SANDIA REPORT
SAND2016-0484
Unlimited Release
Printed January 2016

MELCOR/CONTAIN LMR Implementation Report – FY15 Progress

Humphries, Larry L. and Louie, David L.Y.

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865) 576-8401
Facsimile: (865) 576-5728
E-Mail: reports@adonis.osti.gov
Online ordering: http://www.osti.gov/bridge

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd.
Springfield, VA 22161

Telephone: (800) 553-6847
Facsimile: (703) 605-6900
E-Mail: orders@ntis.fedworld.gov
Online order: http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online

SAND2016-0484
Unlimited Release
Printed January 2016

MELCOR/CONTAIN LMR Implementation Report – FY15 Progress

Humphries, Larry L. and Louie, David L.Y.

Severe Accident Analysis Department
Sandia National Laboratories
P.O. Box 5800
Albuquerque, New Mexico 87185-MS0748

Abstract

This report describes the progress of the CONTAIN-LMR sodium physics and chemistry models to be implemented into MELCOR 2.1. It also describes the progress to implement these models into CONTAIN 2 as well. In the past two years, the implementation included the addition of sodium equations of state and sodium properties from two different sources. The first source is based on the previous work done by Idaho National Laboratory by modifying MELCOR to include liquid lithium equation of state as a working fluid to model the nuclear fusion safety research. The second source uses properties generated for the SIMMER code. Testing and results from this implementation of sodium properties are given. In addition, the CONTAIN-LMR code was derived from an early version of CONTAIN code. Many physical models that were developed since this early version of CONTAIN are not captured by this early code version. Therefore, CONTAIN 2 is being updated with the sodium models in CONTAIN-LMR in order to facilitate verification of these models with the MELCOR code. Although CONTAIN 2, which represents the latest development of CONTAIN, now contains many of the sodium specific models, this work is not complete due to challenges from the lower cell architecture in CONTAIN 2, which is different from CONTAIN-LMR. This implementation should be completed in the coming year, while sodium models from CONTAIN-LMR are being integrated into MELCOR. For testing, CONTAIN decks have been developed for verification and validation use.

In terms of implementing the sodium models into MELCOR, a separate sodium model branch was created for this document. Because of massive development in the main stream MELCOR 2.1 code and the requirement to merge the latest code version into this branch, the integration of the sodium models were re-directed to implement the sodium chemistry models first. This change led to delays of the actual implementation. For aid in the future implementation of sodium models, a new sodium chemistry package was created. Thus reporting for the implementation of the sodium chemistry is discussed in this report.

ACKNOWLEDGMENTS

This work was overseen and managed by Matthew R. Denman (Sandia National Laboratories). In addition, we appreciated that Brad J. Merrill of Idaho National Laboratory (INL) for providing us the modification of the MELCOR source code for the sodium works done at INL. We also appreciated that Andrew J. Clark participated the review of this document. This work is supported by the Office of Nuclear Energy of U.S. Department of Energy M3AR-13SN1701031.

CONTENTS

FIGURES

TABLES

NOMENCLATURE

BRISC	Burner Reactor Integrated Safety Code of Laboratory
BUR	Burn Package designator for MELCOR code
C	Chemical
CAV	Cavity Package designator for MELCOR code
CVH	Control Volume Hydrodynamics Package designator for MELCOR Code
DCH	Direct containment heating
DOE	Department of Energy
EOS	Equation of state
FSD	Fusion Safety Database
FY	Fiscal Year
INL	Idaho National Laboratory
JAEA	Japan Atomic Energy Agency
LDRD	Laboratory Directed Research and Development
LMR	Liquid metal reactor
LWR	Light water reactor
NAM	NaModel Package (new) for MELCOR code
ndry	Number of dry nodes in SLAM
NRC	Nuclear Regulatory Commission
P	Physical
PNC	Power Reactor and Nuclear Fuel Development Corporation
SLAM	Sodium limestone ablation model
SNL	Sandia National Laboratories
STD	Standard test deck

Symbol – Applicable only for Chapters 2 and 3

A_G	Fitted constants for Equation V-7
A_L	Fitted constants for Equation A-7
A_{SAT}	Fitted constants for Equation A-6
B_{SAT}	Fitted constants for Equation L-4
b_L	Fitted constants for Equation A-8
C_{SAT}	Heat capacity along the saturation curve
C_P	Heat capacity at constant pressure
C_V	Heat capacity at constant volume
C_{VG}	Specific heat at constant volume at dilute vapor
c_{SAT}	Fitted constants for Equation L-7
d_L	Fitted constants for Equation V-2
E_{coh}	Cohensive energy for Equation L-10
e	Specific internal energy
e_{liq}^{D}	Specific internal energy of infinitely dilute vapor
G_a	Fitted constants for Equation V-10
H	Enthalpy

h_p	Planck constant
k	Thermal conductivity
mw	Molecular weight
N	number of atoms, 2.62×10^{25}
n	Young's fitted value for sodium defined in Equation L-9
Q	Correction factor for sodium in Equation L-9
S	Entropy defined in Equation L-9
T	Temperature
u	Temperature ratio defined in Equation V-6
u_L	Specific internal energy ratio to molten state in Equation V-2
v	Specific volume
α_p	Volumetric thermal expansion coefficient
α_{SAT}	Thermal expansion coefficient along the saturation curve
β_S	Adiabatic compressibility
$\beta_{S,m}$	Adiabatic compressibility at the melting point
β_T	Isothermal compressibility
δ	Constant defined in Equation L-9
Δ	Change in quantity in Equation V-4
ϵ	Constant defined in Equation L-9
γ	Thermal pressure coefficient
γ_{SAT}	Thermal pressure coefficient along the saturation curve
γ_g^C	Constant defined in Equation V-10
κ	Boltzmann's constant
θ	Temperature ratio defined in Equation L-14
ψ'	Variable defined in Equation V-7
ρ	Density
τ	Temperature difference variable for Equation L-7
μ	Viscosity
σ	Surface tension
ν	Sonic velocity as given in Equation L-14
ξ_L	Specific internal energy ratio to critical state in Equation V-2

Subscript

AVG	Average
C	Critical state
g	Vapor
NA	Sodium
l	Liquid
liq	Molten solid state at 371 K
SAT	Saturation

Symbol – Applicable only for Chapters 4 and 5

c_l	Liquid specific heat
c_{vf}	Vapor specific heat at constant pressure
Cx	Cx, where x is 1 to 5 are the stoichiometric coefficients for Reaction (5-9)

D_O	Diffusion constant
E_{spray}	Energy of spray fire in Equation (5-12)
f_1	Fraction of total oxygen consumed, see Equation 5-13
f_2	Fraction of sensible heat from the reaction to the pool
f_3	Fraction of the Na_2O product that enters the pool as a solid
f_4	Fraction of Na_2O_2 product that enters the pool as a solid
$F_{peroxide}$	Mole fraction of sodium peroxide in Equation (5-10)
g	Gas or vapor
h( )	Scratch array for reals or doubles
ih( )	Scratch array for integers
ah( )	Scratch array for characters
lh( )	Scratch array for logical
k_l	Conductivity of the liquid
l	Liquid
k_{vf}	Conductivity of the vapor evaluated at the film temperature
Matx	Matx, where x is from 1 to 5, are the materials for Reaction (5-9)
P	Pressure
$\mathbf{P_p}$	Pool pressure
q(reaction)	Specific reaction energy per unit mass of reaction product
q_{bot}	Heat flux through the bottom of pool at the onset of film boiling
q_{chf}	Critical heat flux
$q_{chf,s}$	Critical heat flux for a subcooled pool
S	Mole fraction ratio of sodium peroxide to its monoxide in Equation (5-11)
T_{bot}	Layer temperature below the pool layer
T_{chf}	Critical heat flux temperature
T_{film}	Film temperature
T_{pool}	Pool layer temperature
T_{sat}	Saturation temperature
X	Distance of the layer in SLAM (see Figure 5-2)
λ	Heat of vaporization of sodium at T_{sat}
λ'	Coefficient defined in Equation 5-18
ρ_l	Liquid density
ρ_v	Density of the vapor
ρ_{vf}	Density of the vapor evaluated at the film temperature
μ_{vf}	Viscosity of the vapor evaluated at the film temperature
δ	Thickness of the layer region defined in Figure 5-1
σ	Sodium liquid surface tension at T_{sat}
Ψ	Coefficient defined in Equation 5-17

Intentionally Leave Blank

1 INTRODUCTION

A sodium coolant accident analysis code is necessary to provide reactor designers and regulators with a means to perform severe accident analyses for future liquid metal reactor (LMR) application, such as sodium fast reactors (SFRs). A gap analysis of the ability for computer codes and models in the U.S. to support the licensing of SFRs identified a gap in the current ability to model source terms and accidents involving the containment [Schmidt 2011]. A subsequent review of gaps involving sodium technology, accident sequences and initiators, source terms, codes and models, and fuels and materials [Denman et al., 2012] identified this gap as a high priority gap, requiring a near term action .

MELCOR and CONTAIN, which are currently employed by the U.S. Nuclear Regulatory Commission (NRC) for light water reactor (LWR) licensing, have been traditionally used for level 2 and level 3 probabilistic risk analyses as well as for the containment design basis accident analysis. Recent endeavors, in part due to increases in containment-reactor pressure vessel coupling through the use of passive safety systems, MELCOR has been employed for the containment design basis accident analysis as well [Tills 2008, Tills 2009 and Tills 2010].

To meet future design basis analysis needs [Schmidt 2011], new models will be added to the MELCOR code for simulation of LMR designs. Existing models developed for separate effects codes will be integrated into the MELCOR architecture. In particular, many LMR models were added to the CONTAIN code (version 1.11) as part of CONTAIN-LMR code released in the 1990s [Murata 1993, Scholtyssek 1994]. This work will integrate those CONTAIN code capabilities that feasibly fit within the code architecture. Among the LMR code capabilities to be considered are models for:

- sodium pool and spray fires,
- treating two condensable (sodium and water) simultaneously,
- sodium atmosphere and pool chemistry,
- sodium condensation on aerosols,
- heat transfer from oxide core-debris beds (lower priority due to the current focus on metallic fuel) and to sodium pools, and
- sodium-concrete interactions.

Implementation of such models for the sodium reactor simulation into an actively maintained, full-featured integrated severe accident code fills a significant gap in the capability for providing the necessary analysis tools. This project will close this gap by implementing, improving, and verifying model development efforts into the MELCOR source code. The current scope will focus on the following original implementation goals:

Phase 1 (FY13): Implement sodium Equations of State (EOS) as a working fluid for a MELCOR calculation from:

- The fusion safety database
- The SIMMER-III/SAS4A Code

Phase 2 (FY14): Examine and test changes to the CONTAIN-LMR Implemented by Japan Atomic Energy Agency (JAEA), specifically:

- Aerosol Condensation
- Implementation of the capability for simultaneous sodium and water condensation modeling

Phase 3 (FY15): Implementation and Validation of CONTAIN physics models [Jeppson 1986]

- Sodium Spray Fires (including new test data)
- Sodium Pool Modeling
- Sodium Pool Fires

Phase 4 (FY16): Implementation and Validation of CONTAIN chemistry models

- Debris Bed/Concrete Cavity Interactions
- Sodium Pool Chemistry
- Atmospheric Chemistry

Note that beginning with MELCOR 2.0, the code architecture and input formats are significantly different than its predecessor, MELCOR 1.8.6. MELCOR 2.0 utilizes many features of FORTRAN 95 such as dynamic memory management and user defined types, which allows for future changes in compilers and hardware. With MELCOR, the working fluid field is modeled as water. Thus the equation of state is strictly applicable for water and steam. Because a single fluid field is allowed in a given problem, the use of a different fluid model requires that the property model for the new fluid must be defined to replace that of the water. This was accomplished in FY13 and current development is focused on extending MELCOR to include sodium specific models as described in this chapter.

This report summarizes what was completed in FY13 to FY15. In FY13, the implementation and testing of sodium properties into MELOCR 2.1 was completed. Some minor issues related to the property implementation were identified, and these issues will be addressed by the end of the 2016. Chapter 2 and Chapter 3 document the tasks completed in FY13. In FY14, the design documentation of the sodium physics models from CONTAIN-LMR was developed. Here CONTAIN 2 was revitalized, which represented the last version of CONTAIN development before CONTAIN's discontinuation. In FY15, to the process of porting the sodium models into CONTAIN 2 was initiated to createCONTAIN2-LMR. Because a number of new MELCOR models had been developed during FY15, including the resuspension model of aerosol in MELCOR 2.1, much of the implementation effort on the two-condensable option (if possible) has been postponed till FY16. We will discuss our approach to implement this option in this report, along with all sodium models to be included in FY16. Chapter 4 discusses the upgrade/modification done to CONTAIN-LMR and CONTAIN 2 which allow them to run within the MELCOR code development environment, and the development of CONTAIN 2-LMR. In Chapter 4, the lessons learned on migrating sodium models from CONTAIN-LMR into CONTAIN 2 are discussed. The lessons learned will benefit our understanding of the coding in versions of CONTAIN to be used for the implementation to MELCOR. Chapter 5 describes the

actual implementation that took place in FY15, and the implementation strategy for FY16. Included in Chapter 5 is the discussion of the merging issue for the main MELCOR trunk and the sodium branch.

Intentionally Leave Blank

2 LIQUID METAL PHYSICAL PROPERTIES

To accommodate sodium as the working fluid field in MELCOR, the sodium thermophysical properties, such as enthalpy, heat capacity, heat of fusion, vapor pressure, heat of vaporization, density, thermal conductivity, thermal diffusivity, viscosity and thermal expansion must be provided to replace those currently used for water. The equation of state (EOS) for water is based on the polynomials in a tabular format. These polynomials relate pressure, specific internal energy, specific entropy and heat capacity to temperature and density, and are expressed analytically in terms of the Helmholtz free energy. In MELCOR, additional thermodynamic properties are derived from the thermodynamic relationships involving Helmholtz free energy, such as fluid internal energy, enthalpy, entropy, specific heat, and derivatives of pressure with respect to temperature and density. The resulting EOS is valid for temperature $\geq$ 273.15 K and for pressure $\leq$ 100 MPa. Water surface tension is calculated in Subroutine tHS_HSBOIL. Additional thermodynamic properties of water can be found in Module M_H2O. This module also contains the single phase EOS for water, which is modeled in Subroutine tH2O_H2O1PH. The mixed-phase (or 2-phase) EOS for water is modeled in subroutine tH2O_H2O2PH. The binary diffusion coefficient for water vapor in a gas mixture is defined in Module M_NCG.

There are a number of data sources for sodium properties that can be considered for implementing into MELCOR. For supporting fusion safety research, Idaho National Laboratory (INL) modified MELCOR 1.8.5 to include lithium and other metallic fluid [Merrill 2000]. This database is called herein the Fusion Safety Database (FSD). The second database (SIMMER) from the SIMMER-III data was considered in the Burner Reactor Integrated Safety Code of Laboratory (BRISC) Laboratory Directed Research and Development (LDRD) at Sandia National Laboratories (SNL). This work was leveraged in SNL's FY13 efforts. Subsequent work will also include the EOS from SAS4a, but this work does not leverage a historical effort like the FSD and SIMMER EOSs. To isolate from the main development of MELCOR 2.1, a separate sodium branch for this development was created. Once all sodium specific implementations are done, these models will be integrated into the main MELCOR 2.1.

2.1 Fusion Safety Database (FSD)

Implementation of non-water fluids into MELCOR have been explored in the past. Earlier work performed at INL allowed the modeling of lithium fires with MELCOR 1.8.5. This modification permitted MELCOR to access properties from the fusion safety data set, which was originally designed for the ATHENA code and is an extension of the RELAP5 environmental library [Merrill 2000]. It includes 13 fluids: water, hydrogen, lithium, potassium, helium, nitrogen, sodium, sodium-potassium, lithium-lead, etc. Code modifications were made to allow evaluation of the equation of state for an array of potential materials. These models were updated to FORTRAN 95 and tested within the code. Also, several interpolation routines used in the MELCOR 1.8.5 implementation were proprietary and new ones have been used. This is our first approach for implementation of liquid metal properties and has already been started, as many of the 1.8.5 models have been ported to the MELCOR 2.1 code as part of this project and shows great promise.

FSD_EOS module contains property interpolation and correlations for processing the input data file as described in NaLibrary program. In this FSD_EOS module, surface tension, thermal conductivity, viscosity, and critical heat flux correlations are also given for various fluids as described above. Examples of the transport property for the liquid sodium modeled in FSD_EOS are given as:

Viscosity, Pa-s [Gierszewski 1980]:

$$\mu = \frac{3.24\times10^{-3}e^{508/T}}{T^{0.4925}} \quad [2\text{-}1]$$

Thermal Conductivity, W/m-K [Gierszewski 1980]

$$k = 110 - 0.0645 \cdot T + 1.173 \times 10^{-5}T^2 \quad [2\text{-}2]$$

Surface Tension, N/m (curve fitted)

$$\sigma = 0.235115 - 1 \times 10^{-4}T \quad [2\text{-}3]$$

Note that the development of the FSD set requires the user to provide a property input file in order to utilize the FSD_EOS and other program files for this database. The required input file must be named such that it matches the desired fluid to be simulated. Section 3.2 describes more details of the filename requirement. The required unformatted input file contains the thermodynamic properties of simulated fluid. The input file is generated by running the NaLibrary Program. A brief description is given below:

NaLibrary

For the FSD data set, the input data file for sodium is required. A FORTRAN program written by J.E. Tolli of EG&G Idaho, Inc. in September 1991 can be used to produce this input data file. This program requires the user to provide an input containing temperatures and pressures in metric units.

The program generates tables of selected thermodynamic properties as functions of temperature and pressure for both saturation and single phase conditions, liquid and/or vapor states for sodium using the soft-sphere model free energy equation [Young 1977, Blink 1982]. The output of this program provides the triple and critical temperatures, pressures and volumes; saturation properties (temperature versus pressure) tables of temperature, pressure, specific volume, internal energy, thermal expansion, isothermal compressibility, specific heat and entropy all the input range of temperature and pressure; saturation properties (pressure versus temperature) tables of the same property parameters as listed before. It is followed by the thermodynamic properties tables for specific volume, internal energy, thermal expansion, isothermal compressibility, specific heat and entropy. To limit the file size, this program generates the output file in binary form. This output file is input to the multi-fluid MELCOR code. As previously described, MELCOR determines the fluid type by the name of the property data

filename. However, the entropy correlations are obtained from the soft-sphere model as described in the NaLibrary Program. (See Appendix A for the list of the tables input/outputs by this program.)

2.2 SIMMER Database

Our second approach builds off the BRISC LDRD performed at SNL. The BRISC approach took sodium coolant properties and directly implemented them into a branch version of the MELCOR 1.8.6 code. Due to the limited scope, the implementation was performed as a proof of concept which restricted to the complete implementation, validation, verification, full test case development, etc. The property model was based on an analytic EOS model developed for the SIMMER-III code. Many of the code modifications that are required for this modeling approach are identical or extensions to those modifications developed for the previously described FSD approach. As part of this project, the code changes made to the 1.8.6 code were ported to MELCOR 2.1. Debugging is in progress.

Some of the properties and EOS as described by Fink and Leibowitz for the inclusion of the sodium in SAS4A, a severe accident code for liquid metal analyses [Cahalan 1994, Dunn 2012], were also implemented as SIMMER database. For the liquid sodium, much of the thermodynamic and transport properties are derived from saturated condition. Table 2-1 shows the liquid sodium property equations/correlations modeled in MELCOR. For the vapor sodium, examples of the thermodynamic and transport properties are given in Table 2-2. Additional properties and values used in the property determination are given in Table 2-3. Most of the correlations shown in these tables were originated mainly from the SIMMER data set.

Table 2-1 Liquid Sodium Thermophysical Properties Modeled*

Property [reference]	Equation	Range	Equation #
Melting temperature (K), Eq.(L-1) [Fink 1979]	$T_{liq} = 370.98$		L-1
Melting density (kg/m^3), Eq.(L-2)	$\rho_{liq} = 927.63$		L-2
Molten specific internal energy (J/kg)	$e_{liq} = 2.06494 \times 10^5$		L-3
Density (kg/m^3) [Morita 1998a]	$\rho_l = \rho_{liq} \cdot \left[1 + \sum_{i=1}^{3} B_{SAT,i}\left(T - T_{liq}\right)^i\right]$	$T_{liq} < T \leq B_{SAT,4} T_C$	L-4a
	$\rho_l = \rho_C \cdot \left[1 + B_{SAT,5} \cdot \tau^{0.5} + B_{SAT,6} \cdot \tau^2\right]$	$B_{SAT,4} \cdot T_C < T \leq T_C$	L-4b

Property [reference]	Equation	Range	Equation #
	$\rho_l = \rho_C$ where $B_{SAT,i}$ = fitted constants, where i is from 1 to 3 for L-4a $\tau = T_C - T$	$T > T_C$	L-4c
Specific heat at constant pressure (J/kg-K) [Fink 1979]	$C_P = C_{SAT} \cdot T \cdot \alpha_P \cdot \gamma_{SAT}/\rho_l$ where $\gamma_{SAT} = \left(\frac{dP}{dT}\right)_{SAT}$. Note this derivative is taken as $\left(\frac{dP}{dT}\right)_{SAT} = p_v \cdot \left(b_{L,2} - \frac{b_{L,3}}{T^2} + \frac{b_{L,4}}{T}\right)$ where $b_{L,i}$ is given by p_g		L-5
Specific heat at constant volume (J/kg-K) [Fink 2979]	$C_V = C_P \cdot \beta_S/\beta_T$		L-6
Specific internal energy (J/kg) [Morita 1998a]	$e_l = e_{liq} \cdot \left[\sum_{i=1}^{3} C_{SAT,i}(T - T_{liq})^i\right]$ $e_l = e_C \cdot \left[1 - C_{SAT,5} \cdot \tau^{0.5} + CB_{SAT,6} \cdot \tau^2\right]$ where $C_{SAT,i}$ = fitted constants, where i is from 1 to 3 for L-7a $\tau = T_C - T$	$T_{liq} < T \leq C_{SAT,4} T_C$ $C_{SAT,4} \cdot T_C < T \leq T_C$	L-7a L-7b
Enthalpy (kJ/kg) [Fink 1995]	$H_l = -365.77 + 1.6582 \cdot T - 4.2375 \times 10^{-4} T^2 + 1.4847 \times 10^{-7} T^3 + 2992.6/T$ $H_l = 2128.4 + 0.86496 \cdot T$	$T_{liq} \leq T < 2000K$ $2000K \leq T \leq T_C$	L-8a L-8b

Property [reference]	Equation	Range	Equation #
Entropy (J/K) [Young 1977, Blink 1982]	$S = \frac{dH_z}{dT} = \frac{E - A}{T}$ $= N\kappa\left[\frac{3}{2} + \ln\left(\frac{\frac{e}{N \cdot \left(h_p\sqrt{N/(2\pi\kappa)}\right)^3}T^{1.5}}{\rho_l}\right) + \frac{1}{3}(n+4)Q \cdot (\rho_l\delta)^{n/9}\left(\frac{\epsilon}{\kappa T}\right)^{1/3}\right]$ where N = number of atoms, 2.62×10^{25} κ = Boltzmann's constant, 1.3806×10^{-23} J/K h_p = Planck constant, 6.62618×10^{-34} J s $\delta = 7.12349 \times 10^{-30}$ Q = correction factor, 0.95 for sodium $\epsilon = 6.21332 \times 10^{-19}$ J/atom for sodium $n = 8$ for sodium	$4 \le n \le 12$	L-9
	$E = N\kappa T\left[\frac{3}{2} + c_n\rho_l^{n/3}\left(\frac{\epsilon}{\kappa T}\right) + \frac{1}{6}(n+4)Q\rho_l^{n/9}\left(\frac{\epsilon}{\kappa T}\right)^{1/3} - \rho_l^m\left(\frac{\epsilon}{\kappa T}\right)\right] + E_{coh}$		L-10a
	$A = N\kappa T\left[-\ln\left(\frac{Ve}{n\lambda^3}\right) + c_n\rho_l^{n/3}\left(\frac{\epsilon}{\kappa T}\right) + \frac{1}{2}(n+4)Q\rho_l^{n/9}\left(\frac{\epsilon}{\kappa T}\right)^{1/3} - \rho_l^m\left(\frac{\epsilon}{\kappa T}\right)\right] + E_{coh}$ where E_{coh} = cohensive energy Note the first term in Eq.(L-10b) can be reduced when determining the right hand side of Eq.(L-9)		L-10b
Derivative of Partial pressure [Morita 1998a]	$\frac{dP}{d\rho_l} = \frac{R \cdot T}{(1-\rho_l)^2} - \frac{a(T)\left(\frac{2}{\rho_l} + a_{G,3}\right)}{\left(\frac{1}{\rho_l} + a_{G,3}\right)^2}$ where $a(T) = a_{G,2}\left(\frac{T}{T_C}\right)^{a_{G,4}}$ $a(T) = a_{G,2}\left[1 + a_{G,4}\left(\frac{T}{T_C} - 1\right)\right]$	$T < T_C$ $T \ge T_C$	L-11a

Property [reference]	Equation	Range	Equation #
	$\frac{dP}{dT} = \frac{C_P - C_V}{T}\left(\frac{dP}{d\rho_l}\right)^{0.5} \rho_l$		L-11b
Volumetric thermal expansion coefficient (K^{-1}) [Fink 1979]	$\alpha_p = \alpha_{SAT} + \beta_T \cdot \gamma_{SAT}$ where $\alpha_{SAT} = -\left(\frac{d\rho_l}{dT}\right)_{SAT} / \rho_l$ Note that $\left(\frac{d\rho_l}{dT}\right)_{SAT}$ is calculated by taking the derivative of ρ_l equation above.		L-12
Isothermal compressibility (1/Pa) [Fink 1979]	$\beta_T = \frac{\beta_S C_{SAT} + \frac{T}{\rho_l}\alpha_{SAT}(\alpha_{SAT} + \beta_S \gamma_{SAT})}{C_{SAT} - \frac{T}{\rho_l}\gamma_{SAT}(\alpha_{SAT} + \beta_S \gamma_{SAT})}$ where $C_{SAT} = \left(\frac{dH_l}{dT}\right)_{SAT} - \frac{\gamma_{SAT}}{\rho_l}$ $\left(\frac{dH_l}{dT}\right)_{SAT} =$ $35.206 - 2 \cdot 7.0513 \times 10^{-3} T + 3 \cdot 2.5711 \times 10^{-6} T^2 + 12480/T^2$ $18.525 + 0.5 \cdot 43.402 \cdot 0.32227 \cdot (1 - T/T_C)^{0.32227-1}$ 18.525	 $T < 1644K$ $1644K \leq T \leq T_C$ $T > T_C$	L-13
Adiabatic compressibility (1/Pa) [Fink 1979]	$\beta_S = 1/(\rho_l v^2)$ where $v = 2660.7 - 0.3766\,T - 9.0356 \times 10^{-5} T^2$ $\beta_S = \beta_{S,m}\left(1 + \frac{\theta}{b}\right)/(1 - \theta)$ Where $\beta_{S,m}$= adiabatic compressibility at the melting point, 1.717×10^{-4} MPa^{-1} $\theta = \frac{T - T_{liq}}{T_C - T_{liq}}$	371K ≤ T ≤1700K 1700K <T < T_C	L-14a L-14b

*See Table 4 of [Morita 1998b] for the values of the fitted constants and other numerical values provided. Note that ρ_{liq} is calculated from v_{liq} in this reference.

