to rise up the axis of CV, and return down the outer radius of the CV after convective cooling by the CV walls. The cross-sectional area of the inner zone of cells equals that of the pool surface area in Test LA-5, which is 2 m^2. The outer zones have a cross-sectional area of 41.8 m^2. The height of the control volumes is 3.6 m, except the bottom head volume that has a height of 2.2 m. The MELCOR flow paths have been assigned the same flow areas as that of the corresponding control volumes that these flow paths connect. No form losses were assigned to the vertical flow paths, but form losses of 0.4 were assigned to the radial flow paths to simulate 90 ° turns in the convective flow pattern.

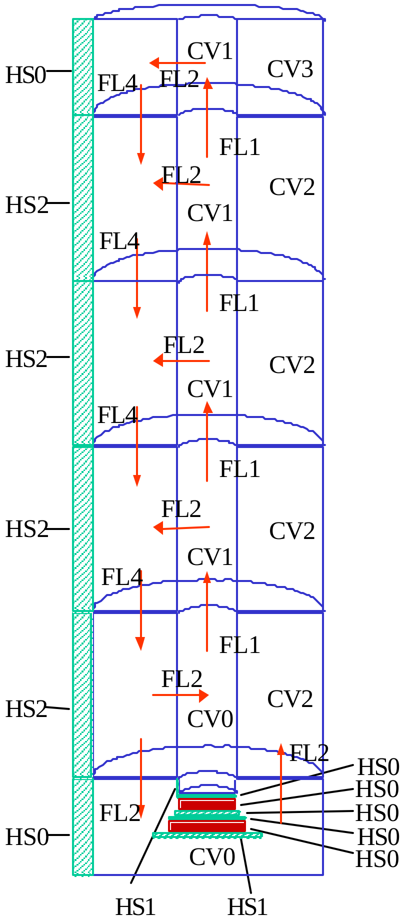

Figure 9. Schematic of MELCOR CSTF containment vessel model.

Heat structures representing the walls of the CV were attached to each of the outer control volumes. The four heat structures representing the walls of the cylindrical portion of the CV have surface areas of 98.5 m^2, and thickness of 0.01862 m. The heat structures that represent the top and bottom heads of the vessel have surface areas of 63 m^2, with a thickness of 0.02839 m and 0.01862 m, respectively. These vessel thicknesses were set to give the reported CV weight, with the top head heat structure sized to include the internal CV components attached to the top of the CV. Because thermal properties for the CV wall material, called out as SA-212-B carbon steel by Reference [18], were not readily available, the CV wall material was assumed to be 316 SS. The exterior convective boundary condition for these walls was convection to the ambient with a heat transfer coefficient of 1.6 W/m^2-K. This heat transfer coefficient accounted for 25.4-mm layer of fiberglass insulation covering the exterior of the CV, which has a thermal conductivity of 0.047 W/m-K [18], and an assumed air convection coefficient of 10 W/m^2-K.

The reaction pan/catch pan assembly was modeled by seven heat structures. The bottom and side of the reaction pan were model as two heat structures. The catch pans and firebrick layers were model by the remaining five heat structures. Heat transfer from the side-wall of the reaction pan was by convection to the atmosphere of the control volume, calculated by the standard MELCOR convection package, and by radiant heat loss to the CV lower head (effective emissivity of 0.17, which is an emissivity of 0.3 per surface assumed [19]). Heat transfer from the bottom of the reaction pan through the catch pan assemble was by gap conduction between heat structures and conduction through heat structures. The gap conductance includes a gap coefficient (typically 10-20 W/m^2-K, which varied per experiment to match measured pool temperatures during the lithium charging period of a test), and thermal radiation. This gap conductance is scaled by the square root of the gap temperature during a calculation, to reflect the fact that the gap gas conductivity would scale as the square root of temperature. The effective emissivity of the gap was set at 0.3, which assumes an emissivity value of 0.3 for the SS316 and 1.0 for the firebrick. Temperature dependent material properties of density, specific heat, thermal conductivity, and emissivity for SS316 where obtained from Reference [19]. Temperature dependent material properties for Missouri firebrick were obtained from Reference [20], and those for KAOWOOL from Reference [21].

A final modeling issue that needs to be addressed relates to the MELCOR model for mass and heat transfer at the surface of the lithium pool described in Section 3.2. As can be seen from Equation 34 of Section 3.2, there is a convective and radiative component to the heat transfer between the pool and atmosphere in a MELCOR non-equilibrium control volume. The problem with the radiative component is that it assumes perfect absorption of the thermal radiation by the atmosphere in the volume. This is probably a good assumption for a water-vapor atmosphere, but not for an air atmosphere. Future modifications will be made to this MELCOR model that will more accurately estimate the quantity of heat being absorbed by the vapor. But for this benchmarking study, the radiative heat transfer component of Equation 34 was set to zero, and a user defined MELCOR control function employed to predict the radiative heat transfer between the pool surface and the CV walls for the following benchmark studies.

4.3 Test LA-4 Benchmark Study Results

Comparisons of MELCOR predictions with data from Test LA-4 are contained in Figures 10 through 17. Figure 10 contains the averaged pool temperature from Test LA-4 and MELCOR predicted pool temperatures. As can be seen, excellent agreement was achieved between two MELCOR predictions and test results. The averaged pool temperature of Figure 10 is that of thermocouple locations 2 through 4 of Figure 4. The pool temperature at 2.54 cm above the reaction pan floor was not included in this average because this thermocouple resides in a region of the pool that experienced a large thermal gradient. The MELCOR predictions correspond to this average temperature because the lithium pool of the MELCOR model was a lumped representation of the pool, that is the pool was not sub-divided into a multiple MELCOR control volumes. The MELCOR results were arbitrarily terminated at 3200 seconds, which is the time at which the reaction pan failed in this test. Two MELCOR predictions have been included. The first case, designated natural convection at the pool interface (NCPI), disallowed forced convection heat transfer at the pool surface. This was accomplished by setting sensitivity coefficient 4407(2) to zero in the user input. The second case, designated forced convection at the pool interface (FCPI), allowed forced convection heat transfer at the pool surface. Because these MELCOR cases gave similar results and because the forced convection heat transfer coefficient is a factor of three larger than the natural convection heat transfer coefficient, it could be concluded that convective heat transfer did not play an important role in Test LA-4. In fact, predictions for this test were not very sensitive to any of the heat transfer modeling assumptions detailed in Section 4.2, for example the gap conductance coefficient between the reaction pan and the firebrick layer beneath the pan. This is probably the consequence of the KAOWOOL and firebrick insulating layers that surround the reaction pan of Test LA-4. As can also be seen, the MELCOR prediction for forced convection at the pool interface gives slightly better pool temperatures than does the case with natural turbulent convection.

Based on MELCOR predictions, the lithium reactions of Test LA-4 produced 170 MJ of energy in the first 3200 seconds of the test. Of this energy, 9 MJ was conducted through the bottom of the reaction pan, 7 MJ was convected from the outer surface of the KAOWOOL to the CV atmosphere, 5 MJ was radiated from the KAOWOOL outer surface to the CV walls, 20 MJ was convected from the inner surface of the reaction pan to the atmosphere, 5 MJ produced lithium pool vaporization, and 46 MJ was radiated from the surface of the pool to the CV walls. To obtain the pool surface radiative rate, an emissivity coefficient of 1.0 was used. This value was adopted because: 1) emissivity data could not be found for Li_2O and Li_3N, 2) it gave the best fit to the data for the above modeling assumptions, and 3) photographs in Reference [18] of the residue from the reaction pan appears to be a “grayish” color, which suggests a high surface emissivity coefficient.

Figures 11 and 12 contain the gmoles per unit area (m^2) of oxygen and nitrogen gas consumed during Test LA-4. As can be seen, both MELCOR cases examined agreed well with the measured value from the experiment. Of interest here are the results for nitrogen for two reasons. First, the rate of consumption of nitrogen decreases to nearly zero in the data between test times of 2700 seconds and 3200 seconds. Note that this transition occurs prior to the failure of the reaction pan during this experiment. The corresponding temperature range is about 1300 K to 1350 K. According to Reference [12], two important phenomena occur for lithium nitride (Li_3N) near this temperature range: 1) lithium nitride melts at 1082 K, and 2) the phase equilibrium constant for the lithium-nitrogen reaction becomes 1.0 at a temperature of about 1225 K, suggesting that above this temperature Li_3N should start to decompose. Second, to achieve the measured rate of nitrogen consumption the reaction rate equation (Equation 27 for nitrogen) had to be multiplied by a factor of 0.3. This suggests that only 30% of the mass diffusing through the gas boundary layer actually reacts with the lithium. In contrast, the measured rate of oxygen consumption matched the predicted rate (Equation 27) without the use of a multiplying factor.

Figures 13 and 14 contain comparisons for CV average atmosphere temperature and average wall temperature, respectively, for Test LA-4. Fair agreement was obtained with the two MELCOR cases examined. The important features of these figures are that the MELCOR prediction for natural convection heat transfer at the pool surface more accurately matches the test data, and that very little heat transfer occurred to the CV atmosphere and walls during the first 3200 seconds of Test LA-4. The measured CV atmosphere temperature rise was 4 K, and the CV wall temperature rise was 1.7 K. This compares to a 7 K and 10 K rise for the CV atmosphere temperature and 1.8 K and 2.0 K rise for the CV wall temperature for the MELCOR cases examined.

The energy transferred directly to the CV wall by thermal radiation was about 51 MJ and that transferred to the CV atmosphere by convection was 31 MJ. If the heat transfer to the atmosphere were to be reduced by 50% to match the measured atmosphere temperature rise, the pool and CV wall temperature rise would increase. Therefore, it could be argued that heat of reaction used in this new MELCOR model is about 9% too high (i.e., 0.5x31 MJ/170 MJ = 0.09). Consequently, a third MELCOR case was examined to test this hypothesis. This case is similar to Case 1 but has a reduced heat of reaction (NCRH). Figures 15 through 17 contain the results of this case, compared to Test LA-4 data. As can be seen, excellent agreement was obtained for this case. It could be argued that a similar result could have been obtained by increasing the heat transfer through the bottom of the reaction pan and catch pan assembly. It is unfortunate that measured catch pan temperatures for this test where not reported, because this would help answer this point. However, for this argument to be correct, the heat transfer through this structure would have to be increased by a factor of 170% since the predicted energy flow through the bottom of the reaction pan was only 9 MJ for this test. The same could be said about increasing the radial heat flow through the side-wall of the reaction pan.

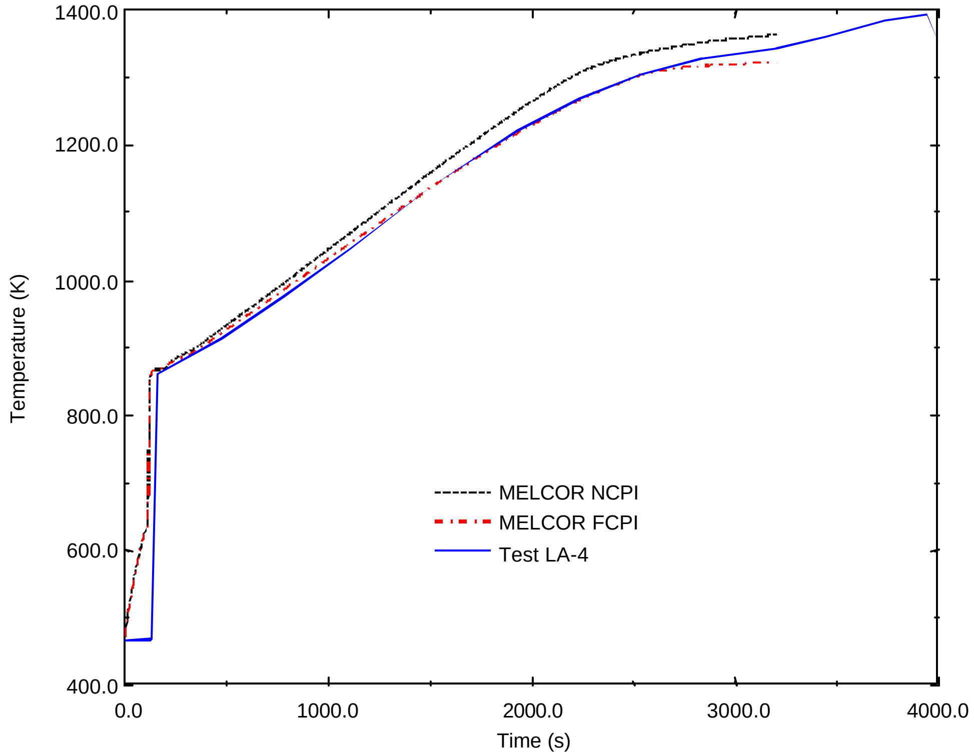

Figure 10. Comparison of MELCOR predicted pool temperature with data from Test LA-4.

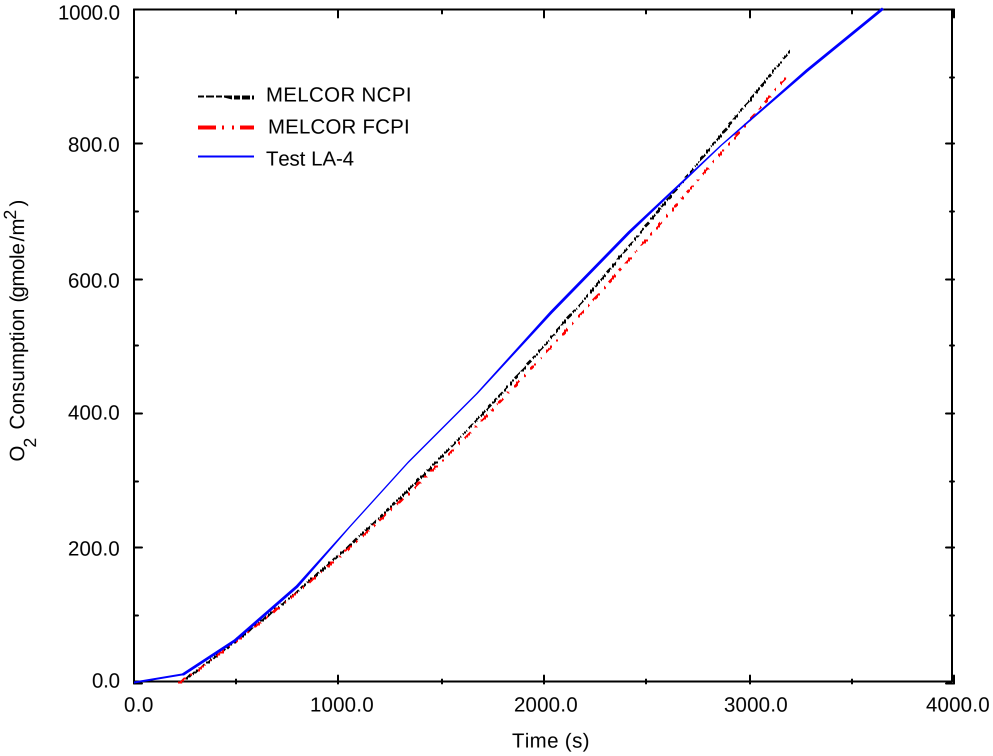

Figure 11. Comparison of MELCOR predicted oxygen consumption with that measured during Test LA-4.

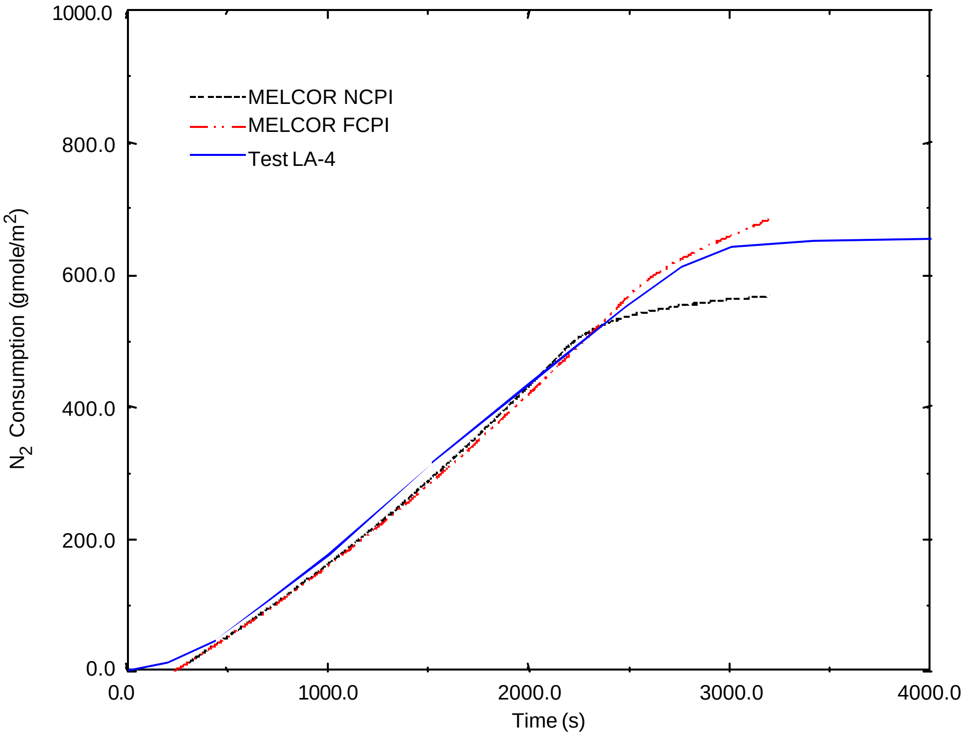

Figure 12. Comparison of MELCOR predicted nitrogen consumption with that measured during Test LA-4.

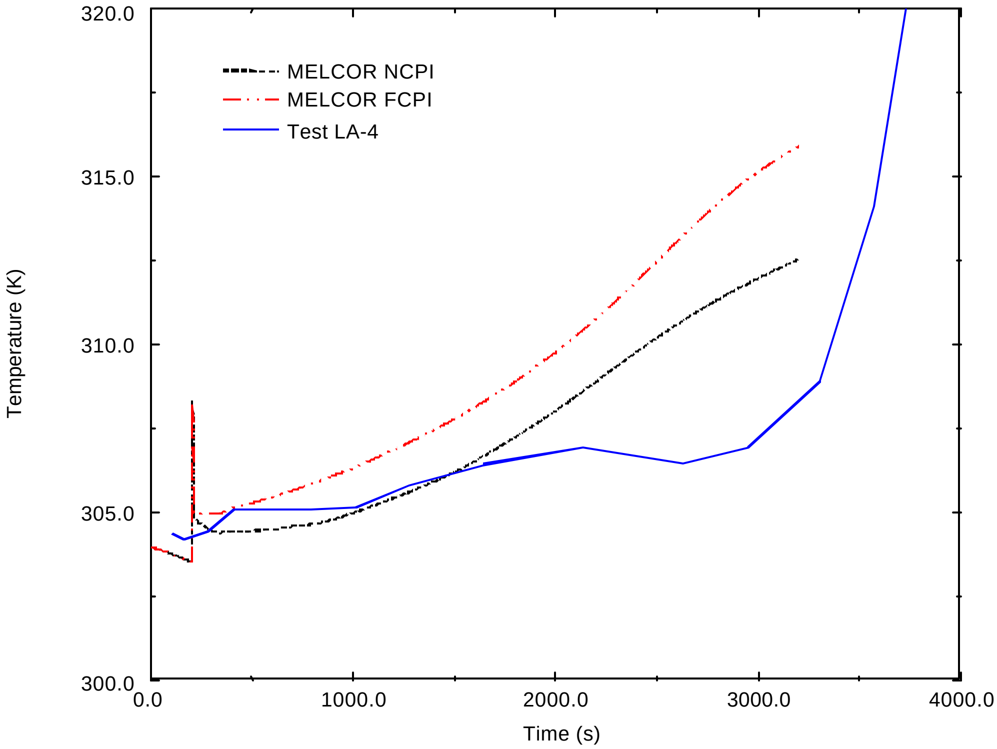

Figure 13. Comparison of MELCOR predicted CV atmosphere temperature with that measured during Test LA-4.

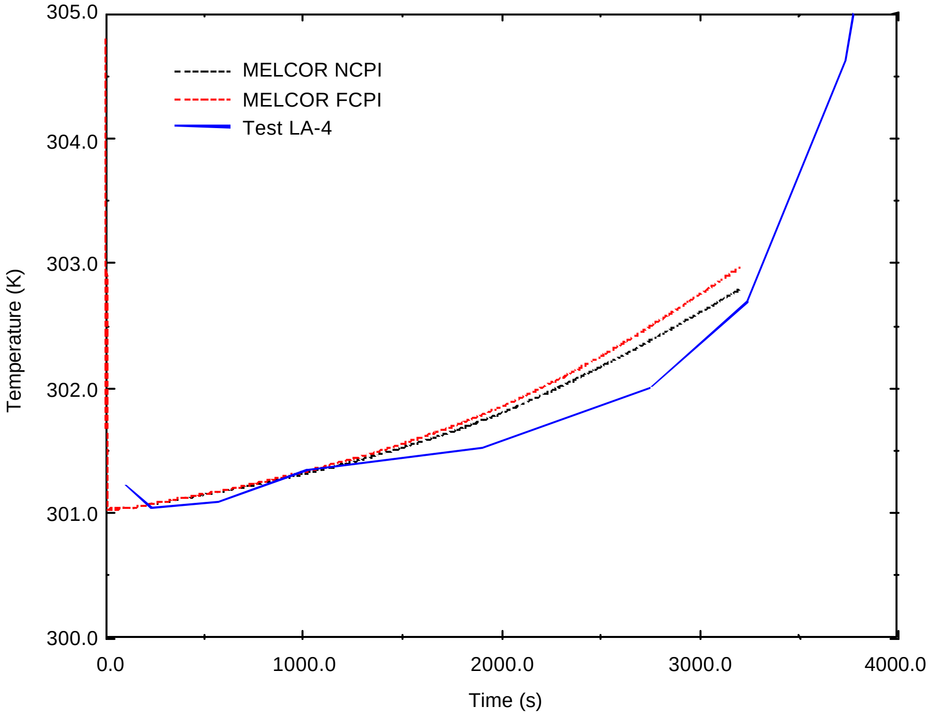

Figure 14. Comparison of MELCOR predicted CV wall temperature with that measured during Test LA-4.