Table 2-2 Vapor Sodium Thermophysical Properties Modeled

Property [reference]	Equation	Range	Equation #
Density (kg/m^3) [Morita 1998a]	$\rho_g = \rho_C / \exp\left[\sum_{i=1}^{6} B_{G,i}(1-\tau)^{n_{G,i}}\right]$ where $B_{G,i}$ = fitting constants as a function of i $n_{G,i}$ = fitting exponents as a function of i $\tau = T/T_C$		V-1
Specific liquid volume (m^3/kg) [Morita 1998a]	$v_l = v_{liq}\left[1 + \sum_{i=1}^{3} d_{L,i}(u_L - 1)^i\right]$	$e_{liq} < e \le d_{L,4}e_{liq}$	V-2a
	$v_l = v_C\left[1 + d_{L,5}(1-\xi_L)^{0.5} d_{L,6}(1-\xi_L)^2\right]$ where $u_L = e/e_{liq}$ $\xi_L = e/e_C$ $d_{L,i}$ = fitted constants, i = 1 to 6	$d_{L,4}e_{liq} < e \le e_C$	V-2b
Specific internal energy (J/kg) [Morita 1998a]	$e_{vg} = C_{VG}(T - T_{liq}) + e^D_{liq} + \frac{y_B R \cdot T}{1+y_B}\left(1 - \frac{d_{G,2}}{T}\right) - \frac{A_{G,2}(1-A_{G,4})\psi}{A_{G,3}}\ln(1 + A_{G,3}\rho_v)$ where $y_B = \frac{1 + 2x_G - \sqrt{1+8x_G}}{2(x_G - 1)}$ $x_G = \frac{\rho_g \cdot k_G R \cdot T}{1 - A_{G,1}\rho_g}$ $k_G = \exp\left(d_{G,1} + \frac{d_{G,2}}{T}\right)$		V-3
	$\psi = \left(\frac{T}{T_C}\right)^{A_{G,4}}$	$T < T_C$	
	$\psi = 1$	$T \ge T_C$	
Specific enthalpy (kJ/kg) [Fink 1995]	$H_g = H_l + \Delta H_g$	$T < 2000K$	V-4a
	$H_g = H_{AVG} + 0.5\,\Delta H_g$	$T \ge 2000K$	V-4b

Property [reference]	Equation	Range	Equation #
	where $\Delta H_g = 393.37\left(1-\frac{T}{T_C}\right)+4398.6\left(1-\frac{T}{T_C}\right)^{0.29302}$ $H_{AVG} = 2128.4 + 0.86496 \cdot T$		
Entropy (J/kg-K) [Young 1977, Blink 1982]	$\frac{dH_z}{dT} = \frac{E-A}{T} = Nk\left[\frac{3}{2} + \ln\left(\frac{\frac{e}{\left(N\cdot\left(h_p\sqrt{N/(2\pi k)}\right)^3\right)}T^{1.5}}{\rho_v}\right) + \frac{1}{3}(n+4)Q\cdot(\rho_v\delta)^{n/9}\left(\frac{\epsilon}{kT}\right)^{1/3}\left(\frac{\epsilon}{kT}\right)^{1/3}\right]$ See Eq.(L-10) for the definitions of the variables, except replacing ρ_l with ρ_g.		V-5
Derivative of specific energy (J/kg-K) [Fink 1979]	$\left(\frac{de_g}{dT}\right)_{SAT} = \left(\frac{dH_g}{dT}+\frac{dH_v}{dT}\right)/mw_{Na}$	$T < 1644K$	V-6a
	$\left(\frac{de_g}{dT}\right)_{SAT} = \frac{18.525 - 6.9936\cdot(1-u)^{-0.6777}}{mw_{Na}}$	$1644K \le T \le T_C$	V-6b
	Where $\frac{dH_g}{dT} = 35.206 - 2\cdot 7.0513\times10^{-3}T + 3\cdot 2.5711\times10^{-6}T^2 + \frac{124280}{T^2}$		V-6c
	$\frac{dH_g}{dt} = \frac{4186}{T_C}(5.557012 - 0.4\cdot 31.25992\cdot u^{-0.6})$ $u = 1 - T/T_C$ $mw_{Na} =$ molecular weight of sodium		V-6d
Specific heat at constant volume (J/kg-K) [Morita 1998a]	$C_V = C_{VG} - \frac{A_{G,2}(1-A_{G,4})\psi'}{A_{G,3}}\ln(1+A_{G,3}\rho_g) + \frac{y_B R}{1+y_B}\left\{1+\frac{1-y_B}{1+3y_B}\left(1-\frac{d_{G,2}}{T}\right)\right\}$ where		V-7

Property [reference]	Equation	Range	Equation #
	$\psi' = \frac{A_{G,4}}{T_C}\left(\frac{T}{T_C}\right)^{A_{G,4}-1}$ $\psi' = 0$	$T < T_C$ $T \geq T_C$	
Specific heat at constant pressure (J/kg-K)	$C_P = C_V + \frac{T \cdot {\alpha_P}^2}{\rho_{gv}\beta_T}$		V-8
Volumetric thermal expansion coefficient (K^{-1}) [Fink 1979]	$\alpha_p = \alpha_{SAT}\left(1 - \gamma_{SAT}/\gamma_g\right)^{-1}$ where $\alpha_{SAT} = \left(\frac{d\rho_g}{dT}\right)_{SAT}/\rho_g$ $\gamma_{SAT} = \left(\frac{dP}{dT}\right)_{SAT}$		V-9
Thermal-pressure coefficient (MPa/K) [Fink 1995]	$\gamma_g = \left(-\frac{G_{a,1}}{T^2} + \frac{G_{a,3}}{T} + G_{a,4} - 2 \cdot e \cdot T\right) EXP\left(G_{a,1} + \frac{G_{a,2}}{T} + G_{a,3}\ln(T) + G_{a,4} \cdot T + G_{a,5} \cdot T^2\right)$ where $G_{a,1} = 8.35307, G_{a,2} = -12905.6, G_{a,3} = -0.45824, G_{a,4} = 2.0949 \times 10^{-3}$, *and* $G_{a,5} = -5.0786 \times 10^{-7}$	T < 1600K	V-10a
	$\gamma_g = \gamma_g^C - 2.5696 \times 10^{-3}\left(1 - \frac{T}{T_C}\right)^{0.5} + 3.5628 \times 10^{-5}\left(1 - \frac{T}{T_C}\right)$ Where $\gamma_g^C = 0.046893$	T > 1600K	V-10b
Isothermal compressibility (1/Pa) [Fink 1979]	$\beta_T = \alpha_p/\gamma_g$		V-11

Table 2-3 Additional EOS Properties Modeled

Property [reference]	**Equation**	**Range**	**Equation #**
Critical temperature (K) [Fink 1979]	$T_C = 2509.46$		A-1
Critical density (kg/m^3) [Fink 1979]	$\rho_C = 214.1\ \mathrm{kg/m^3}$		A-2
Critical specific internal energy (J/kg)	$e_C = 4.13028 \times 10^6$		A-3
Specific internal energy of infinitely dilute vapor (J/kg)	$e_{liq}^D = 4.67732 \times 10^6$		A-4
Specific heat at constant volume for dilute vapor (J/kg-K)	$C_{VG} = 399.177$		A-5
Saturation temperature (K) as a function pressure[Morita 1998a]	$T_{SAT} = \dfrac{1}{A_{SAT,1} + \sum_{i=2}^{4} A_{SAT,i}\left(\ln(P)\right)^{i-1}}$ where $A_{SAT,i}$ = fitted constants, i = 1 to 4		A-6
Saturation temperature (K) as a function specific internal energy [Morita 1998a]	$T_{SAT} = T_{liq}\left[1 + \sum_{i=1}^{3} A_{L,i} \cdot u^i\right]$	$e_{liq} < e \leq A_{L,4} e_{liq}$	A-7a
	$T_{SAT} = T_C \cdot \left[1 - A_{L,5} w^2 - A_{L,6} w^3\right]$ where $u = \left(e/e_{liq} - 1\right)$ $w = (1 - e/e_C)$ $A_{L,i}$ = fitted constants, i = 1 to 6	$A_{L,4} e_{liq} < e \leq e_C$	A-7b
Saturation vapor pressure (Pa) [Morita 1998a]	$p_g = \exp\left[b_{L,1} + b_{L,2} T + \dfrac{b_{L,3}}{T} + b_{L,4} \ln\left(\dfrac{T}{T_C}\right)\right]$ where $b_{L,i}$ = fitted constants, i = 1 to 4 T = liquid temperature	$T \geq T_{liq}$	A-8

3 MELCOR CODE MODIFICATION AND TESTING

3.1 Code Modification

MELCOR 2.1 was modified (as Revision 5311) to add liquid metal fluid properties and EOS. A separate MELCOR branch has been created for this modification (Revision 5311). However, a large number of code modifications have been done since this revision, merging to current MELCOR trunk is in progress. Section 5 will discuss the significant improvement of MELCOR 2.1 in FY15, which altered the implementation of the sodium models into MELCOR.

Following the same general code modification performed by INL for the lithium fluid replacement in MELCOR [Merrill 2000], the water EOS and other property function and table lookup in MELCOR must be re-directed to the appropriate routines or tabular look-up for the sodium properties. Significant effort was made to structure the supporting code changes and input requirements for the SIMMER database and FSD database so that they were as similar as possible to simplify code maintenance requirements and user input. To activate the liquid metal capability, an unformatted file must be present in order to activate a particular fluid's equation of state for the simulation. An array of 20 (NATNAM_Eos(20)) is set up for the fluid type and an array (FILNAM(20)) is set up for each of the corresponding file name for the fluid type. The corresponding fluid type and file name are presented in Table 3-1. As shown in this table, there are unfilled slots in the array for future expansion. Without any matching filename in the working directory where the MELGEN and MELCOR input files are located, the default MELCOR fluid, water will be used. Fluid 1, water as shown in Table 3-1, is for the water properties provided by the FSD data set. Fluid 20, sodium, is for the SIMMER data set. Unlike the FSD data set, there is no need to input property data files for the SIMMER data set, though a dummy file is required to designate the use of the SIMMER model in a manner consistent to the FSD formulation.

Table 3-1 Corresponding Input Filename to Fluid Identifier

Fluid Material [#]	File Name	Fluid Material	File Name	Fluid Material	File Name
H2O [1]	TPFH2O	H2 [2]	TPFH2	Li [3]	TPFLI
K [4]	TPFK	He [5]	TPFHE	N2 [6]	TPFN2
Na [7]	TPFNA[1]	NaK [8]	TPFNAK	LiPb [9]	TPFLIPB
FLIBE [10]	TPFFI	Na [20]	SIMMER[2]		

[1]Refer to FSD data set
[2]Refer to SIMMER data set

In order for MELCOR to model a fluid other than water, new subroutines for calculating the equations of state must be added. Table 3-2 shows the list of the files required to be modified to include both the FSD and SIMMER sodium data sets. In this table, a brief description is provided to include what is being changed and added to the source code. As shown in Table 3-2, the majority of the files modified resided in the EOS package of the MELCOR source. The major changes were in module M_H2O, where the transport property routines of the FSD data and initialization of the working fluid, other than the standard MELCOR water is done.

A number of new files are included in MELCOR, as shown in Table 3-3. As shown in this table, the new files are color coded to identify which database these files belong to. For the FSD data set, the module FSD_EOS contains many routines and functions that account for the property data, in addition to those thermodynamic data that are provided in the input data file (see Section 2.1 for the details of the NaLibrary program). For the SIMMER data set, both NALIQUID and NAVAPOR modules contain the majority of the property correlations (see Section 2.2).

Table 3-2 MELCOR Files Modified

Package/File Name	Description
COR/COR_CORABS [s]	Call TH2O_VAPOREMISS for the vapor emissivity of the fluid
CVH/CVH_GenerateDB [s]	Add use statement of M_MFLDATA and M_MFLBLS in various subroutines to allow diagnostic messages, redefine saturation temperature of the fluid by adding 100 to triple point of the fluid
CVH/CVH_CVHDBE [s]	Provide fluid name to outputs
CVH/THYDR_FLCOK[s]	Add multi-fluid model, and define Henery-Fauske Subcooled pool routine (GCSUB) for the FSD property model.
CVH/THYDR_FLCOK_NEWMODEL [s]	Add multi-fluid model, and define Henery-Fauske Subcooled pool routine (GCSUB) for the FSD property model.
EOS/THYDR_CVTHRM[s]	Add RHOL values for different fluids – multi-fluid model
EOS/EOS_CVTNEQ[s]	Add variable H2OPHX, and other variables for multi-fluid capability
EOS/EOS_CVTNQE[m]	Add variable H2OPHX, and other variables for multi-fluid capability
EOS/CVH_CVTSAT[m]	Redefine PMAX1,PMIN1,TMIN for multi-fluid properties
EOS/TEOS_CVTSVE[s]	Add variable H2OPH as passing variable, and other variables for multi-fluid capability
EOS/CVH_CVTWGE[m]	Add variable H2OPHX, and other variables for multi-fluid capability
EOS/H2O_H2ODBE[s]	Minor change to H2O package call summary to output
EOS/TH2O_H2OEPT[s]	Add multi-fluid capability to allow single phase properties return, in addition to water
EOS/TH2O_H2OEST[s]	Add multi-fluid capability to allow the determination of the thermodynamic state of the fluid
EOS/TH2O_H2OESU[s]	Redefine RU variable for fluid other than water
EOS/TH2O_H2OSAT[s]	Add multi-fluid capability to allow the determination of the saturation properties of fluid, in addition to water
ESF/TCND_CVTWGE[s]	Add variable H2OPHX, and other variables for multi-fluid capability
EXEC/MEG_RW	Add call to INIT_MFLUIDS[s] for the multi-fluid capability to PREPARETOINPUT[s], and add multi-fluid capability in other routines in this file.
EXEC/EXEC_MXXPBD[s]	Add call to INIT_MFLUIDS[s] for the multi-fluid capability
FP/FP_PLOTMANAGER[s]	Add plotting variables for multi-fluid for the FP Package
HS/THS_HSBOIL[s]	Add the call to TH2O_SURFTENSION[f] for the multi-fluid capability
HS/THS_HSCNDS[s]	Add the call to TH2O_SURFTENSION[f] for the multi-fluid capability
HS/THS_HSDMTC[s]	Add pass variable DIFFUS for the mass transfer coefficient calculations – multi-fluid, in addition to water
HS/THS_HSLHX[s]	Add call to TH2O_SURFTENSION[f] for multi-fluid capability
HS/THS_HSTRAN[s]	Add multi-fluid option key, use M_MFLBLS[m], FSD parameters, mass transfer calculations for other fluids other than water.
M_ARGUMENTS[m]	Add USENQE = 0 from M_H2O[m] for the multi-fluid capability
M_CONST[m]	Modified variable constants (mainly for critical conditions) for the multi-fluid capability
M_EXECRTN[m]	Modified the use statement for M_ARGUMENT[m] usage
M_H2O [m]	Add multi-fluid option key, and add use statement for M_MFLDATA, M_MFTBLS. Modifications to a number subroutines/functions in the module to add functionality of the multi-fluid capability: TH2O_H2O1PH[s], TH2O_H2O1PH[s], TH2O_SPECIFICVOL[f], TH2O_WATERHEATCONDUCT[f], TH2O_WATERVISC[f], TH2O_SURFTENSION[f], tH2O_WATERSATTEMP[f], TH2O_VAPOREMISS[f], TH2O_VAPORCOMPRESS[f], TH2O_VAPORCP[f], TH2O_VAPORHEAT[f], TEOS_H2O2PL[s], INIT_MFLUIDS[s] and FLUIDEOS[s]. INIT_MFLUID allows the selection of the fluid data set (nfluid=0, standard

	MELCOR fluid is used; nfluid=20, use SIMMER sodium data set; and else use FSD data set that could be more than sodium). FLUIDEOS is only used for the FSD data set.
M_H2OD1[m], M_H2OD2 [m], M_H2OD3[m], M_H2OD4[m]	Replace minor Fortran directive
M_H2O_VARS[m]	A module contains water data moved from original M_H2O module.
M_NCG[m]	Add multi-fluid check and related changes, including M_MFTBLS and M_MFLDATA use statements. Add "use FSD_EOS" statement and other calls to tNCG_GetHeatConduct[f], tNCG_GetVisc[f], and tNCG_DefProps[f],
RN1_ENTRAINMENT[s]	Add multi-fluid model, replacing surface tension correlation with a call to tH2O_SurfTension[f]
RN1_RN1HYG[s]	Add multi-fluid model, replacing surface tension correlation with a call to tH2O_SurfTension[f]
RN2_RN2DFP[s]	Add multi-fluid model, replacing surface tension correlation with a call to tH2O_SurfTension[f]

Table 3-3 New MELCOR Files

File Name*	Description
M_H2O_VARS[m]	A module that contains data sets from the original M_H2O module. It utilizes M_MFLDATA [m] and M_MFTLS[m], and re-do minor FORTRAN directories.
M_MFLDATA [m]	A module to deal with multiply fluid properties. Define MATNAM_Eos array to hold the name of the fluid, and the corresponding FILNAM array for each fluid defined. A number of empty slots can be filled for future development.
M_MFTBLS[m]	Set up the multi-fluid variables and identify the number of fluids, and data size for mainly the INL data usage.
M_Na[m]	A module to deal with sodium properties from BRISC: NA1PH[s], NA2PD[s], NA_INITTABLES[s], and NA2PL[s].
M_STCOM [m]	Replace common block STCOM that FSD_EOS shared
M_STD2XC [m]	Replace common block STD2XC that FSD_EOS shared
M_STH2XC[m]	Replace common block STH2XC that FSD_EOS shared
NA1[m]	A module for thermodynamic properties of sodium: P_VAPOR[f], DPDR_VAPOR[f], PV_NA[f], DPRHO[f], DPTEMP[f], EVAPFUN[f], CVVAPFUN[f], TEMPDENSATL[f], CUBIC[s], DTDP[f], DVDP[f], DRHOVDT_SAT[f], DRHOLDT[f], and DPDVL[f].
NAEOS[m]	A module that deals with critical, saturation and other thermal related properties for sodium. This module contains a number of functions: PSATFUN[f], DPDTSATL[f], GAMMA_SAT[f], TSAT33[f] and TSATP[f]. It is used by NALIQUID and NAVAPOR modules.
NALIQUID[m]	A module describes densities, heat capacity and other thermodynamic properties for liquid sodium. This module contains a number of functions that are function of temperature: DENL[f], CV_LIQ[f], ALPHA_LIQ[f], CP_LIQ[f], ALPHA_SAT[f], DRHOLDT_SAT_ANALYTIC[f], ENTHL[f], ENTHLOLD[f], ENTHLNEW[f], ENTHLINV[f], DENTH_LIQUID[f], CSAT_L[f], SLIQUID[f], DELDT_SAT_ARGONNE[f], BETA_T[f], BETA_S[f], DPR_LIQ[f] and DPT_LIQ[f].
NAVAPOR[m]	A module describes densities, heat capacity and other thermodynamic properties for vapor sodium. This module contains a number of functions that are function of temperature: CV_VAP[f], DENV[f], ENTHV[f], ALPHA_SATV[f], DVGDT_SAT2[f], DRGDT_SAT[f], BETA_T_VAP[f], ALPHAP[f], GAMMA_V[f], GAMMA_V_1600[f], SVAPOR[f], VS[f], DP_VAPOR_DT[f], DP_VAPOR_DV[f], DEVDT_SAT_ARGONNE[f], DH_NA_V[f], H_NA_L[f], HVAP_NA[f], H_NA_AVG[f], H_NA_V[f], CSAT_V[f], DP_VAPOR_DR[f], CP_VAP[f] and DPT_VAP[f].
FSD_EOS [m]	A main module for the INL data set. It contains a number of routines and functions that are used for the multi-fluid capability. ICMPNX [s] provides access through input data file. ICMPNXSIMMER [s] provides access few data in the routine. STREAD routine reads and initializes the data tables. Water property routines include STH2XL, STH2X3, STH2X4, STRTP, STRSAT, STRPX and STRX. Non-water property routines include STUPX4 and STRPX. Other specified routines are: SURFTN-surface tension, THCOND-thermal conductivity, VISCOS-dynamic viscosity, and GCSUB-critical mass flux for a given fluid.

*The bracket next to each file indicates: m = module, s = subroutine and f = function

**Red colored file is for FSD data set and Blue colored file is for SIMMER data set.

Table 3-4 New Control Functions Defined for the Working Fluid

Control Function Parameter	Description
CVH-CVP(NameCV)	Specific heat at constant volume for liquid in control volume NameCV (units = J/kg/K)
CVH-CPP(NameCV)	Specific heat at constant pressure for liquid in control volume NameCV (units = J/kg/K)
CVH-CVA(NameCV)	Specific heat at constant volume for vapor in control volume NameCV (units = J/kg/K)
CVH-CPA(NameCV)	Specific heat at constant pressure for vapor in control volume NameCV (units = J/kg/K)
CVH-BETATP(NameCV)	Isothermal compressibility for liquid in control volume NameCV (units = 1/Pa)
CVH-BETATA(NameCV)	Isothermal compressibility for vapor in control volume NameCV (units = 1/Pa)
CVH-SP(NameCV)	Specific entropy for liquid in control volume NameCV (units= J/kg)
CVH-SA(NameCV)	Specific entropy for vapor in control volume NameCV (units = J/kg)
CVH-ALPHAA(NameCV)	Volumetric thermal expansion for vapor in control volume NameCV (units = 1/K)
CVH-ALPHAP(NameCV)	Volumetric thermal expansion for liquid in control volume NameCV (units = 1/K)
CVH-THCP(NameCV)	Thermal conductivity for liquid in control volume NameCV (units = W/m-K)
CVH-THCA(NameCV)	Thermal conductivity for vapor in control volume NameCV (units = W/m-K)
CVH-VISCP(NameCV)	Viscosity for liquid in control volume NameCV (units = Pa-s)
CVH-VISCA(NameCV)	Viscosity for vapor in control volume NameCV (units = Pa-s)

3.2 Testing and Results

To support the developmental phase of the liquid metal property implementation, a set of three simple test problems were created. Each test was selected to test the model implemented, except the first test which is to ensure the water property has not been altered for the LWR application:

(a) Water test to demonstrate that liquid metal properties implemented would not affect the current water properties modeled when liquid metal is not invoked.
(b) Sodium SIMMER (BRISC) test – Testing the sodium properties in SIMMER and SAS4A database.
(c) Sodium FSD test – Testing the sodium properties in FSD database. Note that much of the thermodynamic properties are generated using the NaLibrary program.

Except the first test, the comparison to the second and third tests is done by using tabular data and correlations from various sources and codes.

A simple test problem which contains a single test volume with a working fluid (water or sodium) in a closed system is subjected to external enthalpy sources. This test is particularly challenging because it covers a very broad range of test conditions extending from very low pressure near freezing point to near critical pressures. Although the test problem did not run to completion for all three cases due to small time steps, the resulting plots from these runs demonstrate that the addition of working fluid other than water is possible for MELCOR. Note that this test problem was created to test the extreme conditions of the fluid properties. In the future, a refinement of the test problem will be done to represent the physical conditions encountered in severe accident situations.

3.2.1 Water Test Case

A test problem was created to ensure that the default water properties have not been modified. This test problem was terminated by MELCOR at about 1.7×10^4 seconds due to a very small subcycle timestep due to the reach of the supercritical conditions for water. Despite this issue, the results of this water test problem are presented in the plots given below. As indicated before, the thermodynamic condition of the test problem is at saturation. Figure 3-1 shows the water mass as a function of time. As the control volume heats up, the liquid mass decreases while the vapor mass increases. Figure 3-2 plots the pool and atmosphere entropies. Figure 3-3 plots the pressure versus temperature for the problem. As indicated in this figure, MELCOR predicts the saturation temperature up to the point near the supercritical temperature before the code was terminated. Thus this problem is to test the extreme conditions of the coolant properties.

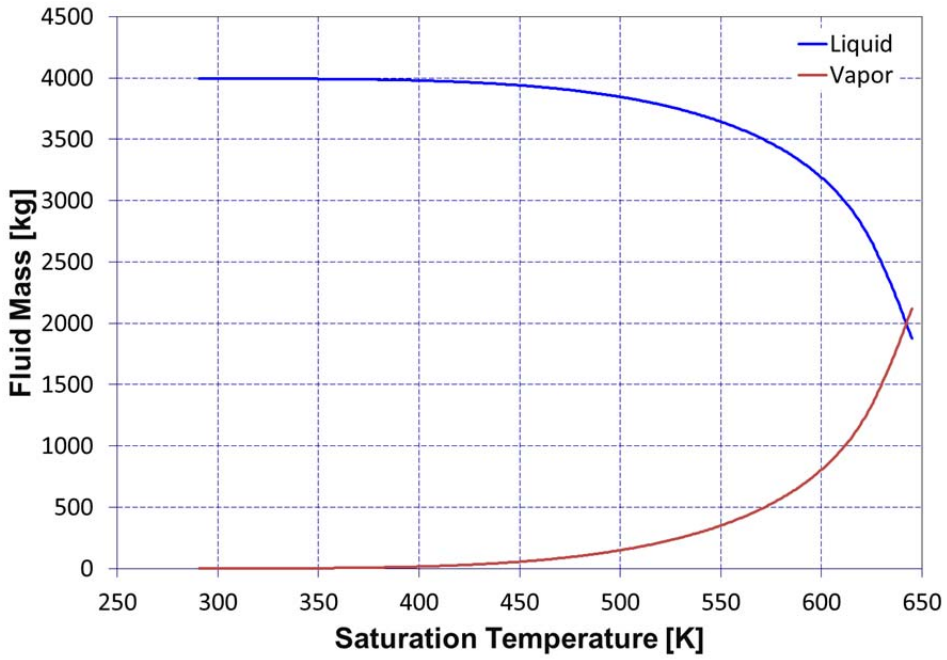


Figure 3-1 Calculated Water Mass in Problem

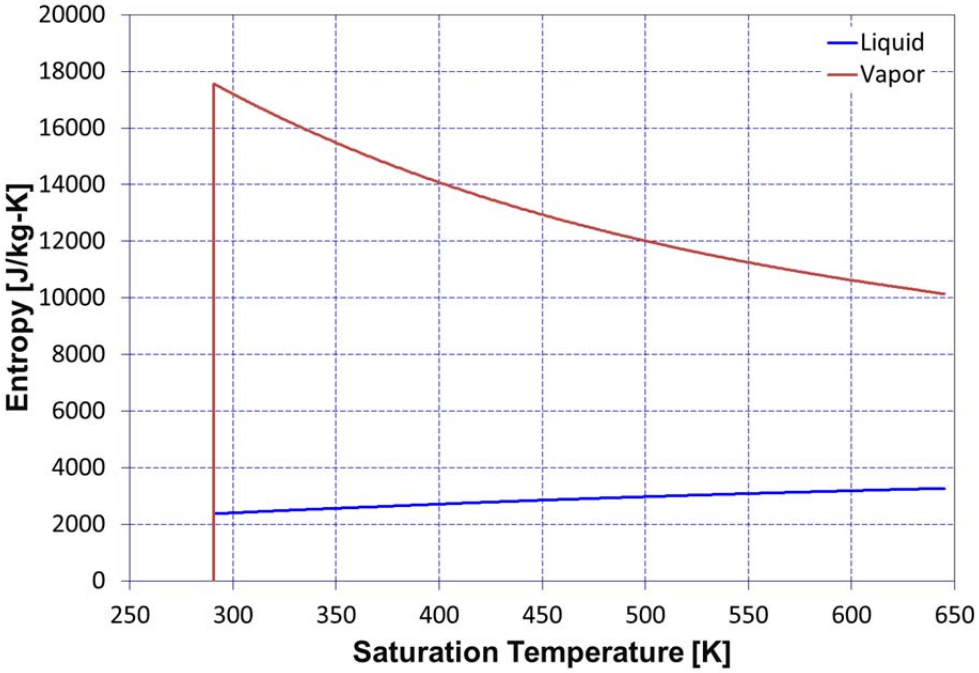


Figure 3-2 Entropy versus Saturation Temperatures

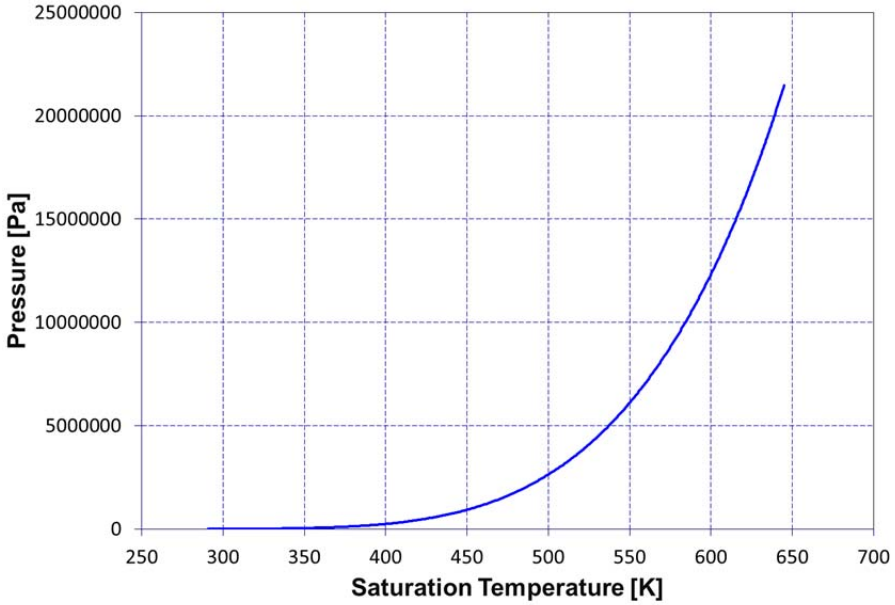


Figure 3-3 Pressure versus Temperature

3.2.2 Sodium

The implementation of the EOS and other thermophysical properties of sodium for the SIMMER/SAS4A and FSD data sets were tested with the previously described test problem. Note that original SIMMER/SAS4A input was aborted ungracefully at about 3×10^4 seconds. It was thought that the rate of enthalpy sources introduced was too large. So, it was scaled down from 4×10^5 J/kg to 1×10^5 J/kg at a specific time interval. The following figures show the results of the revised input for the SIMMER/SAS4A dataset as indicated as SIMMER, although the run was continued beyond 3.8×10^4 seconds, but at a very small timestep of 10^{-7} second. Figure 3-4 shows the sodium mass versus saturation temperatures. As the temperature increases, the sodium liquid vaporizes as shown in this figure. The corresponding entropies for the liquid and vapor sodium are shown in Figure 3-5. The pressure-temperature plot is given in Figure 3-6. The third test was conducted for the FSD data set. Figure 3-7 shows the sodium mass versus saturation temperature for this data set. This test was stopped much earlier than the test problem for the SIMMER/SAS4A data set. Figure 3-8 shows the entropy versus temperature for the FSD data set while Figure 3-9 plots the pressure as a function of the saturation temperature.

To benchmark the implemented sodium properties for both the SIMMER/SAS4A and FSD data sets, comparisons to the existing tabular data available from various SIMMER/SAS4A sources [Fink 1979 and Fink 1995] and FSD sources from equations given in this report and those correlations documented in the NaLibrary Program. The rest of the plots shown in this section contain comparison data from various references, denoted as symbols with MELCOR calculated values, denoted as line curves. Figure 3-10 plots the saturation temperature-pressure curves with the comparison of the available data from various references. As shown in this figure, calculated values for both data sets match closely with the references. As indicated earlier, both FSD and SIMMER/SAS4A runs were stopped before the end of the problem. For the FSD test case, it stops at a saturation temperature of about 1500 K and 1×10^6 Pa pressure as shown in this figure. On the other hand, the SIMMER test case runs to about 2500 K saturation temperature and 2.2×10^7 Pa pressure. Figure 3-11 presents the density versus temperatures. For the liquid density, it is necessary to correct the amount of voids or bubbles in the pool. As shown in this figure, the calculated densities for both liquid and vapor follow similarly with the reference values. Figure 3-12 shows the same plots as in Figure 3-11, except the SIMMER data was generated using the original SIMMER input with higher enthalpy sources (4×10^5 J/kg, instead of 1×10^5 J/kg). In terms of the liquid specific heat, both Figure 3-13 and Figure 3-14 plot this property at constant pressure and volume, respectively. These variables are particularly important in determining the stability of transient conditions. As shown in these figures, the calculated SIMMER data set closely matches to the reference values. Both Figure 3-15 and Figure 3-16 show the vapor specific heat at constant pressure and volume, respectively. As shown in these figures, the calculated SIMMER data matches closely at lower temperatures. At high temperatures, MELCOR underestimates these properties for the SIMMER/SAS4A data set. The calculated MELCOR FSD data do not match to the SIMMER reference at all. The primary reason for this is because of the soft-sphere model for the FSD vapor specific heat correlation where the differentiation of the Helmholz free energy [Young 1977, Blink 1979] may not be suitable for low pressure and temperature conditions.

In terms of transport properties, Figure 3-17 and Figure 3-18 show the viscosity and thermal conductivity curves, respectively. As shown in these figures, MELCOR calculated values are closely matched to the references.

For further analysis of the implemented SIMMER/SAS4A data set in MELCOR, the next three plots (Figure 3-19, Figure 3-20 and Figure 3-21) for isothermal compressibility, volumetric thermal expansion, and heat of vaporization, respectively. As shown in Figure 3-19, MELCOR predicts closely with the reference values. Similarly MELCOR predicts closely for the volumetric thermal expansion, except at lower temperatures. For the heat of vaporization, MELCOR calculated values are closely matched at lower temperatures, but slightly underestimate the values at higher temperatures.

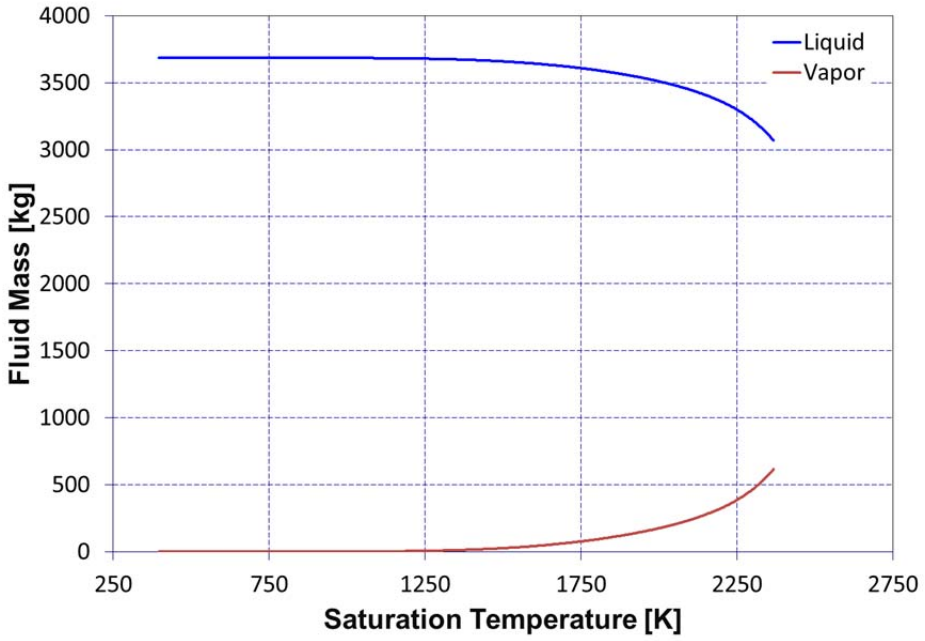


Figure 3-4 Fluid Mass versus Temperature for the SIMMER/SAS4A Database

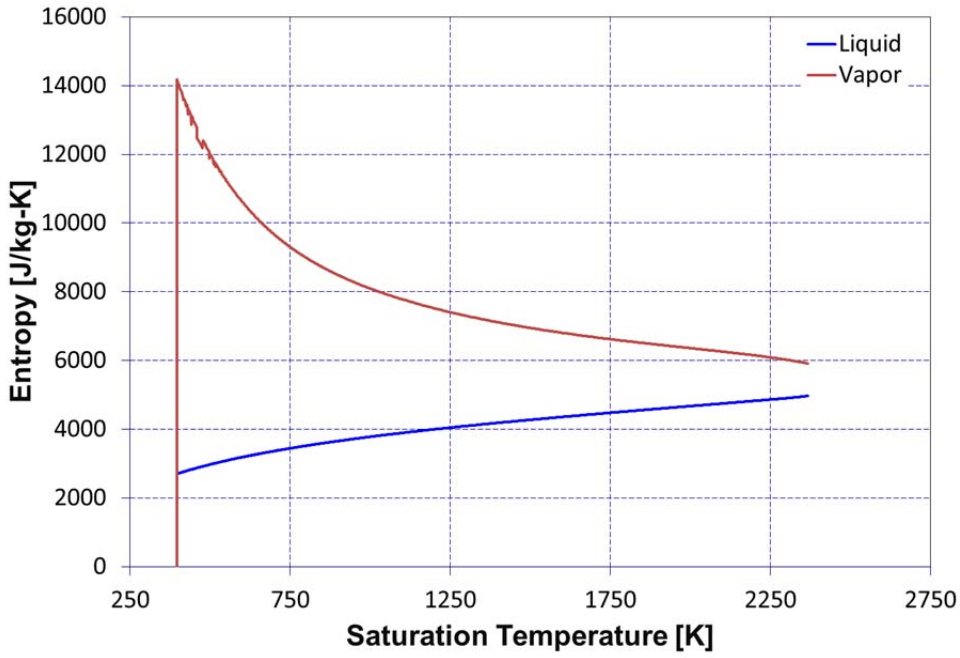


Figure 3-5 Entropy versus Temperature for the SIMMER/SAS4A Database

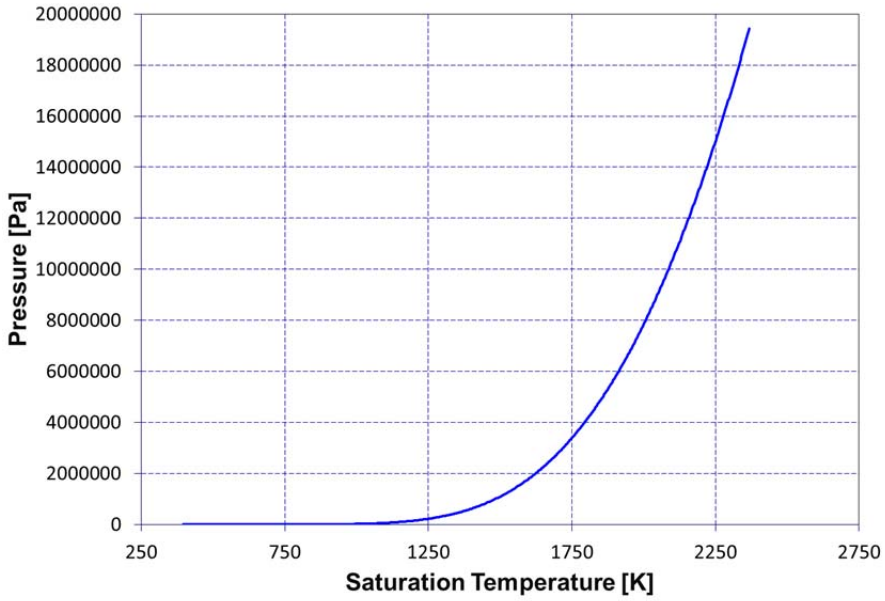


Figure 3-6 Pressure versus Temperature for the SIMMER/SAS4A Database

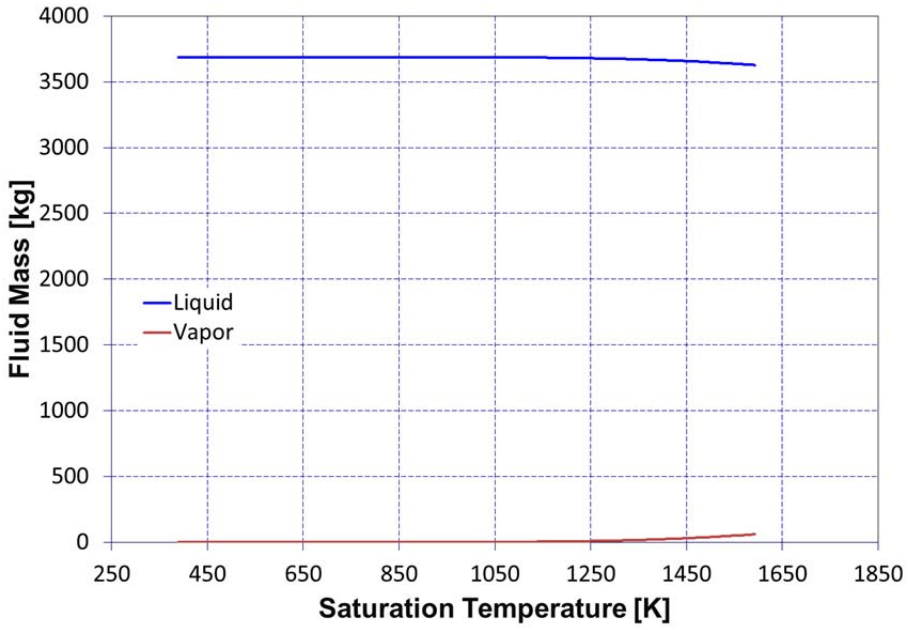


Figure 3-7 Fluid Mass versus Temperature for the FSD Database

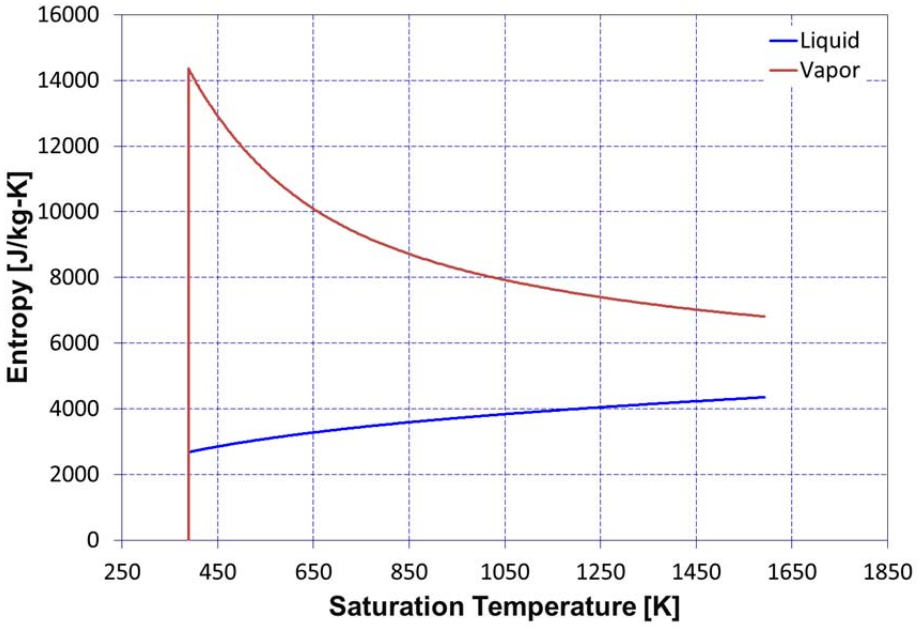


Figure 3-8 Entropy versus Temperature for the FSD Database

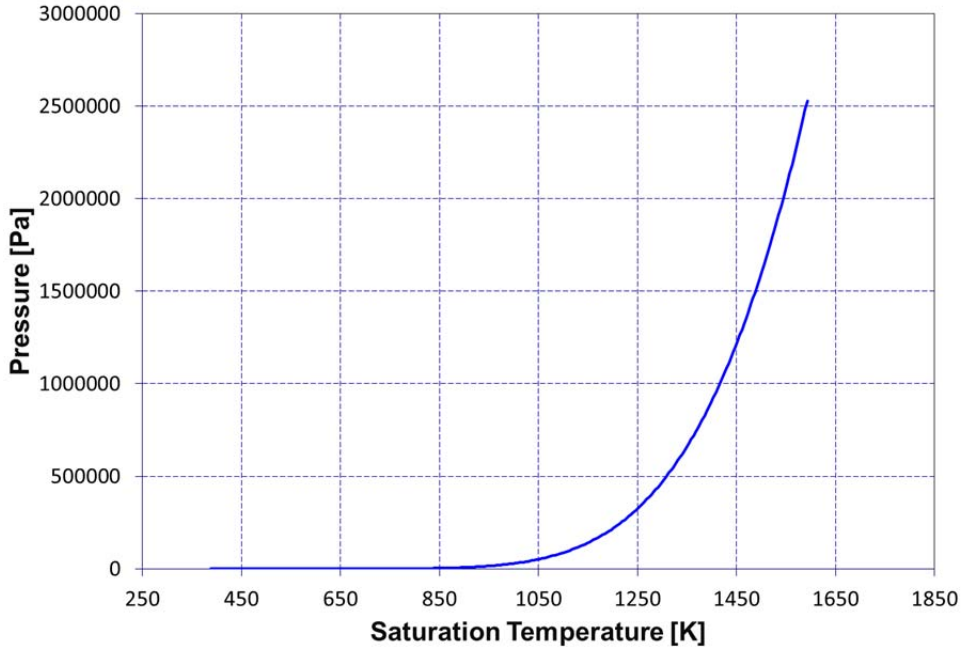


Figure 3-9 Pressure versus Temperature for the FSD Database

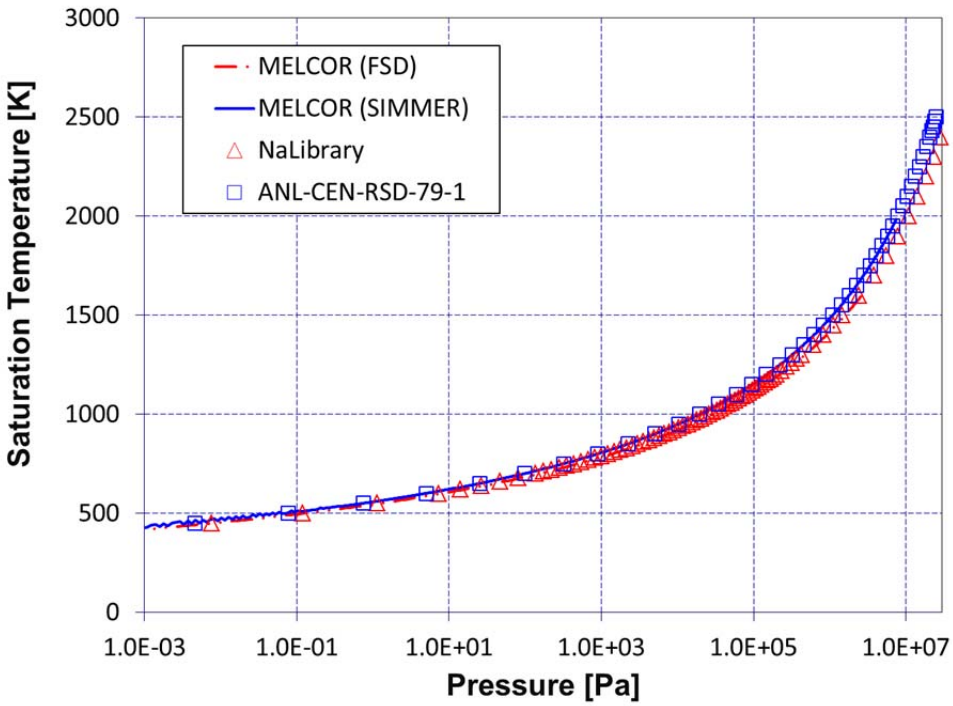


Figure 3-10 Saturation Temperature versus Pressure

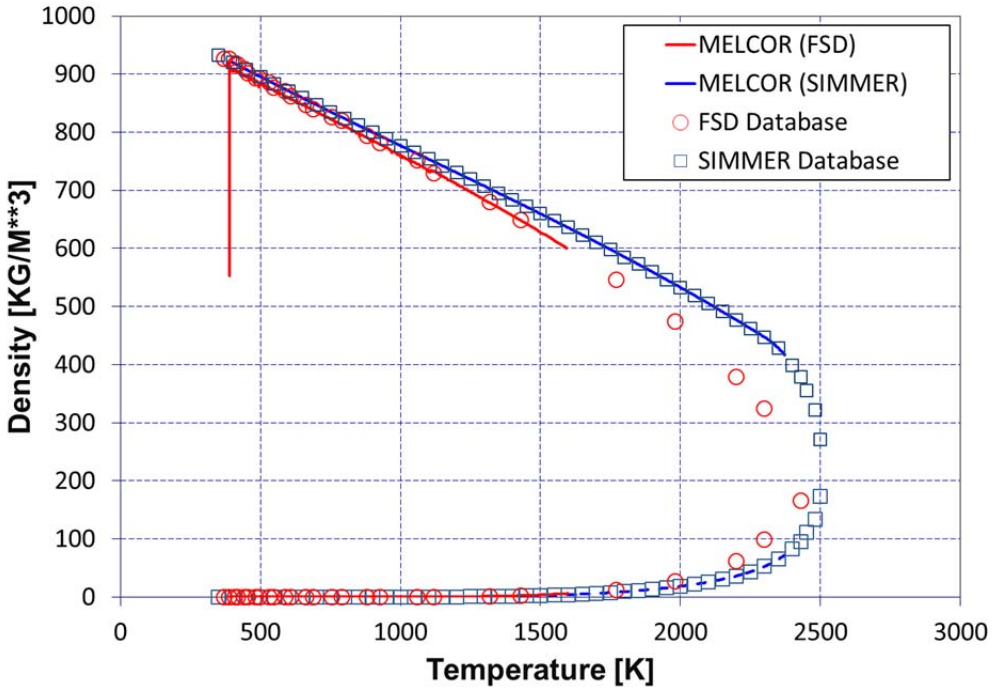


Figure 3-11 Density versus Temperature

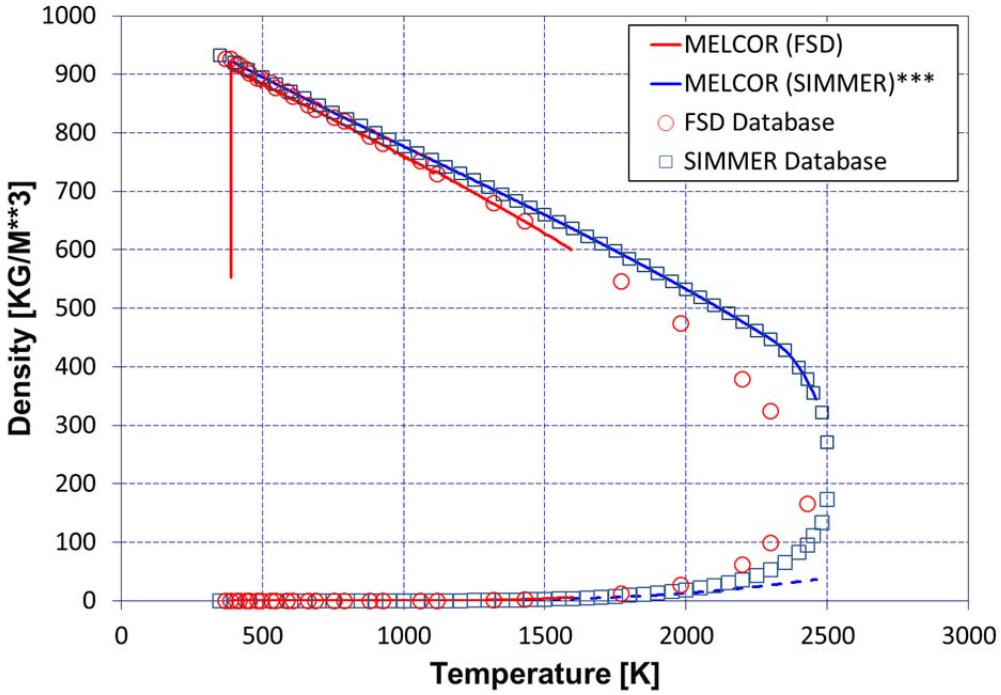


Figure 3-12 Density versus Temperature ***Input with a higher enthalpy source for the SIMMER/SAS4A case.