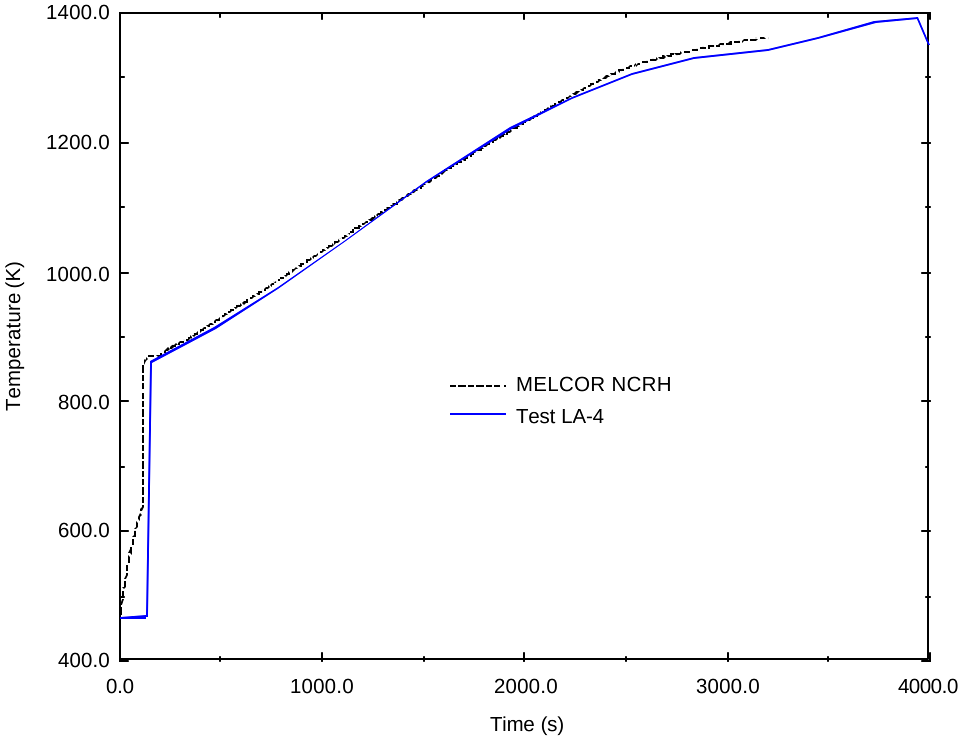

Figure 15. Comparison of MELCOR reduced heat of reaction case predicted lithium pool temperature with that measured during Test LA-4.

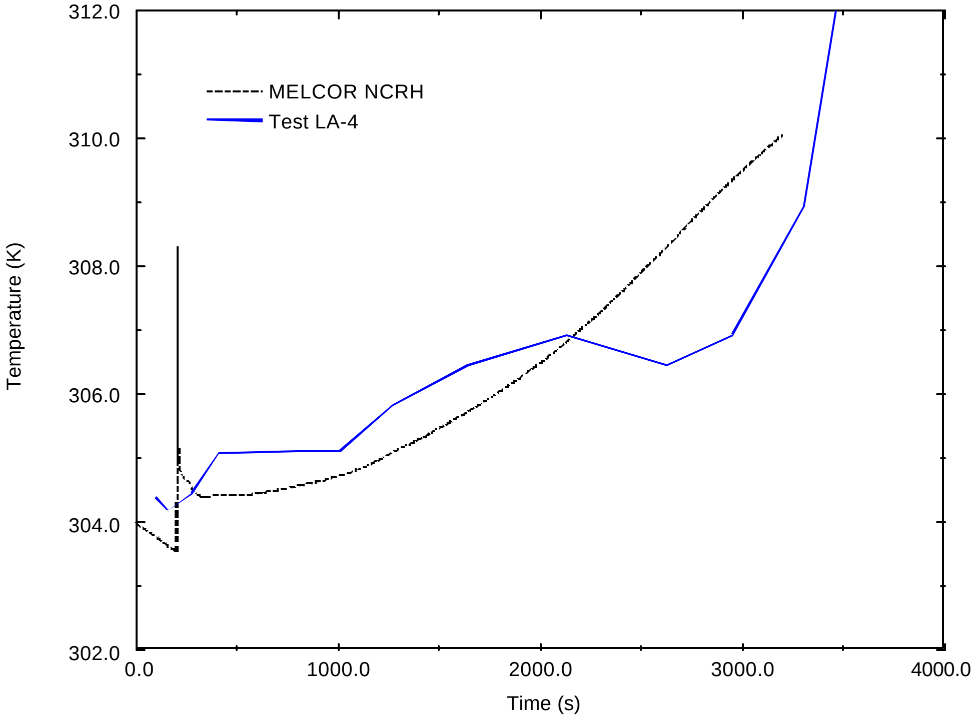

Figure 16. Comparison of MELCOR reduced heat of reaction case predicted CV atmosphere temperature with that measured during Test LA-4.

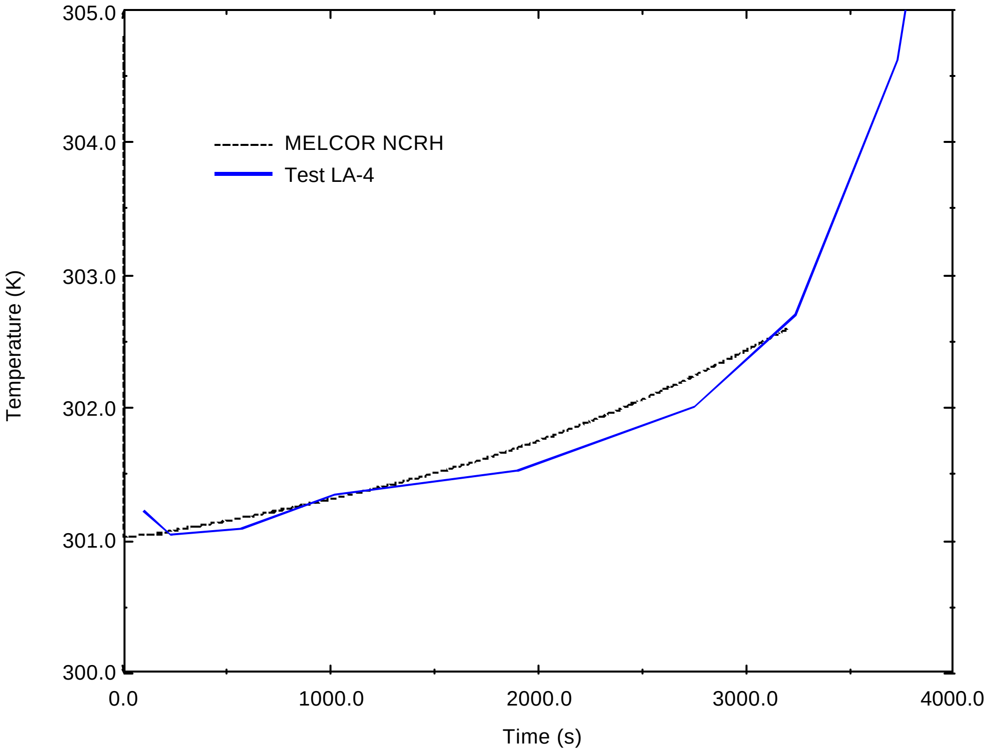

Figure 17. Comparison of MELCOR reduced heat of reaction case predicted CV wall temperature with that measured during Test LA-4.

4.4 Test LA-5 Benchmark Study Results

Results for Test LA-5 appear in Figures 18 through 23. Figure 18 contains a comparison of Test LA5 average pool temperature from thermocouples 1 and 2 (note Figure 3) with MELCOR predictions. The same three cases examined for Test LA4 were examined for Test LA-5, which are: 1) natural convection at the pool atmosphere interface (NCPI), 2) forced convection at the pool atmosphere interface (FCPI), and 3) natural convection at the pool atmosphere interface plus a reduced heat of reaction (NCRH). The comparison for all three cases is very good. However, to follow the measured temperature before 900 seconds and after 1700 seconds three modeling changes had to be made to the MELCOR model used for analyzing Test LA-4. The first change dealt with the difference in reaction pan dimensions between Test LA-4 and Test LA-5, and removal of the KOAWOOL insulation from the reaction pan side-wall. The second modeling change affected the rate at which the lithium reacts with oxygen and nitrogen during the first 900 seconds. As reported by Reference [18]: “During the initial 900 seconds, localized flames were observed as small, ‘popcorn kernel’-like fires on the surface of the pool. Lithium appeared to wick up on these “kernels,” burn and slowly grow in diameter to cover the pool surface.” To simulate the growth of the reaction zone an Arrhenius type reaction relationships were derived from the oxygen and nitrogen consumption data for Test LA-5 during this period of time. These relationships are:

$$\Gamma_{O2} = 1.8x10^{6}\ e^{-15360/T} \qquad (38)$$

for oxygen, where Γ_{O2} is the oxygen consumption rate (kg/m^2-s), and T is the pool temperature (K), and for nitrogen

$$\Gamma_{N2} = 5.8x10^{8}\ e^{-18780/T} \qquad (39)$$

where Γ_{N2} is the nitrogen consumption rate (kg/m^2-s). As such, these relationships compensate for the inability of a one-dimensional heat conduction model to accurately represent a three-dimensional heat conduction problem. The final modeling change dealt with the adopted pool surface emissivity. For Test LA4 a constant value of 1.0 was used. However, for Test LA-5 an emissivity that varied with temperature had to be used to match test data. The emissivity as a function of pool temperature used appears in Figure 19.

Figures 20 and 21 contain the gmoles per unit area (m^2) of oxygen and nitrogen gas consumed during Test LA-5. As can be seen, the MELCOR cases agree well with the test data. The consumption of nitrogen for Test LA-5 is similar to that of Test LA-4, that is the measured rate of nitrogen consumption occurrs at 30% of the predicted rate from Equation 27 for nitrogen. However, there is a difference in the temperature at which the reaction starts to cease. In Test LA-5 this temperature was 1200 K (i.e., pool temperature a 2250 seconds), compared to 1300 K for Test LA-4. It also appears that the temperature range over which the decrease occurs is larger for Test LA-5. To match the measured oxygen consumption for test LA-5, the predicted rate of consumption had to be multiplied by a factor of 0.6. This differed from Test LA-4 that did not require any multiplying factor applied to Equation 27. It is also of interest to note that the lithium-oxygen reaction starts to decrease before the termination of Test LA-5 (i.e., at a time of 3200 seconds and a temperature of 1350 K).

Figures 22 and 23 contain comparisons of average CV atmosphere and wall temperatures for Test LA-5. Fairly good agreement was obtained with the MELCOR cases examined. Cases 2 and 3, which are forced convection heat transfer at the pool interface (FCPI) and natural convection at the pool interface plus a reduction in heat of reaction (NCRH), bracket the measured CV atmosphere and CV wall temperature rise. However, Case 1, which is natural convection at the pool interface (NCPI), appears to more accurately match both temperatures. The measured CV atmosphere temperature rise is 50 K, while the predicted rise for Cases 1 through 3 is 48 K, 60 K, and 40 K, respectively. Similarly the measure CV wall temperature rise is 20 K, while the predicted rise for Cases 1 through 3 is 17 K, 22 K, and 15 K, respectively.

The energy balance for Test LA-5 is that of the 1410 MJ of energy liberated, 410 MJ was radiated from the pool surface to the CV walls, 210 MJ was conducted through the bottom of the reaction pan into the catch pan assembly, 73 MJ was convected to the atmosphere from the reaction pan side wall, 115 MJ was radiated from the reaction pan side-wall, and 155 MJ produced lithium vaporization at the pool surface. This division of energy suggests a shift in importance of heat transfer path between Test LA-4 and Test LA-5 from the side-wall of the reaction pan to the bottom of the reaction pan. For Test LA-5 the energy split was 15% through the bottom of the reaction pan, and 13% through the side-wall of the reaction pan. For Test LA-4, this energy split was 5.3% and 18.8%, respectively. Radiation from the pool surface accounted for 29% of the energy transferred in Test LA-5 and 27% in Test LA-4. This is interesting given the dramatic difference in the modeling assumptions used regarding pool surface emissivity between Test LA-4 and Test LA-5. The fraction of energy produced by the lithium-nitrogen reaction for Test LA-5 was 45% of the total, compared to only 33% for Test LA-4. This is due to the lower lithium-oxygen reaction rate measured for Test LA-5.

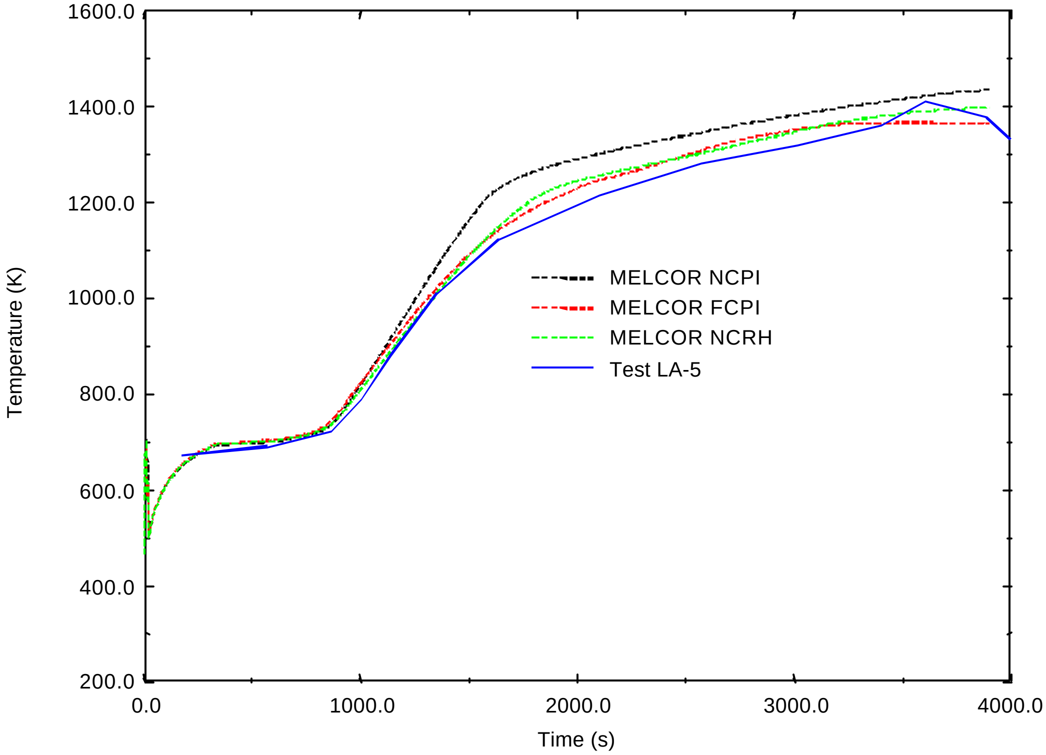

Figure 18. Comparison of MELCOR predicted pool temperature with data from Test LA-5.

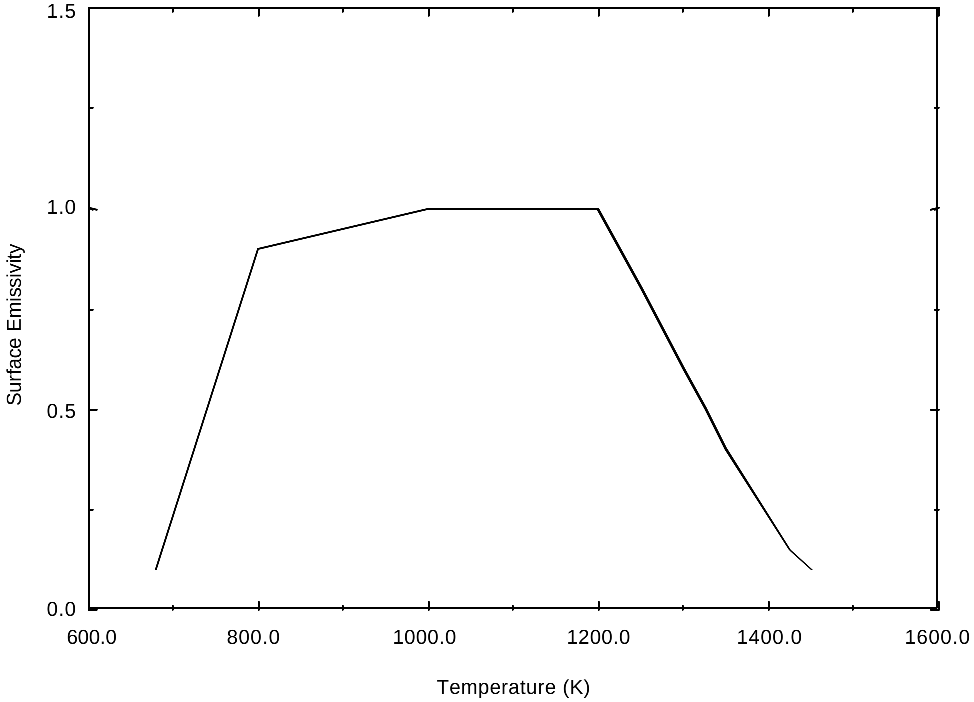

Figure 19. Pool surface emissivity use by MELCOR to match pool temperature data from Test LA-5.

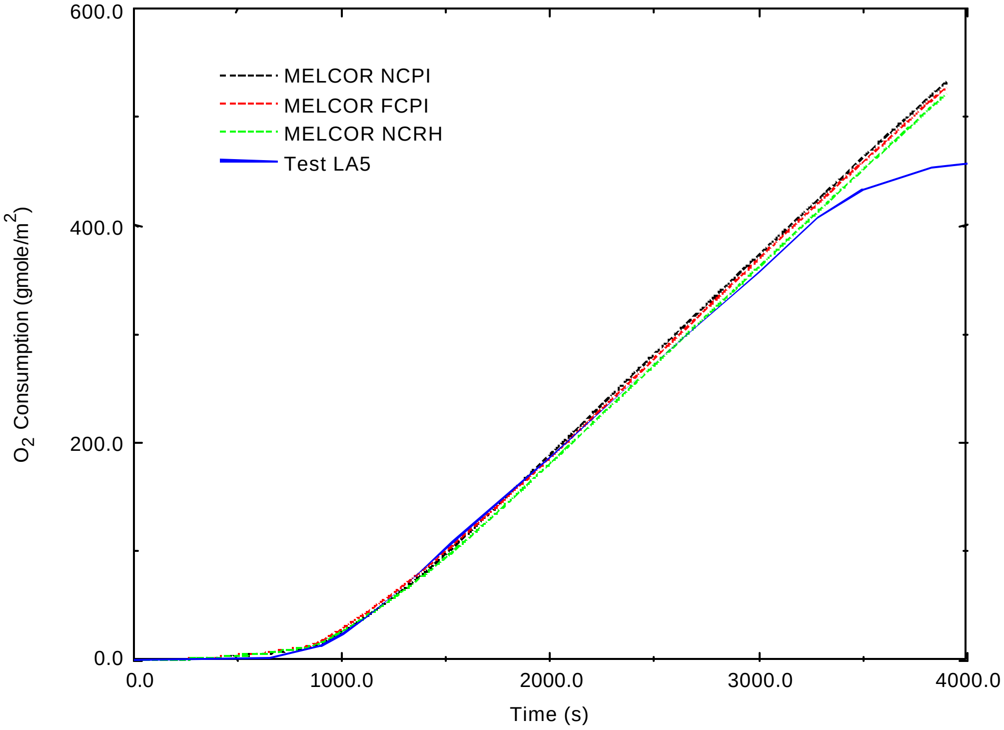

Figure 20. Comparison of MELCOR predicted oxygen consumption with that measured during Test LA-5.

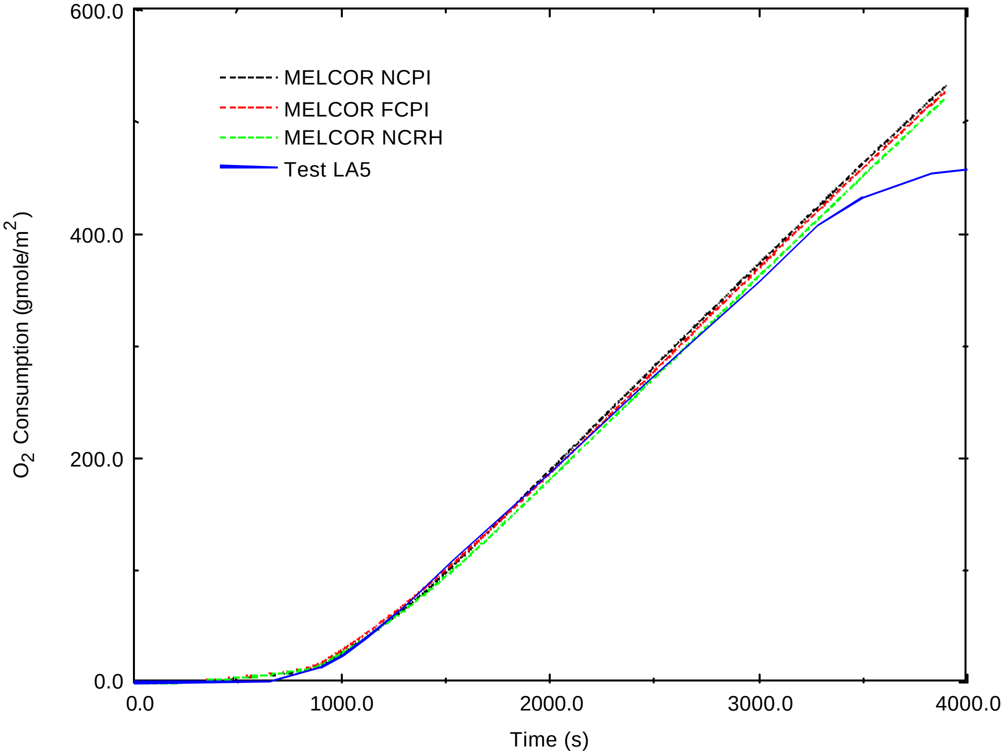

Figure 21. Comparison of MELCOR predicted nitrogen consumption with that measured during Test LA-5.

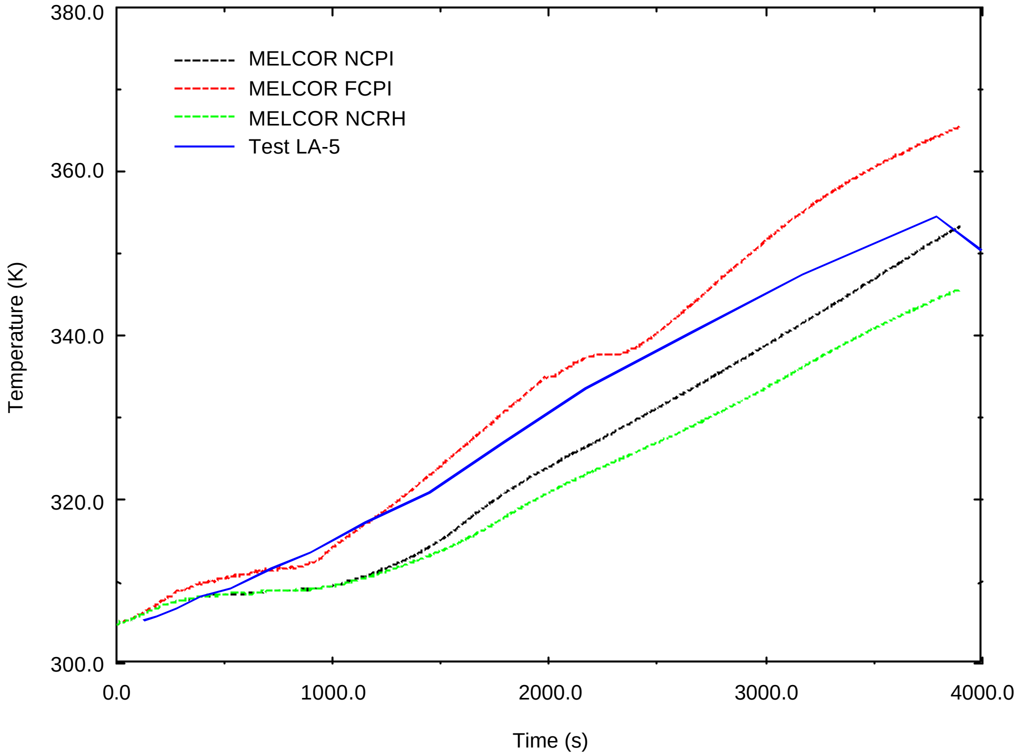

Figure 22. Comparison of MELCOR predicted CV atmosphere temperature with that measured during Test LA-5.

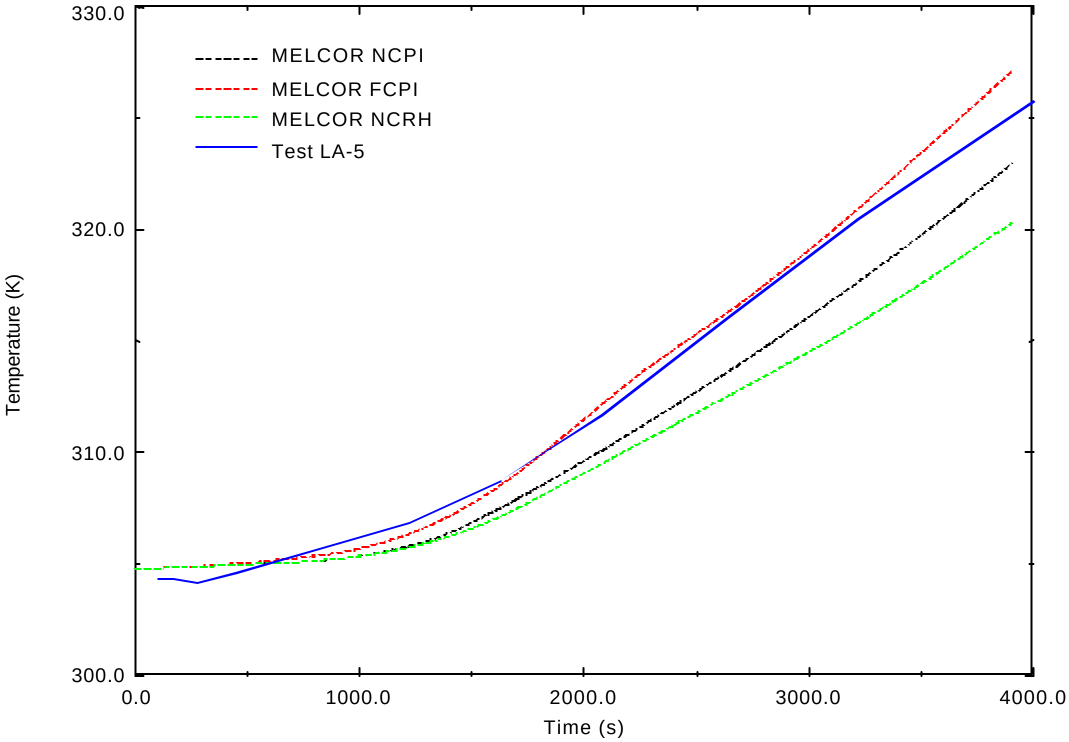

Figure 23. Comparison of MELCOR predicted CV wall temperature with that measured during Test LA-5.

5. CONCLUSIONS AND RECOMENDATIONS

This report documents the effort required to incorporate a new lithium-air fire model in the MELCOR code. This effort includes the introduction of a lithium Equation-of-State accomplished by modifying MELCOR to access the Fusion Safety fluid property database, and by adding subroutines to MELCOR that predict the rate at which lithium reacts with air. These modifications not only allow the Fusion Safety group at the INEEL to examine safety issue of MFE and IFE reactor designs that propose liquid lithium as a coolant, but for other liquid metal and molten salt coolant designs as well.

Two lithium-air reaction tests conducted at the HEDL were selected to benchmark this new lithium capability in MELCOR. These tests were conducted in a large containment vessel. Lithium pool temperature, vessel atmosphere, vessel wall, and oxygen and nitrogen consumption were measured. The two lithium-air spill tests examined were designated Tests LA-4 and LA-5. Test LA-4 was designated as a deep pool spill test, where 43 kg of liquid lithium at a temperature of 600 °C was transferred into a reaction pan, resulting in a lithium pool with a diameter to depth ratio of 0.89. Test LA-5 was designated as a shallow pool spill test, where 100 kg of liquid lithium at a temperature of 500 °C was transferred into a reaction pan, resulting lithium pool diameter to depth ratio of 16.

Three MELCOR cases where compared to data from Test LA-4 and LA-5. Case 1 disabled the forced convection heat transfer option between the pool and the atmosphere of non-equilibrium MELCOR control volume, resulting in only natural convective heat transfer being predicted between the pool and the atmosphere. The second case was the standard pool to atmosphere heat transfer model in MELCOR which accepts the greater heat transfer coefficient predicted from the heat transfer regimes of forced convection, turbulent natural convection, and laminar natural convection heat transfer. The third case was the same as Case 1, but the heat of reaction of the lithium-air reactions was reduced by 9%. Based on the results obtained, the use of the standard MELCOR pool-atmosphere interface heat transfer model (Case 2) is not recommended for use in conjunction with the new MELCOR lithium fire capability for three reasons. First, the use of forced convection (Equation 35) for mass transfer at the pool surface is not consistent with the used of turbulent natural convection (Equation 25) for oxygen and nitrogen mass transport at the surface of the pool. Second, the forced convection correlation depends on predicted atmosphere velocity. This velocity depends on the level of modeling detail the user undertakes in modeling the natural convective flow patterns above the pool, a calculation for which MELCOR is not well suited. Third, overall the results of Case 1 were a better match to measured pool temperature, CV atmosphere temperature, and CV wall temperature for these tests. While Case 3 gave better results for Test LA-4 and for the pool temperature for Test LA-5, but this was not the case for CV atmosphere and wall temperature measurements for Test LA-5. Since heat conduction through the reaction pan is a more dominate heat transfer path for Test LA-5 and because measurements of catch pan temperatures were not reported by Reference [18], the adoption of the heat of reaction multiplier from Case 3 can not be recommended at this time for inclusion in the lithium fire model.

In order to match the nitrogen consumption data for Tests LA-4 and LA-5, the proposed model (Equation 27 for nitrogen) had to be multiplied by a factor of 0.3. The oxygen consumption rate for Test LA-4 did not require a multiplier to match the data, but the consumption rate for LA-5 required a multiplier of 0.6. For both tests the nitrogen reaction ceased prior to the end of the test. In Test LA-4 the lithium-nitrogen reaction began to cease above 1300 K, and for Test LA-5 the reaction began to cease above 1200 K. The oxygen reaction in Test LA-4 did not show an indication of ceasing during the test, but in Test LA-5 the reaction began to cease above a temperature of 1350 K. The reason for the nitrogen rate limit is unclear but it is not believed to be due to the existence of a film or crust on the pool surface for two reasons. First, oxygen did not require a rate limit to match the data from Test LA-4, and the binary diffusion coefficients for oxygen and nitrogen vary by less than a percent and not by a factor of three. Second, there is no difference in the magnitude of the rate limit for nitrogen between Test LA-4 and Test LA-5. It is possible that the 70% reduction in lithium-nitrogen reaction is due to surface kinetics, that is by the rate at which liquid lithium will actually react with nitrogen. However, a plausible explanation for the oxygen rate limit in Test LA-5 is the existence of a film or crust. A thin crust could result in the 40% reduction in the arrival rate of oxygen to the liquid lithium surface between Test LA-4 and Test LA-5, but at the same time not dramatically affect the rate of nitrogen consumption which is limited by surface kinetics and not nitrogen rate of arrival. However, whether a thin crust formed during the course of events in Test LA-5 or a thick patchy crust formed during the first 900 seconds of this test can not be determined. These tests also indicated that the nitrogen reaction decreases after reaching 1300 K in Test LA-4 and 1200 K in test LA-5. For Test LA-4 this decrease is believed to be due to the fact that the thermochemical conditions of this test above 1300 K were no longer favorable to the formation of Li_3N. In Test LA-5, this decrease could be due to the build up of a crust on the pool surface. Because these conclusions are speculative and because of the lack of reproducible data, it was decided that a heuristic approach will be taken for simulating the above phenomena in this new MELCOR capability at this time. This approach assumes that the consumption rates scale as the pool diameter to depth ratio. That is, based on the results obtained empirically from the benchmarking studies in Section 4.3 and 4.4, the rate lithium-oxygen reaction rate (Equation 27) becomes:

$$\Gamma_{O2} = f_{O2} \, K_{m_{O2}} \, r_{O2} \qquad (40)$$

where the rate limiting factor φ is defined as

$$j_{O2} = 1.0 - 0.4\left(D/h - 0.89\right)/15.11 \qquad (41)$$

For nitrogen, this equation becomes:

$$\Gamma_{N2} = f_{N2} \, K_{m_{N2}} \, r_{N2} \qquad (42)$$

where the rate limiting factor φ is defined as

$$j_{N2} = 0.3\,e^{-dT/t} \tag{43}$$

and where

$$dT = \max(0.0,\ T - T_I)$$

$$T = \text{lithium pool temperature (K)}$$

$$T_I = 1300.0 - 100.0\ \left(D/h - 0.89\right)/15.11\ (K)$$

$$t = 25.0 + 25.0\ \left(D/h - 0.89\right)/15.11\,(K)$$

Regarding aerosols, the fraction of Li_2O and Li_3N produced that will be suspended as an aerosol is:

$$j_s = 0.1 - 0.066\,(D/h - 0.89)/15.11 \tag{44}$$

The pool diameter to depth ratio (D/h) in a given control volume is calculated by MELCOR during the course of an analysis.

The final recommendation that needs to be addressed is pool radiant heat transfer. For the benchmark studies the radiant heat transfer rate was specified by user defined control functions that removed heat from the pool and deposited this heat on the walls of the CV. It will be up to the user to replicate this modeling approach. The pool emissivity used for different pool configurations, that is shallow verses deep pools, are reported in Sections 4.3 and 4.4. The physical explanation for a variable emissivity in Test LA-5 is not clear. There are least two possible explanations. First, it could be that after the nitrogen reaction ceases in Test LA-5 that only Li_2O is produced, causing a change in surface emissivity. Reference [22] states that Li_3N has a black-gray appearance while Li_2O has a white appearance. Second, it could also be that the surface emissivity does not change but a crust develops which adds resistance to heat flow from the pool surface, and drops the surface temperature below the measured pool temperature. For example, at 3900 seconds in Test LA-5 the surface radiant heat flux is about $3.2\text{x}10^4$ W/m^2, the pool temperature is 1420 K, and the wall temperature is 311 K. These conditions could also exist for a surface emissivity of 1.0 if a 3.0 cm crust had developed on the pool surface with a surface temperature of 871 K. This crust thickness was estimated by using a thermal conductivity for a Li_2O reported by Reference [23] of 1.73 W/m-K.

The modifications described in this report are intended as initial models for addressing lithium-air reaction safety concerns for future fusion reactors. Given the importance of this issue to the use of lithium in fusion reactors and the nature of these models recommend here, additional theoretical and experimental work must be undertaken to improve the recommendations presented in this section.

6. References

1. R. M. Summers, et al., *MELCOR 1.8.0: A Computer Code for Nuclear Reactor Severe Accident Source Term and Risk Assessment Analyses*, NUREG/CR-5531 and SAND90-0364, USNRC Report, Sandia National Laboratory, January 1991.
2. Keenen, Keyes, Hill, and Moore, 1969, *Steam Tables*, John Wiley & Sons, Inc.
3. K. E. Carlson, P. A. Roth, and V. H. Ransom, *ATHENA Code Manual, Volume 1 and 2*, EGG-RTH-7397, September 1986.
4. J. E. Tolli, *New Fluids Capability for the RELAP5 Based Integrated Code*, unpublished EG&G technical report, Idaho National Engineering Laboratory, September 1989.
5. J. E. Tolli, *Overview of Property Formulations for Helium, Nitrogen, Lithium, and Lithium-lead in ATHENA/MOD1 with Comparison of Calculated Properties to Measured Properties*, EGG-FSP-10245, Idaho National Engineering Laboratory, April 1992.
6. *RELAP5/MOD3 Code Manual*, NUREG/CR-5535, INEL-95/0174, Idaho National Engineering Laboratory, August 1995.
7. R. O. Gauntt, et al., *MELCOR Computer Code Reference Manuals, Version 1.8.4*, NUREG/CR-6119, Vol.2, Rev 1., SAND07-2398, July 1997.
8. Brad Merrill, Richard L. Moore, Steve T. Polkinghorne, and David A. Petti, "Modifications to the MELCOR Code for Application in Fusion Accident Analyses," 5th International Symposium on Fusion Nuclear Technology, Rome, Italy, September 19-24, 1999.
9. G. J. Van Wylen, R. E. Sonntag, 1965, *Fundamentals of Classical Thermodynamics*, John Wiley & Sons, Inc., pg. 336.
10. Ibid, pg. 337.
11. J. G. Collier, *Convective Boiling and Condensation*, McGraw-Hill, New York, pg. 316, 1972.
12. A. Roine, "Outokumpu HSC Chemistry for Windows, Chemical Reaction and Equilibrium Software with Extensive Thermochemical® Database," Version 4.0, Outokumpu Research Oy, P.O. Box 60, FIN-28-101 Pori, Finland, June 30, 1999.
13. J. R. Welty, C. E. Wicks, and R. E. Wilson, *Fundamentals of Momentum, Heat and Mass Transfer*, John Wiley & Sons, Inc., New York, 1969, pp. 534-558.
14. J. G. Collier, J. R. Thome, *Convective Boiling and Condensation*, 3rd edition, Claredon Press, Oxford, 1994, pg. 442.
15. T. T. Utschig and M. L. Corradini, "Lithium Safety in Fusion Systems (ITER)," 5th International Symposium on Fusion Nuclear Technology, Rome, Italy, September 19-24, 1999.
16. R. C. Reid, J. M. Prausnitz, and B. E. Poling, *The Properties of Gases and Liquids*, Fourth Edition, McGraw-Hill, Inc., New York, 1987, pg. 587.
17. J. R. Welty, C. E. Wicks, and R. E. Wilson, *Fundamentals of Momentum, Heat and Mass Transfer*, John Wiley & Sons, Inc., New York, 1969, pg. 460.
18. D. W. Jeppson, *Results and Code Prediction Comparisons of Lithium-air Reaction and Aerosol Behavior Tests*, HEDL-TME 85-25, Hanford Engineering Development Laboratory, March 1986.

19. H.-W. Bartels, et al., *Safety Analysis Data List-2 (SADL-2)*, S 81 RI 97-05004 W1.1, version 2,1, Section 4, ITER San Diego JWS, June 30, 1997.
20. J. R. Welty, C. E. Wicks, and R. E. Wilson, *Fundamentals of Momentum, Heat and Mass Transfer*, 2nd Edition, John Wiley & Sons, Inc., New York, 1969, pg. 730.
21. Sunwacork Insulation, *KAOWOOL Physical Properties*, Sunwacork Webpage, www.sunwacork.com.jk/hewpage8.htm, January 2000.
22. R. C. Weast, *CRC Handbook of Chemistry and Physics*, Florida:CRC Press, Inc., Boca Raton, 1979.
23. Y. S. Touloukian, *Thermal Conductivity: Nonmetallic Solids*, Vol. 2, IFI/Plenum, New York, 1970.