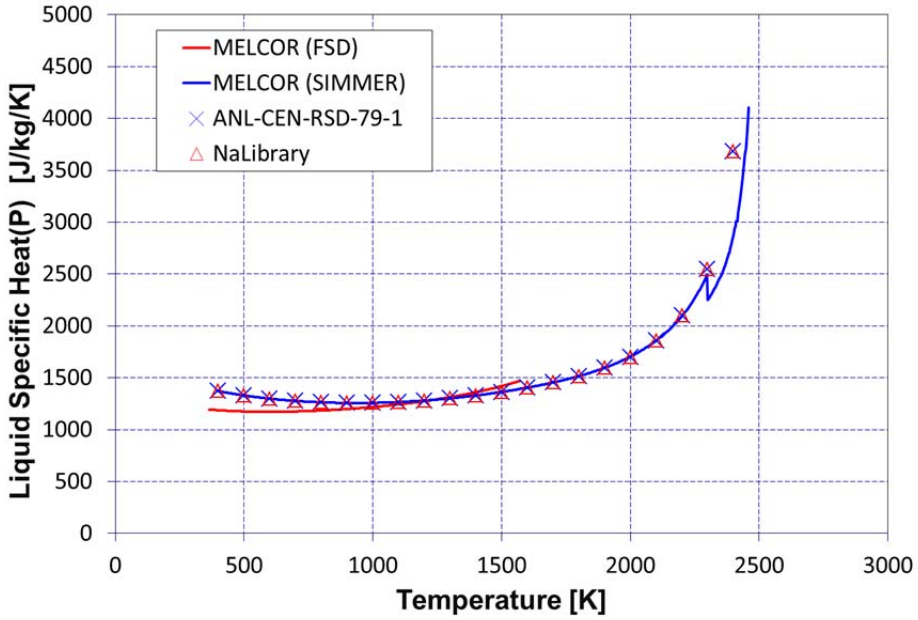


Figure 3-13 Liquid Specific Heat at Constant Pressure

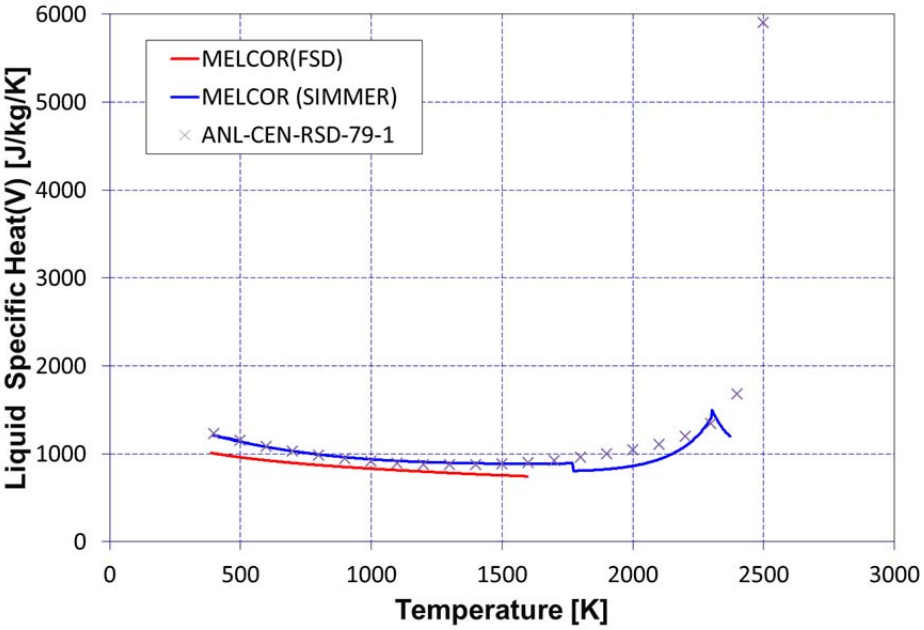


Figure 3-14 Liquid Specific Heat at Constant Volume

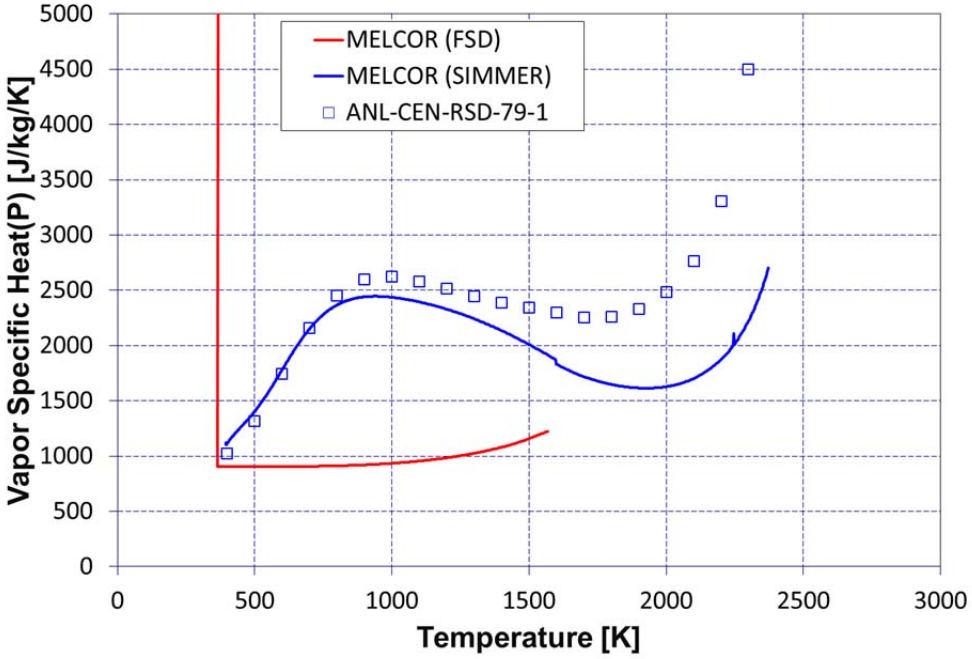


Figure 3-15 Vapor Specific Heat at Constant Pressure

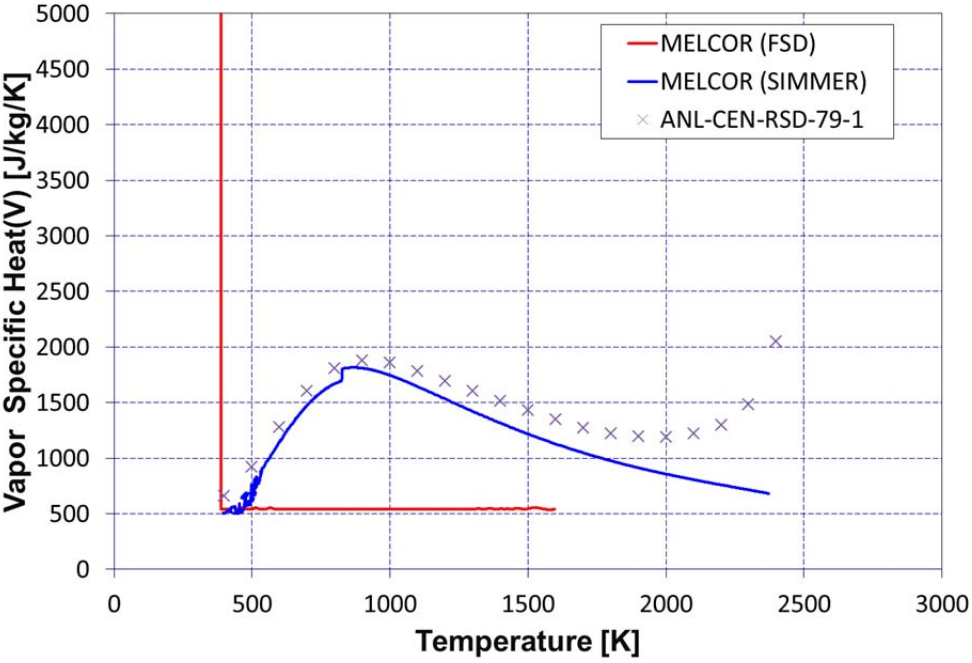


Figure 3-16 Vapor Specific Heat at Constant Volume

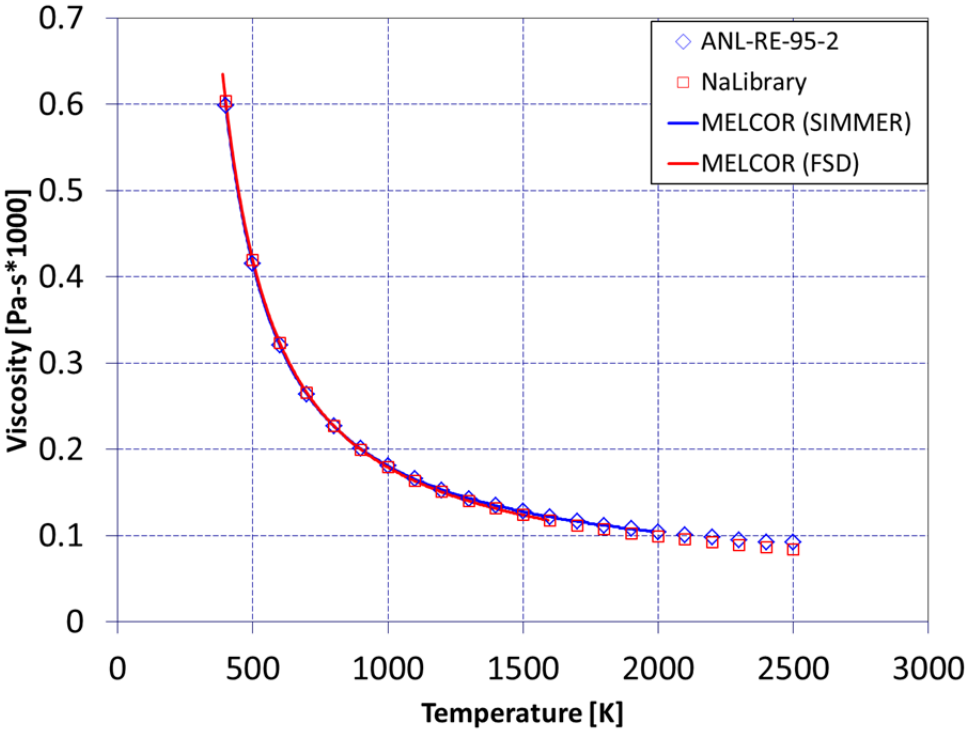


Figure 3-17 Viscosity

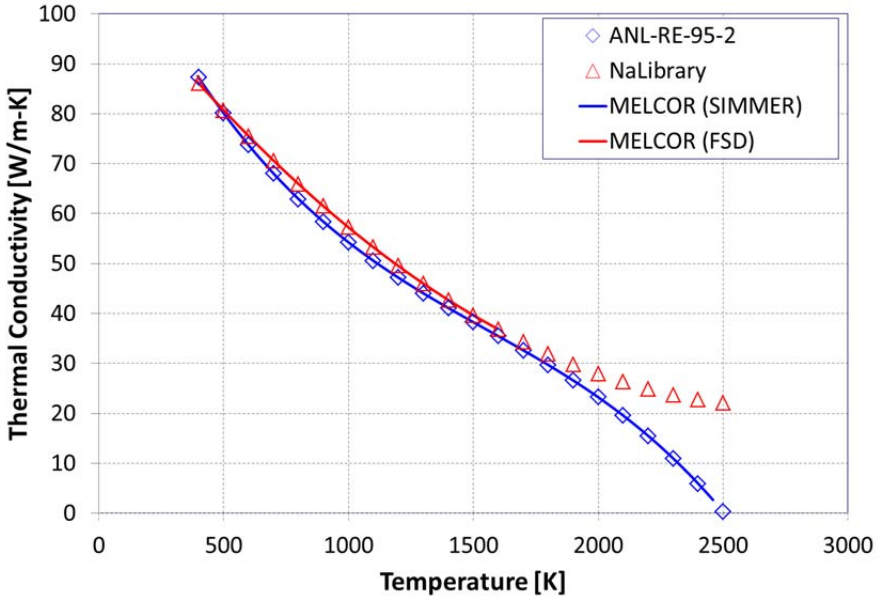


Figure 3-18 Thermal Conductivity

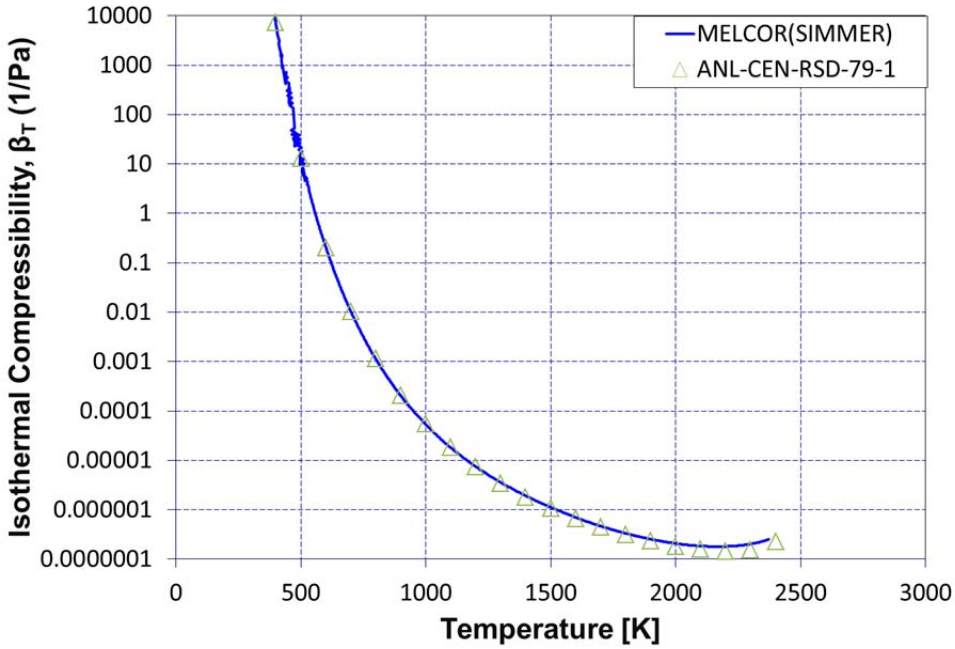


Figure 3-19 Isothermal Compressibility

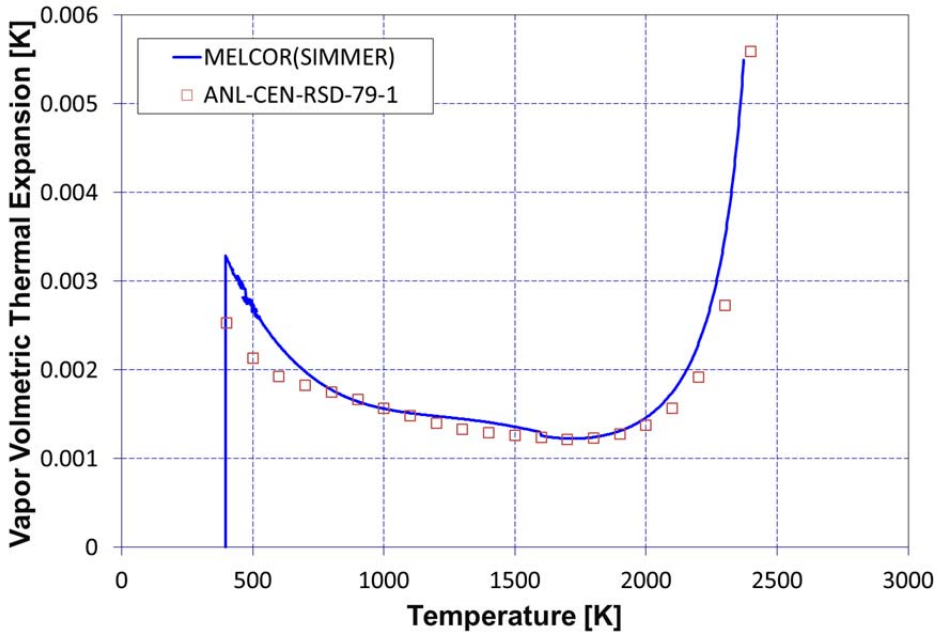


Figure 3-20 Volumetric Thermal Expansion

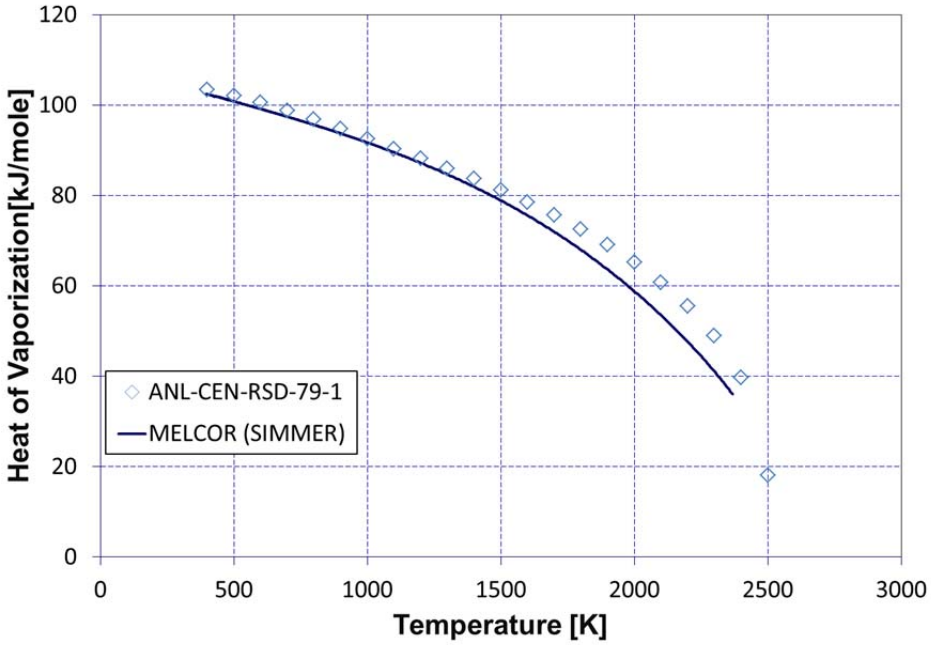


Figure 3-21 Calculated Heat of Vaporization

3.2.3 Discussion

The results of these tests, as described in more detail in the previous section, demonstrate that these models are able to reproduce the thermophysical properties upon which they are based over a wide range of conditions. However, there are still improvements that must be made to improve code numeric at both high and low saturation pressures. Also, currently the SIMMER models appear to perform better at high pressure whereas the FSD models perform better at low pressure.

The test results also indicate that additional refinements will be necessary to ensure that the properties MELCOR calculated for either data sets (FSD or SIMMER/SAS4A) are numerically stable over the full range of liquid states, particularly when iterations are required. As mentioned before test cases presented here did not run to completion. In one case when a higher enthalpy source was used, MELCOR aborted ungracefully. The small time step on the order of 10^{-7} s is also unacceptable. Therefore, further evaluations of the correlations selected as shown in Chapter 2 (see Table 2-1 to

Table 2-3) are necessary for the SIMMER/SAS4A data set. Also it is necessary to ensure that the variables passed (mainly the temperature) are correctly used as intended for the property functions. The range of the correlations implemented requires a close examination of out-of-range issue, which may yield unrealistic results. Error trapping is required to ensure that extrapolation outside the valid range includes error messages or remediation. The current iteration scheme used for the water properties may be examined to ensure that it is valid for evaluating fluid other than water.

Additional test problems may be required to test out each of the implemented property correlations, particularly for different packages of the MELCOR code, for example, condensation and vaporization of sodium on heat structures (HS Package) and aerosols (RN Package). In addition, other functionality of sodium fluid in the problems is needed to be checked out. This checking is in progress.

As a part of follow-on activity, a comparison of the FSD and SIMMER/SAS4A data set should be performed to identify any difference between the two data sets. For example, to explain the difference in the vapor specific heat calculation, Figure 3-15 and Figure 3-16 are provided. It is our intention to meet with FSD developer who can assist us in some of the issues we have encountered during testing of the sodium properties.

4 CONTAIN UPGRADE AND CONTAIN2-LMR

This chapter documents the code changes and upgrades that are necessary to bring CONTAIN 2 [Murata 1997] and CONTAIN-LMR [Murata 1993] codes to modern Software Quality Assurance practices as used in MELCOR code development environment currently employed. Both CONTAIN codes were developed in FORTRAN 77 or early versions of the compilers with the designation of older computer platforms, such as CRAY, CDC, and UNIX. Much of the system interfaces to these older computer platforms were altered to adapt the code to the current development environment.

CONTAIN development was started in 1984 and continued to 1997 before the development work was completely stopped. Table 4-1 shows the history of the major CONTAIN development milestones. As shown in this table, CONTAIN 2 represents the latest code version with significant improvement over CONTAIN-LMR, which was based on CONTAIN 1.11. Therefore, there is incentive to improve CONTAIN 2 with sodium coolant capability. Although CONTAIN 2 is designed for the LWR applications, it does contain many of the sodium models as described in SAND91-1490 [Murata 1993], such as sodium pool fire model, atmospheric chemistry model and sodium spray fire model (see Table 4-2). Many of these models were incomplete. In addition, unlike CONTAIN-LMR in which only sodium coolant can be simulated, the updated CONTAIN2-LMR allows the user to choose the coolant to be simulated. Thus the LWR capability is preserved.

Table 4-1. CONTAIN Code Release History [Murata 1997]

Version	Year	Improvement
1.0	1984	First release of code
1.06	1987	CORCON-MOD2, VANESA, water dropout, aerosol physics model and transport, radiation net enclosure model
1.11	1991	Moving-grid technique for solving aerosol growth by water vapor condensation
1.12	1991	Additions of DCH modeling, reactor cavity models for high pressure debris dispersal and vessel blowdown, …
1.2	1995	Film flow on wall structures, energy and mass conservation tracking, CORCON-MOD3, fission product decay data library
2.0	1997	Improvements in the DCH and hydrogen burn models

*Note that CONTAIN-LMR was derived from CONTAIN 1.11 [Murata 1993].

The CONTAIN-LMR source code used for this work was originated from Power Reactor and Nuclear Fuel Development Corporation (PNC) of Japan. It is believed that this version of code was the first version the PNC received from SNL, but might have been modified by PNC. However, many of the sodium models are similarly found in SAND91-1490 [Murata 1993]. In addition, this first version of CONTAIN-LMR may not have the latest physics models introduced into CONTAIN 2, which was the last version of the CONTAIN to be developed. The source code was received as a single file, which was difficult to debug and modify. File splitting was used to separate individual files into a file for a subroutine, function or small set into individual files. Figure 4-1 shows the top calling sequence of CONTAIN-LMR, where many of the sodium models were identified. As shown in this figure, the sodium models identified include those in the atmosphere calling routines as well as for the lower cell calling routines.

Figure 4-1. Top Level Call Sequence of CONTAIN-LMR.

```
Program main
{
       call input

       call contrl
}

subroutine input
{
       call setma [define the particular machine]
       call timdat [call system time and date]
       call gloset [initialize global common blocks]
       call redef [call restart and redefine parameters through inputs]
       call cpusnd [call cpu time and time]

       call iglobl [read global variable inputs]

       call celset [cell level setup]
       call icell [read cell level variable inputs]
}

subroutine contrl
{
       call output
       call cpusnd
       call chozdt [first step is determined]
       call glrest [set global parameters]
            call zero to zero out all arrays

       call clcntr [cell level main controls]
       call glcntr [global level main controls]

       call output
}

subroutine clcntr [it is looping over all cells]
{
       call nxtcel [next cell information]
       call setgas [reset gas properties]
       call celldt [choose time step]
       call clrest [reset atmosphere quantities and deposition cals]

       call rhcntr [control routine for radiative HT]
       call rbcntr [lower cell controls] - since this routine does not pass ncell, so it must be
                                           explicit declared
            call phydt [allocate timestep for lower-cell physics]
            call atmlcr [heat transfer from lower-cell to atmosphere/structure]
            call laysrc [lower cell explicit source]
            call bctset [set B.C. for lower cells]
            call concrm [SLAM model and other CORCON models] - this routine calls:
            {
                      call slinpt [initialize boundary layer, SLAM chemistry data, concrete
                                   regions] - this routine calls:
                      {
                             call slchem [read chemical reaction data]
                             call concpt [detemines concrete array pointers]
                             call slcoor [initialize SLAM coordine system]
                             call tranb [store concrete storage array, ch()]
                      }
                      call trana [loads storage array for the concrete calculations]
                      call stime [determine timestep for this model]
                      call slam [physics of the sodium-concrete interactions] - this routine
                                                        calls:
                      {
                                   call coneqs [estimate water release and HT calculations for
                                                        the interactions]
                                   call natcon [calculates SLAM physics]
```

```
                                call wtrrls [calculates water release]

                        }
                        call tranb [loads concrete storage array, ch()]
                        call tranc [transports mass and energy of SLAM]
                   }
              call pfire [sodium pool fire model]
              call pmhxfr [computer heat/mass transfer between pool and atmosphere]
              call htset [interlayer heat transfer coefficients]
              call fpheat [compute fission product heating to layers]
              call tabhet [load volumetric heat source arrays]
              call hxlow [compute conduction and heat transfer among lower cell layers]
              call concre [load physics of concrete layers]
              call interm [load physics of interm layers]
              call pool [load physics of pool]
              call atmosp [load physics of atmosphere layers] -currently there is no coding in
                                                                    this routine
              once the layers are done
                   call cvtoat [process cavity physics]
                   call bctset [set B.C. for lower cell for radiation HT]

        call ccntrl [atmospheric control routine] - this routine calls following:
        {
                 call soratm [atmospheric external source]
                 call engctl [call engineered systems]
                 call soratm [fission product atmospheric source]
                 call fpsurf [calculate HT for FPs to/from structures]
                 call chemrx [sodium atmosphere chemistry models]
                      *** reaction: na + h2o => naoh + 0.5 h2
                      call chmrep [aerosol or within aerosol deposit and condensable film]
                      *** reaction: 2*na + h2o => na2o + h2
                      call chmgas [reaction with gases]
                      call chmaer [reaction of aerosols and gases -h2o]
                      call chmaer [reaction of aerosols and gases -sodium]
                      call chmdep [reaction of deposits and film with gases]
                      *** reaction: 2*na + (1-0.5x)*o2 => x*na2o + (1-0.5x)*na2o2
                      call chmgas [reaction with gases]
                      call chmaer [reaction of aerosols and gases]
                      call chmdep [reaction of deposits and film with gases]
                      *** reaction: na2o2 + 2*na => 2*na2o
                      call chmrep
                      call chmaer
                      call chmdep
                      *** reaction: na2o + h2o => 2*naoh
                      call chmrep
                      call chmaer
                      call chmdep
                      *** reaction: na2o2 + h2o => 2*naoh + 0.5*o2
                      call chmrep
                      call chmaer
                      call chmdep
                 call hburn [burn models]
                 call qxlos [calculate heat structure temperatures]
                 call spray [sodium spray model]
                      call sorspr [sodium spray external sources]
                      call velt [calculate terminal velocity of spray droplets]
        }

        call sor [sources]
        call hotemp [set host temperature in atmosphere]
        call savcel [store cell information]
}

subroutine glcntr
{
        call acntrl [aerosol control] - this routine calls the following:
        {
                call aersl [aerosol physic calculation]
        }
        call dchbin [process DCH bins]
        call trpflw [calculates gas and debris velocities]
```

```
        call webdrp [calulcates the mass distribution for debris drop]
        call rpv [main routine for RPV DCH models]
        call entfra [user defined fraction for debris dispersion]
        call entrat [entranment rate models for debris]
        call gsourc [convert and consolidate sources]
        call decay [estimate decay of FPs]
        call fpmove [redistribute FPs]
        call flow [atmosphere flow model]
        call hmpwr [calculate host decay power]
}

subroutine prcell
{
    call pratm [print atmosphere information]
    call rbout [print lower-cell information, including sodium pool fire]
    call praero [print simple aerosol output]
    call praero [print detailed aerosol output]
    call htmout [print heat structure information, including degassing]
    call prspry [print sodium spray fire information]
}
```

Currently the MELCOR code development environment is using Microsoft Visual Studio ® 2008 with INTEL Visual FORTAN Compiler ® XE 2011. To enable both CONTAIN 2 and CONTAIN-LMR codes to run in these environments, the following code modifications were made:

- All passing arrays through subroutine calls, such as (*) and (1) were converted to the explicit size of the arrays in order to compile without errors.
- Scratch arrays such as h(…), ih(…), ah(…) and lh(…) had been increased in sizes in order to function correctly.
- Other bugfixes were also performed, such as “divide by zero”, syntax errors, and uninitialized variables.
- To be consistent with CONTAIN 2 structures, CONTAIN-LMR source code was converted from the upper cases to lower cases.
- Fortran 77 data and programming structures were untouched to minimize the effort.
- Because of the large common blocks used without converting them into modules, implicit none feature was not implemented.

To ensure proper working conditions, both CONTAIN codes were tested with a number of test inputs on hand. For CONTAIN 2, a series of the standard test decks (STDs) were used, since CONTAIN 2 is designed for the LWR applications. On the other hand, CONTAIN-LMR code is only used for fast reactor applications. The discussion of the CONTAIN testing is provided in Appendix B of this document.

In FY15, no new validation test was conducted. This validation is used to verify CONTAIN models before they are implemented into MELCOR. This validation will include porting the existing sodium models from CONTAIN-LMR that have not been included in CONTAIN 2 to CONTAIN 2 code (see Table 4-2). To enhance CONTAIN 2 to include all sodium models from CONTAIN-LMR, a number of the CONTAIN-LMR routines have been ported to CONTAIN 2 and a number of interfaces have been added and modified to ensure that the sodium models were implemented properly. The implementation was completed in FY15, but the testing and debugging were not yet completed. The primary reason for the partial completion level is that there were significant differences in the lower cell models between CONTAIN-LMR (based on

CONTAIN 1.11) and CONTAIN 2 as evidenced in Table 4-1. In addition, starting with CONTAIN 2, stored arrays and the bookkeeping for the lower cells were handled differently from CONTAIN-LMR. The resumption of this CONTAIN2-LMR will be continued in the latter part of FY16. This code as an upgrade to CONTAIN 2 will be used as a tool for the code-to-code comparison when the sodium models have also been implemented in MELCOR 2.1 (see Table 4-3). At the same time, this validation is also used after the models have been incorporated in MELCOR to ensure consistent results. The validation testing for the sodium models are discussed in Chapter 5 for the document.

Table 4-2 Sodium Model Coding in CONTAIN Codes

No	Code Description	
	CONTAIN-LMR*	**CONTAIN2**
1	ACNTRL routine formulates a flag, ICONDN, which designates the condensing component, such as water or sodium (see Two Condensable Option Section for more details).	This model is not available.
2	It contains SPRAY routine to allow the simulation of sodium spray fire.	It also contains SPRAY routine to allow the simulation of sodium spray fire, which is similar to that of CONTAIN-LMR.
3	PMHXFR routine contains heat transfer equations for evaporation and condensation of the coolant in the pool. It also models sodium coolant. BOILER routine contains a number of boiling/film boiling equations, and critical heat flux equations for sodium.	Although both PMHXFR and BOILER routines exist in this version, no sodium correlation is included.
4	PFIRE routine calculates sodium pool fire, which is based on SOFIRE-II code. Limited the burned sodium to ½ of the initial mass at a given timestep. In tracking, it distributes the mass and energies between the atmosphere and the pool. It includes Na_2O, Na_2O_2, Na and O_2. IPFIRE routine is the input processing for the sodium pool fire.	Similarly PFIRE routine as described in CONTAIN-LMR calculates the sodium pool fire.
5	SLAM (Sodium Limestone Ablation Model) routine contains the physics of the sodium-concrete interactions	Although SLAM routine exists, but there is no coding in the file.
6	NFPCHM is a flag to designate a pool chemistry model call to PCHEM routine. However, this routine does not exist. According to SAND91-1490, six of the eight chemical reactions in the pool have been modeled in the SLAM model. In fact, SLCHEM routine shows the coefficients for the reactions for the SLAM model. The actual reaction calculation routine is REACSL, which takes these coefficients to perform reactions. See SAND91-1490, Equation 8-20 to identify the reactions modeled.	Similar to CONTAIN-LMR code, these calls are there, but there is no PCHEM routine.
7	CHEMRX routine models the sodium atmosphere chemistry. It references HEDL-TC-730. It also contains more sets of heat of reaction equations for different reactants. Account for reactions with gases (CHMGAS routine), and with aerosols and gases (CHMAER routine), consider contact reactions within one aerosol particles or within an aerosol deposit or condensable film (CHMREP routine), react deposits or film with gas (CHMDEP).	CHEMRE routine models the sodium atmosphere chemistry. It references HEDL-TC-730. It does not have the heat of reactions equations as described in CONTAIN-LMR. In this routine, it calculates the gas reactions and call AERREA routine for aerosol related reactions.

*See Figure 4-1 for the calling sequence and top tree level of the code.

Table 4-3. Major Models in CONTAIN 2 and MELCOR 2.1 for the Containment Analysis

Model	CONTAIN 2	MELCOR 2.1
In-vessel modeling	None	Yes, COR package
Containment analysis (including engineered safety systems, such as containment sprays, fan cooler, ice condenser, heat exchanger, liquid transport system)	Yes	Yes
Corium concrete interaction	CORCON-MOD3	CORCON-MOD3
Aerosol physics*	MAEROS (tracked by species)	MAEROS (Class concept)
Fission products	Fission product decay library	DCH package

*MELCOR employs 17 default classes to group radionuclides from reactors, and other important materials, such as water and concrete. Aerosols are tracked by groups. However, users can re-define new classes for each of the materials to be tracked, other than those in the default classes.

5 IMPLEMENTATION OF SODIUM MODELS TO MELCOR

This chapter describes the implementation effort performed in FY15. Because there were major code developments in MELCOR 2.1 during FY15, many sodium model implementation that was originally scheduled were postponed till FY16. The only major code developments that could affect this work include:

- Aerosol resuspension
- Deposited aerosols are tracked by bins, rather than a single scalar value

With that, the design and data structure for the sodium models to be implemented in FY15 includes:

- Atmospheric chemistry
- Sodium spray fire
- Sodium pool fire

The remaining sodium models to be incorporated in FY16 would be:

- Sodium-concrete interaction
- Two-condensable option (see Section 5.1 for the modified model)
- Sodium pool physics and corium-concrete interaction (if time permits)

This chapter describes the sodium models in SAND91-1490 and those FORTRAN source codes in the CONTAIN-LMR code that was obtained from PNC. In SAND91-1490, it describes a number of models (see Table 5-1). As shown in this table, the types of the models are identified, including the implementation time frame. In terms of the model implementation as shown in Table 5-1, only the first three models will be implemented in FY15. The remaining models will be addressed in FY16. Note that the models in the CONTAIN-LMR code are similar to those models described in SAND91-1490. However, CONTAIN 2, which is developed at SNL, contains coding for many sodium models which may be included in SAND91-1490 (see Table 5-1).

Table 5-1 Implementation Status of CONTAIN-LMR Sodium Models for MELCOR

No	Model	Physical(P)/ Chemical(C) Type	Description
1	Atmospheric Chemistry	C	This model allows atmospheric constituents to interact chemically to form stable compound. The chemical reactions consider including those for sodium. The designs of the MELGEN input and data structure have been done.
2	Sodium Spray Fires	P/C	This model allows the treatment of the combustion of sodium spray resulting from an energetic event that causes droplets of sodium spraying out of the reactor system. The designs of the MELGEN input and data structure have been done.
3	Sodium Pool Fires	P/C	This pool fire model simulates the chemical reaction between sodium located in a pool and the oxygen in the atmosphere above the pool. The sodium pool physics model allows boiling of sodium pool. It also models

			the heat transfer between the pool and the debris layers such as CORCON layers. So this physics model may not be implemented, unless there is resource available. The designs of the MELGEN input and data structure have been done.
4	Two-Condensable Option/Condensate Removal	P	This option allows the modeling of the condensation, evaporation and boiling of both sodium and water within a single calculation. This model allows removal of condensate from the atmosphere. The fully implementation of this model to MELCOR may not be possible, because the current limitation on treating condensable in MELCOR (see Section 5.1 for more details).
5	Sodium Concrete Interaction	C	As a part of the pool chemistry, only eight major chemical reactions are considered in this model. In terms of sodium-concrete interactions, SLAM model from CONTAIN will be implemented The constituents considered including those species from the sodium-concrete interaction, and those sodium with water content in the concrete. The design of this model and data structure has not been done.
6	Debris Bed/Concrete Cavity Interaction	P/C	This model allows the modeling of the debris bed in the cavity where sodium pool can be present. The physical interaction of the sodium pool and the debris bed can be challenged, such as the heat transfer aspects of the interaction. In addition, the chemical aspect of the interaction can also be occurred. This model may not be implemented, unless resource and time are available.

In each section of this chapter, the model is described first, followed by a discussion of the CONTAIN-LMR coding of the model. Finally, the implementation approach to migrate the CONTAIN-LMR model into MELCOR is given in Section 5.2.

5.1 Review of CONTAIN-LMR Models

A number of sodium chemistry reaction models will be added to MELCOR based on models found in CONTAIN-LMR [Murata 93]. The chemistry models include: atmosphere chemistry, sodium spray and sodium pool. The descriptions of these three chemistry models are described in this section in details. For complete details of the chemistry models, refer to the CONTAIN-LMR manual [Murata 93]. Additionally, this section describes the models of the two-condensable option and the sodium-concrete interaction from CONTAIN-LMR. The details of the implementation of these two models are not specified here. These implementations will be provided in FY16. If time permits, the debris interaction model from CONTAIN-LMR may be implemented in MELCOR. Though code architecture differs significantly between MELCOR and CONTAIN-LMR, these models will be coded similarly and code-to-code validation will be performed to assure that the modeling was performed correctly.

5.1.1 Atmospheric Chemistry

When a breach of the primary coolant system occurs, the potential chemical reactions of sodium coolant with atmospheric constituents in the containment are of great interests in the liquid metal reactors. Depending on the accident scenarios, the sodium coolant may occur as vapors that could react chemically with any oxygen or water present. These reactions are generally exothermic, which can add thermal load to the containment system. In addition, any hydrogen generated by the sodium chemical reactions may pose additional consequence, such as hydrogen explosions. A number of sodium chemistry reactions are considered.

5.1.1.1 Gas Chemistry

The first reaction considered is:

$$\mathbf{Na + H_2O\ (l)\ \rightarrow NaOH + \frac{1}{2}H_2} \quad (5\text{-}1)$$

Reaction (5-1) is assumed to occur only for liquid phase water and sodium in contact on an aerosol particle, mingling aerosol deposits and condensate films on surfaces. Because the water is required to be liquid, the experimentally observed inhibiting effect of oxygen on reactions of water vapor and sodium is assumed to be inapplicable. This requirement assumes that either the temperature is relatively low (below the critical point of water) or the presence of liquid water is traceable to numerical effects and the amount is not significant. As shown in this reaction, hydroxide is expected to be the principal reaction product with water at low temperatures or with excess water. Conversion from hydroxide to monoxide is not modeled.

For this reaction, Na species include Na (g) and NaOH. Other materials involve H_2O(l) and H_2.

The second reaction is:

$$\mathbf{2\ Na + H_2O\ (g)\ \rightarrow Na_2O + H_2} \quad (5\text{-}2)$$

This reaction is used when the phase of water is not liquid. It is presumably correct at high temperatures with excess sodium. At low temperatures with excess sodium, the use of reaction (5-2) may produce excess hydrogen per mole of water. This reaction is also appropriate when water vapor is present, particularly when there is an excess of water vapor over oxygen. In this case, the water vapor is assumed to react not only with sodium vapor in the atmosphere, but also with sodium in aerosol form or in the form of aerosol deposits or films on surfaces. However, the reaction rate for reaction (5-2) at the surface with water is assumed to be limited by the evaporation rate of water from the surface. The sodium species include Na and Na_2O. Other species include H_2O (g) and H_2.

After the reactions with water, if any, oxygen in the atmosphere is assumed to react with sodium to form the monoxide and peroxide, respectively as follow:

$$\mathbf{2\ Na + \frac{1}{2}O_2\ \rightarrow Na_2O} \quad (5\text{-}3)$$

$$\mathbf{2\ Na + O_2\ \rightarrow Na_2O_2} \quad (5\text{-}4)$$

For reaction (5-3) and (5-4) for monoxide versus peroxide as products, this fraction relies on the input fraction parameter "FRNA2O" which represents the fraction of monoxide in the total reactions with oxygen . Reactions (5-1) to (5-4) also are assumed to occur with sodium aerosols, sodium aerosol deposits and sodium films in that orders.

Two subsequent reactions take place when peroxide and monoxide have been formed. The first subsequent reaction is for peroxide reacting with sodium:

$$\mathbf{Na_2O_2 + 2\ Na\ \rightarrow 2\ Na_2O} \quad (5\text{-}5)$$

This reaction is always assumed to occur if the peroxide and condensed sodium are in contact as a consequence of being present on the same aerosol particle or as a consequence of the mingling of the aerosol deposits and condensate film on a surface. The order is aerosol particles then aerosol deposits.

Sodium monoxide and peroxide can react with water to form sodium hydroxide:

$$\mathbf{Na_2O + H_2O\ (g)\ \rightarrow 2\ NaOH} \tag{5-6}$$

$$\mathbf{Na_2O_2 + H_2O\ (g)\ \rightarrow 2\ NaOH + \frac{1}{2}O_2} \tag{5-7}$$

Water vapor is assumed to react with aerosol particles and aerosol deposits in that order. Again, the user should note that while the hydroxide is expected to be the principal reaction product with water at low temperatures or with excess water, the possible subsequent conversion of the hydroxide to the monoxide is not modeled if conditions change. The chemical reaction models presented here assumes that all reaction heat is retained only by the gases present or by the structures; it ignores the increase in the heat content of the aerosols or aerosol deposits due to an increase in temperature above the temperature of the formation. The heat generated by the surface reactions is assumed to be deposited at surface nodes of the structures involved. This treatment is regarded as conservative.