Appendix A

FORTRAN Listing of MELCOR Modifications to Allow MELCOR to Access the Fusion Safety Fluid Property Database

```
    SUBROUTINE H2O1PH
C
C     EVALUATE EOS (SINGLE PHASE - LIQUID OR VAPOR)
C                                       SLT 5/19/84
C
C*********************************************************************
C
C     DATA FROM
C                  ATHENA FLUID PROPERTY PACKAGE
C
C
C       INPUT VARIABLES IN COMMON BLOCK H2OCB1
C               T=TEMPERATURE IN K
C               RHO=DENSITY IN KG/M3
C
C       OUTPUT VARIABLES IN COMMON BLOCK H2OCB1
C               P=PRESSURE IN PA (=NT/M2)
C               E=ENERGY IN JOULES/KG
C               S=ENTROPY IN JOULES/(KG-K)
C               H=ENTHALPY IN JOULES/KG
C               CV=SPECIFIC HEAT IN JOULES/(KG-K)
C               DPT=PRESSURE DERIVATIVE WRT TEMPERATURE PA/K
C               DPR=PRESSURE DERIVATIVE WRT DENSITY PA-M3/KG
C               DHT=ENTHALPY DERIVATIVE WRT TEMPERATURE JOULES/(KG-K)
C               IERR=ERROR INDICATOR FLAG
C                   =0 NO ERROR
C                  =1 T TOO HIGH (.GT.TMAX) - NOT ERROR, USE JANAF DATA
C                   =2 T TOO LOW (.LT.TMIN)
C                   =-1 RHO TOO HIGH FOR GIVEN T
C                   =-2 RHO TOO LOW (.LE.0)
C                   =-3 IN UNPHYSICAL REGION WITH DPR.LT.0
C
C
C
C*********************************************************************
C
C     DO CALCULATION OF Q AND QR=DQ/DR IN DOUBLE PRECISION
C     ON SHORT-WORD MACHINES TO AVOID CANCELLATION IN P
C
*-
C
      COMMON /H2OCB1/ P,E,S,H,CV,DPT,DPR,DHT,T,RHO,IERR
      COMMON /H2OCB2/ CLEST,CL1PH,CL2PH,CLSAT,CNTIT1,CNTIT2,CNFAL
     1 ,TLEST,TL1PH,TL2PH,TLSAT
      COMMON /H2OCB3/ KUNIT
      COMMON /H2OCB4/ TSATOR,KPASE
      COMMON /H2OCBP/ ANSPH,RVAP,RLIQ,AMV,AML,ALPHA
C
C
C     THE NEXT COMMON BLOCK CONTAINS THE INTERNAL ENERGY AND ENTROPY
C     INTEGRATION CONSTANTS IN MKS UNITS
C
C
C      PARAMETER (RLICE=991.064148D0, RVICE=4.83837730D-3)
C      PARAMETER (ELICE= -2.230824D3, EVICE=2.0691669D6)
C
      COMMON / EOSPARM /  TCRIT, RCRIT, PCRIT, TSATO, TSINV, CVLFUS
```

```
     .,            TRIPT, TRIPP, TRIPR, TMIN,  PMIN,  RMIN
     .,            TMAX,   PMAX, RMAX,  NPTS,  INIT,  NFLUID
C
      REAL*8 TCRIT, RCRIT, PCRIT, TSATO, TSINV, CVLFUS
     .,      TRIPT, TRIPP, TRIPR, TMIN,  PMIN,  RMIN
     .,      TMAX,   PMAX,  RMAX
C
      REAL*8 PZEDC, PTMAX, PRMAX, PONE, PZERO
      LOGICAL ERR
C
      PZEDC = TSATO
      PZERO = 0.D0
      PONE  = 1.D0
      PTMAX = TMAX
      PRMAX = RMAX
      RLICE = TRIPR
C
C     UPDATE CALL COUNT
      KPASE = 1
      CL1PH = CL1PH + PONE
      ANSPH = PONE
      TK = T
      R = RHO
      IERR = 0
C
C
C     TEST TABLE LIMITS ******************************************
C
C     =======================
C     CHECK T
C     =======================
C
      IF (TK .EQ. PZEDC) THEN
       GO TO 40
C
C     LOW T =====================
      ELSEIF (TK .LT. PZEDC) THEN
       IERR = 2
       TK = PZEDC
C
C     HIGH T ====================
      ELSEIF (TK .GT. PTMAX) THEN
       IERR = 1
       TK = PTMAX
      ENDIF
C
   40 CONTINUE
C
C
C     =======================
C     CHECK RHO
C     =======================
C
C     LOW RHO ==============
      IF (R .LE. PZERO) THEN
       R = 1.D-20
       IERR = -2
```

```
C
C     HIGH RHO =============
      ELSE
       IF (R .GT. PRMAX) THEN
          R = PRMAX
          IERR = -1
       ENDIF
      ENDIF
C
      CALL FLUIDEOS(NFLUID, R, TK, P, E, S, DPT, DPR, CV, ERR)
C
C     ================================================================
C     FINISHED
C     ================================================================
C
C
C     CHECK FLAG ***************************************************
C
      IF (IERR .NE. 0) THEN
C
C        THE FOLLOWING PATCHS ARE TO ALLOW ITERATIONS TO FUNCTION
C        OUTSIDE OF DEFINED PARAMETER RANGE ONLY
C
       IF(IERR .EQ. -1) THEN
C
C           ===============================
C           PATCH TO EXTEND TO HIGH DENSITY
C           ===============================
C
          Q = (P - TK*DPT) / (R*R)
          E = E + Q * (RHO - R)
          P = P + DPR * (RHO - R)
          IF (T .EQ. TK) GO TO 220
       ENDIF
C
       IF (T .GT. PTMAX) THEN
C
C           ====================================
C           PATCH TO EXTEND TO HIGH TEMPERATURES
C           ====================================
C
C           MELCOR HOT STEAM FOR T.GT.1589 (VARIABLE CV PERFECT GAS)
          P = P + DPT * (T - TK)
          DHT = LOG(T / TK)
          E = E + CV * (T - TK)
          S = S + CV * DHT
          IF (IERR .EQ. 1) THEN
             IERR = 0
             GO TO 240
                      ENDIF
                    ENDIF
C
C     LOW T =====================
C
C  ================= MERRILL MODS FOR FREEZING
C            HEAT OF FUSION TREATED OVER ONE DEGREE KELVIN
C            EXTRAPOLATE AT CONSTANT PROPERTIES BELOW FREEZING
```

```
C
      IF (IERR .EQ. 2)  THEN
       EK = E
       PK = P
       SK = S
C
       TKM1 = TK-1.
C
C  ---------- VAPOR
       IF(R.LT.RLICE) THEN
       P = PK + DPT*(T-TK)
       E = EK + CV*(T-TK)
       S = SK + CV*LOG(T/TK)
                  ELSE
C  ---------- ICE
       DPT = PK/TK
       P = PK + DPT*(T-TK)
       IF(T.GT.TKM1)      THEN
       CV = CVLFUS
       E  = EK + CV*(T-TK)
       S  = SK + CV*LOG(T/TK)
                          ELSE
       E  = EK - CVLFUS + CV*(T-TKM1)
       S  = SK + CV*LOG(T/TKM1)
     .              + CVLFUS*LOG(TKM1/TK)
                         ENDIF
C
                  ENDIF
       IERR = 0
C
                  ENDIF
C
  220    CONTINUE
C
                   ENDIF
C
  240    CONTINUE
       IF (DPR .LT. PZERO) IERR = -3
C
      H = E + P/ R
      DHT = CV + DPT/ R
C
C     ***************************************************************
      RETURN
      END
*DECK H2O2PH
CMP$ /H2O2PHMJ/         1/04/93 14:31:06 SLT
      SUBROUTINE H2O2PH
C
C     EVALUATE STEAM TABLES (TWO-PHASE - LIQUID-VAPOR)
C     PART OF H2OEST ROUTINE
C     THIS SUBROUTINE EVALUATES PHASE LINE PROPERTIES FROM TABLES
C
C**********************************************************************
*
C       IF KPASE=2 ON RETURN HAVE TWO-PHASE STATE
C       IF KPASE=1 ON RETURN HAVE ONE-PHASE STATE AND NO
```

```
C                      THERMODYNAMIC VARIABLES HAVE BEEN DEFINED
C
C     WARNING   THIS SUBROUTINE HAS INTERNAL UNITS IN CGS.
C               ALL EXTERNAL VARIABLES ARE IN SI.
C
C                      WRITTEN BY S.L.THOMPSON, SNLA, 6444  7/20/83
C
C             RESTRUCTURED BY J.F.CUDERMAN II, SNLA, 6415 JUN 87
C             INTERPOLATION OF RLIQ CHANGED BY R.K.COLE/6418 JAN 90
C                                        Last mod SLT 11/25/92
C**********************************************************************
*
C
      COMMON /H2OCB1/ P,E,S,H,CV,DPT,DPR,DHT,T,RHO,IERR
      COMMON /H2OCB2/ CLEST,CL1PH,CL2PH,CLSAT,CNTIT1,CNTIT2,CNFAL
     1 ,TLEST,TL1PH,TL2PH,TLSAT
      COMMON /H2OCB4/ TSATOR,KPASE
      COMMON /H2OCB5/ RL(376),RV(376),PV(376)
      COMMON /H2OCB6/  EL(376),EV(376)
     1,DPTL(376),DPTV(376),SL(376),SV(376)
     2,CVL(376),CVV(376),DPRL(376),DPRV(376)
C
C     THE NEXT COMMON BLOCK CONTAINS THE INTERNAL ENERGY AND
C     ENTROPY INTEGRATION CONSTANTS
C
      COMMON /H2OZER/ EH2OZR , SH2OZR
C
C     NEXT TWO COMMON BLOCKS CONTAIN OTHER TWO-PHASE PROPERTIES
C     WHICH MAY BE OF INTEREST TO USER OF ROUTINES. THEY ARE NOT
C     EMPLOYED BY THE REST OF THE H2O PACKAGE
C     SEE H2OEST FOR VARIABLE DEFINITIONS
C
      COMMON /H2OCBP/ ANSPH,RVAP,RLIQ,AMV,AML,ALPHA
      COMMON /H2OCBQ/ ELIQ,EVAP,SLIQ,SVAP,CLIQ,CVAP
     1               ,PTLIQ,PTVAP,PRLIQ,PRVAP
C
C
      PARAMETER ( PZERO = 0.0D0 , PHALF = 0.5D0 , PONE = 1.0D0 )
C
      COMMON / EOSPARM /  TCRIT, RCRIT, PCRIT, TSATO, TSINV, CVLFUS
     .,            TRIPT, TRIPP, TRIPR, TMIN,  PMIN,  RMIN
     .,            TMAX,   PMAX, RMAX,  NPTS,  INIT,  NFLUID
C
      REAL*8 TCRIT, RCRIT, PCRIT, TSATO, TSINV, CVLFUS
     .,      TRIPT, TRIPP, TRIPR, TMIN,  PMIN,  RMIN
     .,      TMAX,   PMAX,  RMAX
C
      REAL*8 TINV, OTINV, TKM, TKM1, TKM2, SQTKM2
C
      IF(INIT.EQ.0) THEN
      CALL INITFLD
      INIT = 1
                ENDIF
C
      TINV = TSINV
      OTINV = 1.D0/TSINV
      TKM = TSATO
```

```
      TKM1 = TKM-1.
C
C
C     ======================
C     CHECK FOR SINGLE PHASE
C     ======================
C
      TK = T
      IF (TK .GE. TCRIT) GO TO 1
C
C
C     ===============================================================
C     MORE THAN ONE PHASE PRESENT
C     ===============================================================
C
C     UPDATE CALL COUNT
      CL2PH = CL2PH + PONE
      ANSPH = PZERO
      R = RHO
      DT = TK - TKM
      N = DT*OTINV + PONE
      IERR = 0
C
      IF (DT .GT. PZERO) THEN
C
C
C     ==============================
C     DETERMINE NUMBER OF PHASES
C     ==============================
C
      N2 = N + 1
      DT = (DT - (N-1)*TINV) * OTINV
      DT1 = PONE - DT
C
      IF (N .LT. 373) THEN
C      RLIQ FROM QUADRATIC THROUGH N,N+1,N+2
       DT2 = PONE + DT1
       RLIQ = PHALF*DT1*DT2*RL(N) + DT*DT2*RL(N+1) -
     1          PHALF*DT1*DT*RL(N+2)
       RVAP = RV(N)*DT1 + RV(N2)*DT
       IF (RLIQ.LE.R .OR. RVAP.GE.R) GO TO 1
      ELSE IF (N .LT. 374) THEN
       DTM1 = PONE + DT
C      RLIQ FROM QUADRATIC THROUGH N-1,N,N+1
       RLIQ = DT1*DTM1*RL(N) + PHALF*DT*DTM1*RL(N+1) -
     1          PHALF*DT1*DT*RL(N-1)
       RVAP = RV(N)*DT1 + RV(N2)*DT
       IF (RLIQ.LE.R .OR. RVAP.GE.R) GO TO 1
      ELSE
       CALL H2ORNC(TK,RLIQ,RVAP)
       IF (RLIQ.LT.R .OR. RVAP.GT.R .OR. RVAP.GE.RLIQ) GO TO 1
       DT = (RV(N) - RVAP) / (RV(N) - RV(N2))
       DT1 = PONE - DT
      ENDIF
C
C
C     TWO-PHASE STATE *********************************************
```

```
C     ************************************************************
C
C     ==========================================
C     DETERMINE PROPERTIES OF EACH PHASE AND MIX
C     ==========================================
C
C
      PVAP = PV(N)*DT1 + PV(N2)*DT
C
      ELIQ = EL(N)*DT1 + EL(N2)*DT
      EVAP = EV(N)*DT1 + EV(N2)*DT
C
      SLIQ = SL(N)*DT1 + SL(N2)*DT
      SVAP = SV(N)*DT1 + SV(N2)*DT
C
      CLIQ = CVL(N)*DT1 + CVL(N2)*DT
      CVAP = CVV(N)*DT1 + CVV(N2)*DT
C
      PTLIQ = DPTL(N)*DT1 + DPTL(N2)*DT
      PTVAP = DPTV(N)*DT1 + DPTV(N2)*DT
C
      PRLIQ = DPRL(N)*DT1 + DPRL(N2)*DT
      PRVAP = DPRV(N)*DT1 + DPRV(N2)*DT
C
                   ELSE
C
C  ================= MERRILL MODS FOR FREEZING
C            HEAT OF FUSION TREATED OVER ONE DEGREE KELVIN
C            EXTRAPOLATE AT CONSTANT PROPERTIES BELOW FREEZING
C
      PVAP  = PV(1) + DPTV(1)*(TK-TKM)
C
      CVAP  = CVV(1)
      EVAP  = EV(1) + CVAP*(TK-TKM)
      SVAP  = SV(1) + CVAP*LOG(TK/TKM)
C
      IF(TK.GT.TKM1) THEN
      CLIQ  = CVLFUS
      ELIQ  = EL(1) + CLIQ*(TK-TKM)
      SLIQ  = SL(1) + CLIQ*LOG(TK/TKM)
                   ELSE
      CLIQ  = CVL(1)
      ELIQ  = EL(1) - CVLFUS
     .              + CLIQ*(TK-TKM1)
      SLIQ  = SL(1) + CLIQ*LOG(TK/TKM1)
     .              + CVLFUS*LOG(TKM1/TKM)
                   ENDIF
C
CCCCC      PTLIQ = DPTL(1)
      PTLIQ = PV(1)/TKM
      PTVAP = DPTV(1)
C
      PRLIQ = DPRL(1)
      PRVAP = DPRV(1)
C
      RLIQ  = RL(1)
      RVAP  = RV(1)
```

```
C       RVAP  = RV(1)*(TKM/TK)*(PVAP/PV(1))
C
      IF(R.LT.RVAP .OR. R.GT.RLIQ) GO TO 1
C
                   ENDIF
      KPASE = 2
C
      ODR = PONE / (RLIQ - RVAP)
      ORHO = PONE / R
C
      DPTS = RLIQ * RVAP * ODR * (SVAP - SLIQ)
C
      AMV = RVAP * ORHO * (ODR*(RLIQ - R))
      AML = PONE - AMV
C
      IF (AML .LE. 0.1D0) THEN
       AML = RLIQ * ORHO * (ODR*(R - RVAP))
       AMV = PONE - AML
      ENDIF
C
      DRLIQ = (DPTS - PTLIQ) / PRLIQ
      DRVAP = (DPTS - PTVAP) / PRVAP
C
      CVS = (ELIQ - EVAP) * ORHO * (ODR*ODR)
     1* (RLIQ*(R - RLIQ)*DRVAP + RVAP*(RVAP - R)*DRLIQ)
     2+ AML*(CLIQ + (PVAP - TK*PTLIQ)*DRLIQ / (RLIQ*RLIQ))
     3+ AMV*(CVAP + (PVAP - TK*PTVAP)*DRVAP / (RVAP*RVAP))
C
      DPRS = PZERO
      PS = PVAP
      ES = AML*ELIQ + AMV*EVAP
      SS = AML*SLIQ + AMV*SVAP
      ALPHA = R*AMV / RVAP
C
C     ************************************************************
C     RESET ANSWERS FOR RETURN
C     ************************************************************
C
C     ===========================
C     PUT IN SI UNITS
C     ===========================
C
      P = PS
      E = ES
      S = SS
      CV = CVS
      DPT = DPTS
      DPR = DPRS
      H = E + P/RHO
      DHT = CV + DPT/RHO
C
C     ===========================
      RETURN
C     ===========================
C
C
C     SINGLE-PHASE STATE *****************************************
```

```
C     ************************************************************
C
    1 CONTINUE
      KPASE = 1
C
C
C     ===========================
      RETURN
C     ===========================
C
      END
*DECK MPPRP1
      SUBROUTINE MPPRP1 (IDMAT, X1, X2, Y, JERROR)
C
C     THIS GROUP OF MATERIAL PROPERTIES SUBROUTINE CALCULATES
C       THE PROPERTIES FOR VARIOUS MATERIALS.
C
C           WRITTEN BY JOHN ORMAN FOR MELCOR  4/15/85
C
C     INPUT:
C       IDMAT - MATERIAL ID NUMBER
C       X1    - FIRST INDEPENDENT VARIABLE
C       X2    - SECOND INDEPENDENT VARIABLE
C
C     OUTPUT:
C       Y     - DEPENDENT VARIABLE OF MATERIAL IDMAT
C       JERROR - ERROR FLAG, = 0 IF INPUT PARAMETERS O.K.
C                           .NE. 0  IF INPUT PARAMETERS OFF-SCALE
C
C
C     WRITTEN FOR MATERIAL PROPERTIES BY JOHN ORMAN   4/15/85
C
      SAVE
      EXTERNAL MPPRPL
C
*- INCLUDE BLANK
C                                    GENERIC DOUBLE PRECISION VERSION
C     MAIN DATABASE COMMON BLOCKS
C     REL = REALS, INT = INTEGERS, CHR = CHARACTERS, LOG = LOGICALS
C     NUM*** = DIMENSION OF ARRAY
C     NED*** = PHYSICS PACKAGES WORKING + SCRATCH STORAGE
C     IXXXCS = POINTER TO START OF SCRATCH STORAGE
C     I***ES = POINTER TO START OF EDIT PACKAGE SCRATCH STORAGE
C     XREALX, INTEGE, LOGICA, CHARAC STORAGE ARRAYS
C
C     NUMREL = MAXIMUM NUMBER OF REALS            (DIMENSION OF XREALX)
      PARAMETER (NUMREL = 500000)
C     NUMINT = MAXIMUM NUMBER OF INTEGERS         (DIMENSION OF INTEGE)
      PARAMETER (NUMINT = 50000)
C     NUMLOG = MAXIMUM NUMBER OF LOGICALS         (DIMENSION OF LOGICA)
      PARAMETER (NUMLOG = 5000)
C     NUMCHR = MAXIMUM NUMBER OF CHARACTER*8 VARIABLES  (DIM OF CHARAC)
      PARAMETER (NUMCHR = 30000)
C
C     DECLARING DREALX IN COMMON ENSURES THAT STORAGE BEGINS AT THE
C     CORRECT 8-BYTE BOUNDARY ON THOSE MACHINES THAT CARE
C
```

```
      COMMON /DBREAL/ NEDREL, IRELCS, IRELES
      COMMON /      /                         DREALX(NUMREL/2)
      DOUBLE PRECISION DREALX
      COMMON /DBINTG/ NEDINT, IINTCS, IINTES, INTEGE(NUMINT)
      COMMON /DBLOGC/ NEDLOG, ILOGCS, ILOGES, LOGICA(NUMLOG)
      LOGICAL LOGICA
      COMMON /DBCHAD/ NEDCHR, ICHRCS, ICHRES
      COMMON /DBCHAR/                         CHARAC(NUMCHR)
      CHARACTER*8 CHARAC
C
C     SINGLE PRECISION REALS
C
      DIMENSION XREALX(NUMREL)
      EQUIVALENCE (XREALX(1), DREALX(1))
C
C     VARIABLE PRECISION REALS - DOUBLE FOR 32 BIT MACHINES
C                                SINGLE FOR 64 BIT MACHINES
C
      DOUBLE PRECISION VREALX(NUMREL/2)
      EQUIVALENCE (VREALX(1), DREALX(1))
*-
C
*- INCLUDE MPDB
C
      COMMON /MPDB/  IRMPF , IRMPN , IIMPF , IIMPN ,
     1                ILMPF , ILMPN , ICMPF , ICMPN
C
C     IRMPF  = THE FIRST ELEMENT OF THE REAL ARRAY CONTAINING
C              THE MATERIAL PROPERTIES DATA.
C     IRMPN  = THE NUMBER OF ELEMENTS OF THE REAL ARRAY
C              CONTAINING MATERIAL PROPERTIES DATA.
C
C     IIMPF  = THE FIRST ELEMENT OF THE INTEGER ARRAY CONTAINING
C              THE MATERIAL PROPERTIES DATA.
C     IIMPN  = THE NUMBER OF ELEMENTS OF THE INTEGER ARRAY
C              CONTAINING MATERIAL PROPERTIES DATA.
C
C     ILMPF  = THE FIRST ELEMENT OF THE LOGICAL ARRAY CONTAINING
C              THE MATERIAL PROPERTIES DATA.
C     ILMPN  = THE NUMBER OF ELEMENTS OF THE LOGICAL ARRAY
C              CONTAINING MATERIAL PROPERTIES DATA.
C
C     ICMPF  = THE FIRST ELEMENT OF THE CHARACTER (*8) ARRAY
C              CONTAINING THE MATERIAL PROPERTIES DATA.
C     ICMPN  = THE NUMBER OF ELEMENTS OF THE CHARACTER (*8) ARRAY
C              CONTAINING MATERIAL PROPERTIES DATA.
C
*-
C
*- INCLUDE MPPRNO
C
C          COMMON BLOCK /MPPRNO/ HAS ORDERING OF PROPERTIES IN LIST
C          GIVEN IN /MPPRNM/
C
      COMMON /MPPRNO/ NUMENH, NUMTMP, NUMCPS, NUMTHC, NUMVIS, NUMDIF,
     1                NUMRHO
C
```

```
C                  NUMENH = 1       ENTHALPY
C                  NUMTMP = 2       TEMPERATURE
C                  NUMCPS = 3       SPECIFIC HEAT
C                  NUMTHC = 4       THERMAL CONDUCTIVITY
C                  NUMVIS = 5       VISCOSITY
C                  NUMDIF = 6       DIFFUSIVITY
C                  NUMRHO = 7       DENSITY
C
*-
C
*- INCLUDE MPMATS
C
C        THE COMMON BLOCK /MPMATS/ CONTAINS MATERIAL ID NUMBERS
C        USED AS INPUT PARAMETERS FOR CALLS TO THE MPAAAB SUBROUTINES.
C        ADD URANIUM METAL (MUMETL) AND GRAPHITE (MGRAPH) TO MATERIALS
C
      PARAMETER (NMPPNT = 35)
C
      COMMON /MPMATS/ MWATER, MSTEAM, MAIR, MH2, MHE, MAR, MD2,
     1                MZR, MZRO2, MUO2, MSS, MSSOX, MB4C, MAGINC,
     2                MEU, MUMETL, MGRAPH, MCON, MO2, MCO2, MCO,
     3                MN2, MNO, MN2O, MNH3, MC2H2, MCH4, MC2H4,
     4                MUF6, MALUM, MAL2O3, MCADM, MSS304, MLIAL, MUAL
C
C           MWATER = MATERIAL ID NUMBER FOR WATER
C           MSTEAM = MATERIAL ID NUMBER FOR STEAM
C           MAIR   = MATERIAL ID NUMBER FOR AIR
C           MH2    = MATERIAL ID NUMBER FOR HYDROGEN
C           MHE    = MATERIAL ID NUMBER FOR HELIUM
C           MAR    = MATERIAL ID NUMBER FOR ARGON
C           MD2    = MATERIAL ID NUMBER FOR DEUTERIUM
C           MZR    = MATERIAL ID NUMBER FOR ZIRCALOY
C           MZRO2  = MATERIAL ID NUMBER FOR ZIRCONIUM DIOXIDE
C           MUO2   = MATERIAL ID NUMBER FOR URANIUM DIOXIDE
C           MSS    = MATERIAL ID NUMBER FOR STAINLESS STEEL
C           MSSOX  = MATERIAL ID FOR STAINLESS STEEL OXIDE
C           MB4C   = MATERIAL ID NUMBER FOR BORON CARBIDE
C           MAGINC = MATERIAL ID FOR AG-IN-CD
C           MEU    = MATERIAL ID FOR EUTECTIC OF ZIRCALOY AND UO2
C           MUMETL = MATERIAL ID FOR URANIUM METAL
C           MGRAPH = MATERIAL ID FOR GRAPHITE
C           MCON   = MATERIAL ID FOR CONCRETE
C           MO2    = MATERIAL ID FOR OXYGEN
C           MCO2   = MATERIAL ID FOR CARBON DIOXIDE
C           MCO    = MATERIAL ID FOR CARBON MONOXIDE
C           MN2    = MATERIAL ID FOR NITROGEN
C           MNO    = MATERIAL ID FOR NITRIC OXIDE
C           MN2O   = MATERIAL ID FOR NITROUS OXIDE
C           MNH3   = MATERIAL ID FOR AMMONIA
C           MC2H2  = MATERIAL ID FOR ACETYLENE
C           MCH4   = MATERIAL ID FOR METHANE
C           MC2H4  = MATERIAL ID FOR ETHYLENE
C           MUF6   = MATERIAL ID FOR URANIUM HEXAFLUORIDE
C           MALUM  = MATERIAL ID FOR ALUMINUM
C           MAL2O3 = MATERIAL ID FOR ALUMINUM OXIDE (AL2O3)
C           MCADM = MATERIAL ID FOR CADMIUM
C           MSS304 = MATERIAL ID FOR STAINLESS STEEL 304
```

```
C            MLIAL = MATERIAL ID FOR LITHIUM-ALUMINUM
C            MUAL = MATERIAL ID FOR URANIUM-ALUMINUM
C
      DIMENSION MMPPNT(NMPPNT)
      EQUIVALENCE (MMPPNT(1),MWATER)
C
      COMMON / EOSPARM /  TCRIT, RCRIT, PCRIT, TSATO, TSINV, CVLFUS
     .,             TRIPT, TRIPP, TRIPR, TMIN,  PMIN,  RMIN
     .,             TMAX,   PMAX, RMAX,  NPTS,  INIT,  NFLUID
C
      REAL*8 TCRIT, RCRIT, PCRIT, TSATO, TSINV, CVLFUS
     .,      TRIPT, TRIPP, TRIPR, TMIN,  PMIN,  RMIN
     .,      TMAX,   PMAX,  RMAX
      CHARACTER*8 STATE
      LOGICAL ERR
*-
C
      ENTRY MPENH1 (IDMAT, X1, X2, Y, JERROR)
C
      IPRNUM = NUMENH
      GO TO 10
C
      ENTRY MPTMP1 (IDMAT, X1, X2, Y, JERROR)
C
      IPRNUM = NUMTMP
      GO TO 10
C
      ENTRY MPCPS1 (IDMAT, X1, X2, Y, JERROR)
C
      IPRNUM = NUMCPS
      GO TO 10
      ENTRY MPTHC1 (IDMAT, X1, X2, Y, JERROR)
C
C              CHECK FOR SPECIAL MATERIALS
C
      IF (IDMAT.EQ.MWATER .OR. IDMAT.EQ.MSTEAM) THEN
C
      ITYPE  = 1
      JERROR = 0
      PRES   = 1.E5
      TEMP   = X1
      CALL H2OSAT(ITYPE,TSAT,PRES,RLIQS,RVAPS,JERROR)
      IF(IDMAT.EQ.MWATER) THEN
      STATE = 'liquid'
      RHO   = RLIQS
                        ELSE
      STATE = 'vapor'
      RHO   = RVAPS*TSAT/TEMP
                        ENDIF
      CALL THCOND (NFLUID, TEMP, PRES, RHO, STATE, Y, ERR)
      IF(ERR) JERROR = 1
      GO TO 100
                                              ENDIF
C
      IPRNUM = NUMTHC
      GO TO 10
      ENTRY MPVIS1 (IDMAT, X1, X2, Y, JERROR)
```

```
C
C             CHECK FOR SPECIAL MATERIALS
C
      IF (IDMAT.EQ.MWATER .OR. IDMAT.EQ.MSTEAM) THEN
C
      ITYPE  = 1
      JERROR = 0
      PRES   = 1.E5
      TEMP   = X1
      CALL H2OSAT(ITYPE,TSAT,PRES,RLIQS,RVAPS,JERROR)
      IF(IDMAT.EQ.MWATER) THEN
      STATE = 'liquid'
      RHO   = RLIQS
                          ELSE
      STATE = 'vapor'
      RHO   = RVAPS*TSAT/TEMP
                          ENDIF
      CALL VISCOS (NFLUID, TEMP, PRES, RHO, TSAT, STATE, Y, ERR)
      IF(ERR) JERROR = 1
      GO TO 100
                                                ENDIF
C
      IPRNUM = NUMVIS
      GO TO 10
      ENTRY MPDIF1 (IDMAT, X1, X2, Y, JERROR)
C
      IPRNUM = NUMDIF
      GO TO 10
      ENTRY MPRHO1 (IDMAT, X1, X2, Y, JERROR)
C
C             CHECK FOR SPECIAL MATERIALS
C
      IF (IDMAT .EQ. MSTEAM) THEN
C
         CALL MPRHSM(X1, X2, Y, JERROR)
C
         GO TO 100
      ENDIF
C
      IF (IDMAT .EQ. MAIR) THEN
C
         CALL MPRHAI(X1, X2, Y, JERROR)
C
         GO TO 100
      ENDIF
C
      IPRNUM = NUMRHO
      GO TO 10
C
10    CONTINUE
C
C       COMPUTE NUMBER OF SCRATCH STORAGE AVAILABLE
C
      NSCR = NEDREL - IRELCS + 1
C
C          PASS MATERIAL PROPERTIES
C         ALSO PASS SCRATCH STORAGE
```

```
C
      CALL MPPRPL(IDMAT, IPRNUM, X1, X2, Y, JERROR,
     1 IRMPN , XREALX(IRMPF) ,
     2 IIMPN , INTEGE(IIMPF) ,
     3 ILMPN,  LOGICA(ILMPF),
     4 ICMPN,  CHARAC(ICMPF),
     5 NSCR, XREALX(IRELCS) )
C
100   CONTINUE
C
      RETURN
      END
*DECK MPVIS
      SUBROUTINE MPVIS( EPS, SIG, XMOLW, TEMP, CEVISC )
C
C**********************************************************************
*
C   EVALUATES CHAPMAN-ENSKOG VALUES FOR GAS TRANSPORT PROPERTIES
C
C   ENTRY MPVIS RETURNS VALUE FOR VISCOSITY
C   ENTRY MPDIF RETURNS VALUE FOR BINARY DIFFUSIVITY
C
C   TABLES COPIED FROM TABLE B-2 IN BIRD, STEWART AND LIGHTFOOT,
C     'TRANSPORT PHENOMENA', WILEY, (1960)
C
C   ATTRIBUTED TO HIRSCHFELDER, BIRD, AND SPOTZ
C      CHEM. REVS. 44, 205 (1949)
C
C   EXTRAPOLATION USED ABOVE KT/EPS=100. GIVES BETTER THAN .1 PERCENT
C      FIT TO ALL DATA ABOVE KT/EPS=10.
C
C   EXTRAPOLATION USED BELOW KT/EPS=0.3 GIVES BETTER THAN .5 PERCENT
C      FIT TO ALL DATA BELOW KT/EPS=0.5
C
C                          CODED BY R.K. COLE, JR, SANDIA, NOVEMBER
1990
C**********************************************************************
*
C
      INTRINSIC MAX, MIN, SQRT
C
C   PARAMETERS IN EQUATIONS
C                          PUB COEFF   ATM TO PA   CM2/S TO M2/S
      PARAMETER ( PCEDIF = 1.8583D-3 * 1.01325D5 *    1.0D-4    )
C                          PUB COEFF   G/CM.S TO KG/M.S
      PARAMETER ( PCEVIS = 2.6693D-5 *      1.0D-1     )
C
C   PARAMETERS FOR PLACE-FINDING IN TABLES
      PARAMETER (  P0 = 0.0D0, PPT1 = 0.1D0,  P1 = 1.0D0,  P5 = 5.0D0,
     1            P10 = 10.D0,  P15 = 15.D0, P20 = 20.D0, P60 = 60.D0,
     2            P69 = 69.D0 )
C
C   PARAMETERS FOR G-MOL TO KG-MOL CONVERSIONS
      PARAMETER ( P1000 = 1000.D0 , PPT001 = 0.001D0 )
C
C   PARAMETERS FOR EXTRAPOLATION OF OMEGA AS C*T**PEXP FOR LARGE T
      PARAMETER ( PEXPVH = -0.145D0, PEXPDH = -0.155D0 )
```

```
C   PARAMETERS FOR EXTRAPOLATION OF OMEGA AS C*T**PEXP FOR SMALL T
      PARAMETER ( PEXPVL = -0.40D0,  PEXPDL = -0.50D0  )
C
C   TABLES OF OMEGA FUNCTIONS (COLLISION INTEGRALS)
      PARAMETER ( NCETBL = 79 )
      DIMENSION XKTE(NCETBL), OMEGAV(NCETBL), OMEGAD(NCETBL)
C
C   KT/EPSILON VALUES
C
      DATA XKTE/ 0.30D0, 0.35D0, 0.40D0, 0.45D0,
     1   0.50D0, 0.55D0, 0.60D0, 0.65D0, 0.70D0,
     2   0.75D0, 0.80D0, 0.85D0, 0.90D0, 0.95D0,
     3   1.00D0, 1.05D0, 1.10D0, 1.15D0, 1.20D0,
     4   1.25D0, 1.30D0, 1.35D0, 1.40D0, 1.45D0,
     5   1.50D0, 1.55D0, 1.60D0, 1.65D0, 1.70D0,
     6   1.75D0, 1.80D0, 1.85D0, 1.90D0, 1.95D0,
     7   2.0D0,  2.1D0,  2.2D0,  2.3D0,  2.4D0,
     8   2.5D0,  2.6D0,  2.7D0,  2.8D0,  2.9D0,
     9   3.0D0,  3.1D0,  3.2D0,  3.3D0,  3.4D0,
     A   3.5D0,  3.6D0,  3.7D0,  3.8D0,  3.9D0,
     B   4.0D0,  4.1D0,  4.2D0,  4.3D0,  4.4D0,
     C   4.5D0,  4.6D0,  4.7D0,  4.8D0,  4.9D0,
     D   5.D0,   6.D0,   7.D0,   8.D0,   9.D0,
     E   10.D0, 20.D0,  30.D0,  40.D0,  50.D0,
     F   60.D0, 70.D0,  80.D0,  90.D0, 100.D0/
C
C   COLLISION INTEGRALS FOR VISCOSITY
C
      DATA OMEGAV/ 2.785D0,  2.628D0,  2.492D0,  2.368D0,
     1   2.257D0,  2.156D0,  2.065D0,  1.982D0,  1.908D0,
     2   1.841D0,  1.780D0,  1.725D0,  1.675D0,  1.629D0,
     3   1.587D0,  1.549D0,  1.514D0,  1.482D0,  1.452D0,
     4   1.424D0,  1.399D0,  1.375D0,  1.353D0,  1.333D0,
     5   1.314D0,  1.296D0,  1.279D0,  1.264D0,  1.248D0,
     6   1.234D0,  1.221D0,  1.209D0,  1.197D0,  1.186D0,
     7   1.175D0,  1.156D0,  1.138D0,  1.122D0,  1.107D0,
     8   1.093D0,  1.081D0,  1.069D0,  1.058D0,  1.048D0,
     9   1.039D0,  1.030D0,  1.022D0,  1.014D0,  1.007D0,
     A   0.9999D0, 0.9932D0, 0.9870D0, 0.9811D0, 0.9755D0,
     B   0.9700D0, 0.9649D0, 0.9600D0, 0.9553D0, 0.9507D0,
     C   0.9464D0, 0.9422D0, 0.9382D0, 0.9343D0, 0.9305D0,
     D   0.9269D0, 0.8963D0, 0.8727D0, 0.8538D0, 0.8379D0,
     E   0.8242D0, 0.7432D0, 0.7005D0, 0.6718D0, 0.6504D0,
     F   0.6335D0, 0.6194D0, 0.6076D0, 0.5973D0, 0.5882D0/
C
C   COLLISION INTEGRALS FOR MASS DIFFUSIVITY
C
      DATA OMEGAD/ 2.662D0,  2.476D0,  2.318D0,  2.184D0,
     1   2.066D0,  1.966D0,  1.877D0,  1.798D0,  1.729D0,
     2   1.667D0,  1.612D0,  1.562D0,  1.517D0,  1.476D0,
     3   1.439D0,  1.406D0,  1.375D0,  1.346D0,  1.320D0,
     4   1.296D0,  1.273D0,  1.253D0,  1.233D0,  1.215D0,
     5   1.198D0,  1.182D0,  1.167D0,  1.153D0,  1.140D0,
     6   1.128D0,  1.116D0,  1.105D0,  1.094D0,  1.084D0,
     7   1.075D0,  1.057D0,  1.041D0,  1.026D0,  1.012D0,
     8   0.9996D0, 0.9878D0, 0.9770D0, 0.9672D0, 0.9576D0,
     9   0.9490D0, 0.9406D0, 0.9328D0, 0.9256D0, 0.9186D0,
```

```
     A   0.9120D0, 0.9058D0, 0.8998D0, 0.8942D0, 0.8888D0,
     B   0.8836D0, 0.8788D0, 0.8740D0, 0.8694D0, 0.8652D0,
     C   0.8610D0, 0.8568D0, 0.8530D0, 0.8492D0, 0.8456D0,
     D   0.8422D0, 0.8124D0, 0.7896D0, 0.7712D0, 0.7556D0,
     E   0.7424D0, 0.6640D0, 0.6232D0, 0.5960D0, 0.5756D0,
     F   0.5596D0, 0.5464D0, 0.5352D0, 0.5256D0, 0.5170D0/
C
C   GET INDEX IN TABLE AND INTERPOLATE OMEGA
C
      RKTE = TEMP/EPS
      IND = MIN( P20*RKTE-P5, P10*RKTE+P15, RKTE+P60, PPT1*RKTE+P69 )
      IF ( IND .LT. 1 ) THEN
         OMEGA = OMEGAV(1)*(RKTE/XKTE(1))**PEXPVL
      ELSE IF ( IND .LT. NCETBL ) THEN
         FRAC = MAX(P0,MIN(P1,
     1         (RKTE-XKTE(IND))/(XKTE(IND+1)-XKTE(IND)) ))
         OMEGA = OMEGAV(IND) + FRAC*(OMEGAV(IND+1)-OMEGAV(IND))
      ELSE
         OMEGA = OMEGAV(NCETBL)*(RKTE/XKTE(NCETBL))**PEXPVH
      END IF
C
      CEVISC = PCEVIS*SQRT(P1000*XMOLW*TEMP)/(SIG*SIG*OMEGA)
C
      RETURN
C
      ENTRY MPDIF( EPS, SIG, XMOLW1, XMOLW2, PRESS, TEMP, CEDIFF )
C
C**********************************************************************
*
C   ENTRY MPGCED RETURNS VALUE FOR BINARY DIFFUSIVITY
C**********************************************************************
*
C
C   GET INDEX IN TABLE AND INTERPOLATE OMEGA
C
      RKTE = TEMP/EPS
      IND = MIN( P20*RKTE-P5, P10*RKTE+P15, RKTE+P60, PPT1*RKTE+P69 )
      IF ( IND .LT. 1 ) THEN
         OMEGA = OMEGAD(1)*(RKTE/XKTE(1))**PEXPDL
      ELSE IF ( IND .LT. NCETBL ) THEN
         FRAC = MAX(P0,MIN(P1,
     1         (RKTE-XKTE(IND))/(XKTE(IND+1)-XKTE(IND)) ))
         OMEGA = OMEGAD(IND) + FRAC*(OMEGAD(IND+1)-OMEGAD(IND))
      ELSE
         OMEGA = OMEGAD(NCETBL)*(RKTE/XKTE(NCETBL))**PEXPDH
      END IF
C
C * * WILKE & LEE FORMULA - PG 587 OF REID, PROPERTIES OF GASES &
LIQUIDS
C
      SQRMAB = SQRT(2./(PPT001/XMOLW1+PPT001/XMOLW2))
      PCDIF  = (3.03-0.98/SQRMAB)*1.01325E-2
      CEDIFF = PCDIF*TEMP**1.5/(PRESS*SQRMAB*SIG*SIG*OMEGA)
C
C      CEDIFF = PCEDIF*SQRT(TEMP**3*(PPT001/XMOLW1+PPT001/XMOLW2))/
C     1  (PRESS*SIG*SIG*OMEGA)
C
```

```
      RETURN
      END
      SUBROUTINE FLCOK (
C       INPUT
     1  NUMFL, NUMVOL, KACTFL, IFLDNR, VLFLN, ALP, PROPSN, CDCHK,
     1  NUMMAT,  PVOL,   VSAT,  TSPCN, TEMPS,
C       INPUT AND OUTPUT
     2  JVSPEC, IVSPEC, VSPEC, FLOWC, JVSPO)
C
C     FLCOK CALCUATES THE CHOKED FLOW VELOCITIES
C
C***FLCOK MODIFIED: 8-DEC-87, 29-JAN-90
C
      INTRINSIC ABS, EXP, MAX, MIN, SIGN, SQRT
C
C     INPUT
C
C       NUMFL = NUMBER OF FLOW PATHS
C       NUMVOL = NUMBER OF CONTROL VOLUMES
C       KACTFL(NUMFL) = ACTIVE/INACTIVE FLOW PATH SWITCH =0 THEN ACTIVE
C       IFLDNR(2,NUMFL) = POOL/ATMOS DONOR CONTROL VOLUME NUMBER
C       VLFLN(2,NUMFL) = POOL/ATMOS FLOW PATH VELOCITY
C       ALP(NUMFL) = VOID FRACTION
C       PROPSN(NPRP,NUMVOL) = THERMODYNAMIC PROPERTIES
C       CDCHK(2,NUMFL) = FORWARD AND REVERSE DISCHARGE COEFFICIENTS
C       NUMMAT = NUMBER OF MATERIALS
C       PVOL(NUMVOL) = PRESSURES
C       VSAT(KS1DIM,KS2DIM,NUMVOL) = SATURATION DATA
C       TSPCN(KSP1DM,NUMMAT,NUMVOL) = SPECIFIC THERMO DATA
C
C     OUTPUT
C       JVSPEC = GLOBAL SPECIFIED VELOCITY FLAG.  EQUAL TO THE
C                NUMBER OF FLOWPATHS WITH SPECIFIED VELOCITIES
C       IVSPEC(NUMFL) = POINTER TO THE SPECIFIED VELOCITY FLOWPATH
C                       NUMBER.
C                       IVSPEC(N), N=1,JVSPEC IS DEFINED AND
C                       1 .LE. IVSPEC(N) .LE. NUMFL
C       VSPEC(NUMFL) = SPECIFIED VELOCITY.  DEFINED ONLY FOR INDICES
C                       1 THROUGH JVSPEC
C
*- INCLUDE SC4402
C     4402 CONTAINS THE THRESHOLD VELOCITY FOR CRITICAL FLOW TESTING
      COMMON /F4402 / I4402  , C4402(1) , NA4402
      COMMON /CP4402/ JP4402 , IS4402(1) , IE4402(1) , MA4402
      COMMON /CF4402/ CI4402
      CHARACTER CI4402*19
      EQUIVALENCE (C4402(1),CVELMN)
C
*-
*- INCLUDE CVTPRP
C     THIS PARAMETER STATEMENT CONTAINS POINTERS TO THERMODYNAMIC
C     PROPERTIES STORED IN ARRAYS PROPSO(NPRP,NNVOL) AND
C     PROPSN(NPRP,NNVOL).
C
C      KPVOD POINTS TO THE POOL VOID FRACTION
C      KCSLPL POINTS TO THE LIQUIDS SPEED OF SOUND IN THE POOL
C      KCSVPL POINTS TO THE VAPORS SPEED OF SOUND IN THE POOL
```

```
C      KCSATM POINTS TO THE SPEED OF SOUND IN THE ATMOSPHERE
C      KRHOLQ POINTS TO THE LIQUID DENSITY IN THE POOL
C      KRHOVP POINTS TO THE VAPOR DENSITY IN THE POOL
C      KRHOAT POINTS TO THE ATMOSPHERE DENSITY (EXCLUDING FOG)
C      NPRP IS THE FIRST DIMENSION OF THE ARRAY
      PARAMETER ( KPVOD = 1 , KCSLPL = 2 , KCSVPL = 3 ,
     1  KCSATM = 4 , KRHOLQ = 5 , KRHOVP = 6 , KRHOAT = 7 , NPRP = 7 )
*-
*- INCLUDE SATDAT
C     THIS PARAMETER STATMENT CONTAINS THE POINTERS TO THE SATURATION
C       DATA STORED IN ARRAYS VSATO(KS1DIM,KS2DIM,NNVOL)
C       AND VSATN(KS1DIM,KS2DIM,NNVOL)
C
C     FIRST DIMENSION: DEPENDENT VARIABLE
C       KSRHOL = LIQUID WATER DENSITY
C       KSRHOV = STEAM DENSITY
C       KSHL   = LIQUID WATER ENTHALPY
C       KSHV   = STEAM ENTHALPY
C       KSEL   = LIQUID WATER INTERNAL ENERGY
C       KSEV   = STEAM INTERNAL ENERGY
C       KSCPL  = CONSTANT PRESSURE SPECIFIC HEAT OF LIQUID WATER
C       KSCPV  = CONSTANT PRESSURE SPECIFIC HEAT OF STEAM
C       KSPS   = SATURATION PRESSURE
C       KSTS   = SATURATION TEMPERATURE
C       KSHLV  = HEAT OF VAPORIZATION
C       KS1DIM = FIRST DIMENSION OF SATURATION DATA ARRAY
C     SECOND DIMENSION: INDEPENDENT VARIABLE
C       KSCVP  = CONTROL VOLUME PRESSURE
C       KSSPP  = STEAM PARTIAL PRESSURE
C       KSPLT  = POOL TEMPERATURE
C       KSATT  = ATMOSPHERE TEMPERATURE
C       KS2DIM = SECOND DIMENSION OF SATURATION DATA ARRAY
C
      PARAMETER(
     1  KSRHOL =  1, KSRHOV =  2, KSHL   =  3, KSHV   =  4,
     2  KSEL   =  5, KSEV   =  6, KSCPL  =  7, KSCPV  =  8,
     3  KSPS   =  9, KSTS   = 10, KSHLV  = 11, KS1DIM = 11,