5.1.1.2 Combustion of Sodium Hydrogen Jets

CONTAIN-LMR models the deflagration of hydrogen in the presence of sodium aerosol particles as ignition sources. In this model, CONTAIN-LMR utilizes the standing flame model for hydrogen burn. If the standing flame model is active in the current volume, each flow path into the volume is monitored for temperatures and concentrations of hydrogen and sodium. If the flow entering has a temperature greater than 533.1 K, a hydrogen mole fraction greater than 0.1, and a sodium density greater than 0.006 kg per cubic meter of hydrogen, and there is at least 8% molar oxygen in the atmosphere, a burn is initiated. If sufficient oxygen is present, all of the hydrogen entering with the sodium is consumed.

$$\mathbf{H_2(g) + 0.5O_2(g)\ \rightarrow H_2O(g)} \tag{5-8}$$

where the reaction energy for Reaction (5-8) is 1.43×10^8 J/kg of hydrogen consumed.

5.1.1.3 CONTAIN-LMR Coding

For this model, the main subroutine is CHEMRX, which is a controlling routine that calls CHMGAS for reactions with gases, CHMAER for reactions with aerosols and gases, CHMREP for reactions within aerosol particles or within aerosol deposit or condensable film, and CHMDEP for reactions of deposits or film with gases. Below are the summary of the routine functions:

- CHEMRX routine – set up reaction energy based on mole fraction, mw, and gas species
  - Reaction: H_2+0.5 O_2=H_2O (sodium-hydrogen jet)
  - Reaction: Na+H_2O = NaOH+0.5 H_2, call CHMREP first

- Reaction: 2 Na+H_2O = Na_2O+H_2, call CHMGAS first for reacted gas, call CHMAER twice for aerosols with gas, call CHMDEP for reacted deposits with gas
- Reaction: 2 Na +0.5 O_2=Na_2O, 2 Na+O_2=Na_2O_2, call CHMGAS, CHMAER, CHMDEP
- Reaction: Na_2O_2+2 Na=2 Na_2O, call CHMREP, CHMAER, CHMDEP
- Reaction: Na_2O+H_2O=2 NaOH, call CHMREP, CHMAER, CHMDEP
- Reaction: Na_2O_2+H_2O=2 NaOH+0.5 O_2, call CHMREP, CHMAER, CHMDEP

- CHMAER, CHMREP, CHMDEP, CHMGAS routines have similar input requirement:
  - General reaction:
    $$C1 \times Mat1 + C2 \times Mat2 = C3 \times Mat3 + C4 \times Mat4 + C5 \times Mat5 \quad (5\text{-}9)$$
    where Mat1 is aerosol, Mat2 is gas, Mat3 and Mat4 are aerosols, and Mat5 is gas. All C1 to C5 in this general reaction are the stoichiometric coefficients.
  - General interface variables are slightly different among these routines

Based on the above, all these subroutines will be implemented to MELCOR, with proper interface variables to be communicated with other packages in MELCOR.

5.1.2 Sodium Spray Fire

Of the two basic types of sodium fires postulated in sodium cooled fast reactors, spray and pool sodium fires, spray fires are generally considered to be more energetic. This is due to the fact that a sodium spray always burns at a higher rate than a sodium pool containing the identical amount of sodium because of the large surface area of the droplet versus the pool surface area. Pipe breaks are often postulated for developing sodium spray fire. The sodium released through the break is usually assumed to eject upward and impinge on the ceiling of the room, where a sodium liquid is formed and then breaks up to form droplets [Tsai 1980]. These droplets form a sodium spray. The interaction of the sodium spray with oxygen and available moisture in the atmosphere of the room creates the sodium spray fire phenomena.

The model for the sodium spray fire is based on the phenomenological model used in NACOM, a code developed and tested at Brookhaven National Laboratory [Tsai 1980]. However, CONTAIN-LMR did not include the sodium reaction with water vapor as in NACOM. Note that this model is also available in CONTAIN 2 (see Table 4-2). In this model, an initial size distribution is determined from a correlation using a specified mean droplet diameter. An assumption is used to state the trajectory of the droplets, which is assumed to be downward flow, with a terminal velocity. The combustion rate of the spray fire is integrated over the droplet's fall to obtain the total sodium burned mass, as functions of droplet size, fall velocity and atmospheric conditions.

In this model, the user specifies the mean droplet diameter for the sodium spray then an initial size distribution is determined using the Nukiyama-Tanasama correlation [Tsai 1980]. The current default mean droplet diameter is set at 0.001 m. This model also requires a user input fall height 'HITE'. In addition, this model requires the user to specify the mole fraction of Na_2O_2 produced by the spray fire. This mole fraction is currently set at 1.0 as default. Two main reactions with oxygen are modeled as shown in Reactions (5-3) and (5-4).

The combustion energy is computed based on the mole fraction of sodium (Na) to peroxide (Na_2O_2) as given by the following correlation:

$$\mathbf{S} = \frac{\mathbf{1.3478 \cdot F_{peroxide}}}{\mathbf{1.6957 - 0.3479 \cdot F_{peroxide}}} \quad (5\text{-}10)$$

Heat combustion, E_{spray} (J) is then calculated as

$$\mathbf{E_{spray}} = \mathbf{(1 - S) \cdot 9.1797 \times 10^6 + S \cdot 10.46 \times 10^6} \quad (5\text{-}11)$$

The duration of this source and the available oxygen determines the combustion time and the amount of the by-products and heat content to be generated. In the absence of better information regarding the kinetics of the oxidation process, the value of 1 for the ratio of peroxide over the sum of the peroxide and monoxide is often used. Tests have shown that the peroxide indeed dominate the reaction products, particularly when combusting in air.

5.1.2.1 CONTAIN Coding

In both CONTAIN-LMR and CONTAIN 2, the SPRAY routine documents the sodium spray fire model. The input routine for this model is through ISPRAY.

In the SPRAY routine, the following are included:

- Droplets are assumed at 1.015×10^5 Pa and saturation temperature.
- Mass fraction of Na_2O_2 is estimated based on the user specified input.
- Heat is estimated based on the above mass fraction.
- Selection of drop size distribution is based on the user specified mean droplet diameter.
- Determination of the spray source is based on the user specified data.
- Integration of the droplet fall and reactions is estimated.

The SPRAY routine only calls VELT routine for estimating the terminal velocity and Reynolds number. The SPRAY routine also calls SORSPR routine for the spray source.

For the implementation, only SPRAY and VELT subroutines will be ported over to MELCOR, with proper interface variables for communicating with other packages in MELCOR.

5.1.3 Sodium Pool Fire

This sodium pool fire model is taken from the SOFIRE II code developed from the results of pool fire tests [Beiriger 1973]. SOFIRE II model was based on the verification of experiments, which included a large test vessel in a series of thermodynamic parameter tests to study the effect of oxygen concentration on the system pressure, sodium burning rates and heat transfer rates. This vessel has a diameter of 3.05 m (10 ft), with a height of 9.14 m(30 ft) and contains 62.3 m^3 (2200 ft^3) of gas at the standard condition. In the lower section of the vessel, a 0.5574 m^2 (6 ft^2) steel pan was installed on a spider off the floor of the vessel. The pan was insulated with fire brick and mounted below a feed line from an external sodium preheat tank. Thermocouples were mounted in or on the sodium pool volume, steel pan, pan insulation, gas volume and vessel

walls. This experiment is referred to as a one-cell experiment. A two-cell experiment was also used to validate this model [Beiriger 1973].

The main pool fire reaction for this model is given as:

$$(\mathbf{1}+\mathbf{f_1})\cdot\mathbf{2}\cdot\mathbf{Na}+\ \mathbf{O_2}\rightarrow\mathbf{2}\cdot\mathbf{f_1}\cdot\mathbf{Na_2O}+(\mathbf{1}-\mathbf{f_1})\cdot\mathbf{Na_2O_2}+\mathbf{q(reaction)} \quad (5\text{-}12)$$

where f_1= fraction of total oxygen consumed that reacts to form monoxide and q(reaction) is 9.04540×10^6 J/kg and 1.09746×0^7 J/kg for the monoxide and peroxide, respectively [Murata 1993].

The above reaction requires oxygen in the air to diffuse to the sodium pool. CONTAIN-LMR uses a diffusion constant, D_O (m^2/s) for oxygen-nitrogen mixtures different than that of SOFIRE II [Murata 1993]:

$$\mathbf{D_O}=\mathbf{6.4315}\times\mathbf{10^{-5}}\frac{\mathbf{T_{film}}^{\mathbf{1.823}}}{\mathbf{P}} \quad (5\text{-}13)$$

where T_{film} = average temperature of the pool and atmosphere (K) and P = system pressure (Pa).

In this pool fire model, it is required to allocate the amount of the products and reaction energy to the pool and to the atmosphere layer of the cell. Thus, additional fractional inputs may be provided. The fractional inputs include:

- f_2 is the fraction of sensible heat from the reaction to the pool. The remainder will be directed to the atmosphere layer of the cell.
- f_3 is the fraction of Na_2O product that enters the pool as a solid after the fire. The remainder will be directed to the atmosphere as aerosols.
- f_4 is the fraction of Na_2O_2 product that enters the pool as a solid after the fire. The remainder will be allocated to the atmosphere as aerosols.

Note that the sodium burning rates calculated by this model depend on the temperature differences between the pool and the atmosphere. This difference is assumed to set up turbulent natural convection above the pool surface – the greater the differential, the greater the burning rate would be. Thus radiative heat transfer between the pool and its surrounding may result in differences in the burning rate.

A sodium pool may form in the reactor cavity area, which can play an essential role in LMR accident analyses. The modeling described here is limited to the heat transfer models within the sodium pool with hot surfaces, such as hot debris. Although sodium pool chemistry can take place, it is deferred to a topic on the sodium pool chemistry.

This model is associated with the sodium pool in the reactor cavity area. The lower cell input must be invoked in order to use this model. This modeling is to include any heat transfer equations that are specifically designed for sodium forming a lower cell pool.

In boiling heat transfer from the sodium pool, the saturation temperature (T_{sat}) is a function of the pool pressure ($\mathbf{P_p}$ in Pascals). It is given as [Murata 1993]:

$$\mathbf{T_{sat}} = \frac{-1.2023\times10^4}{\log_{10}(P_p/10^6)-8.1185} \tag{5-14}$$

For film boiling heat transfer, particularly for the surface below the sodium pool, T_{bot}(K) at the Leidenfrost point is given by [Murata 1993]. All units are in the MKS system:

$$\mathbf{T_{bot} - T_{sat}} = \frac{\Psi\lambda}{1-0.5c_{vf}\Psi} \tag{5-15}$$

where Ψ is given as:

$$\mathbf{\Psi = 0.086806}\left[\frac{\rho_{vf}\mu_{vf}^{1/3}}{k_{vf}}\right]\rho_l^{-5/6}\sigma^{1/2} \tag{5-16}$$

where λ is the heat of vaporization of sodium at T_{sat}, c_{vf} is the vapor specific heat at constant pressure, ρ_{vf}, μ_{vf} and k_{vf} are the density, viscosity and conductivity of the vapor evaluated at the film temperature (where this temperature is average of T_{bot} and T_{pool}). σ is the sodium liquid surface tension evaluated at T_{sat}. ρ_l is the pool density. Thus the heat flux at the onset of film boiling is given:

$$\mathbf{q_{bot} = 0.15927 \cdot \rho_{vf}\,\lambda'}\left(\frac{\sigma}{\rho_l}\right)^{1/4} \tag{5-17}$$

where

$$\mathbf{\lambda' = \lambda + 0.5c_{vf}(T_{bot} - T_{sat})} \tag{5-18}$$

The critical heat flux used in CONTAIN-LMR is given by:

$$\mathbf{q_{chf} = 0.24775\,\lambda \cdot \rho_v^{1/2}\left(1 + 163.54 \cdot P_p^{-0.4}\right)(\sigma\rho_l)^{1/4}} \tag{5-19}$$

The equation used for the critical heat flux temperature (T_{chf}), the surface temperature below the sodium pool evaluated at q_{chf} is defined as

$$\mathbf{T_{chf} - T_{sat} = 1814.6}\left[\frac{q_{chf}}{k_l}\right]^{0.3}\left[\frac{\lambda\rho_v}{c_l P}\right]^{0.7}\left[\frac{\sigma}{\rho_l}\right]^{1/2} \tag{5-20}$$

The relation before this critical heat flux and the critical heat flux for a subcooled pool is given as

$$q_{chf,s} = q_{chf}\left[1 + 0.10714\ \frac{\left(\frac{\rho_l}{\rho_v}\right)^{3/4}\frac{c_l}{\lambda}(T_{sat}-T_{pool})}{1+163.54\ P^{-0.4}}\right] \quad (5\text{-}21)$$

5.1.3.1 CONTAIN Coding

Sodium Pool Fire

This sodium pool fire model is called within the lower-cell physics routine. In RHCNTR subroutine, where the lower cell controls are done, it calls lower cell layers to set up the physics such as the intermediate and concrete layers. Then it calls the pool layer, which calls the PFIRE routine to perform sodium pool fire calculations using Eq. (5-13) and the fractional inputs above. The calculations include mass and energy estimate of the reaction, including the estimate of the reactants, sodium from the pool and the oxygen from the atmosphere, via Eq. (5-14) for the diffusion rate, and the products, sodium monoxide and sodium peroxide. The allocation of the product masses to the pool and atmosphere are functions of the user-defined values or by default.

Sodium Pool Physics

In addition, CONTAIN-LMR contains a physic model that is associated with the sodium pool in the reactor cavity area. The lower cell input must be invoked in order to use this model. ICELL routine is the input controller for the cell level input models. It calls REBPLT routine a number of times for lower cell layers, such as concrete, intermediate, pool and atmosphere. REBPLT routine contains a number of sodium specified properties (such as surface tension for sodium), which includes the call of BOILER routine. Boiler routine includes correlations for boiling, film boiling, and critical heat flux equations for sodium. Critical temperature for the critical heat flux is also calculated in this routine. This model is important for the core-concrete interactions.

Based on the above, PFIRE subroutine will be ported over to MELCOR with proper interface variables to communicate with other packages in MELCOR.

5.1.4 Two Condensable Option

In most system codes (e.g., CONTAIN and MELCOR), only a single coolant is permitted at one time. The introduction of sodium for the coolant in CONTAIN-LMR will pose issues relating condensation processes of both sodium and water simultaneously. To address the problem of modeling of the condensation, evaporation and boiling of both sodium and water within a single calculation, the two-condensable option was implemented in CONTAIN-LMR. In fact, the CONTAIN code architecture prevents a completely general treatment of two condensables in the calculation. To permit two condensables, such as water and sodium, only atmosphere thermodynamics and flow, and aerosol condensation are allowed.

Two-condensable option in CONTAIN-LMR is intended to treat both sodium and water simultaneously [Murata 1993]. The treatment of this option for the atmosphere thermodynamics and flow is available for either thermal or fast reactor (which includes the LMRs). This general treatment includes modeling of the condensate dynamics within the aerosol model.

Since the code limitation only permits a single condensable in the atmosphere, the other condensable, if present, is treated as an ideal gas. This designation is required in order for the model to work. The two-condensable option permits condensation onto aerosols and deposition

on surfaces. Within a cell (or control volume in MELCOR), the specified cell-level condensable is allowed to condense on surfaces, and the other condensable is treated as an ideal gas requiring its atmosphere properties (such as viscosity and conductivity). In addition, if both chemical reactions and aerosol condensation for the other condensable is modeled, the ideal gas assumption should be adequate. In CONTAIN, cell-level models are restricted with respect to the condensable used. These models are not extended to the two-condensable option, which includes SPARC pool scrubbing model for aerosols, spray, ice condenser and fan cooler engineered systems. Allowing the presence of other condensable and other materials in liquid pools is treated in CONTAIN-LMR; however, it is assumed that this would not affect the transfer rate of the cell-level condensable and transfer through flow between pools.

As pointed out later on for the chemical reactions, CONTAIN-LMR assumes that chemical reactions take place among repositories associated with the atmosphere. In this model, there is an imposed limitation on the reaction rates, which is at most ½ of any gaseous reactant or atmosphere condensable to be allowed to react per system timestep. If any sodium is sufficiently cold to preclude significant vapor-phase transport and is not settling out rapidly on surfaces, the reaction rate may be controlled by the evaporation rate of the water films (a slow process). Also chemical reactions between atmosphere and surface films or deposits that depend on the gas transport or condensables from the atmosphere to the surface in general are assumed to occur instantaneously. However, the reactions of sodium in the atmosphere and surface water are not included in the model due to the low sodium pressure. By default, the reactions between sodium in the atmosphere and surface water require the transport of water vapor from the surface through evaporation, which controls the rate of the reaction. Thus the evaporation rate is important. Note that this can only occur, if water is the cell-level condensate.

Aerosol condensation within this condensable option depends on the condensation model used. The old method relies on the fixed grid method to estimate the water condensation, which is used to calculate the aerosol population change in the aerosol size classes due to condensation, but did not consider hygroscopic effects. The second method, adapted from CONTAIN 1.1 is the moving grid method, which allows modeling hygroscopic or Kelvin effects. This second method should minimize the numerical diffusion instability. In terms of input requirement, keyword SOLAER is an option for AEROSOL global block to deal when both water and sodium are declared as condensable. Two possible cases if SOLAER option is invoked: a) water uses fixed grid, and b) sodium is not active; or c) water uses moving grid and sodium uses fixed grid. The latter case allows the modeling of both water and sodium in a single problem. However, for the new moving grid method for aerosol condensation, aerosol nucleation is permitted for these conditions.

5.1.4.1 CONTAIN Coding

In CONTAIN-LMR, subroutine ACNTRL provides a control for aerosol modeling. A flag, ICONDN is used to identify the condensation phenomena (see Table 5-2). This routine contains logic according the value of ICONDN as described in Table 5-2. It calls the AERSL routine, which is a driver routine for the multicomponent aerosol module. Within this routine is the model of the moving grid formulation. Note that the moving grid method is only applicable for water used. Sodium is still using the fixed grid method. In the moving grid formulation, the

subroutine CONDEN is called. CONDEN routine controls the condensation calculation for aerosol.

Table 5-2 ICONDN Flag Description

ICONDN Value	Description
1	Only water is a condensing component
2	Only sodium is a condensing component
3	Sodium is the only condensing component in the present cell (control volume in MELCOR) although water can condense in others
4	Both water and sodium are condensing components. Note that this requires the moving grid method be available for water

5.1.5 Sodium-Concrete Interaction

When sodium leaks onto a concrete floor, there is a potential chemical reaction between the sodium and the concrete material. Although the concrete is normally lined with steel to protect against the direct contact of the sodium, there are heat transfers between the liquid sodium and the liners that could potentially heat up the concrete floor, which will cause dry out to the concrete. Both carbon dioxide and moisture released from the concrete can interact with sodium if the liner is penetrated. Thus the sodium-concrete interaction can occur. The model treatment in CONTAIN-LMR is based on the experiments done at SNL – sodium limestone ablation model (SLAM) [Suo-Anttila 1983, Westrich 1983]. Only a brief description is provided in this report and the reader is encouraged to read the previously referenced reports.

SLAM uses a nodalized representation of the concrete with models for heat transfer, water migration, water and CO_2 evolution, and chemical ablation of exposed concrete surface (see Figure 5-1). As shown in Figure 5-1, SLAM consists of three regions. The topmost region is the pool region, but the nodalization is associated with the boundary layer where the ablation occurs. Below this region is the dry concrete region. Also shown in this figure, a number of constituents can involve within SLAM, which includes SiO_2, H_2O, Na, H_2, NaOH, Na_2SiO_3, Na_2CO_3, Na_2O, CaO, $CaCO_3$, CO_2, graphite, $MgCO_3$, MgO, inerts, steel and UO_2. The major reactions considered in SLAM are given later. In SLAM, the boundary layer consists of 12 nodes, while the dry region consists of 15 nodes or more. Each node has the same thickness or size, which varies with the changing dimensions of the dry concrete region. A variable "del1" is the thickness of the boundary layer and dry concrete regions. This variable is subjected to change in terms of increasing or decreasing in the course of a problem. The initial del1 is 0.003 m. The dry concrete region will increase when the thermal penetration rate of the concrete exceeds the ablation rate and will decrease when the converse is true. The bottom-most region is the wet concrete region where evaporable water may still be found in the concrete as shown in this figure. The number of nodes depends on the number of dry nodes which is given by 50 – ndry + 2.

With these three regions, SLAM computes each region as time passes and penetration occurs, which each region will change its size and position. The coordinate system of SLAM uses the

moving Eulerian system (see more details in [Suo-Anttila 1983]). Below is a brief description of each region.

Pool Region: It contains a sodium pool region with all of the reaction products from the sodium-concrete interaction. Materials are assumed to be well mixed and virtually isothermal. The pool changes in composition which results in swelling with time during penetration. The swelling is caused by the addition of gases and reaction products of lower density than the reactants.

Dry Region: It contains the dehydrated concrete region and the boundary layer of the pool region. Almost all of the important reactions occur within the boundary layer of the dry region. At the interface, the ablation is presumed to occur by two mechanisms: dissolution and ablation. This region can swell or shrink (it moves with the penetration front).

Wet Region: It is the concrete region that contains water. The distribution of the water is important because it determines the amount which can be evaporated and available for the reactions with sodium at the boundary layer.

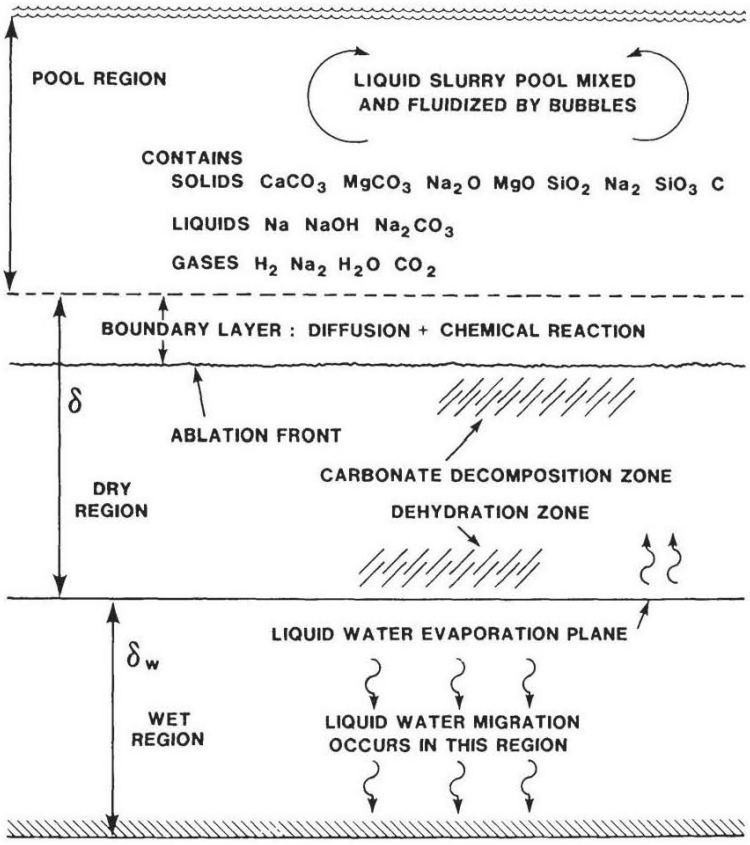


Figure 5-1 Schematic Diagram of SLAM [Suo-Anttila 1983]

Major reactions considered in SLAM are [Suo-Anttila 1983]:

$$\mathbf{H_2O + Na \rightarrow NaOH + 1/2\ H_2} \quad (5\text{-}22)$$

$$\mathbf{CO_2 + 2\ Na \rightarrow 4\ Na_2O\ + C} \quad (5\text{-}23)$$

$$3\,CaCO_3 + 4\,Na \rightarrow 2\,Na_2CO_3 + 3\,CaO + C \quad (5\text{-}24)$$

$$3\,MgCO_3 + 4\,Na \rightarrow 2\,Na_2CO_3 + 3\,MgO + C \quad (5\text{-}25)$$

$$2\,NaOH + CaCO_3 \rightarrow CaO + H_2O + Na_2CO_3 \quad (5\text{-}26)$$

$$2\,NaOH + SiO_2 \rightarrow Na_2SiO_2 + H_2O \quad (5\text{-}27)$$

Note SLAM is designed for the limestone concrete, which has ignored the possible reaction forming Na_4SiO_4, because small quantity of silicates presents in the limestone concrete. The reaction with silicates would provide a significant heat source in comparison to the carbonates.

The coordinate system used in SLAM is represented in Figure 5-2. As shown in this figure, SLAM uses a 1-D nodalization to compute all three regions. The moving boundaries at the interfaces of the three regions are shown. For details, refer to [Suo-Anttila 1983].

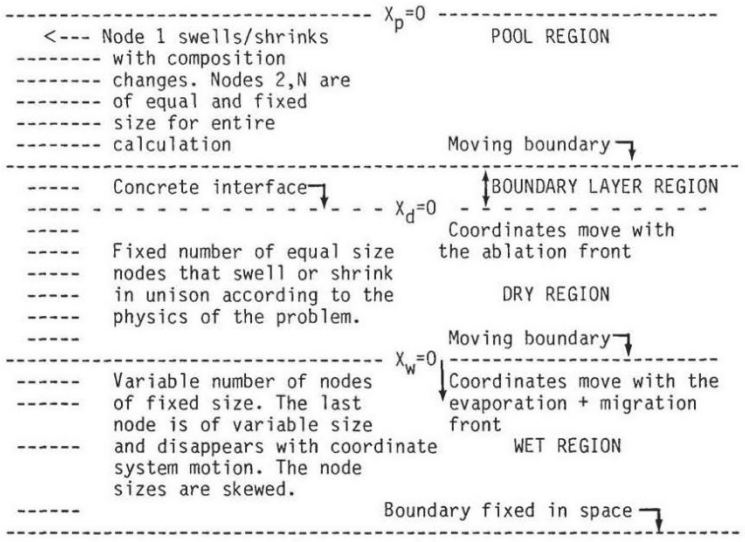


Figure 5-2. The SLAM Systems (subscripts p, d, and w refer to pool, dry, and wet respectively) [Suo-Anttila 1983]

Note that this model has not been fully described in this section.

Based on [Murata 1993], the SLAM model in CONTAIN-LMR may not be fully functional:

- An option to disable concrete ablation and allow only concrete outgassing
- The outgassing option assumes that the concrete is covered with an impenetrable barrier, representing the liner shell.
- The gases removed from the concrete through the SLAM outgassing option are not placed into the atmosphere of a cell, but simply disappear from the problem.

Once testing of the models using an input model for CONTAIN-LMR is done, a more descriptive discussion of the models will be provided in FY 16.

5.1.5.1 CONTAIN Coding

The SLAM model is only available in the lower cell of the CONTAIN-LMR. It is called by CONCRM routine (see Figure 4-1). This routine also calls other CORCON models. Once called, SLINPT subroutine is called to initialize the boundary layer, SLAM chemistry data and concrete information. Within SLINPT, it calls SLCHEM for reading chemical reaction data, then CONCPT for determining the concrete array pointers, then SLCOOR for initializing SLAM coordinate system. There are a number of specific subroutines that are important for SLAM:

- Storage array routines:
    - TRANA loads storage array for the concrete calculations, and it is called by SLINPT.
    - TRANB loads concrete storage array after SLAM physics routine (SLAM) is called
    - TRANC loads the transport information about the mass and energy from SLAM calculations
- STIME routine determines the timestep for this model
- SLCHEM routine contains all data for the chemistry models and other constants
- SLAM routine is the physics and chemistry model, which computes the sodium-concrete interactions. It calculates starting from the pool to the wet regions of the model. This routine calls:
    - CONEQS for estimating the water release and heat transfer calculations for the model
    - NATCOM for calculating the SLAM physics
    - WTRRLS for calculating the water release

To activate this model, NA-CONC keyword is used within the lower cell inputs of CONTAIN-LMR. The keyword SLAM must be used in order to activate the sodium-concrete ablation model. To allow outgassing, the keyword H2O-MIGR is used. In order to activate SLAM, the keyword FAILURE with a failure temperature for the intermediate layer of the lower cell is required. This is a prerequisite to start SLAM. The output of the SLAM includes:

- Average dry region temperature
- Wet-dry interface temperature
- Concrete reaction heat
- Concrete surface heat flux
- Heat flux into the wet zone
- Penetration depth
- Ablation velocity
- Dry zone depth

- Dry zone growth rate
- Dry zone heat sources
- Wet-dry interface heat flux
- Wet-dry interface water partial pressure
- Interface water evaporative flux
- Flux of bound water from the dry zone
- Integral of evaporative and bound water flux
- Flux of bound CO_2 from the dry region
- Integral of CO_2 flux

As indicated before, the implementation of this model has not yet begun. Thus no input and result will be described in Appendix B.

5.2 MELCOR Modification

This section describes the actual modification to be done for number of CONTAIN-LMR sodium models. Note that only the chemistry models are described in more detail in comparison to the two-condensable option and the sodium-concrete interaction model.

To be more efficient and better managing the sodium related models, a new package "Sodium Chemistry" (NAC) package will be added to MELCOR. In order to activate this package, "NFLUID" must be either 7 for the FSD database or 20 for the SIMMER database as described in Table 3-1 for the sodium coolant. This package will include a number of subroutines from CONTAIN-LMR, which include SPRAY, CHEMRX, CHMAER, CHMGAS, CHMREP, CHMDEP and PFIRE. All these subroutines will interface with CVH and RN package variables for transferring chemistry related processes (both heat and mass), including sodium, oxygen, water and the creation of the by-products of sodium burn resulting from the reactions. Figure 5-3 summarizes the chemistry models in Sections 5.1.1 to 5.1.3. As shown in this figure, a total of five aerosol species are identified, including the reactants and the sodium by-products. Note that the number of default classes in the RN package is shown in Table 5-3. The five RN classes to be tracked within NAC package include H_2O, Na, NaOH, Na_2O, and Na_2O_2 aerosols. Note that there should be more than five aerosols if including the by-products from the SLAM model (see Section 5.1.5). For now, only five aerosol mappings from NAC are included.

For the MELGEN input processing, the calling of the NAC package will be included in Meg_RW.f90. NAC_ReadmelgenInput should be called within ReadMelgenInput subroutine. The following are the current designed input records for the NAC package:

- NAC_INPUT
  - Test if Nfluid=7 or Nfluid=20
- NAC_ATMCHEM
- NAC_SPRAY
- NAC_PFIRE
- NAC_COND (not yet developed)

- NAC_SLAM (not yet developed)
- NAC_SC (not yet developed)

Note that NAC_COND, NAC_SLAM and NAC_SC have not been developed yet. They will be added in the latter part of FY16.

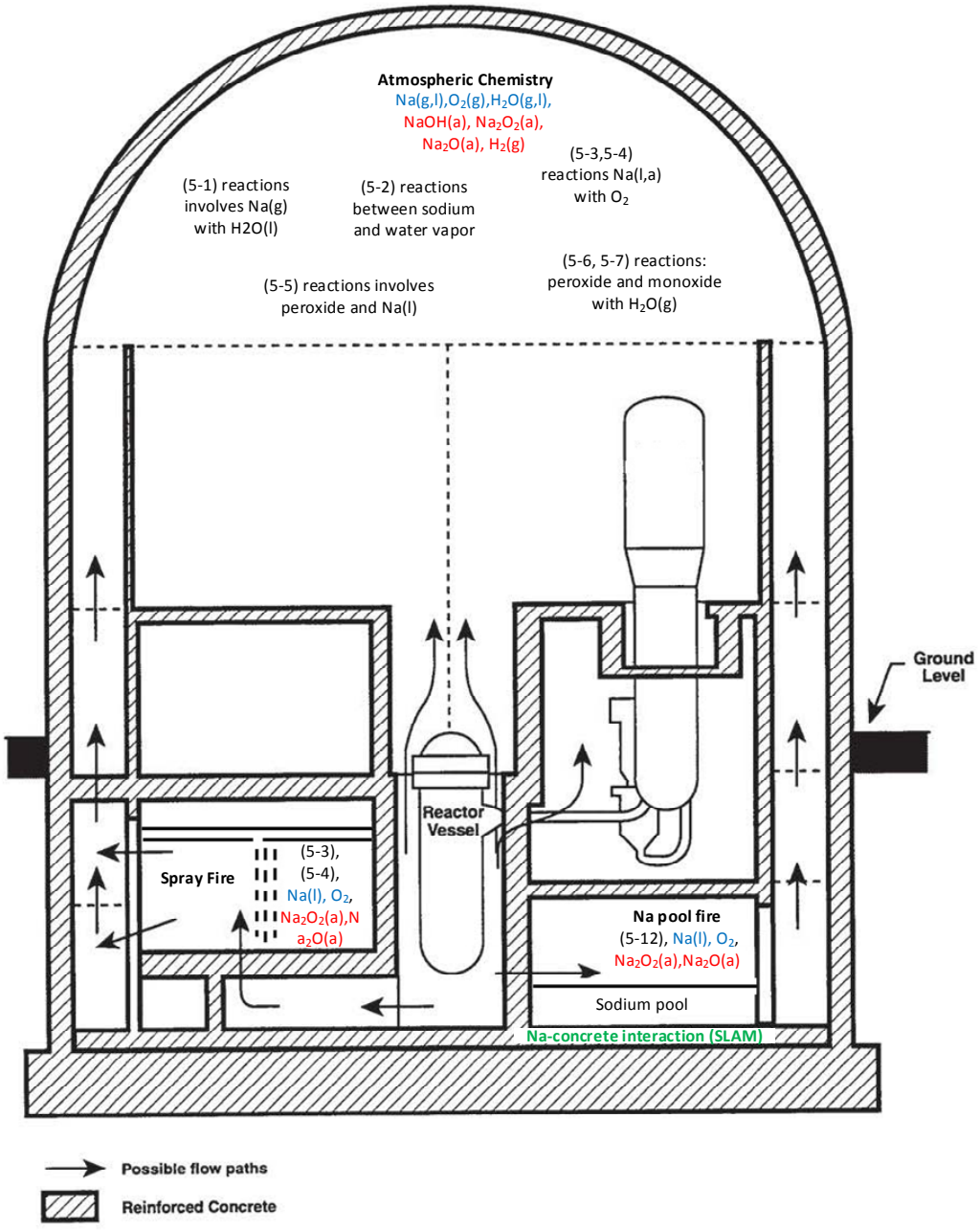


Figure 5-3. Graphical Representation of the Sodium Chemistry Models for Atmosphere, Spray and Pool.

() are the equation (reaction) number in the text, while a, l and g stand for aerosol, liquid and gas respectively.

Since the sodium chemistry models in CONTAIN-LMR are usually in the form of correlations, it is recommended that some of the coefficients of the correlations be placed in sensitivity coefficients, allowing the user to over-write as necessary to model his or her specified problem.

Since there is a standard numbering of the package sensitivity coefficients in MELCOR, it was decided to use 8100-8499 as the range for sodium specified models. Other numbers have been used or reserved for other package uses.

Table 5-3. RN Class Compositions

Class	Class Name	Chemical Group	Representative	Member Elements
1	XE	Noble Gas	Xe	He, Ne, Ar, Kr, Xe, Rn, H, N
2	CS	Alkali Metals	Cs	Li, Na, K, Rb, Cs, Fr, Cu
3	BA	Alkaline Earths	Ba	Be, Mg, Ca, Sr, Ba, Ra, Es, Fm
4	I2	Halogens	I_2	F, Cl, Br, I, At
5	TE	Chalcogens	Te	O, S, Se, Te, Po
6	RU	Platinoids	Ru	Ru, Rh, Pd, Re, Os, Ir, Pt, Au, Ni
7	MO	Early Transition Elements	Mo	V, Cr, Fe, Co, Mn, Nb, Mo, Tc, Ta, W
8	CE	Tetravalent	Ce	Ti, Zr, Hf, Ce, Th, Pa, Np, Pu, C
9	LA	Trivalents	La	Al, Sc, Y, La, Ac, Pr, Nd, Pm, Sm, Eu, Gd, Tb, Dy, Ho, Er, Tm, Yb, Lu, Am, Cm, Bk, Cf
10	UO2	Uranium	UO_2	U
11	CD	More Volatile Main Group	Cd	Cd, Hg, Zn, As, Sb, Pb, Tl, Bi
12	AG	Less Volatile Main Group	Ag	Ga, Ge, In, Sn, Ag
13	BO2	Boron	BO_2	B, Si, P
14	H2O	Water	H_2O	H_2O
15	CON	Concrete	CON	- – -
16	CSI	Cesium iodide	CsI	CsI
17	CSM	Cesium Molybdate	CsM[1]	CsM[1]

[1]Cesium Molybdate (Cs2MoO4) is represented in MELCOR as CSM in order to satisfy the three character naming limitation in MELCOR

The subroutines for MELCOR execution have not been completed yet. It is being updated. The intent is to place the call for NAC package near the end of the package call sequence in MELCOR execution. In addition, the number of modules within the current MELCOR will be used, including:

- MCFTF – for control function and tabular function calls
- M_CVHFP – for CVH, HS, flow path and other MELCOR package information and data
- M_H2O – for NFLUID information and the property information

- M_NCG – for number of materials, and pointers for all gaseous materials in the problem
- M_SouCVH – for sourcing in CVH
- M_RN1 – for Radionuclide class structure information and data

5.2.1 NAC Input Record

In order to model sodium chemical reactions, the LMR reactor type must be invoked. This keyword is only used when liquid metal property is invoked through the use of the property data input file. Once detected, "LMR" reactor type is created. Note that this reactor type is different from the reactor type in the COR package, since the current sodium chemistry is valid only for the containment analysis use. Thus it is not expected that COR package is invoked.
The MELGEN input records for all chemistry models are given within the NAC package inputs. The NAC_GenerateDB subroutine will be placed as the last package call within GenerateDB subroutine within the EXEC package. This is subject to change. As required, NAC_INPUT record is necessary.

NAC_INPUT – Activation Record
Required.

This record activates the NAC package in MELCOR. When the NAC_INPUT record is absent, then by default, the activation switch is set to not active. It is required that additional aerosol classes must be defined in RN1_CC record in order for any aerosols to be appeared in the problem if this package is invoked. If this package is active and NFLUID is not equal to 7 or 20, a diagnostic message will be output and code stopped.

(1) IACTV

Activation switch for the NAC package. Optional field.

(a) 0 or ACTIVE
RN package Active

(b) 1 or NOTACTIVE
RN package Not Active

(type = integer / character*9, default=0 (active), units = none)

Example

```
NAC_INPUT ACTIVE
```

This record specifies the activation of the sodium chemistry models in MELCOR. In order to use this package records, NFLUID =7 or NFLUID = 20 must be set when the liquid metal fluid property is invoked. In addition, all sodium chemistry models are control volume specified models; therefore, it is necessary to specify the specific CVH volumes to contain these models.

The following subsections describe the MELGEN input records that are optional if the desired sodium models are invoked.

5.2.2 RN Class Mapping

Since aerosols would be generated from the sodium chemistry models, it is necessary to map the sodium product aerosols with the classes identified in the RN package. Table 5-3 lists the classes of the current identification of the RN classes. The mapping is done within the M_NAC module to track each of the RN classes to be modeled. A subroutine "NAC_RNClassConnect" will be used to map classes in the RN package and those classes specified in this record.

5.2.2.1 MELGEN Input Record

Based on the sodium chemistry models described here, only five aerosols have been identified: Na, H_2O, NaOH, Na_2O_2 and Na_2O. As shown in Table 5-3, Na is included in Class 2 of RN Package, since Class 2 is a radionuclide class. For the sodium reactors, sodium is not radioactive, so the use of Class 2 for sodium may be appropriate. A treatment for the sodium should be done in a new class. H_2O is in Class 14. No specified class can be assigned to the rest of the sodium compounds since they are products of the reactions. Therefore, it is necessary for the users to declare new classes for these sodium compounds. MELCOR will check during MELGEN execution if there are not declared and provide a warning message in the diagnostic file. Thus, no sodium compound aerosols would be generated and tracked. Thus this input represents all classes to be tracked for the entire problem, so it is necessary to map all reactants and by-products aerosol classes for the problem.

NAC_RNCLASS – Aerosol mapping
Optional

This record permits the mapping of the classes from the RN package to NAC package. Note that for sodium aerosol, the vapor pressure curve may be required to model the phase change of sodium. Thus SC7110 may be modified for sodium in the new RN class. Similarly, the molecular weight may be input. Only five input variables are required at this time. This list will be expanded as more sodium models are added.

(1) NaCL1
RN class number for water. Default is 14 (see Table x)
(type = integer, default = 14, units = dimensionless)

(2) NaCL2
RN class number for sodium. No default
(type = integer, default = none, units = dimensionless)

(3) NaCL3
RN class number for NaOH. No default.
(type = integer, default = none, units = dimensionless)

(4) NaCL4
RN class number for Na2O2. No default
(type = integer, default = none, units = dimensionless)

(5) NaCL5
RN class number for Na2O. No default

(type = integer, default = none, units = dimensionless)

5.2.3 Atmospheric Chemistry

The models representing the atmospheric chemistry for sodium are contained in CHEMRX, CHMAER, CHMGAS, CHMREP and CHMDEP. CHEMRX is the controlling subroutine for all chemistry models in the atmosphere (see Section 5.1.1 for more details). All these subroutines will be resided in the NAC package. The details where these subroutines will be called have not been completed yet. However, the interfaces are being developed to interface with other MELCOR packages. Since this model will interface with CVH, HS and RN packages, multiple calls are expected.

5.2.3.1 MELGEN Input Record

This section describes the MELGEN input variables for this model. This record requires the specification of the CVH volumes for which this model is included. This record only models the sodium chemistry in the atmosphere of the CVH volume, including on the surfaces such as heat structures and aerosols, except the floor when the sodium pool is presented.

To invoke this model for atmosphere chemistry and for combustion of sodium-hydrogen jets, minimal input is required. When this record is included, it expects the number of CVH volumes to contain this model. The required fraction FRNA2O, the fraction of sodium that produces the monoxide versus the peroxide in the atmospheric reactions of sodium with oxygen is an input variable.

NAC_ATMCHEM – Sodium Atmosphere Chemistry Model
Optional

This model allows the modeling of the atmosphere chemistry in a given control volume if the sodium is present in the atmosphere.

(1) NUM

The number of control volumes to include this model

(type = integer, default = none, units = dimensionless)

The following data are input as a table with length NUM:

(1) NC

Table row index.

(type = integer, default = none, units = none)

(2) CVHNAME

The name of the CVH volume.

(type = character, default = none, units = dimensionless)

(3) FRNA2O

The fraction of sodium that produces Na2O versus Na2O2 with oxygen (Eq. 1-3).

(type = real, default = 0.5, units = dimensionless)

5.2.4 Sodium Spray Fires

This model is an atmospheric model, which requires a sodium source to be specified either from a table or control function. Both mass and temperature are required for the source. Thus it is necessary to specify the CVH volume in which the spray fire is located. Multiple CVH volumes with spray fires can be modeled. The SPRAY subroutine from CONTAIN-LMR will be ported to the NAC package. tHydr_CVHRN1 routine in CVH package may be used to direct the chemical energy the gas space.

5.2.4.1 MELGEN Input Record

To invoke this model for sodium spray fire, a number of input variables are required. Each spray fire model requires the user to input the fall height and mean sodium droplet diameter. These parameters are used to calculate the reaction time and reaction area in the spray fire. Note that the default height is set to CVH height, while the default mean sodium droplet diameter is set at 0.001 m.

NAC_SPRAY – Sodium Spray Fire Model
Optional

This model allows the modeling of the sodium spray fire in a given control volume if the sodium spray source is given.

(1) NUM
The number of control volumes to include this model
(type = integer, default = none, units = dimensionless)

The following data are input as a table with length NUM:

(1) NC
Table row index.
(type = integer, default = none, units = none)

(2) CVHNAME
The name of the CVH volume.
(type = character, default = none, units = none)

(3) HITE
Fall height of sodium spray. Default is CVH height.
(type = real, default = CVH volume height, units = m)

(4) DME
Mean sodium droplet diameter.
(type = real, default = 0.001, units = m)

(5) SOU-TYPE
Sodium spray source type: TF or CF. Default is TF. Note that two tables are expected: mass and temperature/enthalpy

(type = character, default = TF, units = none)

(6) MASS-NAME

Name of the TF or CF for the mass source.

(type = character, default = none, units = kg/s)

(7) THERM-NAME

Name of the TF or CF for the temperature of the source.

(type = character, default = none, units = temperature)

5.2.5 Sodium Pool Fires

This model contains two sub-models: one for the fire and the other for pool heat transfer specified. The latter model is directed for the debris concrete interaction.

Sodium Pool Fire

To implement this model into MELCOR, PFIRE will be ported to MELCOR. In terms of heat generation and heat transfer for this model this model relies on subroutine tHS_HSBOIL for handling the boiling heat transfer for pools. For the chemical energy generated by the pool fire, the current CONTAIN model assumes that the user specifies the fraction of the chemical energy to pool. Then the rest of the energy would be directed to the gas space of the control volume. The similar methodology will be used for now until a more realistic model is developed for splitting this reaction energy. Using the same treatment as in the sodium spray fire model, tHydr_CVHRN1 routine may be used to direct the chemical energy from the pool fire to the gas space as well as to the pool space. Another logical place for this energy between the pool and atmosphere is in tHydr_CVHRN3 routine. In this routine, calling subroutine tHydr_CVHBL may be useful to account for the mass and energy of the sodium pool fire.

Sodium Pool Physics

In MELCOR, Subroutine tHS_HSBOIL handles the boiling heat transfer for pools. Thus, the equations modeled in the BOILER routine of CONTAIN-LMR will be ported to tHS_HSBOIL routine of MELCOR. However, many constants as shown in the equations in Section 3.1 will be included as sensitivity coefficients. tHS_HSBOIL routine is called by tHS_HSTRAN routine, which handles many heat and mass transfer from and to pool for MELCOR. The other two routines that call tHS_HSBOIL are COR_CORCNV and COR_CORRN1. Since the implementation task for this work is related to the containment analyses; therefore, no change will be made for these COR related routines. This model may not be implemented since it is related to the core-concrete interaction phenomena.

5.2.5.1 MELGEN Input Record

To invoke this model, a number of input variables are required to model sodium pool fires.

NAC_PFIRE – Sodium Pool Fire Model

Optional

This model allows the modeling of the sodium pool fire in a given control volume. A number of fraction inputs can be input

(1) NUM

The number of control volumes to include this model

(type = integer, default = none, units = dimensionless)

The following data are input as a table with length NUM:

(1) NC

Table row index.

(type = integer, default = none, units = none)

(2) CVHNAME

The name of the CVH volume.

(type = character, default = none, units = none)

(3) FO2

Fraction of the oxygen consumed that reacts to form monoxide. 1-FO2 is the remaining oxygen fraction for the reaction to form peroxide. Default is 0.5.

(type = real, default = 0.5, units = none)

(4) FHEAT

Fraction of the sensible heat from the reactions to be added to the pool. The balance will go to the atmosphere. Default is 1.0

(type = real, default = 1.0, units = none)

(5) FNA2O

Fraction of the Na_2O remains in the pool. The balance will be applied to the atmosphere as aerosols. Default is 1.

(type = real, default = 1.0, units = none)

(6) FNA2O2

Fraction of the Na_2O_2 remains in the pool. The balance will be applied to the atmosphere as aerosols. Default is 0.

(type = real, default = 0.0, units = none)

5.2.6 Two-Condensable Option

In MELCOR, the current LWR version of the code treats water as only a condensable material. The suspended water droplets in the atmosphere are referred to as fog. The atmosphere also includes water vapor and noncondensible gases. In pool, it includes liquid water and water vapor bubbles. Therefore, the current code architecture in MELCOR only allows a single coolant in the problem. When sodium property implementation into MELCOR replaces the water coolant, this prohibits another condensable to be modeled in the problem. The reason is that the replacement method used was to mask over the water coolant property variables, so that the sodium coolant property data can be used instead. This masking may reduce the possibility of adding additional condensable in the problem, which is not the case in CONTAIN code (see Section 4). To model similar way as in CONTAIN, substantial code modeling to MELCOR may be required for the Two-Condensable option. Therefore, this option may not be implemented, and instead we propose to model the following:

- Condensable will be sodium for the LMR application
- Water will be modeled as aerosols (traceable gas), since aerosol can be modeled as vapor or liquid, depending on its vapor pressure

The implementation of this model has not been started. This model will be implemented in the later part of FY16.

5.2.7 Sodium-Concrete Interaction

This model relates to the interaction of the sodium pool and the concrete floor below it. The major subroutine SLAM will be ported to MELCOR, along with calling routines which includes NATCON for sodium concrete interaction models and WTRRLS routine for mass diffusion and advection of the water release from the concrete.

The actual implementation of this model has not been started yet. This model will be implemented in the later part of FY16.

6 SUMMARY/CONCLUSIONS

This is a progress report for the MELCOR/CONTAIN LMR Integration Project, M3AR-13SN1701031. This report describes the implementation progress of the sodium models from CONTAN-LMR code to be implemented into MELCOR 2.1. Additionally, preliminary testing for verification and validation of the selected sodium models for CONTAIN-LMR has been conducted. Although CONTAIN-LMR utilized the earlier version of CONTAIN code, additions of the missing sodium models from CONTAIN-LMR to CONTAIN 2 would represent a latest CONTAIN development tool for the code-to-code comparison to MELCOR 2.1, when the sodium models have been implemented. Section 4 describes the effort made to develop CONTAIN2-LMR code. However, it is nearly complete, because of the issues related to the lower cell modeling and interfaces. Additional time is required to fully test CONTAIN2-LMR with the CONTAIN 2 standard tests and the sodium specific test decks as shown in the FY14 report (SAND2014-19183). In Section 5, the implementation progress into MELCOR 2.1 is described. In Section 5.1, all sodium models planned to implement into MELCOR are discussed – atmospheric chemistry, sodium spray fire and sodium pool fire. The data structures and MELGEN input record formats have been developed. They are being implemented into MELCOR 2.1. Additionally, the two-condensable option and sodium-concrete interaction models have been discussed in terms of the models and CONTAIN coding. However, the actual implementation has not been started.

Since this report is a "living" document, the prior year's accomplishment is also included in this report, such as the preliminary implementation of the sodium thermophysical properties into MELCOR 2.1 from both works done from INL as FSD database and the BRISC LDRD as SIMMER database. To minimize the impact, the implementation of the FSD was done by utilizing the detection of the data input file as a way to invoking the FSD. This methodology has been adapted currently for this work, but it may subject modification as the project becomes mature. The implementation of the initial stage of the sodium properties into MELCOR has been completed- all essential interfaces to replace water with sodium and all input processing. Additional work was done to ensure the FORTRAN coding of the sodium property subroutines, modules and functions are within the same code development guidelines as in MELCOR and is easily maintained. Documentation on the usage of the multi-fluid version of MELCOR is being developed. The equation sets and pertinent information on the sodium properties implemented into MELCOR have been provided in this report. Preliminary testing was performed to ensure that the sodium properties have been implemented properly. However, refinement of the properties and any interpolation of the ranges of the property correlation are still needed. In addition, a refinement of the sodium properties implemented in MELCOR will be performed with additional testing of the sodium models over a broad range of possible states, including the condensation of sodium on surfaces (i.e., aerosol surfaces and heat structure surfaces).

In the coming FY, it is intended to implement all sodium models as described in Chapter 5 of this report into MELCOR 2.1. All sodium models, except the debris concrete interaction model should be completely implemented into MELCOR 2.1. These models will be validated with a number of available experiments and those experiments done in SNL.

Intentionally Leave Blank

7 REFERENCES

[Blink 1982] Blink, J.A., **Lithium Equation-Of-State**, UCID-19890 (DE83 017988), Lawrence Livermore Laboratory, Livermore California, December 30, 1982.

[Beiriger 1973] Beiriger, P., et.al, **SOFIRE II User Report, AI-AEC-13055,** Atomics International Division, Canoga Park, California 1973.

[Cahalan 1994] Cahalan, J.E., et al., "Advanced LMR Safety Analysis Capabilities in the SASSYS-1 and SAS4A Computer Codes," Proceedings of the International Topical Meeting on Advanced Reactors Safety, Pittsburgh PA, 1994.

[Denman 2012] Denman, M.R., et al., **Sodium Fast Reactor Safety and Licensing Research Plan - Volume I,** SAND2012-4260, Sandia National Laboratories, Albuquerque, New Mexico, May 2012.

[Dunn 2012] Dunn, F.E., **The SAS4A/SASSYS-1 Safety Analysis Code System – Chapter 12:** Sodium **Voiding Model,** ANL/NE-12/4, Argonne National Laboratory, Argonne, Illinois, January 2012.

[Fink 1979] Fink, J.K., and Leibowitz, L., **Thermophysical Properties of Sodium,** ANL-CEN-RSD-79-1, Argonne National Laboratory, Argonne, Illinois, May 1979.

[Fink 1995] Fink, J.K., and Leibowitz, L., **Thermodynamic and Transport Properties of Sodium Liquid and Vapor**, ANL/RE-95/2, 1995.

[Gierszewski 1980] Gierszewski, P., et.al, **Property Correlations for Lithium, Sodium, Helium, FLIBE and Water in Fusion Reactor Applications**, PFC-RR-80-12, Fusion Blanket and Structure Group, Massachusetts Institute of Technology, August 1980.

[Jeppson 1986] Jeppson, D.W., **Results and Code Predictions of the SOFICOV Large-Scale Sodium Fire Spray Test SAl,** HEDL-TME 85-20, Hanford Engineering Development Laboratory, Richland, Washington, February 1986.

[Humphries 2015a] Humphries, L.L., Cole, R.K., Louie, D.L., Figueroa, V.G., and Young, M.F., **MELCOR Computer Code Manuals – Vol.1: Primer and Users' Guide, Version 2.1.6840 2015**, SAND2015-6691R, Sandia National Laboratories, Albuquerque, NM, August 2015.

[Humphries 2015b] Humphries, L.L., Cole, R.K., Louie, D.L., Figueroa, V.G., and Young, M.F., **MELCOR Computer Code Manuals- Vol.2: Reference Manual, Version 2.1.6840 2015**, SAND2015-6692 R, Sandia National Laboratories, Albuquerque, NM, August 2015.

[Humphries 2015c] Humphries, L.L., Louie, D.L., Figueroa, V.G., Young, M.F., Weber, S., Ross, K., Phillips, J. and Jun, R.J., **MELCOR Computer Code Manuals-Vol.3: MELCOR Assessment Problems, Version 2.1 6850 2015**, SAND2015-6693 R, Sandia National Laboratories, Albuquerque, NM, August 2015.

[Merrill 2000] Merrill, B.J., **Modifications Made to the MELCOR Code for Analyzing Lithium Fires in Fusion Reactors**, INEEL/EXT-2000-00489, Idaho National Engineering and Environmental Laboratory, Idaho Falls, Idaho, April 2000.

[Morita 1998a] Morita, K., Fischer, E.A., "Thermodynamic Properties and Equations of State for Fast Reactor Safety Analysis, Part1: Analytic equation-of-state model," Nuclear Engineering and Design, Volume 183, pp. 177-191, 1998.

[Morita 1998b] Morita, K., et.al, "Thermodynamic Properties and Equations of State for Fast Reactor Safety Analysis, Part II: Properties of Fast Reactor Materials," Nuclear Engineering and Design, Volume 183, pp. 193-211, 1998.

[Murata 1993] Murata, K.K., et al., **CONTAIN LMR/1B-Mod.1, A Computer Code for Containment Analysis of Accidents in Liquid-Metal Cooled Nuclear Reactors**, SAND91-1490, Sandia National Laboratories, Albuquerque, New Mexico, 1993.

[Murata 1997] Murata, K.K., et al., **Code Manual for CONTAIN 2.0: A Computer Code for Nuclear Reactor Containment Analysis,** SAND97-1735, NUREG/Cr-6533, Sandia National Laboratories, Albuquerque, New Mexico, December 1997.

[Schmidt 2011] Schmidt, R., et.al, **Sodium Fast Reactor Gaps Analysis of Computer Codes and Models for Accident Analysis and Reactor Safety,** SAND2011-4145, Sandia National Laboratories, Albuquerque, New Mexico, June 2011.

[Scholtyssek 1994] Scholtyssek, W., and Murata, K., **Sodium Spray and Jet Fire Model Development within the CONTAIN-LMR Code,** SAND93-2200C, Sandia National Laboratories, Albuquerque, New Mexico, 1994.

[Souto 1994] Souto, F.J., et.al, **MELCOR 1.8.2 Assessment: Aerosol Experiments ABCOVE AB5, AB6, AB7 and LACE LA2,** SAND94-2166, Sandia National Laboratories, Albuquerque, New Mexico 1994.

[Suo-Anttila 1983] Suo-Anttila, A.,SLAM – **A Sodium-Limestone Concrete Ablation Model**, NUREG/CR-3379, SAND83-7114, Sandia National Laboratories, Albuquerque, New Mexico, December 1983.

[Tills 2008] Tills, J., et al., **An Assessment of MELCOR 1.86: Design Basis Accident Tests of the Carolinas Virginia Tube Reactor (CVTR)**

Containment (Including Selected Separate Effects Tests), SAND2008-1224, Sandia National Laboratories, Albuquerque, New Mexico, February 2008.

[Tills 2009] Tills, J., et al., **Application of the MELCOR Code to Design Basis PWR Large Dry Containment Analysis,** SAND2009-2858, Sandia National Laboratories, Albuquerque, New Mexico, May 2009.

[Tills 2010] Tills, J., **MELCOR DBA Containment Audit Calculations for the ESBWR Plant (Final),** a partial fulfillment of the NRC Contract Q4157 Task Order 5, Sandia National Laboratories, Albuquerque, New Mexico, November 2010.

[Tsai 1980] S.S. Tsai, **The NACOM Code for Analysis of Postulated Sodium Spray Fires in LMFBRs,** NUREG/CR-1405, Brookhaven National Laboratory, Upton, NY March 1980.

[Westrich 1983] Westrich, H., et al, **Laboratory-Scale Sodium-Carbonate Aggregate Concrete Interactions**, NUREG/CR-3401, SAND83-1502, Sandia National Laboratories, Albuquerque, New Mexico, July 1983.

[Young 1977] Young, D.A., **A Soft-Sphere Model for Liquid Metals,** UCRL-52352, Lawrence Livermore Laboratory, Livermore, California, November 8 , 1977.

Intentionally Leave Blank

APPENDIX A: INPUT/OUPTUT FILES FOR NALIBRARY PROGRAM

This appendix tabulates the input and output for the NaLibrary Program that is used with FSD data set. Table A-1 shows the INL data input file, Table A-2 shows output of the data limits and table statistics. Table A-3 shows an example of the saturation property table of temperature versus pressure. Table A-4 shows an example of the thermodynamic property table.