     4  KSCVP  =  1, KSSPP  =  2, KSPLT  =  3, KSATT  =  4,
     5  KS2DIM =  4 )
*-
*- INCLUDE CVHSP
C      THIS PARAMETER STATMENT CONTAINS POINTERS TO THE SPECIFIC
C      THERMODYNAMIC QUANTITIES IN ARRAYS TSPCN AND TSPCO.
C      DIMENSION:  TSPCN(KSP1DM,NUMMAT,NUMVOL)
C
C      FIRST INDEX
C        KSPRHO = MASS DENSITY (KG/M**3)
C        KSPH   = SPECIFIC ENTHALPY (J/KG/K)
C        KSPE   = SPECIFIC INTERNAL ENERGY (J/KG/K)
C        KSP1DM = FIRST DIMENSION
C      SECOND INDEX
C        MATERIAL NUMBER
C      THIRD INDEX
C        VOLUME NUMBER
       PARAMETER ( KSPRHO = 1 , KSPH = 2 , KSPE = 3 , KSP1DM = 3 )
*-
      PARAMETER ( PSI2PA = 6.894757D3 , PLB2KG = .4535924D0 ,
```

```
     1             PBTU2J = 1.055870D3 , P15    = 15.D0      ,
     1             P3000  = 3000.D0    , P300   = 300.D0     ,
     2             P10    = 10.D0      , P200   = 200.D0     ,
     2             P170   = 170.D0     , PTWO   = 2.D0       ,
     3             PFT2M  = .3048D0    , P1000  = 1000.D0    ,
     4             PZERO  = 0.D0       , PONE   = 1.D0       ,
     5             PKG2LB = PONE/PLB2KG, PJ2BTU = PONE/PBTU2J,
     6             PA2PSI = PONE/PSI2PA, PM2FT  = PONE/PFT2M )
      DIMENSION KACTFL(NUMFL), IFLDNR(2,NUMFL), VLFLN(2,NUMFL),
     1 ALP(NUMFL), PROPSN(NPRP,NUMVOL), CDCHK(2,NUMFL),
     2 TSPCN(KSP1DM,NUMMAT,NUMVOL), PVOL(NUMVOL),TEMPS(2,NUMVOL),
     3 VSAT(KS1DIM, KS2DIM, NUMVOL)
      DIMENSION IVSPEC(NUMFL), VSPEC(NUMFL)
C
      DIMENSION FLOWC(*)
C
C/////////CANDIDATE FOR SENSITIVITY COEFFICIENT//////
C   FACTOR BY WHICH CRITICAL FLOW MUST BE EXCEEDED TO INVOKE CHOKING
      PARAMETER (CFFACT=1.03D0)
C/////////CANDIDATE FOR SENSITIVITY COEFFICIENT//////
C
C   **********************************************
C   * DATA FOR CRITICAL FLOW FITS FROM RETRAN-02  *
C   * NP-1850-CCM, VOLUME 1, SECTION 1V.3         *
C   * ALL FITS USE PSIA, BTU/LBM, LBM/FT2.S       *
C   **********************************************
C
      DIMENSION H1(6,6), H2(6,6), XM1(6,6), XM2(6,6), A(9)
C
C   //////////////////////////////////////////////////////////////////
C   / EXTENDED HENRY-FAUSKE.  SUB-COOLED LIQUID, H .GE. 170 BTU/LBM /
C   //////////////////////////////////////////////////////////////////
C
C   REGION 1:  P .LE. 300 PSIA
C   GCRIT = SUM(J=0,5) SUM(I=0,5) H1(I+1,J+1) * P**J * H**I
      DATA ((H1(I, J), I=1, 6), J=1, 6) /
     1      0.11971131D+05, -0.29275019D+03,  0.13088631D+01,
     2      0.96555931D-02, -0.85871644D-04,  0.16193695D-06,
     3     -0.25664444D+03,  0.40495998D+02, -0.60219912D+00,
     4      0.34802345D-02, -0.85504973D-05,  0.71823710D-08,
     5     -0.11154016D+03,  0.17440962D+01, -0.93144283D-02,
     6      0.15567712D-04,  0.13551325D-07, -0.41240653D-10,
     7      0.14181940D+01, -0.25888201D-01,  0.17982760D-03,
     8     -0.57489343D-06,  0.81339877D-09, -0.38055461D-12,
     9     -0.57956498D-02,  0.11065153D-03, -0.81656598D-06,
     A      0.28585300D-08, -0.46830222D-11,  0.28432048D-14,
     B      0.77436012D-05, -0.15104216D-06,  0.11454533D-08,
     C     -0.41616729D-11,  0.71809283D-14, -0.46890691D-17  /
C
C   REGION 2:  300 PSIA .LT. P .LE. 3000 PSIA
C   GCRIT = SUM(J=0,5) SUM(I=0,5) H1(I+1,J+1) * P**J * H**I
      DATA ((H2(I, J), I=1, 6), J=1, 6) /
     1      0.11996419D+05, -0.66516773D+02,  0.25523516D+00,
     2     -0.44088343D-03, -0.53252751D-06,  0.76399011D-09,
     3      0.33614071D+02, -0.21670146D+00,  0.18031256D-02,
     4     -0.57669789D-05,  0.98378584D-08, -0.64276860D-11,
     5     -0.34555139D-01,  0.48840252D-03, -0.35661004D-05,
```

```
     6      0.10431626D-07, -0.15532591D-10,  0.93866385D-14,
     7      0.27341308D-04, -0.43249560D-06,  0.28854007D-08,
     8     -0.78849169D-11,  0.10650136D-13, -0.59412445D-17,
     9     -0.10430915D-07,  0.16802644D-09, -0.10625928D-11,
     A      0.27918999D-14, -0.35327078D-17,  0.18329776D-20,
     B      0.13430502D-11, -0.21923832D-13,  0.13604749D-15,
     C     -0.35206568D-18,  0.43073516D-21, -0.21322858D-24  /
C
C   ///////////////////////////
C   / MOODY.  SATURATED WATER /
C   ///////////////////////////
C
C   REGION 1:  P .LE. 200 PSIA
C   GCRIT = EXP ( SUM(J=0,5) SUM(I=0,5) XM1(I+1,J+1) * P**J * H**I )
      DATA ((XM1(I, J), I=1, 6), J=1, 6) /
     1      0.75398883D+01, -0.27349762D-01,  0.77033614D-04,
     2     -0.11597165D-06,  0.85314613D-10, -0.24171538D-13,
     3      0.48635447D+00, -0.25172525D-02,  0.66198222D-05,
     4     -0.84665377D-08,  0.52492322D-11, -0.12636194D-14,
     5     -0.14054847D-01,  0.90215743D-04, -0.25893147D-06,
     6      0.35752133D-09, -0.23790918D-12,  0.61256651D-16,
     7      0.18252651D-03, -0.12541056D-05,  0.36860723D-08,
     8     -0.51915992D-11,  0.35151649D-14, -0.91925726D-18,
     9     -0.10492510D-05,  0.74318133D-08, -0.22032243D-10,
     A      0.31249946D-13, -0.21287482D-16,  0.55968113D-20,
     B      0.21537617D-08, -0.15500316D-10,  0.46118179D-13,
     C     -0.65601247D-16,  0.44796907D-19, -0.11802990D-22  /
C
C   REGION 2:  200 PSIA .LT. P .LE. 3000 PSIA
C   GCRIT = SUM(J=0,5) SUM(I=0,5) XM2(I+1,J+1) * P**J * H**I
      DATA ((XM2(I, J), I=1, 6), J=1, 6) /
     1      0.64582892D+04, -0.59379818D+02,  0.19040194D+00,
     2     -0.27991119D-03,  0.19327093D-06, -0.50859277D-10,
     3      0.15724915D+03, -0.76566784D+00,  0.14863253D-02,
     4     -0.13833255D-05,  0.60309542D-09, -0.95031339D-13,
     5     -0.16101131D+00,  0.11162190D-02, -0.27575375D-05,
     6      0.31653794D-08, -0.17199274D-11,  0.35699646D-15,
     7     -0.14741137D-04, -0.18726780D-06,  0.85169692D-09,
     8     -0.12626170D-11,  0.79444415D-15, -0.18158912D-18,
     9      0.61947560D-07, -0.22806979D-09,  0.30284416D-12,
     A     -0.16803993D-15,  0.29596473D-19,  0.17431814D-23,
     B     -0.19206668D-10,  0.88455810D-13, -0.16287505D-15,
     C      0.15109240D-18, -0.71294533D-22,  0.13825335D-25  /
C
C   ///////////////////////
C   / TRANSITION ENTHALPY /
C   ///////////////////////
C
C   HTRN = SUM(J=0,8) A(J+1) * P**J
      DATA (A(J), J=1, 9) /
     1      0.17045231D+03,  0.14778460D+01, -0.46053540D-02,
     2      0.89234086D-05, -0.10088781D-07,  0.67625795D-11,
     3     -0.26445490D-14,  0.55672345D-18, -0.48685744D-22  /
 C
      COMMON / EOSPARM /  TCRIT, RCRIT, PCRIT, TSATO, TSINV, CVLFUS
     .,             TRIPT, TRIPP, TRIPR, TMIN,  PMIN,  RMIN
     .,             TMAX,   PMAX, RMAX,  NPTS,  INIT,  NFLUID
```

```
C
      REAL*8 TCRIT, RCRIT, PCRIT, TSATO, TSINV, CVLFUS
     .,      TRIPT, TRIPP, TRIPR, TMIN,  PMIN,  RMIN
     .,      TMAX,   PMAX,  RMAX
C
C
      DO 100 NFL = 1 , NUMFL
CC        DO 110 MFL = 1 , JVSPEC
CC          IF ( IVSPEC(MFL) .EQ. NFL ) GO TO 100
CC 110    CONTINUE
C       THIS FLOW PATH HAS NOT BEEN PREVIOUSLY DEFINED
      IF ( ABS(VLFLN(2,NFL)) .GT. 0.*CVELMN .OR.
     1       ABS(VLFLN(1,NFL)) .GT. 0.*CVELMN ) THEN
C         MAY BE CHOKING
C
C///////////////////////////////////////////////
C/// THIS CLEARLY NEEDS SOME THOUGH ABOUT    ///
C/// THE POSSIBILITY OF COUNTER-CURRENT FLOW ///
C///////////////////////////////////////////////
C
        IF ( ABS(VLFLN(2,NFL)) .GT. PZERO ) THEN
          IDONOR = IFLDNR(2,NFL)
        ELSE
          IDONOR = IFLDNR(1,NFL)
        ENDIF
        IF ( ALP(NFL) .GE. PONE ) THEN
C           ATMOSPHERE ONLY - USE SOUND SPEED WITH DISCHARGE
COEFFICIENT
          CS = PROPSN(KCSATM,IDONOR)
C          ACCOUNT FOR EXPANSION AND COOLING TO THROAT
C          WEAK FCN OF GAMMA (+- 5 PERCENT FOR 1.1 .LE. GAMMA .LE. 1.8)
          GAMMA = 1.4
          EXPON = (GAMMA+PONE)/(PTWO*(GAMMA-PONE))
          CS = CS * (PTWO/(GAMMA+PONE))**EXPON
          IF ( VLFLN(2,NFL) .GT. PZERO ) THEN
            CS = CS*CDCHK(1,NFL)
          ELSE
            CS = CS*CDCHK(2,NFL)
          ENDIF
          FLOWC(NFL) = CS*TSPCN(KSPRHO,3,IDONOR)
          IF ( ABS(VLFLN(2,NFL)) .GT. CFFACT*CS ) THEN
            JVSPEC = JVSPEC + 1
            IVSPEC(JVSPEC) = NFL
            VSPEC (JVSPEC) = SIGN(CS,VLFLN(2,NFL))
          ENDIF
        ELSE
C
C           USE RETRAN (HENRY-FAUSKE/MOODY) MODEL FOR POOL
C           THIS MODEL IS IN TERMS OF BRITISH UNITS.  CONVERT TO
C           BRITISH UNITS, THEN BACK TO SI
C
          RHOPOL = TSPCN(KSPRHO,1,IDONOR)
          IF (RHOPOL .LE. PZERO) RHOPOL = P1000
C
          IF(NFLUID.EQ.1) THEN
          PLIQ = PVOL(IDONOR)
C////////////////////////////////
```

```
C/// WHAT IF NO POOL IN DONOR ///
C////////////////////////////////
          HSPEC = TSPCN(KSPH,1,IDONOR)
C
C           CONVERT FROM SI TO FT, BTU, LB
C
          PSI = PLIQ*PA2PSI
          PPSI = MAX( P15 , MIN ( P3000 , PSI ) )
C
          BTULB = PLB2KG*PJ2BTU*HSPEC
          HBTULB = MAX ( P170 , BTULB )
C
C          LIMIT H TO SATURATED VAPOR TO PREVENT WILD EXTRAPOLATION
          HBTULB = MIN ( HBTULB,
     1                    PLB2KG*PJ2BTU*VSAT(KSHV,KSCVP,IDONOR) )
C
          HTRN = A(8) + A(9)*PPSI
          DO 1050,J=7,1,-1
            HTRN = A(J) + HTRN*PPSI
1050        CONTINUE
C
          HSBTU = PLB2KG*PJ2BTU*VSAT(KSHL,KSCVP,IDONOR)
          HTRN = MIN ( HTRN , HSBTU-P10 )
C
          IF ( HBTULB .LE. HSBTU ) THEN
C            ------------------------------
C            EVALUATE EXTENDED HENRY FAUSKE
C            ------------------------------
            XP = PONE
            FLOWHF = PZERO
            IF ( PPSI .LE. P300 ) THEN
            DO 1150,J=1,6
              X = H1(5,J) + H1(6,J)*HBTULB
              DO 1160,I=4,1,-1
                X = H1(I,J) + HBTULB*X
1160              CONTINUE
              FLOWHF = FLOWHF + XP*X
              XP = XP*PPSI
1150            CONTINUE
C
            ELSE
            DO 2150,J=1,6
              X = H2(5,J) + H2(6,J)*HBTULB
              DO 2160,I=4,1,-1
                X = H2(I,J) + HBTULB*X
2160              CONTINUE
              FLOWHF = FLOWHF + XP*X
              XP = XP*PPSI
2150            CONTINUE
            END IF
C
          END IF
C
          IF ( HBTULB .LE. HTRN ) THEN
C            -----------------------------------
C            ACCEPT EXTENDED HENRY FAUSKE RESULT
C            -----------------------------------
```

```
            FLOWPL = FLOWHF
C
          ELSE
C            --------------
C            EVALUATE MOODY
C            --------------
            XP = PONE
            FLOWM = PZERO
            IF ( PPSI .LE. P200 ) THEN
            DO 3150,J=1,6
              X = XM1(5,J) + XM1(6,J)*HBTULB
              DO 3160,I=4,1,-1
                X = XM1(I,J) + HBTULB*X
3160              CONTINUE
              FLOWM = FLOWM + XP*X
              XP = XP*PPSI
3150            CONTINUE
            FLOWM = EXP(FLOWM)
C
            ELSE
            DO 4150,J=1,6
              X = XM2(5,J) + XM2(6,J)*HBTULB
              DO 4160,I=4,1,-1
                X = XM2(I,J) + HBTULB*X
4160              CONTINUE
              FLOWM = FLOWM + XP*X
              XP = XP*PPSI
4150            CONTINUE
            END IF
C
            IF ( HBTULB .GE. HSBTU ) THEN
C              -------------------
C              ACCEPT MOODY RESULT
C              -------------------
            FLOWPL = FLOWM
C
            ELSE
C              --------------------------
C              CONSTRUCT WEIGHTED AVERAGE
C              --------------------------
            FLOWPL = FLOWHF +
     1          (HBTULB-HTRN)*(FLOWM-FLOWHF)/(HSBTU-HTRN)
            END IF
C
          END IF
C
          IF ( BTULB .LT. P170 ) THEN
C            INTERPOLATE TO ORIFICE EQUATION AT H = 0
            FLOWO = PKG2LB*PFT2M*PFT2M*SQRT( PTWO*RHOPOL*PLIQ )
            FLOWPL = FLOWO + BTULB*(FLOWPL-FLOWO)/P170
          END IF
C
          IF ( PSI .GT. P3000 ) FLOWPL=FLOWPL*SQRT (PSI/P3000)
C
C           NOW CONVERT BACK TO SI UNITS
C
          FLOWC(NFL) = FLOWPL*PLB2KG*PM2FT*PM2FT
```

```
                          ELSE
C
C * * CALL HENRY-FAUSKE ROUTINE
C
          PLOSS = 0.
          TPOL  = MIN(TEMPS(1,IDONOR),VSAT(KSTS,KSCVP,IDONOR)-.001)
          CALL GCSUB(PVOL(IDONOR),TPOL,PLOSS,FLOWC(NFL))
C         FLOWC(NFL) = RHOPOL*PROPSN(KCSLPL,IDONOR)
                          ENDIF
C
C           CONVERT MASS FLOW RATE PER UNIT AREA TO VELOCITY PER UNIT
C           AREA
C
          IF ( ALP(NFL) .LE. PZERO ) THEN
C             POOL ONLY - USE RETRAN MODEL WITH DISCHARGE COEFFICIENT
            CS = FLOWC(NFL)/RHOPOL
            IF ( VLFLN(1,NFL) .GT. PZERO ) THEN
            CS = CS*CDCHK(1,NFL)
            ELSE
            CS = CS*CDCHK(2,NFL)
            ENDIF
            FLOWC(NFL) = CS*RHOPOL
            IF ( ABS(VLFLN(1,NFL)) .GT. CFFACT*CS ) THEN
            JVSPEC = JVSPEC + 1
            IVSPEC(JVSPEC) = NFL
            VSPEC (JVSPEC) = SIGN(CS,VLFLN(1,NFL))
            ENDIF
          ELSE
C             MIXED ATMOS AND POOL FLOW
            CSVAP  = PROPSN(KCSATM,IDONOR)
            IF (CSVAP .LE. PZERO) CSVAP = P200
C            ACCOUNT FOR EXPANSION AND COOLING TO THROAT
C            WEAK FCN OF GAMMA (+- 5 PERCENT FOR 1.1 .LE. GAMMA .LE.
1.8
            GAMMA = 1.4
            EXPON = (GAMMA+PONE)/(PTWO*(GAMMA-PONE))
            CSVAP = CSVAP * (PTWO/(GAMMA+PONE))**EXPON
            RHOVAP = PROPSN(KRHOAT,IDONOR)
            IF (RHOVAP .LE. PZERO) RHOVAP = P10
C
            CS = PONE/
     1        ( ALP(NFL)/CSVAP + (PONE-ALP(NFL))*RHOPOL/FLOWC(NFL) )
C
            FLOW = ALP(NFL)*VLFLN(2,NFL)*RHOVAP +
     1        (PONE-ALP(NFL))*VLFLN(1,NFL)*RHOPOL
C
            IF ( FLOW .GT. PZERO ) THEN
            CS = CS*CDCHK(1,NFL)
            ELSE
            CS = CS*CDCHK(2,NFL)
            ENDIF
            RHOMIX = ALP(NFL)*RHOVAP + (PONE-ALP(NFL))*RHOPOL
            FLOWC(NFL) = RHOMIX*CS
C
      DO 110 MFL = 1 , JVSPEC
      MFLS = MFL
      IF ( IVSPEC(MFL) .EQ. NFL ) GO TO 120
```

```
  110 CONTINUE
C
            IF ( ABS(FLOW) .GT. CFFACT*FLOWC(NFL) ) THEN
C               CHOKED FLOW
            JVSPEC = JVSPEC + 1
            IVSPEC(JVSPEC) = NFL
            VSPEC(JVSPEC) = SIGN(CS,FLOW)
            ENDIF
            GO TO 100
 120          DVSPEC = VSPEC(MFLS)-CS
            VSPEC(MFLS) = SIGN(CS,FLOW)
            IF(CS.NE.0.) DVSPEC = DVSPEC/CS
            IF(ABS(DVSPEC).GT.1.E-3) JVSPO = 0
          ENDIF
        ENDIF
      ENDIF
 100  CONTINUE
      RETURN
      END
      SUBROUTINE INITFLD
C
      COMMON / EOSPARM /  TCRIT, RCRIT, PCRIT, TSATO, TSINV, CVLFUS
     .,            TRIPT, TRIPP, TRIPR, TMIN,  PMIN,  RMIN
     .,            TMAX,   PMAX, RMAX,  NPTS,  INIT,  NFLUID
C
      REAL*8 TCRIT, RCRIT, PCRIT, TSATO, TSINV, CVLFUS
     .,      TRIPT, TRIPP, TRIPR, TMIN,  PMIN,  RMIN
     .,      TMAX,   PMAX,  RMAX
C
      COMMON /SUMPRPS/ SIGLJ(9), EPSLJ(9), FLDMW(9), HFUSN(9), XKSLD(9)
C
      COMMON /H2OCB6/  EL(376),EV(376)
     1,DPTL(376),DPTV(376),SL(376),SV(376)
     2,CVL(376),CVV(376),DPRL(376),DPRV(376)
      COMMON /H2OCB5/ RL(376),RV(376),PV(376)
C
      COMMON /CONST/  BIGNMP, BIGNMM, SMLNMP, SMLNMM, RNDNUM, EXPMIN
     1      , EXPMAX, XLNMX , XLOGMX, GRAVTY, ONE   , PI    , RGAS
     2      , TCRITX, ZERO  , ZERODC, ONETRD, TWOTRD, SBCON , TWO
     3      , HALF
C
C     2002 CONTAINS COEFFICIENTS FOR THE INITIAL GUESS OF THE
C     TEMPERATURE
      COMMON /F2002 / I2002  , C2002(3) , NA2002
C
      PARAMETER (NPROPC = 5, NDMATL = 35)
      COMMON /MPPRO1/ XPRUND, XMPCON(NPROPC, NDMATL)
*- INCLUDE NCGEOS
C
C     NON-CONDENSIBLE GAS EOS COMMON BLOCKS
C
      PARAMETER (IEOS = 27)
C
      COMMON /NCGNAM/ NAMGAS
C
      CHARACTER*8 NAMGAS(IEOS)
C
```

```
      COMMON /NCGEOS/ MAXGAS, NMMAT, RNCG(IEOS), WM(IEOS),
EMAXIM(IEOS),
     2 KH2OP  , KH2OLA , KH2OVA , KH2    , KO2    , KCO2   , KCO    ,
     3 KN2    , KNO    , KN2O   , KNH3   , KC2H2  , KCH4   , KC2H4  ,
     4 KHE    , KAR    , KD2    ,
     5 KGASA  , KGASB  , KGASC  , KGASD  , KGASE  , KGASF  , KGASG  ,
     6 KGASH  , KGASI  , KGASJ  ,
C     SCRATCH STORAGE FOR EOS WORK
     7 XNCGSA(IEOS), XNCGSB(IEOS), XNCGSC(IEOS), XNCGSD(IEOS),
     8 XNCGSE(IEOS), XNCGSF(IEOS), XNCGSG(IEOS), XNCGSH(IEOS)
C
C      DIMENSION TSAT(376)
C
      LOGICAL ERR
      COMMON /TABLES/ TBLS(45000),S(26)
      REAL*8 TBLS, S
      COMMON /MATCOM/ MATNAM(9),FILNAM(9),LFN(9)
      CHARACTER*8 MATNAM,FILNAM
      CHARACTER*80 RECORD(2)
      LOGICAL THERE
C
C  initialization call to wagner's relap5 property routines
      NUSE   = 45000
      NUNIT  = 15
C
C * * * DETERMINE FLUID
C
      DO 10 NF = 1,9
      INQUIRE (FILE=FILNAM(NF),EXIST=THERE)
      IF(THERE) THEN
      NFLUID = NF
      PRINT *,"USING FLUID NAMED = ",MATNAM(NF)
      GO TO 20
                ENDIF
   10 CONTINUE
      PRINT *,"FLUID PROPERTY TABLES NOT FOUND"
      STOP
C
   20 CALL ICMPNX(NUNIT,NUSE,FILNAM(NFLUID),TBLS)
      IF(NUSE.LT.0) THEN
      PRINT *,"CAN'T LOAD PROPERTY TABLES"
      STOP
                    ENDIF
      NPTS     = 376
      ZERODC   = TRIPT
      TSATO    = TRIPT
      TCRITX   = TCRIT
      C2002(1) = RCRIT
C
C * * * SET PSAT CRIT
C
      RHOC = RCRIT
      TC   = TCRIT
      PC   = PCRIT
      CALL FLUIDEOS(NFLUID, RHOC, TC, PV(NPTS), EV(NPTS), SV(NPTS)
     .,                   DPTV(NPTS), DPRV(NPTS), CVV(NPTS), ERR)
      EL(NPTS)   = EV(NPTS)
```

```
      SL(NPTS)   = SV(NPTS)
      DPTL(NPTS) = DPTV(NPTS)
      DPRL(NPTS) = DPRV(NPTS)
      CVL(NPTS)  = CVV(NPTS)
      RV(NPTS)   = RHOC
      RL(NPTS)   = RHOC
C
      TSINV = (TCRIT-TSATO)/FLOAT(NPTS-1)
C
C * * * FIND RHOSAT LIQUID
C
      DO 100 N = 1, NPTS-1
      TSAT = FLOAT(N-1)*TSINV+TSATO
C
      CALL FLUIDEOS(NFLUID, RHOC, TSAT, PV(N) , EDUM, SDUM
     .,                  DDUM1, DDUM2, CDUM, ERR)
      RL(N)  = 1./S(11)
      RV(N)  = 1./S(12)
      EL(N)  = S(13)
      EV(N)  = S(14)
      SL(N)  = S(25)
      SV(N)  = S(26)
C
      VOLAL  = S(11)
      VOLAG  = S(12)
C
      HLS    = S(15)
      HGS    = S(16)
C
      BETAL  = S(17)
      BETAG  = S(18)
C
      CAPAL  = S(19)
      DPRL(N)= VOLAL/CAPAL
      CAPAG  = S(20)
      DPRV(N)= VOLAG/CAPAG
C
      CPLS   = S(21)
      CPGS   = S(22)
C
      CVV(N) = CPGS - TSAT * BETAG**2 * VOLAG /CAPAG
      CVL(N) = CPLS - TSAT * BETAL**2 * VOLAL /CAPAL
C
      DPTV(N)= (CPGS-CVV(N))/(VOLAG*TSAT*BETAG)
      DPTL(N)= (CPLS-CVL(N))/(VOLAL*TSAT*BETAL)
C
  100 CONTINUE
C
      IF(SIGLJ(NFLUID).NE.0.0) THEN
      XMPCON(4,2) = SIGLJ(NFLUID)
      XMPCON(5,2) = EPSLJ(NFLUID)
C                             ELSE
C VALUES OF EPS & SIG FROM EMPIRICAL RELATIONSHIPS, PG 456 OF WELTY,
WICKS
C   & WILSON
      XMPCON(4,2) = 2.44*(TCRIT*1.01325E5/PCRIT)**(1./3.)
      XMPCON(5,2) = 0.77*TCRIT
```

```
                               ENDIF
      DO 200 N=1,3
  200 WM(N)       = FLDMW(NFLUID)*1.D-3
      CVLFUS      = HFUSN(NFLUID)/1.D0
      NAMGAS(KH2OVA) = MATNAM(NFLUID)
      RMAX        = 1.25*RL(1)
      RMIN        = 1.E-20
      TRIPR       = RL(1)
C
      RETURN
      END
c
      BLOCK DATA NAMES
      COMMON /MATCOM/ MATNAM(9),FILNAM(9),LFN(9)
      CHARACTER*8 MATNAM,FILNAM
      DATA MATNAM /'H2O','H2','Li','K','He','N2','Na','Nak'
     .,      'LiPb'/
      DATA FILNAM /'tpfh2o','tpfh2','tpfli','tpfk','tpfhe'
     .,     'tpfn2','tpfna','tpfnak','tpflipb'/
      DATA LFN /3, 2, 2, 1, 2, 2, 2, 3, 4 /
C
      COMMON /SUMPRPS/ SIGLJ(9), EPSLJ(9), FLDMW(9), HFUSN(9), XKSLD(9)
C
C VALUES OF EPS & SIG FROM EMPIRICAL RELATIONSHIPS, PG 456 OF WELTY,
WICKS
C   & WILSON SIG = 2.44*(Tc/Pc)**1/3, EPS = 0.77*Tc
C     FOR Li, Na,  & K
C
      DATA SIGLJ / 2.649, 2.968, 0.000, 0.000, 2.576, 3.681, 0.000
     .,            0.000, 0.000 /
      DATA EPSLJ / 356.0,  33.3, 0.000, 0.000, 10.22,  91.5, 0.000
     .,            0.000, 0.000 /
      DATA FLDMW / 18.02, 2.016, 6.939, 39.10, 4.003, 28.01, 22.99
     .,            62.09, 214.1 /
C * * * DATA FROM CRC HANDBOOK CHEM&PHYS
C      HELIUM EST-DEL ENTROPY=UNIVERAL GAS CONT & DEL H=DELS*T
C             (HIRSHFELDER,CURTISS,BIRD, "MOLECULAR THEORY ...",WILEY,
1954)
C      LiPb - M.C. BILLONE, ANL INTERNAL REPORT, 1992.
      DATA HFUSN / 3.338E5, 5.778E4, 6.636E5, 6.113E4, 4.16E3, 2.575E4
     .,            1.147E5,  0.000, 3.39E4 /
C * * * DATA FROM CRC HANDBOOK CHEM&PHYS & TOULOUKIAN
C      LiPb - M.C. BILLONE, ANL INTERNAL REPORT, 1992.
      DATA XKSLD /    2.2,    0.9, 71.2, 98.5, 0.18, 0.12, 120.0
     .,              48.0,  31.52 /
C
      END
      subroutine fluideos(nmat,rho,temp,pt,utot,stot,dpdt,dpdr,cv,err)
c
c *** dimension for property output
c
      logical err
      common/tables/ tbls(45000),s(26)
      real*8 tbls, s
c
c *** go get properties
c
```

```
      s(1)   = temp
      s(3)   = 1./rho
      if(nmat.eq.1) then
      call sth2x4 (tbls,s,ip,err)
                else
      call stupx4 (tbls,s,ip,err)
                endif
      if(err) then
      write(6,106) s
      return
      endif
c
c *** properties returned
c
      pt     = s(2)
      utot   = s(4)
      qual   = s(9)
c     print*,pt,utot,qual
c
c ***   check for equilibrium two-phase
c
      if ( ip .eq. 2 ) go to 63
c
      vola   = s(3)
      beta   = s(6)
      cappa  = s(7)
      cp     = s(8)
      psat   = s(10)
      cv     = cp - temp * (beta**2) * vola / cappa
      dpdt   = rho*(cp-cv)/(temp*beta)
      dpdr   = 1./(rho*cappa)
      tsat   = s(1)
      stot   = s(24)
c     print*,vola,beta,cappa,cp,psat,cv,dpdt,dpdr,tsat,stot
      return
c
   63 continue
c
      tsat   = s(1)
      psat   = s(2)
c
      volal  = s(11)
      volag  = s(12)
c
      hls    = s(15)
      hgs    = s(16)
c
      betal  = s(17)
      betag  = s(18)
c
      capal  = s(19)
      dpdrl  = volal/capal
      capag  = s(20)
      dpdrg  = volag/capag
c
      cpls   = s(21)
      cpgs   = s(22)
```

```
      cp     = qual * cpgs + (1.00 - qual) * cpls
c
      pprim  = (hgs - hls )/(volag - volal )
      pprim  = pprim /tsat
c
      cvg    = cpgs - tsat * pprim * volag * (2.00*betag-capag*pprim )
      cvl    = cpls - tsat * pprim * volal * (2.00*betal-capal*pprim )
      cv     = qual * cvg + (1.00 - qual) * cvl
c
      dpdtg  = (cpgs-cvg)/(volag*temp*betag)
      dpdtl  = (cpls-cvl)/(volal*temp*betal)
c
      dpdt   = qual*dpdtg + (1.-qual)*dpdtl
      dpdr   = qual*dpdrg + (1.-qual)*dpdrl
C
      stot   = qual*s(26) + (1.-qual)*s(25)
c
c     print*,dpdt,dpdr,dpdtg,dpdtl,dpdrg,dpdrl
c     print *,"fluideos in twophase region"
  102 format(10x,'sth2x2 bound error in compu'/3(/10x,10(1x,1pe11.4)))
  106 format(10x,'sth2x6 bound error in compu'/3(/10x,10(1x,1pe11.4)))
      return
      end
*deck icmpn1
       subroutine icmpnx(nunit,nuse,filnamx,tbls)
c
       dimension tbls(*)
       real*8 tbls
       character*8 filnamx
       integer nunit
C
      COMMON / EOSPARM /  TCRITX, RCRITX, PCRITX, TSATO, TSINV, CVLFUS
     .,            TRIPT, TRIPP, TRIPR, TMINX, PMINX, RMINX
     .,            TMAXX, PMAXX, RMAXX,  NPTS,  INIT,  NFLUID
C
      REAL*8 TCRITX, RCRITX, PCRITX, TSATO, TSINV, CVLFUS
     .,      TRIPT, TRIPP, TRIPR, TMINX,  PMINX,  RMINX
     .,      TMAXX,   PMAXX,  RMAXX
      INTEGER NFLUID, NPTS, INIT
C
c
c  $Id: icmpn1.F,v 1.10 1998/02/26 18:23:57 rqn Exp $
c
c  Controls cross checking of component input and first pass of
c  component initialization.
c
c  Cognizant engineer: rjw.
c
*if def,impnon,1
       implicit none
      include 'machas.h'
      include 'machos.h'
      include 'machss.h'
      include 'stcom.h'
c
       common /sth2xc/ nth2x,nph2x,nsh2x,ns2h2x,klph2x,klp2h2,llph2x,
     & nt5h2x,jplh2x
```

```
       integer nth2x,nph2x,nsh2x,ns2h2x,klph2x,klp2h2,llph2x,nt5h2x,
     & jplh2x
       save /sth2xc/
c
       common /std2xc/ ntd2x,npd2x,nsd2x,ns2d2x,klpd2x,klp2d2,llpd2x,
     & nt5d2x,jpld2x
       integer ntd2x,npd2x,nsd2x,ns2d2x,klpd2x,klp2d2,llpd2x,nt5d2x,
     & jpld2x
       save /std2xc/
c
c  Local variables
       real*8 stceqv(lstcom)
       equivalence (stceqv(1),ttrip)
       logical hsetst,lsfdes,tfail,tpffal,kmultd,lmultd
       integer filx,i,index,inxget,isfdes,ist,isttbl,iv,j,jj,jx,k,
     & kc,kk,l,lc,lcontg,lenlfn,m,matmax,mm,n,nd,nj,nuse,nv,nwleft
       real*8 ajth,ftbnid
       integer nface,ir,ix,num,geom,err
       real*8 dj,rin,rout,delt,rdelt
       character*80 record(2)
       character*80 blnk40
cc
c      integer indexj,indexv
cc
c
c  Data statements
       data  blnk40 /' '/
      include 'machaf.h'
      include 'machof.h'
      include 'machsf.h'
c
c
c  Determine name of file from which thermodynamic properties data
c  are to be obtained.
c
c  Test which thermodynamic property files are needed and bring
c  them into memory.
c
c
c  Find the maximum fluid number required for the problem by checking
c  the material number of each volume.
c
c  Load data into steam tables data block for each required fluid.
         open (unit=nunit,file=filnamx,status='old',
     &     form='unformatted',iostat=k)
*
      call stread(nunit,nuse,record,tbls)
c
c  If fluid is light or heavy water, load required steam table
c  routines common blocks.
         if (filnamx .eq. 'tpfh2o') then
           nth2x = nt
           nph2x = np
           nsh2x = nst
           ns2h2x = nsp
           klph2x = it3bp
           klp2h2 = it4bp
```

```
           llph2x = it5bp
           nt5h2x = nprpnt
           jplh2x = it3p0
         elseif (filnamx .eq. 'tpfd2o') then
           ntd2x = nt
           npd2x = np
           nsd2x = nst
           ns2d2x = nsp
           klpd2x = it3bp
           klp2d2 = it4bp
           llpd2x = it5bp
           nt5d2x = nprpnt
           jpld2x = it3p0
         endif
c
c  Get number of words still available for data storage, and pointer to
c  next available word.
c
c  Close steam tables data file for this fluid.
         close (unit=nunit)
c
C
      TCRITX = TCRIT
      RCRITX = 1.D0/VCRIT
      PCRITX = PCRIT
      TRIPT  = TTRIP
      TRIPP  = PTRIP
      TRIPR  = 1.D0/VTRIP
      TMINX  = TMIN
      PMINX  = PMIN
C     RMINX
      TMAXX  = TMAX
      PMAXX  = PMAX
C     RMAXX
C
       return
       end
       subroutine gcsub(pzero,tzero,ploss,gczero)
*in32 iprop
*in32 lprop
*in32end
c
c $Id: gcsub.F,v 1.5 1998/04/10 20:08:40 rjw Exp $
c
*if def,newnrc
c  Name: gcsub
c
c  Description : Subroutine gcsub computes the critical mass flux for
c  subcooled liquid using the Henry-Fauske model.
c
c  Cognizant engineer: jmk
c
c  Arguments:
c  gcrit - critical mass flux based on throat area (kg/m2-s)
c  dgcdp - derivative of gcrit wrt upstream pressure.
c  hfact - empirical constant in Henry-Fauske model.
c  err - logical variable, true if solution converged.
```

```
c
c  Input parameters:
c  pzero - stagnation pressure (Pa)
c  tzero - upstream liquid temperature (deg K)
c
*if def,impnon,1
      implicit none
c
c  Define local variables.
       real*8 conv,delg,delp,delp1,dgcdp,dgcdp1,dgcdp2,dgtdp,dndp,
     & dpfi,dpmax,dpmin,dpsub,dsldp,dsvdp,dtdps,dvldp,dvvdp,
     & dxdp,dxeqdp,gcrit,gt,hfact,ptmax,ptmin,psubc,pthrot,ratio,
     & rzero,sfg,small,szero,vfg,xeq,svap,sliq
       integer it,iter
       real*4 pzero, tzero, gczero, ploss
c
       parameter (small=1.0d-6,hfact=0.14)
       parameter (conv=0.001d0)
c
      common/tables/ tbls(45000),prop(26)
      real*8 tbls, prop
C
      COMMON / EOSPARM /  TCRITX, RCRITX, PCRITX, TSATO, TSINV, CVLFUS
     .,            TRIPT, TRIPP, TRIPR, TMINX, PMINX, RMINX
     .,            TMAXX, PMAXX, RMAXX,  NPTS,  INIT,  NFLUID
C
      REAL*8 TCRITX, RCRITX, PCRITX, TSATO, TSINV, CVLFUS
     .,      TRIPT, TRIPP, TRIPR, TMINX,  PMINX,  RMINX
     .,      TMAXX,   PMAXX,  RMAXX
      INTEGER NFLUID, NPTS, INIT
C
c
c  Local variables for RELAP5 properties.

       real*8 betaa,betf,betg,cpp,cppf,cppg,hbarr,hsubf,hsubg,
     & kpa,kpaf,kpag,press,psatt,quall,ttt,ubarr,
     & usubff,usubgg,vbarr,vsubff,vsubgg
c
      logical err
c
       equivalence          ( prop( 1), ttt   ), ( prop( 2), press ),
     & ( prop( 3), vbarr ), ( prop( 4), ubarr ), ( prop( 5), hbarr ),
     & ( prop( 6), betaa ), ( prop( 7), kpa   ), ( prop( 8), cpp   ),
     & ( prop( 9), quall ), ( prop(10), psatt ), ( prop(11), vsubff),
     & ( prop(12), vsubgg), ( prop(13), usubff), ( prop(14), usubgg),
     & ( prop(15), hsubf ), ( prop(16), hsubg ), ( prop(17), betf  ),
     & ( prop(18), betg  ), ( prop(19), kpaf  ), ( prop(20), kpag  ),
     & ( prop(21), cppf  ), ( prop(22), cppg  )
c
       err = .false.
c  Stagnation properties.
c  Load in upstream pressure and temperature.
       press = pzero
       ttt   = tzero
       if (nfluid .eq. 1) then
         call sth2x3 (tbls,prop,it,err)
                         else
```

```
          call  strtp (tbls,prop,it,err)
                            endif
c  Set liquid density, entropy, and saturation pressure at liquid temp.
        rzero = 1.0d0/vbarr
        szero = prop(24)
        psubc = prop(10)
c  Iteration loop for throat pressure.
c  Set initial bounds & guess.
        iter = 1
        ptmax = min(psubc,0.99d0*(dble(pzero)+min(0.0d0,dble(ploss))))
        ptmin = 6.12d+2
c  Use Henry-Fauske at saturation to limit pressure undershoot.
        dpfi = 2.571d-2*pzero**(1.1334d0)
        dpsub = pzero - psubc
        dpsub = (dpfi**4 + dpsub**4)**0.25d0
        pthrot = max(ptmin,(dble(pzero)+min(0.0d0,dble(ploss))-dpsub))
c  Step 1:  solve Bernoulli for mass flux.
 10     continue
        gt = sqrt(2.0d0*rzero*(pzero + ploss - pthrot))
c  Set derivative of mass flux wrt throat pressure.
        dgtdp = - rzero/gt
c  Step 2:  compute throat equilibrium quality.
c  Throat saturation properties.
c  Load in pressure and zero quality
        press = pthrot
        quall = 0.0d0
        if (nfluid .eq. 1) then
         call sth2x2(tbls,prop,err)
        else
         call strpx (tbls,prop,err)
        endif
        sliq  = prop(25)
        svap  = prop(26)
        vfg   = vsubgg - vsubff
        sfg   = svap - sliq
        dtdps = vfg / sfg
        dsldp = -vsubff*betf + cppf*(dtdps/ttt)
        xeq = (szero - sliq)/sfg
        xeq = max(small,min(1.0d0,xeq))
c  Apply non-equilibrium factor & set derivative wrt pressure.
        if (xeq .gt. hfact) then
c  Use equilibrium option.
          dndp = 0.0d0
          ratio = 1.0d0
        elseif (hfact .lt. 1000.d0) then
c  Apply non-equilibrium parameter.
          dsvdp  = -vsubgg*betg + cppg*(dtdps/ttt)
          dxeqdp =  -(xeq*dsvdp + (1.0d0- xeq)*dsldp)/sfg
          dxeqdp = min(0.0d0,dxeqdp)
          dndp   = dxeqdp/hfact
          ratio  = min(1.0d0,xeq/hfact)
        else
c  Use frozen option.
          dndp  = 0.0d0
          ratio = 0.0d0
        endif
c  Step 3:  critical mass flux.
```

```
       dvldp = -vsubff*kpaf
c  Derivative of quality wrt pressure.
       dxdp = -ratio*(dsldp/sfg)
c  Critical mass flux.
       gcrit = sqrt(-1.0d0/(dvldp + vfg*dxdp))
c  Approximate derivative wrt pressure.
       dvvdp = vsubgg*(betg*dtdps - kpag)
       dgcdp = 0.5d0*(vfg*dndp*(-dsldp/sfg) + dxdp*dvvdp)*gcrit**3
c  Step 4:  check convergence & update throat pressure.
c  Check convergence.
       dpmax = ptmax - ptmin
       delg = abs(gt - gcrit)
       delg = min(delg,dpmax*abs(dgtdp))
       if (delg/gt .gt. conv) then
       delp1  = (gt - gcrit)/(dgcdp - dgtdp)
       dgcdp2 = 3.0d0*dgcdp/gcrit
       dgcdp1 = max(0.25d0,min(1.5d0,(1.0d0+
0.5d0*dgcdp2*delp1)))*dgcdp
       delp   = (gt - gcrit)/(dgcdp1 - dgtdp)
c  Set iteration limits.
         dpmin = abs(conv*gt/dgtdp)
         if (delp .gt. 0.0d0) then
           ptmin = pthrot
           dpmax = ptmax - ptmin
           pthrot = pthrot + min(delp,0.95d0*dpmax)
         else
           ptmax = pthrot
           dpmax = ptmax - ptmin
           if (abs(gt - gcrit)/gt .gt. 0.25d0) then
             pthrot = pthrot + max(min(-dpmin,delp),-0.95d0*dpmax)
           else
             pthrot = pthrot + max(delp,-0.95d0*dpmax)
           endif
         endif
         iter = iter + 1
         if (iter .lt. 20) then
c  Continue iterating on throat pressure.
           go to 10
         else
c  Did not converge, set err flag and default value.
           err = .true.
           dpsub = min(dpsub,dble(pzero + ploss))
           gt = sqrt(2.0d0*rzero*dpsub)
           dgtdp = rzero/gt
         endif
       endif
c  Use mass flux from momentum solution.  The expression for
c  the critical mass flux can be very sensitive to throat
c  pressure and its derivative very large.
       gczero = gt
       dgcdp  = abs(dgtdp)
*endif
       return
       end
```