Table A-1. Input File

```
tpfna version 1.1.1, tables of thermodynamic properties of sodium

  84 temperatures as follows:

      371.
      400.   450.
      500.   550.
      600.   620.   640.   660.   680.
      700.   710.   720.   730.   740.   750.   760.   770.   780.   790.
      800.   810.   820.   830.   840.   850.   860.   870.   880.   890.
      900.   910.   920.   930.   940.   950.   960.   970.   980.   990.
     1000.  1010.  1020.  1030.  1040.  1050.  1060.  1070.  1080.  1090.
     1100.  1110.  1120.  1130.  1140.  1150.  1160.  1170.  1180.  1190.
     1200.  1220.  1240.  1260.  1280.
     1300.  1350.
     1400.  1450.
     1500.  1600.  1700.  1800.  1900.
     2000.  2100.
     2200.  2300.  2400. 2500. 2510.0 2600. 2800.0 3000.

  28 pressures as follows:

     2.13652E-05
     1.00e-4  5.00e-4
     1.00e-3  5.00e-3
     1.00e-2  5.00e-2
     1.00e-1  5.00e-1
     1.00e+0  5.00e+0
     1.00e+1  5.00e+1
     1.00e+2  5.00e+2
     1.00e+3  5.00e+3
     1.00e+4  5.00e+4
     1.00e+5  5.00e+5
     1.00e+6  5.00e+6
     1.0E+07
     2.0E+07
     25624700.0
     4.e7
     5.e7
```

Table A-2. Output File – Data Limits and Table Statistics

```
Data limits:

                                   temperature      pressure         volume
                                   -----------     -----------     -----------
   minimum allowed input value =     3.71000E+02     1.61000E-06         n/a
   maximum allowed input value =     3.00000E+03     5.00000E+07         n/a
   triple point value          =     3.71000E+02     2.11953E-05     1.07896E-03
   critical point value        =     2.45000E+03     3.20663E+07     6.32824E-03
Table statistics:
   number of temperatures            =    84
   number of pressures               =    28
   number of saturation temperatures =    79
   number of saturation pressures    =    26
   number of words in binary tables  = 15589
```

Table A-3. Output File – Example of Saturation Properties of Sodium (Temperature vs. Pressure)

temperature	pressure	state	specific volume	internal energy	thermal expansion	isothermal compress.	specific heat	entropy
3.71000E+02	2.11953E-05	liquid	1.07896E-03	2.05903E+05	2.96659E-04	1.95989E-10	1.19914E+03	2.62650E+03
		vapor	6.33173E+09	4.86030E+06	2.69542E-03	4.71803E+04	9.04333E+02	1.46920E+04
4.00000E+02	2.45341E-04	liquid	1.08827E-03	2.40575E+05	2.95561E-04	2.02010E-10	1.19228E+03	2.71649E+03
		vapor	5.89763E+08	4.87604E+06	2.50001E-03	4.07598E+03	9.04334E+02	1.38743E+04
4.50000E+02	7.91868E-03	liquid	1.10444E-03	2.99952E+05	2.94700E-04	2.12806E-10	1.18333E+03	2.85637E+03
		vapor	2.05560E+07	4.90316E+06	2.22230E-03	1.26285E+02	9.04345E+02	1.27240E+04
5.00000E+02	1.26705E-01	liquid	1.12083E-03	3.58957E+05	2.94929E-04	2.24191E-10	1.17732E+03	2.98071E+03
		vapor	1.42732E+06	4.93027E+06	2.00031E-03	7.89279E+00	9.04388E+02	1.18163E+04
5.50000E+02	1.21779E+00	liquid	1.13751E-03	4.17723E+05	2.96075E-04	2.36239E-10	1.17371E+03	3.09274E+03
		vapor	1.63321E+05	4.95732E+06	1.81907E-03	8.21304E-01	9.04509E+02	1.10838E+04
6.00000E+02	7.98980E+00	liquid	1.15453E-03	4.76362E+05	2.98021E-04	2.49029E-10	1.17213E+03	3.19478E+03
		vapor	2.71428E+04	4.98426E+06	1.66878E-03	1.25218E-01	9.04791E+02	1.04819E+04
6.20000E+02	1.55566E+01	liquid	1.16144E-03	4.99803E+05	2.99005E-04	2.54372E-10	1.17201E+03	3.23321E+03
		vapor	1.44005E+04	4.99498E+06	1.61576E-03	6.43230E-02	9.04974E+02	1.02704E+04
6.40000E+02	2.90363E+01	liquid	1.16842E-03	5.23244E+05	3.00101E-04	2.59853E-10	1.17215E+03	3.27042E+03
		vapor	7.96092E+03	5.00565E+06	1.56628E-03	3.44702E-02	9.05209E+02	1.00732E+04
6.60000E+02	5.21540E+01	liquid	1.17547E-03	5.46690E+05	3.01306E-04	2.65478E-10	1.17254E+03	3.30650E+03
		vapor	4.56831E+03	5.01627E+06	1.52006E-03	1.91968E-02	9.05507E+02	9.88899E+03
6.80000E+02	9.04556E+01	liquid	1.18259E-03	5.70147E+05	3.02618E-04	2.71254E-10	1.17319E+03	3.34151E+03
		vapor	2.71203E+03	5.02683E+06	1.47685E-03	1.10724E-02	9.05876E+02	9.71656E+03
7.00000E+02	1.51948E+02	liquid	1.18979E-03	5.93619E+05	3.04034E-04	2.77188E-10	1.17407E+03	3.37553E+03
		vapor	1.66068E+03	5.03730E+06	1.43643E-03	6.59448E-03	9.06328E+02	9.55487E+03
7.10000E+02	1.94752E+02	liquid	1.19341E-03	6.05363E+05	3.04781E-04	2.80216E-10	1.17460E+03	3.39219E+03
		vapor	1.31360E+03	5.04251E+06	1.41721E-03	5.14641E-03	9.06588E+02	9.47775E+03
7.20000E+02	2.47871E+02	liquid	1.19706E-03	6.17111E+05	3.05553E-04	2.83287E-10	1.17518E+03	3.40862E+03
		vapor	1.04612E+03	5.04769E+06	1.39862E-03	4.04469E-03	9.06872E+02	9.40298E+03
7.30000E+02	3.13364E+02	liquid	1.20073E-03	6.28866E+05	3.06351E-04	2.86400E-10	1.17582E+03	3.42484E+03
		vapor	8.38526E+02	5.05284E+06	1.38063E-03	3.20035E-03	9.07183E+02	9.33045E+03
7.40000E+02	3.93614E+02	liquid	1.20442E-03	6.40628E+05	3.07175E-04	2.89557E-10	1.17652E+03	3.44084E+03
		vapor	6.76320E+02	5.05797E+06	1.36322E-03	2.54872E-03	9.07520E+02	9.26007E+03
7.50000E+02	4.91364E+02	liquid	1.20813E-03	6.52397E+05	3.08024E-04	2.92759E-10	1.17727E+03	3.45663E+03
		vapor	5.48752E+02	5.06307E+06	1.34639E-03	2.04244E-03	9.07886E+02	9.19174E+03
7.60000E+02	6.09753E+02	liquid	1.21186E-03	6.64174E+05	3.08898E-04	2.96007E-10	1.17808E+03	3.47223E+03
		vapor	4.47799E+02	5.06814E+06	1.33011E-03	1.64653E-03	9.08282E+02	9.12538E+03
7.70000E+02	7.52358E+02	liquid	1.21562E-03	6.75958E+05	3.09797E-04	2.99302E-10	1.17894E+03	3.48764E+03
		vapor	3.67428E+02	5.07317E+06	1.31436E-03	1.33501E-03	9.08710E+02	9.06092E+03
7.80000E+02	9.23231E+02	liquid	1.21939E-03	6.87752E+05	3.10722E-04	3.02645E-10	1.17985E+03	3.50286E+03
		vapor	3.03075E+02	5.07816E+06	1.29913E-03	1.08842E-03	9.09170E+02	8.99827E+03
7.90000E+02	1.12694E+03	liquid	1.22319E-03	6.99556E+05	3.11672E-04	3.06037E-10	1.18081E+03	3.51789E+03
		vapor	2.51261E+02	5.08312E+06	1.28440E-03	8.92108E-04	9.09664E+02	8.93735E+03
8.00000E+02	1.36864E+03	liquid	1.22702E-03	7.11369E+05	3.12647E-04	3.09480E-10	1.18183E+03	3.53275E+03
		vapor	2.09321E+02	5.08804E+06	1.27016E-03	7.34953E-04	9.10193E+02	8.87812E+03

Table A-4. Output File – Example of Thermodynamic Properties of Sodium

pressure	temperature	state	specific volume	internal energy	thermal expansion	isothermal compress.	specific heat	entropy
2.13652E-05	3.71000E+02	liquid	1.07896E-03	2.05903E+05	2.96659E-04	1.95989E-10	1.19914E+03	2.62650E+03
	4.00000E+02	vapor	6.77237E+09	4.87604E+06	2.50000E-03	4.68051E+04	9.04333E+02	1.47572E+04
	4.50000E+02	vapor	7.61892E+09	4.90317E+06	2.22222E-03	4.68051E+04	9.04333E+02	1.48637E+04
	5.00000E+02	vapor	8.46546E+09	4.93030E+06	2.00000E-03	4.68051E+04	9.04333E+02	1.49590E+04
	5.50000E+02	vapor	9.31201E+09	4.95743E+06	1.81818E-03	4.68051E+04	9.04333E+02	1.50452E+04
	6.00000E+02	vapor	1.01586E+10	4.98456E+06	1.66667E-03	4.68051E+04	9.04333E+02	1.51239E+04
	6.20000E+02	vapor	1.04972E+10	4.99541E+06	1.61290E-03	4.68051E+04	9.04333E+02	1.51535E+04
	6.40000E+02	vapor	1.08358E+10	5.00626E+06	1.56250E-03	4.68051E+04	9.04333E+02	1.51823E+04
	6.60000E+02	vapor	1.11744E+10	5.01712E+06	1.51515E-03	4.68051E+04	9.04333E+02	1.52101E+04
	6.80000E+02	vapor	1.15130E+10	5.02797E+06	1.47059E-03	4.68051E+04	9.04332E+02	1.52371E+04
	7.00000E+02	vapor	1.18517E+10	5.03882E+06	1.42857E-03	4.68051E+04	9.04332E+02	1.52633E+04
	7.10000E+02	vapor	1.20210E+10	5.04425E+06	1.40845E-03	4.68051E+04	9.04332E+02	1.52761E+04
	7.20000E+02	vapor	1.21903E+10	5.04967E+06	1.38889E-03	4.68051E+04	9.04332E+02	1.52888E+04
	7.30000E+02	vapor	1.23596E+10	5.05510E+06	1.36986E-03	4.68051E+04	9.04332E+02	1.53012E+04
	7.40000E+02	vapor	1.25289E+10	5.06052E+06	1.35135E-03	4.68051E+04	9.04332E+02	1.53135E+04
	7.50000E+02	vapor	1.26982E+10	5.06595E+06	1.33333E-03	4.68051E+04	9.04332E+02	1.53257E+04
	7.60000E+02	vapor	1.28675E+10	5.07138E+06	1.31579E-03	4.68051E+04	9.04332E+02	1.53377E+04
	7.70000E+02	vapor	1.30368E+10	5.07680E+06	1.29870E-03	4.68051E+04	9.04332E+02	1.53495E+04
	7.80000E+02	vapor	1.32061E+10	5.08223E+06	1.28205E-03	4.68051E+04	9.04332E+02	1.53612E+04
	7.90000E+02	vapor	1.33754E+10	5.08765E+06	1.26582E-03	4.68051E+04	9.04332E+02	1.53727E+04
	8.00000E+02	vapor	1.35447E+10	5.09308E+06	1.25000E-03	4.68051E+04	9.04332E+02	1.53840E+04
	8.10000E+02	vapor	1.37141E+10	5.09851E+06	1.23457E-03	4.68051E+04	9.04332E+02	1.53953E+04
	8.20000E+02	vapor	1.38834E+10	5.10393E+06	1.21951E-03	4.68051E+04	9.04332E+02	1.54064E+04
	8.30000E+02	vapor	1.40527E+10	5.10936E+06	1.20482E-03	4.68051E+04	9.04332E+02	1.54173E+04
	8.40000E+02	vapor	1.42220E+10	5.11478E+06	1.19048E-03	4.68051E+04	9.04332E+02	1.54282E+04
	8.50000E+02	vapor	1.43913E+10	5.12021E+06	1.17647E-03	4.68051E+04	9.04332E+02	1.54389E+04
	8.60000E+02	vapor	1.45606E+10	5.12564E+06	1.16279E-03	4.68051E+04	9.04332E+02	1.54494E+04
	8.70000E+02	vapor	1.47299E+10	5.13106E+06	1.14943E-03	4.68051E+04	9.04332E+02	1.54599E+04
	8.80000E+02	vapor	1.48992E+10	5.13649E+06	1.13636E-03	4.68051E+04	9.04332E+02	1.54702E+04
	8.90000E+02	vapor	1.50685E+10	5.14191E+06	1.12360E-03	4.68051E+04	9.04332E+02	1.54805E+04
	9.00000E+02	vapor	1.52378E+10	5.14734E+06	1.11111E-03	4.68051E+04	9.04332E+02	1.54906E+04
	9.10000E+02	vapor	1.54072E+10	5.15277E+06	1.09890E-03	4.68051E+04	9.04332E+02	1.55006E+04
	9.20000E+02	vapor	1.55765E+10	5.15819E+06	1.08696E-03	4.68051E+04	9.04332E+02	1.55104E+04
	9.30000E+02	vapor	1.57458E+10	5.16362E+06	1.07527E-03	4.68051E+04	9.04332E+02	1.55202E+04
	9.40000E+02	vapor	1.59151E+10	5.16904E+06	1.06383E-03	4.68051E+04	9.04332E+02	1.55299E+04
	9.50000E+02	vapor	1.60844E+10	5.17447E+06	1.05263E-03	4.68051E+04	9.04332E+02	1.55395E+04
	9.60000E+02	vapor	1.62537E+10	5.17990E+06	1.04167E-03	4.68051E+04	9.04332E+02	1.55489E+04
	9.70000E+02	vapor	1.64230E+10	5.18532E+06	1.03093E-03	4.68051E+04	9.04332E+02	1.55583E+04
	9.80000E+02	vapor	1.65923E+10	5.19075E+06	1.02041E-03	4.68051E+04	9.04332E+02	1.55676E+04
	9.90000E+02	vapor	1.67616E+10	5.19617E+06	1.01010E-03	4.68051E+04	9.04332E+02	1.55768E+04
	1.00000E+03	vapor	1.69309E+10	5.20160E+06	1.00000E-03	4.68051E+04	9.04332E+02	1.55858E+04
	1.01000E+03	vapor	1.71002E+10	5.20703E+06	9.90099E-04	4.68051E+04	9.04332E+02	1.55948E+04
	1.02000E+03	vapor	1.72696E+10	5.21245E+06	9.80392E-04	4.68051E+04	9.04332E+02	1.56038E+04
	1.03000E+03	vapor	1.74389E+10	5.21788E+06	9.70874E-04	4.68051E+04	9.04332E+02	1.56126E+04

APPENDIX B: CONTAIN TESTING

This appendix documents the testing done to both CONTAIN-2 and CONTAIN-LMR codes after they were bought to the modern Software Quality Assurance practice. Table B-1 shows the list of the standard test problems were used to ensure CONTAIN-2 were upgraded correctly. For the CONTAIN-LMR code, there were no standard test problems designed for this code. The standard test problems used in CONTAIN-2 were designed for light water reactors, so these tests would not be suitable for testing CONTAIN-LMR code. However, we tested CONTAIN-LMR code with a test problem specifically designed for this code. Additionally, we provided demonstration problems to test out CONTAIN-LMR code for the specific sodium models as described in Chapter 5 of this document. Note that CONTAIN2-LMR also uses these tests as verification tests. This testing is still on-going until sodium testing is completed.

B.1 CONTAIN 2

This section discusses the testing for CONTAIN 2. Since there was no test result available, the testing is done by comparing the last time point calculation results from the outputs to those calculations done by the executables created in March 20, 2008 using a different Window Fortran compiler. Table B.1-1 shows the latest results using the STD tests conducted. As shown in this table, the latest version of CONTAIN 2 is working as indicated.

Table B.1-1. CONTAIN 2 Testing

STD Test No	Description	Run?	Comparison
1	ST01.ac --- condensation model w/forced convection, adapted from ac23(st)	Yes	Result is identical to existing executable*
2	ST02.af --- aerosol fall through check, adapted from af06(st)	Yes	Result is identical to existing executable*
3	ST03.af --- intercell aerosol flow test with fps, adapted from af07(st), and modified for light water reactors.	Yes	Result is identical to existing executable*
4	ST04.cf --- intercell gas flow test (adiabatic flow), adapted from cf09(st), but modified for light water reactors.	Yes	Result is identical to existing executable*
5	ST05.cv --- corcon/vanesa standard problem, adapted from cv04(st)	Yes	Existing executable* aborts on this input. Use last edit from CONTAIN 1.2 testing - results look very similar. Minor differences are observed.
6	ST06.ev --- engineered vent test, adpated from ev05(st)	Yes	Result is identical to existing executable*
7	ST07.ft --- fission product transport, adapted from ft02(st)	Yes	Result is identical to existing executable*
8	ST08.hb --- hydrogen burn test, adapted from hb04(st), but converted to thermal reactor	Yes	Result is identical to existing executable*
9	ST09.ic --- ice condenser test, adapted from ic02(st)	Yes	Result is similar to existing executable*up tp the point when all ices were melted. However, there should not be additional ice to be melted as predicted by the existing executable*. The latest executable predicts the ice melt and accumulation including vapor mass from atmosphere is correct.
10	ST10.ih --- test of fission product decay heating, adapted from ih11(st)	Yes	Result is identical to existing executable*
11	ST11.ih --- test of the engineered safety features, adapted from ih20(st)	Yes	Results are very similar to that of the existing executable*- minor differences in energy, mass and flow rates
12	ST12.ih --- fission product transport, adapted from ih22(st)	Yes	Result is identical to existing executable*

STD Test No	Description	Run?	Comparison
13	ST13.it --- integrated workshop problem, adapted from it01(st)	Yes	Results are very similar to that of the existing executable*- minor differences in energy, mass and flow rates
14	ST14.rh --- radiation enclosure problem, adapted from rh04(st)	Yes	Result is identical to existing executable*
15	ST15.bw --- bwr test, spv and srv with pool boiling, adapted from bw14(st)	Yes	Result is identical to existing executable*
16	ST16.cs --- connected structure option test, adapted from cs01	Yes	Result is identical to existing executable*
17	ST17.ff --- film flow test with fission products, adapted from ff01	Yes	Result is identical to existing executable*
18	ST18.ht --- condensation and ht test problems	Yes	Results are very similar to that of the existing executable* - minor differences in energy, aerosol, mass and flow rates
18a	ST18.ht --- condensation and ht test problems - slightly different cell elevation	Yes	Results are very similar to that of the existing executable*- minor differences in energy, aerosol, mass and flow rates
19	ST19.pt --- pool tracking test with drain-down, adapted from pt01	Yes	Results are very similar to that of the existing executable*- minor differences in energy, aerosol, mass and flow rates
20	ST20.pd --- grand gulf plant deck	Yes	Results are very similar to that of the existing executable*- very minor differences
21	ST21.df --- diffusion frame burn test, adapted from dfb05	Yes	Results are very similar to that of the executable from MELCOR shared drive- very minor differences
22	ST22.eo --- non-ideal equation of state water test, adapted from eo01	Yes	Result is identical to existing executable*
23	ST23.fp --- fission product library test, adapted from fpd01	Yes	Results are very similar to that of the existing executable* - very minor differences in FP masses
24	ST24.lc --- lower cell heat conduction test, adapted from lch01	Yes	Result is identical to existing executable*

*Existing executable means the executable dated March 20, 2008.

B.2 CONTAIN-LMR and CONTAIN2-LMR

In this section, the description of the validation and verification tests to CONTAIN-LMR and CONTAIN2-LMR is provided. Note that CONTAIN2-LMR can model both LWR and sodium reactors. Therefore, Section B.1 can be applied to CONTAIN2-LMR. Since the testing and debugging are not completely done to CONTAIN2-LMR (see Section 4), only the sodium spray fire test was documented in this section for this code.

B.2.1 Two Condensable Option

This section describes the demonstration input for the Two Condensable Option within the atmosphere input options (see Section 5.1.4). This option allows the modeling of both sodium and water condenses in the same problem. The keyword TWOCOND will activate this model. This keyword is placed in the MATERIAL block of the input. However, a keyword of FIRST and SECOND for each of condenses must be declared. Table B.2.1-1 shows the demonstration input deck for exercising this option. This deck was modified from the sample problem documented in SAND91-1490. The keyword to activate this model is "TWOCOND". There was no experiment to be used for verifying or validating this model. Thus no further discussion of the option is provided. Note that this input contains the SLAM model input. If the SLAM model input is not working properly, this input can't use for the verification purpose.

Table B.2.1-1 Demonstration Input for the Two-Condensable Option

```
&& cray
&& slam-corcon-debris-bed test problem
&& (from appendix a of sand91-1490)
&& this problem is a modification of indb4. a .1 see slam
&& timestep is used to control limit cycles and assure
&& consistent slam results. the end time for slam is set
&& to 500 seconds and a transition to corcon is forced at
&& 10000 seconds, because slam cannot operate
&& concurrently with the debris bed, the debris bed is
&& introduced at 500 sec. the bed area has been modified
&& to allow quench at this time, and the bed power
&& increased to allow reasonable corcon interactions.
&&
&& ---------- global input ----------
control ncells=2 ntitl=2 ntzone=8 nsectn = 10 nac=4 eoi
material
compound h2 o2 co co2 h2ol h2ov fe n2 conc uo2 pu
  u mno mgo k2o nav nal uo2 puo2 cao ssox fel
  na2o2 na2o naoh na
  twocond
    first=nav
    second=h2ov
    cellcond nav nav
  eoi
times 60000. 0.
&& ---------- time zones ----------
  5,   10,   1500.
  10.  30.   2400.
  5.   30.   3000.
  10.  60.   5000.
  10. 120.   9500.
  10.  30.  10000.
  10. 120.  16000.
  10. 120.  20000.
&& ----------
1. 1.
longedt 2

fast
flows
  area(1,2)=1. cfc(1,2)=.5 avl(1,2) =.5
  implicit
  dropout
aerosol
  deldif = 1.e-4 densty=300. tgas1=328. pgas1=1.e+5
  diam1 =1.e-8 tgas2=3000. pgas2=10.e+5
  na2o2=1.e-6 .531
  na2o=1.e-6 .531
  naoh=1.e-6 .531
  nal=1.e-6 .531
prlow-cl
prheat
prflow
praer
title
corcon/breeder test problem
        cna002 - mod - Two condensable demonstration
&& ---------- end of global input ----------
cell =2
 control
   nhtm =3 mxslab = 10 numtbc=2 maxtbc=3
   jconc =50 jint =2 jpool = 1 nraycc=50 ndblyr=4
 eoi
 geometry 1570. 9.4
 atmos=3 1.e+5 328.
```

```
        n2=.79 o2=.2 nav=.01
 condense
&& atmchem
&& h-bum
&& ------------ structures ----------------------
struc
&& --------------- cavity wall ---------------
cavwall wall cylinder 10 2 380. 9.4 0. 9.4
 7.30 7.40 7.50 7.60 7.70 7.80 7.90 8.00 8.10 8.20 8.30
 conc conc conc conc conc conc conc conc conc conc conc
&& --------------- cavity wall --------------------
cavwa12 wall cylinder 10 2 380. 9.4 0. 9.4
 7.30 7.40 7.50 7.60 7.70 7.80 7.90 8.00 8.10 8.20 8.30
 conc conc conc conc conc conc conc conc conc conc conc
&& --------------- cavity roof ---------------------
cavroof roof slab 10 2 380. 7.3 0. 167.
 0.00 0.10 0.20 0.30 0.40 0.50 0.60 0.70 0.80 0.90 1.00
 conc conc conc conc conc conc conc conc conc conc conc
&&
rad-heat
  emsvt .8 .8 .8 .8
  gaswal 7.4
eoi
ht-tran on on on on on
low-cell
  geometry 167. && floor area
  decay-ht=14.e+8
  dist-pwr 0. 1. 1. 0. 0. eoi
  concrete
&& concrete composition specification
&& use corcon's crbr concrete type, contain's generic
 compos = 1
  concrete = limena
    rhocon =2250. rebar =.14 tablat = 1650. emconc =.9
  eoi
  1.3e6 && concrete mass
  temp=300.
  physics
   na-conc
    model = slam
    times 0. 500. .1
   eoi
   corcon
    times 5000. 1
    20. 20. 100. 55000.
    geometry 0. 1.
      flatcyl
        0. 7.29 1.5 .5 10. 10. 10 10
    emisiv
      oxide
        time 2
          0. .9 6.e+8 .9
      metal
        time 2
          0. .9 6.e+8 .9
      surrnd
        time 2
          0. .9 6.e+8 .9
    eoi && terminate emisiv option
&& ischem && keyword for ideal solution chemistry
   eoi && terminate corcon option
&&
  eoi && terminate physics
&&
 eoi && terminate the layer
&& --------------- corcon melt layer specification -------------
 interm
  lay-nam = ccmelt
  compos = corcon
    oxides =4
```

```
      cao 2000.
      fe3o4 13128.1
      uo2 25278.
      puo2 1151.
    metals = 1
      fe 19557.
      toxide = 2000.
      tmetal=2500.
      layers=0
 eoi && terminate corcon compos
 temp = 2500.
 physics
   corestat
   0. 70.1851  2441. 0.
 eoi && terminate physics
eoi && terminate the layer
interm
 lay-name =debbed
 temp=2500.
 physics
   deb-bed && debris bed model
     times
       tdbstr=500. && debris start time
       timmlt= 10000. && debris remelt time
     eoi && terminate times block
     height
       total =.1875 && total debris bed height
       dry =.1875 && dry debris bed height
     eoi && terminate height block
     dbarea=50.
     dporos =.5 && bed porosity
     ddia = .005 && particle diameter
     tmpmlt=2550. && remelt temperature
     expmul = .01 && exposed bed ht. multiplier
   eoi && terminate debris input
 eoi && terminate physics
eoi && terminate interm layer
pool
 compos 1 nal 38000.
  temp 873.
  physics
   boil
   ht-coef
    name = atmos
    var-x=time x=2,0.,400000.
    var-y=coef y=2,22.,22.
   eoi
  eoi
eoi && terminate the layer
bc 300.
eoi && terminate lower cell input
&& ---------------------- end cell 2 -------------------------
cell = 1
 control
  nhtm =3 mxslab = 10
 eoi
 geometry 68500. 79.
 atmos=4 1.018e+5 300.
       n2=.74 o2=.15 nav=.01 h2ov=.10
 condense
&& ------------ structures ----------------------
struc
&& --------------- confinement wall --------------------
  conwall wall cylinder 5 6 300. 79. 0. 300. 79.
  25.00 25.08 25.15 25.23 25.30 25.38
  fe fe fe fe fe
&& --------------- confinement wall --------------------
  conwa12 wall cylinder 5 6 300. 79. 0. 300. 79.
  25.00 25.08 25.15 25.23 25.30 25.38
  fe fe fe fe fe
```

```
&& ---------- confinement roof ----------
  conroof roof slab 5 6 300. 7.3 0. 300. 167.
  0.00 0.08 0.15 0.23 0.30 0.38
  fe fe fe fe fe
&&
ht-tran on on on on on
eof
```

B.2.2 Atmosphere Chemistry

This section describes the demonstration input for the atmosphere chemistry. To invoke this model, the ATMCHEM keyword in the cell atmosphere block should be declared. Note that there were two principle chemical reactions considered in this model: $Na + H_2O(l) \rightarrow NaOH + 0.5H_2$ and $2Na + H_2O(v) \rightarrow Na_2O + H_2$. Note that the second equation dominates if water vapor fraction is larger than the oxygen fraction in the atmosphere. Another option is that the user can enter the fraction of sodium that produces the monoxide versus the peroxide. Table B.2.2-1 shows the demonstration input deck for this model. As shown in this table, the model is activated in Cell 2, where the source of the sodium vapor is located. No experiment is available to validate this model. Thus no further discussion for this calculation is provided.