Appendix B

FORTRAN Listing of MELCOR Modifications for Lithium-Air Reaction Kinetics

```
      SUBROUTINE LIFIRE (
C   INPUT
     1  NUMVOL, NUMMAT, NCVVZ, ICVFLG, ICVNUM,
     2  CVVOL, DT, ICVVZP, CVZ, CVMVL, ZPOOL, PVOL, PPART, XMASS,
     3  CVVEL, TEMPS, VSATO, TSPCO, BETA, ETOTL, DELE, DELM )
C
C     LIFIRE CALCULATES THE MASS AND ENERGY TRANSPORT BETWEEN
C     THE POOL AND ATMOSPHERE DURING LITHIUM FIRES - BJM 12/2/99
C
C     INPUT
C       NUMVOL = NUMBER OF CONTROL VOLUMES - DIMENSION PARAMETER
C       NUMMAT = NUMBER OF MATERIALS - DIMENSION PARAMETER
C       NCVVZ  = DIMENSION OF ARRAYS CVZ AND CVMVL
C       ICVFLG = CONTROL VOLUME FLAGS AND POINTERS.REFER TO PARAM
ICVFLG
C       ICVFLG(KCVACT,*) = ACTIVE/INACTIVE SWITCH FOR CONTROL VOLUMES
C       ICVFLG(KCVTHR,*) = CONTROL VOLUME THERMODYNAMICS SWITCH
C       ICVNUM(NUMVOL)   = USER NUMBER OV CONTROL VOLUME
C       CVVOL = VOLUME OF POOL, FOG AND ATMOSPHERE IN CONTROL VOLUME
C       DT    = TIMESTEP
C       ICVVZP= POINTER ARRAY FOR CVZ AND CVMVL
C       CVZ   = ALTITUDES FOR CONTROL VOLUME'S VOLUME-ALTITUDE TABLES
C       CVMVL = VOLUMES FOR CONTROL VOLUME'S VOLUME-ALTITUDE TABLES
C       ZPOOL = POOL ELEVATION
C       PVOL  = TOTAL PRESSURE
C       PPART = PARTIAL PRESSURES
C       XMASS = START-OFF-STEP MASSES
C       CVVEL(2,NUMVOL) = POOL AND ATMOSPHERE CONTROL VOLUME VELOCITIES
C       TEMPS = TEMPERATURES
C       DELE(3,NVOL) = CONTAINS ENERGY CHANGE
C       VSATO(KS1DIM,KS2DIM,NNVOL) = OLD SATURATION DATA
C       TSPCO(KSP1DM,NUMMAT,NUMVOL) = OLD SPECIFIC THERMO DATA
C       CTE(NUMMAT,NUMVOL) = THERMAL EXPANSION COEFFICIENT
C
C     OUTPUT
C       DELM(NUMMAT,NUMVOL) = CHANGE MASSES OF MATERIALS
C       DELE(NUMMAT,NUMVOL) = CHANGE IN INTERNAL ENERGIES OF MATERIALS
C
      EXTERNAL H2OSAT, MEXSTP, MPDFSA, MPDFSH, MPEVAL
      INTRINSIC MAX, MIN, ABS, EXP, LOG, SQRT
C
*- INCLUDE SATDAT
C     THIS PARAMETER STATMENT CONTAINS THE POINTERS TO THE SATURATION
C       DATA STORED IN ARRAYS VSATO(KS1DIM,KS2DIM,NNVOL)
C       AND VSATN(KS1DIM,KS2DIM,NNVOL)
C
C     FIRST DIMENSION: DEPENDENT VARIABLE
C       KSRHOL = LIQUID WATER DENSITY
C       KSRHOV = STEAM DENSITY
C       KSHL   = LIQUID WATER ENTHALPY
C       KSHV   = STEAM ENTHALPY
C       KSEL   = LIQUID WATER INTERNAL ENERGY
C       KSEV   = STEAM INTERNAL ENERGY
C       KSCPL  = CONSTANT PRESSURE SPECIFIC HEAT OF LIQUID WATER
C       KSCPV  = CONSTANT PRESSURE SPECIFIC HEAT OF STEAM
C       KSPS   = SATURATION PRESSURE
C       KSTS   = SATURATION TEMPERATURE
```

```
C        KSHLV  = HEAT OF VAPORIZATION
C        KS1DIM = FIRST DIMENSION OF SATURATION DATA ARRAY
C      SECOND DIMENSION: INDEPENDENT VARIABLE
C        KSCVP  = CONTROL VOLUME PRESSURE
C        KSSPP  = STEAM PARTIAL PRESSURE
C        KSPLT  = POOL TEMPERATURE
C        KSATT  = ATMOSPHERE TEMPERATURE
C        KS2DIM = SECOND DIMENSION OF SATURATION DATA ARRAY
C
      PARAMETER(
     1  KSRHOL =  1, KSRHOV =  2, KSHL   =  3, KSHV   =  4,
     2  KSEL   =  5, KSEV   =  6, KSCPL  =  7, KSCPV  =  8,
     3  KSPS   =  9, KSTS   = 10, KSHLV  = 11, KS1DIM = 11,
     4  KSCVP  =  1, KSSPP  =  2, KSPLT  =  3, KSATT  =  4,
     5  KS2DIM =  4 )
*-
*- INCLUDE CVHSP
C       THIS PARAMETER STATMENT CONTAINS POINTERS TO THE SPECIFIC
C       THERMODYNAMIC QUANTITIES IN ARRAYS TSPCN AND TSPCO.
C       DIMENSION:  TSPCN(KSP1DM,NUMMAT,NUMVOL)
C
C       FIRST INDEX
C         KSPRHO = MASS DENSITY (KG/M**3)
C         KSPH   = SPECIFIC ENTHALPY (J/KG/K)
C         KSPE   = SPECIFIC INTERNAL ENERGY (J/KG/K)
C         KSP1DM = FIRST DIMENSION
C       SECOND INDEX
C         MATERIAL NUMBER
C       THIRD INDEX
C         VOLUME NUMBER
       PARAMETER ( KSPRHO = 1 , KSPH = 2 , KSPE = 3 , KSP1DM = 3 )
*-
*- INCLUDE CVHBOL
C       THIS PARAMETER STATEMENT CONTAINS POINTERS TO THE POOL/ATMOS
C       MASS AND ENTHALPY TRANSFER DATA IN ARRAY BOIL(KBOLDM,NNVOL).
C
C       KBOLM = MASS FLUX FROM POOL TO ATMOS
C       KBOLH = ENTHALPY FLUX FROM POOL TO ATMOS
C       KBOLHA = CONDUCTION AND CONVECTION ENERGY FLUX FROM INTERFACE
C                TO ATMOS
C       KBOLCA = HEAT TRANSFER COEFFICIENT FROM INTERFACE TO ATMOS
C       KBOLDM = FIRST DIMENSION OF BOIL ARRAY
C
      PARAMETER ( KBOLM = 1 , KBOLH = 2 , KBOLHA = 3 , KBOLCA = 4 ,
     1 KBOLDM = 4 )
*-
C
      PARAMETER ( PZERO = 0.D0 , PONE = 1.D0 , PTWO = 2.D0)
C
*- INCLUDE SC4407
C      4407 CONTAINS SENSITIVITY COEFFICIENTS FOR THE POOL/ATMOS
C      HEAT AND MASS TRANSFER
      COMMON /F4407 / I4407  , C4407(12) , NA4407
      COMMON /CP4407/ JP4407 , IS4407(1) , IE4407(1) , MA4407
      COMMON /CF4407/ CI4407
      CHARACTER CI4407*27
      EQUIVALENCE ( C4407(1), VELBUB), (C4407(2),TRACC),
```

```
     1 (C4407(3),CTLATA) , (C4407(4),ETLATA) , (C4407(5),CTLALA),
     1 (C4407(6),ETLALA) , (C4407(7),CTLATP) , (C4407(8),ETLATP),
     1 (C4407(9),CTLALP) , (C4407(10),ETLALP) , (C4407(11),ALPMAX),
     1 (C4407(12),CFRMAX)
C
*-
*- INCLUDE CONST
C
C************** PROCESSOR DEPENDENT NUMBERS
C      BIGNMP (BIGNMM) = LARGEST POSITIVE (NEGATIVE) NUMBER
C      SMLNMP (SMLNMM) = SMALLEST POSITIVE (NEGATIVE) NUMBER
C      RNDNUM IS THE ROUNDOFF ASSOCIATED WITH THE NUMBER 1.
C      EXPMIN (EXPMAX) = LARGEST NEGATIVE (POSITIVE) NUMBER THAT CAN
C       BE USED AS AN ARGUMENT FOR THE EXP FUNCTION
C      XLNMX (XLOGMX) = LARGEST NUMBER THAT CAN BE USED AS AN ARGUMENT
C       FOR THE LN (LOG) FUNCTION
C********************* CONSTANTS
C      GRAVTY IS THE ACCELERATION OF GRAVITY (M/S**2) = 9.80665D0
C      ONE (ZERO,TWO,HALF) HOLDS THE NUMBER ONE (ZERO,TWO,HALF)
C      PI HOLD THE NUMBER PI= 3.141592653589793238462643383279D0
C      RGAS IS THE UNIVERSAL GAS CONSTANT (J/MOLE-K) = 8.31441D0
C      TCRIT HOLDS THE CRITICAL POINT OF WATER (K) = 647.244989D0
C      ZERODC HOLDS THE TEMPERATURE IN KELVIN EQUAL TO ZERO DEGREES
C        CENTIGRADE (K) = 273.15D0
C      ONETRD = 1/3,   TWOTRD = 2/3
C      SBCON = STEPFAN-BOLTZMANN CONSTANT (WATT/M**2-K**4) = 5.67D-8
C
      COMMON /CONST/  BIGNMP, BIGNMM, SMLNMP, SMLNMM, RNDNUM, EXPMIN
     1      , EXPMAX, XLNMX , XLOGMX, GRAVTY, ONE   , PI    , RGAS
     2      , TCRITX, ZERO  , ZERODC, ONETRD, TWOTRD, SBCON , TWO
     3      , HALF
C
*-
*- INCLUDE MXXMON
C      THE ENTRIES ARE USED TO TEST WHETHER A PACKAGE IS ACTIVE
C      OR INACTIVE.  IF A PACKAGE IS ACTIVE, THEN THE ENTRY IS
C      NONZERO.  IF A PACKAGE IS INACTIVE, THEN THE ENTRY IS ZERO.
C      THE FORMAT FOR THE ENTRY IS MONXXX WHERE XXX IS THE THREE
C      OR TWO CHARACTER IDENTIFIER FOR THE PACKAGE
C
C      ADD ACTIVITY FLAG FOR EXTERNAL DATA FILE PACKAGE (MONEDF)
C
      COMMON /MXXMON/  MONNCG, MONCVH, MONFL,  MONCVT, MONCOR,
     1 MONINP, MONCRA, MONCF,  MONTF,  MONSPR, MONMES, MONH2O,
     2 MONHS,  MONBUR, MONDCH, MONCAV, MONMP,  MONRN1, MONRN2,
     3 MONTP,  MONFDI, MONESF, MONBH , MONEDF, MONSC
*-
*- INCLUDE NCGEOS
C
C      NON-CONDENSIBLE GAS EOS COMMON BLOCKS
C
      PARAMETER (IEOS = 27)
C
      COMMON /NCGNAM/ NAMGAS
C
      CHARACTER*8 NAMGAS(IEOS)
C
```

```
      COMMON /NCGEOS/ MAXGAS, NMMAT, RNCG(IEOS), WM(IEOS),
EMAXIM(IEOS),
     2 KH2OP  , KH2OLA , KH2OVA , KH2    , KO2    , KCO2   , KCO    ,
     3 KN2    , KNO    , KN2O   , KNH3   , KC2H2  , KCH4   , KC2H4  ,
     4 KHE    , KAR    , KD2    ,
     5 KGASA  , KGASB  , KGASC  , KGASD  , KGASE  , KGASF  , KGASG  ,
     6 KGASH  , KGASI  , KGASJ  ,
C     SCRATCH STORAGE FOR EOS WORK
     7 XNCGSA(IEOS), XNCGSB(IEOS), XNCGSC(IEOS), XNCGSD(IEOS),
     8 XNCGSE(IEOS), XNCGSF(IEOS), XNCGSG(IEOS), XNCGSH(IEOS)
C
      DIMENSION KPONT(IEOS)
      EQUIVALENCE (KPONT(1),KH2OP)
*-
*- INCLUDE MPPRNO
C
C          COMMON BLOCK /MPPRNO/ HAS ORDERING OF PROPERTIES IN LIST
C          GIVEN IN /MPPRNM/
C
      COMMON /MPPRNO/ NUMENH, NUMTMP, NUMCPS, NUMTHC, NUMVIS, NUMDIF,
     1                NUMRHO
C
C                 NUMENH = 1       ENTHALPY
C                 NUMTMP = 2       TEMPERATURE
C                 NUMCPS = 3       SPECIFIC HEAT
C                 NUMTHC = 4       THERMAL CONDUCTIVITY
C                 NUMVIS = 5       VISCOSITY
C                 NUMDIF = 6       DIFFUSIVITY
C                 NUMRHO = 7       DENSITY
C
*-
*- INCLUDE ICVFLG
C      THIS PARAMETER STATEMENT CONTAINS POINTERS TO THE CONTROL
C      VOLUMES FLAGS AND SWITCHS IN ARRAY ICVFLG(KCFLDM,NNVOL)
C
C      KCVFF POINT TO THE FLOW GEOMETRY FLAG
C      KPFSW POINTS TO THE POOL/NO POOL, FOG/NO FOG SWITCH
C      KCVTHR POINTS TO THE CONTROL VOLUME THERMO SWITCH
C      KCVTYP POINTS TO THE CONTROL VOLUME TYPE
C      KCVACT POINTS TO THE ACTIVE/INACTIVE SWITCH
C      KCFLDM IS THE FIRST DIMENSION OF THE ICVFLG ARRAY
       PARAMETER ( KCVFF = 1 , KPFSW = 2 , KCVTHR = 3 ,
     1 KCVTYP = 4 , KCVACT = 5 , KCFLDM = 5 )
*-
C
      DIMENSION CVVOL(3,NUMVOL), XMASS(NUMMAT,NUMVOL),
     1 ETOTL(NUMMAT,NUMVOL), TEMPS(2,NUMVOL),
     2 ICVVZP(NUMVOL+1), CVZ(NCVVZ),
     3 CVMVL(NCVVZ), ZPOOL(NUMVOL), ICVFLG(KCFLDM,NUMVOL),
     4 CVVEL(2,NUMVOL),
     5 DELM(NUMMAT,NUMVOL), DELE(3,NUMVOL),
     6 VSATO(KS1DIM,KS2DIM,NUMVOL), TSPCO(KSP1DM,NUMMAT,NUMVOL)
       DIMENSION PVOL(NUMVOL), PPART(NUMMAT,NUMVOL), ICVNUM(NUMVOL),
     1 BETA(NUMMAT,NUMVOL)
C
      SAVE IMESAG
C   DIAGNOSTIC MESSAGE COUNTER
```

```
      DATA IMESAG /0/, MAXMSG /50/
C   LOWER LIMIT ON NCG FRACTION FOR DIFFUSION TREATMENT
      DATA EPSNCG /1.0D-6/
C   MULTIPLIER FOR EFFECT OF MCP ON TO-INTERFACE HEAT FLUXES
      DATA DAMPER /2.0D0/
C   MP PARAMETERS
      DATA ICNCRN /-1/
      DATA PDUM /-1.D0/
C
      COMMON / EOSPARM /  TCRIT, RCRIT, PCRIT, TSATO, TSINV, CVLFUS
     .,            TRIPT, TRIPP, TRIPR, TMIN,  PMIN,  RMIN
     .,            TMAX,   PMAX, RMAX,  NPTS,  INIT,  NFLUID
C
      REAL*8 TCRIT, RCRIT, PCRIT, TSATO, TSINV, CVLFUS
     .,      TRIPT, TRIPP, TRIPR, TMIN,  PMIN,  RMIN
     .,      TMAX,   PMAX,  RMAX
C * * *    J/kg-Li    kcal/g mole Li2O * J/kcal * gm/kg / gmole Li
/(gm/gmole LI)
      PARAMETER ( HFLI2O = 142.7 * 4187. * 1000./ 2. / 7. )
      PARAMETER ( HFLI3N =  47.5 * 4187. * 1000./ 3. / 7. )
      PARAMETER ( CPLI3N = 2980., CPLI2O = 2296., CPLI = 4190. )
      PARAMETER ( WMLI2O = 30.E-3, WMLI3N = 35.E-3)
C
C * * * ENTHALPY OF FORMATION VS TEMPERATURE
      LOGICAL TDEPR
      DATA TDEPR /.TRUE./
      LOGICAL LA5
      DATA LA5 / .TRUE./
      DATA FMOBL / 0.067 /
C
      COMMON / FLIAERO / DMLI2O(2,999), DMLI3N(2,999)
C
C      ipass = ipass + 1
C
C * * CHECK FOR REACTANTS
      IF(KO2.EQ.PZERO .AND. KN2.EQ.PZERO) RETURN
C
C
      DO 1000 NVOL = 1,NUMVOL
C
      DMLI2O(1,NVOL) = 0.
      DMLI2O(2,NVOL) = 0.
      DMLI3N(1,NVOL) = 0.
      DMLI3N(2,NVOL) = 0.
C
      IF ( ICVFLG(KCVACT,NVOL) .NE. 0 ) GO TO 1000
C
C      ----------------------------------------
C      DETERMINE IF ATMOSPHERE MATERIALS PRESENT
C      ----------------------------------------
      IF( (XMASS(KO2,NVOL) .EQ. PZERO
     .     .AND. XMASS(KN2,NVOL) .EQ. PZERO)
     .          .OR.  XMASS(1,NVOL) .EQ. PZERO) GO TO 1000
C
C      --------------------------
C      DETERMINE IF POOL IS PRESENT
C      --------------------------
```

```
C
      IF (XMASS(1,NVOL) .GT. PZERO  .AND.
     2        CVVOL(1,NVOL) .GT. PZERO        ) THEN
C
C        ==========================================================
C        GET POOL SURFACE AREA (SAREA), DEPTH OF POOL (DZPOOL), AND
C        DEPTH OF ATMOSPHERE (DZATMS).  NOTE THAT POOL SURFACE
C        CAN BE WITHIN ROUNDOFF OF TOP OR BOTTOM OF VOLUME
C        ==========================================================
C
      DO 110,N = ICVVZP(NVOL)+1 , ICVVZP(NVOL+1)-1
        IF ( CVZ(N) .GE. ZPOOL(NVOL) ) THEN
            SAREA = (CVMVL(N)-CVMVL(N-1))/(CVZ(N)-CVZ(N-1))
            DIA = SQRT( SAREA*4./PI )
            GO TO 111
                                       ENDIF
  110 CONTINUE
  111 CONTINUE
C
        DZPOOL = ZPOOL(NVOL)-CVZ(ICVVZP(NVOL))
        IF (DZPOOL .LE. PZERO)
     1      DZPOOL = CVVOL(1,NVOL)/SAREA
        DZATMS = CVZ(ICVVZP(NVOL+1)-1) - ZPOOL(NVOL)
        IF (DZATMS .LE. PZERO)
     1      DZATMS = (CVVOL(1,NVOL)+CVVOL(2,NVOL))/SAREA
                                                ELSE
      GO TO 1000
                                                ENDIF
       FPDOH = MAX(0.,(DIA/DZPOOL-0.89)/(16.-0.89))
       FPDOH = MIN(1.,FPDOH)

C          ---------------------------------------------------
C          DENSITY, SPECIFIC HEAT, THERMAL EXPANSION COEFFICIENT
C          ---------------------------------------------------
      RHOL = XMASS(1,NVOL)/CVVOL(1,NVOL)
C
      SUMMAS = PZERO
      BETAA  = PZERO
      DO 115,NMAT=3,NUMMAT
      SUMMAS = SUMMAS + XMASS(NMAT,NVOL)
      BETAA  = BETAA  + BETA(NMAT,NVOL)*XMASS(NMAT,NVOL)
  115 CONTINUE
      RHOA = SUMMAS/CVVOL(3,NVOL)
      IF(SUMMAS.NE.ZERO) THEN
      BETAA = BETAA / SUMMAS
                         ELSE
      BETAA = BETA(3,NVOL)
                         ENDIF

C
C          --------------------------
C          PRESSURES AND TEMPERATURES
C          --------------------------
          PTOT = PVOL(NVOL)
          TPOL = TEMPS(1,NVOL)
          TATM = TEMPS(2,NVOL)
C
```

```
C           ---------
C           VISCOSITY
C           ---------
          CALL MPEVAL(NUMVIS,XMASS(1,NVOL),ICNCRN,3,NUMMAT,'CVH',
     1       NUMMAT,PTOT,TATM,VISCA,IGERR)
          IF ( IGERR .NE. 0 ) THEN
             WRITE (NOUT,'(A)') ' *** CALL FROM CVHBL FOR ATMOSPHERE'
             WRITE (NDIAG,'(A)') ' *** CALL FROM CVHBL FOR ATMOSPHERE'
             CALL MEXSTP
     1          ('MATERIAL PROPERTIES ERROR DETECTED IN CVH
PACKAGE',49)
          END IF
C
      GRNO = GRAVTY*BETAA*ABS(TPOL-TATM)*DIA**3*(RHOA/VISCA)**2
C
C * * DETERMINE O2 REACTION RATE
C
      ROXO2 = 0.
      IF(KO2.GT.PZERO) THEN
      IF(XMASS(KO2,NVOL).GT.PZERO) THEN
C
      ICNCRN = KO2
      TAVE = 0.5*(TPOL+TATM)
      CALL MPEVAL(NUMDIF,XMASS(1,NVOL),ICNCRN,3,NUMMAT,'CVH',
     1               NUMMAT,PTOT,TAVE,DCO2,IGERR)
C     1               NUMMAT,PTOT,TPOL,DCO2,IGERR)
C     1               NUMMAT,PTOT,TATM,DCN2,IGERR)
          IF ( IGERR .NE. 0 ) THEN
            IERR = 1
            DCO2 = ZERO
            CALL MESERA('*** MP CALL FROM HSTRAN DAB',' ',' ',0)
                                END IF
C
       SCNO  = VISCA/(RHOA*DCO2)
       SHNO  = 0.13*(GRNO*SCNO)**0.33333
       RHOO2 = XMASS(KO2,NVOL)/CVVOL(3,NVOL)
C
C      ROXO2 = 1.05*RHOO2*DCO2*SHNO/DIA
C      ROXO2 = 0.60*RHOO2*DCO2*SHNO/DIA
       FMO2  = 1.0 - 0.4*FPDOH
       ROXO2 = FMO2*RHOO2*DCO2*SHNO/DIA
       RO2LA5= 1.E30
       IF(LA5) THEN
       RO2LA5= 1.8e6*EXP(-15360./TPOL)
               ENDIF
       ROXMAX= 0.9*XMASS(KO2,NVOL)/(SAREA*DT)
       ROXO2 = MIN(ROXMAX,ROXO2,RO2LA5)
C      if(mod(ipass,20).eq.0) then
C      if(ro2la5.eq.roxo2) then
C      print *,"O2 limitED BY RO2LA5"
C                         endif
C      if(ROXMAX.eq.roxo2) then
C      print *,"O2 limitED BY ROXMAX"
C                         endif
C                             endif
C
                                     ENDIF
```

```
                           ENDIF
       RLIO2 = ROXO2*4.*WM(3)/WM(KO2)
C
C * * DETERMINE N2 REACTION RATE
C
      ROXN2 = 0.
      IF(KN2.GT.PZERO ) THEN
      IF(XMASS(KN2,NVOL).GT.PZERO) THEN
C
      ICNCRN = KN2
      TAVE = 0.5*(TPOL+TATM)
      CALL MPEVAL(NUMDIF,XMASS(1,NVOL),ICNCRN,3,NUMMAT,'CVH',
     1               NUMMAT,PTOT,TAVE,DCN2,IGERR)
C     1               NUMMAT,PTOT,TPOL,DCN2,IGERR)
C     1               NUMMAT,PTOT,TATM,DCN2,IGERR)
          IF ( IGERR .NE. 0 ) THEN
            IERR = 1
            DCN2 = ZERO
            CALL MESERA('*** MP CALL FROM HSTRAN DAB',' ',' ',0)
                              END IF
C
       SCNO  = VISCA/(RHOA*DCN2)
       SHNO  = 0.13*(GRNO*SCNO)**0.33333
       RHON2 = XMASS(KN2,NVOL)/CVVOL(3,NVOL)
C
       ROXN2 = 0.30*RHON2*DCN2*SHNO/DIA
       RN2LA5= 1.E30
       FACN2 = 1.0
       TTRAN = 1300.0 - 100.*FPDOH
       TTAU  =  25. + 25.*FPDOH
       IF(TPOL.GT.TTRAN) FACN2 = EXP(-(TPOL-TTRAN)/TTAU)
       IF(LA5) THEN
       RN2LA5= 5.6E8*EXP(-18780./TPOL)
               ENDIF
       ROXMAX= 0.9*XMASS(KN2,NVOL)/(SAREA*DT)
       ROXN2 = FACN2*MIN(ROXMAX,ROXN2,RN2LA5)
C      if(mod(ipass,20).eq.0) then
C      if(rn2la5.eq.roxn2) then
C      print *,"***********N2 LIMITED BY RN2LA5 "
C                          endif
C      if(ROXMAX.eq.roxn2) then
C      print *,"***********N2 LIMITED BY ROXMAX"
C                          endif
C                             endif
C
                                      ENDIF
                                            ENDIF
       RLIN2 = ROXN2*6.*WM(3)/WM(KN2)
C
       ROXLI = RLIO2 + RLIN2
       IF(TDEPR) THEN
       TPDC = TPOL - 273.
C
       IF(TPOL.LE.453.) THEN
       QLI2O = RLIO2*(286.1+4.568E-3*TPDC-6.752E-6*TPDC**2)/(4.*WM(3))
       QLI3N = RLIN2*(78.52+5.664E-3*TPDC-2.067E-6*TPDC**2)/(6.*WM(3))
                                                    ELSE
```

```
       QLI2O = RLIO2*(289.2+4.343E-3*TPDC-7.357E-6*TPDC**2
     .                     +       8.770E-10*TPDC**3)/(4.*WM(3))
       QLI3N = RLIN2*(82.38+1.263E-2*TPDC-2.295E-5*TPDC**2
     .                     +       1.681E-09*TPDC**3)/(6.*WM(3))
                                                    ENDIF
C
       QOXLI = 4187.*(QLI2O + QLI3N)
                ELSE
       QOXLI = RLIO2*HFLI2O + RLIN2*HFLI3N
                ENDIF
C
       IF(ROXLI.GT.0.9*XMASS(1,NVOL)/(SAREA*DT)) THEN
       FR     = 0.9*XMASS(1,NVOL)/(ROXLI*SAREA*DT)
       ROXO2  = FR*ROXO2
       ROXN2  = FR*ROXN2
       ROXLI  = FR*ROXLI
       QOXLI  = FR*QOXLI
                                       ENDIF
C
       DMLI         = SAREA*ROXLI*DT
       EBURN        = EBURN + SAREA*QOXLI*DT
C
       FMOBL = 0.1-0.066*FPDOH
C
       DELMX           = RLIO2*WMLI2O*SAREA*DT/(2.*WM(3))
       DMLI2O(2,NVOL)  =      FMOBL*DELMX
       DMLI2O(1,NVOL)  = (1.-FMOBL)*DELMX
       DELMX           = RLIN2*WMLI3N*SAREA*DT/(3.*WM(3))
       DMLI3N(2,NVOL)  =      FMOBL*DELMX
       DMLI3N(1,NVOL)  = (1.-FMOBL)*DELMX
C      if(mod(ipass,20).eq.0) then
C      Qfrac = QLI3N/(QLI2O + QLI3N)
C       print *,"Eburn =", EBURN/1.e6, qfrac,FMO2,FMOBL,TTRAN,TTAU
C                              endif
       DELM(1,NVOL) = DELM(1,NVOL) - DMLI
       ELI          = ETOTL(1,NVOL)/XMASS(1,NVOL)
       DELE(1,NVOL) = DELE(1,NVOL) - DMLI*ELI
     .                  + SAREA*QOXLI*DT
       IF(KO2.GT.0 .AND. XMASS(KO2,NVOL).GT.0.0 ) THEN
       DMO2           = SAREA*ROXO2*DT
       DELM(KO2,NVOL) = DELM(KO2,NVOL) - DMO2
       EO2            = ETOTL(KO2,NVOL)/XMASS(KO2,NVOL)
       DELE(2,NVOL)   = DELE(2,NVOL) - DMO2*EO2
                       ENDIF
       IF(KN2.GT.0 .AND. XMASS(KN2,NVOL).GT.0.0) THEN
       DMN2           = SAREA*ROXN2*DT
       DELM(KN2,NVOL) = DELM(KN2,NVOL) - DMN2
       EN2            = ETOTL(KN2,NVOL)/XMASS(KN2,NVOL)
       DELE(2,NVOL)   = DELE(2,NVOL) - DMN2*EN2
                       ENDIF
C
1000  CONTINUE
C
      RETURN
      END
```