Table B.2.2-1 Demonstration Input for Atmosphere Chemistry Model

```
&& cray
&& -------------------------- clmrdb4 ------------------------------
&&           contain lmr/1 test problem db4
&&
&& model atmosphere chemistry
&&      to activate h2 generation, h2ov mole fraction > that of o2
&&
&& original cell 1 is divided into 5 cells
&&
&& --------------------- global input ---------------------------
control ncells=6 ntitl=2 ntzone=7  nsectn=10  nac=4 eoi
material
compound h2 o2 co co2 h2ol h2ov fe n2 conc uo2 pu
   u mno mgo k2o nav nal uo2 puo2 cao ssox  ss fel
   na2o2 na2o naoh na
times 60000.0   0.
&& ---------- time zones ---------------
  5.0   10.0        10.
  10.0   30.0     3000.
  10.0   60.0     5000.
  10.0  120.0     9500.
  10.0   30.0    12000.
  10.0  120.0    15000.
  10.0  120.0    20000.
&& ---------------------------------------
1.0 1.0 1.0 1.0 1.0 1.0  && cell timestep fraction
&&    debug 5 cvtoat medboil pool slam pooldf  1100.  1200.
longedt 2
fast
&& thermo
flows
  area(1,2)=160.0  cfc(1,2)=0.01  avl(1,2)=0.50   && area is the roof area of cell2
  elevcl(2)=4.7  elevcl(1)=20.0  elevfp(1,2)=9.4 elevfp(2,1)=9.4

  area(1,4)=162.7  cfc(1,4)=0.01  avl(1,4)=0.50
  elevcl(1)=20.0  elevcl(4)=16.5  elevfp(1,4)=16.5    elevfp(4,1)=16.5

  area(1,3)=162.7  cfc(1,3)=0.01  avl(1,3)=0.50
  elevcl(1)=20.0  elevcl(3)=16.5  elevfp(1,3)=16.5    elevfp(3,1)=16.5

  area(4,5)=242.5  cfc(4,5)=0.01  avl(4,5)=0.50
  elevcl(4)=16.5 elevcl(5)=5.85  elevfp(4,5)=9.4    elevfp(5,4)=9.4
```

```
  area(3,6)=242.5  cfc(3,6)=0.01  avl(3,6)=0.50
  elevcl(3)=16.5 elevcl(6)=5.85  elevfp(3,6)=9.4    elevfp(6,3)=9.4

  area(2,5)=3.5   cfc(2,5)=0.01  avl(2,5)=0.50
  elevcl(2)=4.7  elevcl(5)=5.85  elevfp(2,5)=5.85    elevfp(5,2)=5.85

  area(2,6)=3.5   cfc(2,6)=0.01  avl(2,6)=0.50
  elevcl(2)=4.7  elevcl(6)=5.85  elevfp(2,6)=5.85    elevfp(6,2)=5.85


  implicit
  dropout
aerosol
   deldif=1.0e-4  densty=300.
   diam1=1.0e-08   tgas2=3000.0   pgas2=10.0e+5
   na2o2=1.0e-6  0.531
   na2o=1.0e-6   0.531
   naoh=1.0e-6   0.531
   nal =1.0e-6   0.531
prlow-cl
prheat
prflow
prburn   && print option for burn
praer
title
   contain lmr - snl
     model atmosphere chemistry
&& ------------ end of global input ---------------
cell=2
control
  nhtm=3  mxslab=10 numtbc=2  maxtbc=3
  jconc=5  jint=1  jpool=1            && nraycc=50 ndblyr=4
  nsoatm=1 nspatm=4     && number of atm source, entry for table
eoi
geometry  1570.0  9.4   && volume and height
atmos=4  1.018e+5  378.0
   n2=0.75  o2=0.10  nav=0.0 h2ov=0.15  && initial condition to allow 2*na+h2o=na2o+h2
   condense
  atmchem
  frna2o=0.5
  h-burn
  eoi
  source=1  && number of table
    nav=4 iflag=1
      t=0.0   250.0 500.0 1000.0
      mass=5.0 5.0 5.0 5.0
      temp=600.0 600.0 600.0 600.0
&& ------------- structures ------------------------
struc
&& ---------------- cavity wall ---------------------
  cavwal1 wall  cylinder  10 2 328. 9.4 0. 9.4
  7.30  7.40  7.50  7.60  7.70  7.80  7.90  8.00  8.10  8.20  8.30
  ss  ss  conc  conc  conc  conc  conc  conc  conc  conc
&& ---------------- cavity wall ---------------------
  cavwal2 wall  cylinder  10 2 328. 9.4 0. 9.4
  7.30  7.40  7.50  7.60  7.70  7.80  7.90  8.00  8.10  8.20  8.30
  ss  ss  conc  conc  conc  conc  conc  conc  conc  conc
&& ---------------- cavity roof ----------------------
  cavroof  roof  slab  10  2  328.  7.3  0.  167.
   0.00  0.10  0.20  0.30  0.40  0.50  0.60  0.70  0.80  0.90  1.00
   ss  ss  conc  conc  conc  conc  conc  conc  conc  conc
&&
rad-heat
     emsvt .8 .8 .8 .65   && nhtm+pool surface
     gaswal   7.4
eoi

ht-tran  on  on  on  on  on
```

```
low-cell
    geometry  167.0  && floor area

&& ---------------- concrete specification, inc. interaction model
    concrete
&&              concrete composition specification
&&      use corcon's crbr concrete type, contain's generic
          compos=1   concrete=limecc rhocon=2250   && concrete=limena   rhocon=2250.
           rebar=.140  tablat=5000.  emconc=0.9
          eoi
&&               concrete mass
          1.3e6
          temp=300.0  && initial temp

        eoi    &&  terminate the layer
&& ---------------- stainless liner specification ----------------
    interm
          lay-nam = liner
          compos = 1 ss  266.87 && assume mass here

          temp=400.

    eoi   &&  terminate the layer

     pool
        compos  1  nal   1000.
           temp  400.
           physics
             boil
         eoi    &&   terminate physics
    eoi   &&  terminate the layer
        bc  300.0  0.85e5
eoi  &&  terminate cell 1 input
&& ----------------------- end cell 1 ------------------------
cell=1
control
  nhtm=1  mxslab=10
eoi
geometry  3540.4  21.2
atmos=3  1.018e+5  378.0
    n2=0.75   o2=0.10  h2ov=0.15
  condense
&& ------------- structures -----------------------
struc

&& ---------------- confinement roof (center)----------------------
 conroof  roof  slab   10  10  328.  14.4 0.0  300.0  167.0
  0.00  0.10  0.20  0.30  0.40  0.50  0.60  0.70  0.80  0.90  1.00
  ss  ss  conc  conc  conc  conc  conc  conc  conc  conc
&&
ht-tran on  on  on  on  on

&& ----------------------- end cell 3 ------------------------
cell=3
control
  nhtm=2  mxslab=10
eoi
geometry  3443.4  14.1
atmos=3  1.018e+5  378.0
    n2=0.75   o2=0.10  h2ov=0.15
  condense
&& ------------- structures -----------------------
struc

&& ---------------- confinement wall --------------------
  conwal3 wall  cylinder   10 10 328. 14.1 0. 300.0 14.1
  14.40  14.50  14.60  14.70  14.80  14.90  15.00  15.10  15.20  15.30 15.4
  ss  ss  conc  conc  conc  conc  conc  conc  conc  conc
&& ---------------- confinement roof ---------------------
  conroof3  roof  slab   10  10  328.  14.1  0.  300.0 242.22
```

```
  0.00  0.10  0.20  0.30  0.40  0.50  0.60  0.70  0.80  0.90  1.00
  ss  ss  conc  conc  conc  conc  conc  conc  conc  conc
&&
ht-tran on  on  on  on  on

&& ------------------------ end cell 4 ------------------------
cell=4
control
 nhtm=2  mxslab=10
eoi
geometry  3443.4   14.1
atmos=3  1.018e+5  378.0
   n2=0.75   o2=0.10  h2ov=0.15
  condense
&& ------------- structures -----------------------
struc
&& ---------------- confinement wall --------------------
 conwal4 wall  cylinder   10 10 328. 14.1 0. 300.0 14.1
 14.40  14.50  14.60  14.70  14.80  14.90  15.00  15.10  15.20  15.30 15.4
 ss  ss  conc  conc  conc  conc  conc  conc  conc  conc

&& ---------------- confinement roof ---------------------
 conroof4  roof  slab   10  10  328.  14.1  0.  300.0 242.22
  0.00  0.10  0.20  0.30  0.40  0.50  0.60  0.70  0.80  0.90  1.00
  ss  ss  conc  conc  conc  conc  conc  conc  conc  conc
&&
ht-tran on  on  on  on  on

&& ------------------------ end cell 5 ------------------------
cell=5
control
 nhtm=1  mxslab=10
eoi
geometry  1721.4  7.1
atmos=3  1.018e+5  378.0
   n2=0.75   o2=0.10  h2ov=0.15
  condense
&& ------------- structures -----------------------
struc

&& ---------------- confinement wall --------------------
 conwal5 wall  cylinder   10 10 328. 7.1 0. 300.0 7.1
 14.40  14.50  14.60  14.70  14.80  14.90  15.00  15.10  15.20  15.30 15.4
 ss  ss  conc  conc  conc  conc  conc  conc  conc  conc

ht-tran on  on  on  on  on

&& ------------------------ end cell 6 ------------------------
cell=6
control
 nhtm=1  mxslab=10
eoi
geometry  1721.43  7.1
atmos=3  1.018e+5  378.0
   n2=0.75   o2=0.10  h2ov=0.15
  condense
&& ------------- structures -----------------------
struc
&& ---------------- confinement wall --------------------
 conwal6 wall  cylinder   10 10 328. 7.1 0. 300.0 7.1
 14.40  14.50  14.60  14.70  14.80  14.90  15.00  15.10  15.20  15.30 15.4
 ss  ss  conc  conc  conc  conc  conc  conc  conc  conc

&&
ht-tran on  on  on  on  on

eof
```

B.2.3 Sodium Spray Fire

This section describes the demonstration input for the sodium spray fire model (see the model details in Section 5.1.2). This model is a part of the atmospheric physics model. To invoke this model, the keyword "SPRAFIRE" is required. Once this model is activated, the user can specific the fall height of the sodium spray, mean sodium droplet diameter, the mole fraction of sodium peroxide by the fire, and the source of the sodium for the spray.

To verify and validate this model, ABCOVE AB5 experiment is used [Souto 1994]. The brief description of the experiment is provided. The primary objective of the ABCOVE test AB5 was to provide experimental data for use in validating aerosol behavior computer codes for the case of a moderate-duration, strong, single-component aerosol source generated by a sodium spray in an air atmosphere. A secondary objective was to provide experimental data on the temperature and pressure in the containment vessel and its atmosphere, for use in validating containment response codes. The experimental apparatus is given in Figure B.2.3-1. As shown in this figure, the experimental vessel is a round headed cylindrical vessel, which are built with steel and surrounded with insulation to minimize the heat loss. The sodium spray is injected about 5.1 m above the vessel bottom. A pan catch is in place to allow aerosol settling and liquid collection.

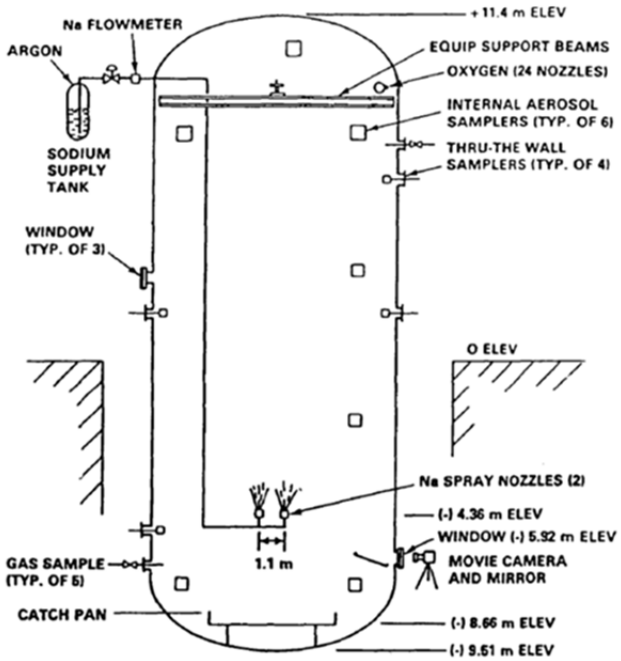


Figure B.2.3-1 CSTF Arrangement for ABCOVE AB5 Test [Souto 1994]

For the CONTAIN model, a single cell is used for this vessel. Walls, floor and roof of the vessel are modeled, including the internal deposition components. A summary of the Test AB5 is provided in Table B.2.3-1. The input deck for this experiment is shown in Table B.2.3-2. As shown in this table, the thermodynamic conditions of the experiment were modeled, including the sources of the sodium and oxygen. Since the aerosol results showed no monoxide formed (60% Na_2O_2 and 40% NaOH), the input value for the peroxide is set to 1.0. In order to model NaOH formation, the water vapor mass of the dew point from the test was included.

The preliminary comparison between calculated results from CONTAIN-LMR, CONTAIN2-LMR and the available test results is shown in Table B.2.3-3. Even though with the amount of water vapors modeled, CONTAIN-LMR and CONTAIN2-LMR did not predict any NaOH production since the sodium spray model did not account for any NaOH formations. In the experiment, all sourced sodium was reacted. However, CONTAIN-LMR and CONTAIN2-LMR estimated that a portion of the sodium did not react and fall into the pool.

Future tests should include turning on the atmospheric chemistry model to account for NaOH formations with the unreacted sodium.

Table B.2.3-1 Test Conditions for ABCOVE AB5 [Souto 1994]

INITIAL CONTAINMENT ATMOSPHERE	***PARAMETER***
Oxygen Concentration	23.3±0.2%
Temperature (mean)	302.25K
Pressure	0.122MPa
Dew Point	289.15±2K
Nominal Leak Rate	1%/day at 68.9kPa
Na SPRAY	***PARAMETER***
Na Spray Rate	256±15g/s
Spray Start Time	13s
Spray Stop Time	885 s
Total Na Sprayed	223±11 kg
Na Temperature	836.15 K
Spray Drop Size, MMD	1030±50 μm
Spray Size Geom. Std. Dev., GSD	1.4
OXYGEN CONCENTRATION	***PARAMETER***
Initial O_2 Concentration	23.3±0.2 vol %
Final O_2 Concentration	19.4±0.2 vol %
Oxygen Injection Start	60 s
Oxygen Injection Stop	840 s
Total O_2	47.6 m^3 (STD)
CONTAINMENT CONDITIONS DURING TESTS	***PARAMETER***
Maximum Average Atmosphere Temperature	552.15 K
Maximum Average Steel Vessel Temperature	366.65 K
Maximum Pressure	213.9 kPa
Final Dew Point	271.65 K
AEROSOL GENERATION	***PARAMETER***
Generation Rate	445 g/s
Mass Ratio, Total Na	1.74
Material Density	2.50 g/cm^3
Initial Suspended Concentration	0
Source Mass Median Dia.	0.50 μm
Source Sigma, σ_g	1.5
Maximum Suspended Mass Concentration	170 g/m^3
Suspended Conc. Steady-State Value	110±17 g/m^3

Table B.2.3-2. Demonstrated Input For ABCOVE AB5

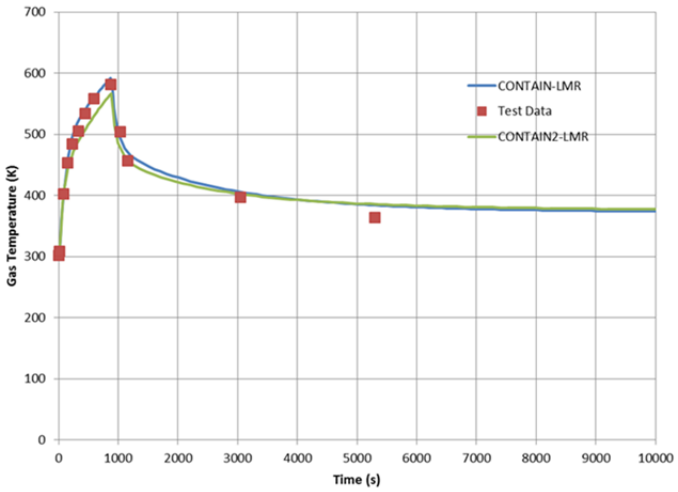
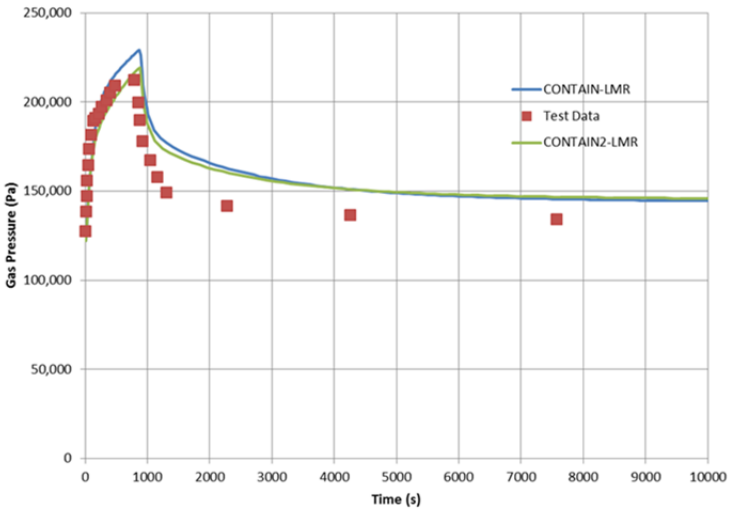
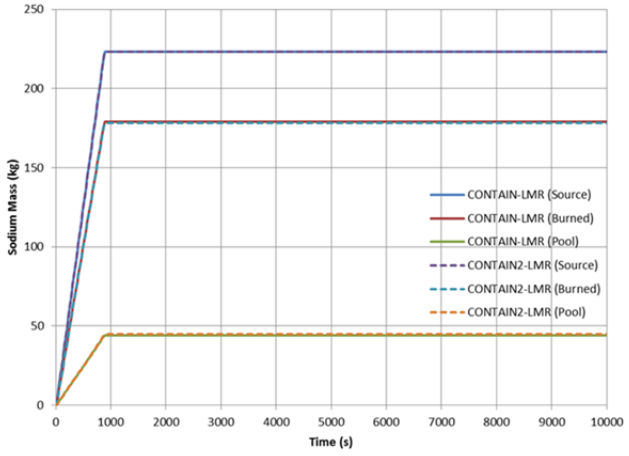
```
&& cray
&& ------------------------- AB5 -----------------------------
&&          contain lmr/1 test problem AB5
&&
&& model sodium spray fire - AB5
&& The experiment was modeled as two cells.
&&     cell 1 represents CSTF
```

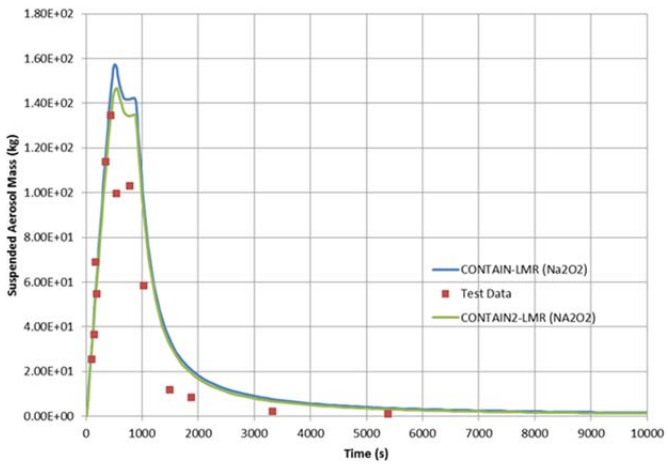
```
&& Heat structures are modeled in 5:
&&     1 the CSTF top head
&&     2 CSTF cylindrical walls
&&     3 internal components for aerosol plating
&&     4 CSTF bottom head
&&     5 internal componets for aerosol settling
&&
&& Sodium is injected at 5.15 m above the tank bottom.
&& a total of 223 kg over 872 s.  All sodium converted to
&& 60%na2o2 and 40%naoh.
&& Results: maximum P=214kPa, MeanT=553.15K, max 843.15K
&& max suspended aerosol density 170g/m3 after 383s initiated Na inj
&& ---------- global input ----------
control ncells=1 ntitl=2 ntzone=4  nsectn=10  nac=4 eoi
material
compound h2 o2 co co2 h2ol h2ov fe n2 conc uo2 pu
  u mno mgo k2o nav nal uo2 puo2 cao ssox  ss fel
  na2o2 na2o naoh na
times 60000.0   0.
&& ---------- time zones ----------
 5.0   10.0       10.
 10.0   30.0     3000.
 10.0   60.0     5000.
 10.0  120.0    10000.
&& ----------
1.0  && cell timestep fraction
&&
longedt 2
fast
thermo

&& aerosol
&& generation rate 445 g/s, material density=2.5g/cc
&& mass median radius=0.25 um, sigma=1.5
aerosol
  densty=2500.
  tgas2=3000.0   pgas2=10.0e+5
  na2o2=0.250e-6  1.5
  na2o=0.250e-6  1.5
  naoh=0.250e-6  1.5
  na  =0.250e-6  1.5
prlow-cl
prheat
prflow
prspray   && print option for sodium spray fire
praer
title
  contain lmr/1 test abcove AB5
    model sodium spray fire
&& ---------- end of global input ----------

&& ---------- cell 1 - CSTF ----------
cell=1  && CSTF
control
 nhtm=5  mxslab=10 numtbc=2  maxtbc=3 nsoatm=1  nspatm=3
 jint=1  jpool=1           && nraycc=50 ndblyr=4
 nsospr=1 nspspr=3     && number of spray source, entry for table
eoi
geometry  852.0  20.3   && volume and height
atmos=3  1.22e+5  302.25  && 0.122 MPa, 302.25 K
  n2=0.752  o2=0.233 h2ov=0.015  && initial o2 23.3vol%
  source=1  && model o2 injection rate, ts=60, tend=840s
         && 47.6 m3 (STD), 0.02241 m3= 1mole -> 2123.67 moles of O2
         && total injected 68 kgs. 0.08718 kg/s
    o2=3 iflag=1
    t=0.0  60.0 840.0
    mass=0.0 0.08718 0.08718
    temp=293.0 293.0 293.0
```

```
&& ------------ structures ----------------------
struc
&& --------------- Top Head ------------------
 tophead  roof  slab  2  0  303.45  7.62  1.0 0.0 63.0   && insulated
  0.00  0.00905 0.0181
  ss  ss
&& ---------------  walls   ------------------
 wall1 wall  cylinder 2  0  301.55  7.62  1.0 0.0 1.9   && insulated
 3.81 3.82145 3.8329
 ss  ss
&&
 wall2 wall  cylinder 2  0  301.55  7.62  1.0 0.0 1.9   && insulated
 3.81 3.82145 3.8329
 ss  ss
&& --------------- Internal   --------------------
 intern1  wall  slab  2  1  302.25  7.62  1.0  232.0
  0.00  0.0017 0.0034
  ss  ss
&& --------------- Internal  --------------------
 intern2  floor  slab  2  1  302.25  7.62  1.0  42.696
  0.00  0.0017 0.0034
  ss  ss

&&
rad-heat
    emsvt .9 .9 .9 0.9 0.9 .65   && nhtm+pool surface
    gaswal   7.4
eoi

ht-tran  on  on  on  on  on

&&
&& sodium spray fire input
&&
&& assume fall height of 10.0, since
sprafire   hite=5.15  dme=0.00103  fna2o2=1.0  && 1030 um, fna2o2, default
 source=1  && number of table
   nal=3 iflag=1
   t=0.0   13.0 885.0
   mass=0.0 0.256 0.256
   temp=836.15 836.15 836.15
 eoi

low-cell
    geometry  45.604  && floor area (bottom head)


&& ---------------- stainless bottom head specification ---------------
    interm
        lay-nam = bothead
        compos = 1 ss  8753.0 && bottom head mass
        temp=301.55
        physics
        ht-coef name=bas-mat var-x=time
          x=2 0.0 10000.0
          y=2 0.0 0.0
        eoi
      eoi
    eoi   &&  terminate the layer

     pool
        compos  1  nal   0.
            temp  301.55
            physics
              boil
         eoi   &&   terminate physics
    eoi   &&  terminate the layer
        bc  300.0  0.85e5
eoi  &&  terminate cell 1 input
```

```
&& ---------- end cell 1 ----------

eof
```

Table B.2.3-3. Calculation Results for ABCOVE AB5

	As shown the gas temperature results, the code matches closely with the experiment. However, not all sodium reacted, which means the gas temperature may be higher for CONTAIN-LMR and CONTAIN2-LMR, if all sodium reacted. Note that although the majority of the reaction energy is added to the atmosphere, a small portion of the reaction energy is also added to the pool. CONTAIN2-LMR tended to underestimate the gas temperature early in time, but matching better later in time.
	The pressure results show some agreements between the codes and experimental data. Similar results about CONTAIN-LMR and CONTAIN2-LMR as in gas temperature for the gas pressure.
	As shown in this figure, the total sourced sodium is 223 kg, and the amount sodium burned is 179 kg for both codes. The unburned sodium was relocated to the pool.



This figure shows that CONTAIN-LMR and CONTAIN2-LMR predicts similar results as from the experiment. However, CONTAIN2-LMR matches closer to the data than that of CONTAIN-LMR.

B.2.4 Sodium Pool Fire

This section describes the demonstration input for the sodium pool fire model (see the model details in Section 5.1.3). An effort has been made to benchmark the model with existing experiment. An experiment used to verify SOFIRE II code was used – one-cell experiment [Beiriger 1973]. In this experiment, a large test vessel (LTV) with a dimension of 3.048 m (10 ft) in diameter, 9.14 m (30 ft) in height and a gas volume of 62.3 m^3 (2200 ft^3). This gas volume contained gases at standard conditions. In the lower portion of the vessel, a 0.55742 m^2 (6 ft^2) steel pan was installed on a spider off the floor of the vessel. The pan was insulated with fire brick and mounted below a feed line from an external sodium pre-heat tank. A number of thermocouples were placed in the experiment. The test conditions of this one-cell experiment are shown in Table B.2.4-1. As shown in this table, three tests were done. The initial sodium mass sourced in the pool is 22.68 kg (50 lbs) for all cases. The only differences among these tests were the initial oxygen content, and pool temperatures which yield slightly different in the system pressure. In addition, this table provides the experimental oxide fraction of peroxide after the experiment. This would be used as an input condition for the CONTAIN-LMR calculation. Because there was no specific additional information about the vessel, other than the dimensions, the heat structures modeled in CONTAIN-LMR for this experiment were assumed. Even though the experiment indicated that the steel pan holding the sodium was insulated with fire brick, a constant and small heat transfer rate from the pan to the base mat was assumed. Tables B.2.4-2 to B.2.4-4 show the input decks for Test 4, Test 5 and Test 6, respectively. The results shown in this section are preliminary. Further evaluations of the models and the experiment may be required to identify any model and/or experiment shortcomings.

For Test 4, CONTAIN-LMR predicts general trend of the experiment data. The finding is provided in Table B.2.4-5.

For Test 5, additional runs were conducted to check the sensitivity of the assumed fraction of monoxide and peroxide. By assuming 100% reaction for the monoxide, the CONTAIN-LMR greed generally with the experiment than the base case and 100% reaction for peroxide. Table B.2.4-6 provided the preliminary finding.

For Test 6, the assumed 50% reaction for peroxide yields higher pressure than the experiment data. The slope of the sodium pool's pool temperature is more pronounce for the experiment than CONTAIN-LMR. Table B.2.4-7 shows the preliminary finding for this test.

In general, the demonstration runs show that the model of the sodium pool fire in CONTAIN-LMR is working. Also the modeling of the aerosol generation from the sodium pool fire is working for CONTAIN-LMR. Additional model refinement and experiment comparison may be needed, particularly if no detail information is available for the experiments shown here.

Table B.2.4-1. Test Conditions and Data of the One-Cell Experiment [Beiriger 1973]

Parameter	Test 4	Test 5	Test 6
Weight of sodium introduced (kg)	22.68	22.68	22.68
Introduced sodium temperature (K)	811	866	866
Initial pool temperature with sodium (K)	566.6	616.5	699.8
Initial vessel oxygen (%)	21	9.25	2.0
Initial vessel pressure (kPa)	121.7	123.0	125.9
Pan Thickness (m)	0.0064	0.0064	0.00025
Sodium peroxide fraction (experimental)	0.39	0.78	-
Oxygen consumed (%)	10.7	5.3	0.34
Average gas temperature rise* (C)	48.3	27.2	7.2
Peak pressure rise (kPa)	19	10.9	2.8
Maximum bulk sodium temperature (K)	977-1037	750	700
*It was reported as in °F, but converted and recalculated using standard condition temperature of 20 °C or 68 °F			

Table B.2.4-2. Test 4 Input for CONTAIN-LMR

```
&&
&& -------------------------- one-cell----------------------------
&&          contain lmr/1 test problem 1cell – Test4
&&
&& model na pool fire –
&& using one cell experiment in AI-AEC-13055
&&   Test 4 – oxide fraction of peroxide = 0.39
&&           f1=1-0.39=0.61
&&          O2 consumed=10.7%
&&          Peak delP=2.77 psi (1.9098e4 Pa)
&&          Bulk Na T=1300-1400 F(917.6-1037.15K)
&&          Gas T Rise=87 F
&&
&&
&& ---------------------- global input ----------------------------
control ncells=1 ntitl=2 ntzone=4  nsectn=10  nac=4 eoi
material
compound h2 o2 co co2 h2ol h2ov fe n2 conc uo2 pu
  u mno mgo k2o nav nal uo2 puo2 cao ssox  ss fel
  na2o2 na2o naoh na
times 60000.0   0.
&& ---------- time zones ----------------
 5.0   10.0       10.
 10.0   20.0     1000.
 10.0   20.0     2000.
 10.0   50.0     4320.

&& ---------------------------------------
0.0    && cell timestep fraction
```

```
longedt 2

fast

thermo

aerosol
  deldif=1.0e-4  densty=300.
  Diam1=1.0e-08   tgas2=3000.0   pgas2=10.0e+5
  na2o2=1.0e-6  0.531
  na2o=1.0e-6   0.531
  naoh=1.0e-6   0.531
  na  =1.0e-6   0.531


prlow-cl
prheat

praer  && aerosol print
title
  contain lmr/1 test4 One-cell
    model pool fire
&& ------------ end of global input ---------------
&& cell 1 models the LTV, height = 30 ft (9.14 m)
&&                      volume = 2200 ft3 (62.3 m3)
&&      1 ft3=28.3168e-3 m3
cell=1
control
 nhtm=3  mxslab=10 numtbc=2  maxtbc=3
   jint=1  jpool=1           &&
eoi
geometry  62.3  9.14   && volume and height
atmos=2  1.21685e+5  293.  && STD: 20 C, 101.325 kPa
  n2=0.79  o2=0.21
  condense
&&  atmchem
&&  h-burn
&& ------------- structures ------------------------
struc
&& ---------------- vessel wall --------------------
&& name   istr  ishape  nslab ibc tint chrl vufac bctr heit
 veswal1 wall cylinder 5      0  293. 9.14 0.0 0.0 9.14 && 566.48 9.14 0.  0.   9.14
 1.524 1.534 1.544 1.555 1.565 1.575
 ss  ss  ss  ss  ss
&& ---------------- cavity wall --------------------
 veswal2 wall cylinder 5      0  293. 9.14 0.0 0.0 9.14 && 566.48 9.14 0.  0.   9.144
 1.524 1.534 1.544 1.555 1.565 1.575
 ss  ss  ss  ss  ss
&& ---------------- vessel roof ---------------------
 vesroof  roof slab    5      0  293. 9.14 0.0 0.0 6.82 && 566.48 9.14 0.  0.   6.82
  0.0     0.01  0.02  0.03  0.04 0.05


  ss  ss  ss  ss  ss
&&
rad-heat
    emsvt .8 .8 .8 .8   && nhtm+pool surface
    gaswal   7.4
eoi
ht-tran  on  on  on  on  on
low-cell
    geometry  0.55742  && floor area, 6 ft2


&& ---------------- stainless liner specification ----------------
    interm
      lay-nam = liner
      compos = 1 ss  28.52 && based on pan thickness of 0.25”x6ft2x503 lb/ft3 density

      temp=293.
```

```
      Physics
       ht-coef name=bas-mat var-x=time
         x=2 0.0 10000.0
         y=2 1.0e+1 1.0e+1
       eoi
      eoi


   eoi   &&  terminate the layer

    pool
       compos  1  nal   22.68 && 50 lbs
          temp  566.5   && 560 F
          physics
&&                boil
&&
&& sodium pool fire –
            poolfire
              ratios  0.61  0.5  0.5 0.5   &&
            eoi

         eoi    &&   terminate physics
   eoi   &&  terminate the layer
       bc  293.
Eoi  &&  terminate cell 1 input
&& ---------------------- end cell 1 ------------------------



eof
```

Table B.2.4-3. Test 5 Input for CONTAIN-LMR

```
&&
&& ------------------------- one-cell-----------------------------
&&           contain lmr/1 test problem 1cell -Test5
&&
&& model na pool fire -
&& using one cell experiment in AI-AEC-13055
&&   Test 5 - oxide fraction of peroxide = 0.78
&&            f1=1-0.78=0.22
&&           initial p=3.2 psig (2.2063e4 Pa)
&&           initial T=300 K (assumed at STD)
&&           peak delP=1.59 psi (1.0962 Pa)
&&           gas T rise=49 F
&&
&&
&& --------------------- global input ----------------------------
control ncells=1 ntitl=2 ntzone=4  nsectn=10  nac=4 eoi
material
compound h2 o2 co co2 h2ol h2ov fe n2 conc uo2 pu
  u mno mgo k2o nav nal uo2 puo2 cao ssox  ss fel
  na2o2 na2o naoh na
times 60000.0   0.
&& ---------- time zones ---------------
 5.0   10.0       10.
 10.0   20.0     1000.
 10.0   20.0     2000.
 10.0   50.0     7200.

&& ----------------------------------------
1.0   && cell timestep fraction

longedt 2

fast

thermo
```

```
aerosol
  deldif=1.0e-4  densty=300.
  diam1=1.0e-08   tgas2=3000.0   pgas2=10.0e+5
  na2o2=1.0e-6  0.531
  na2o=1.0e-6   0.531
  naoh=1.0e-6   0.531
  na  =1.0e-6   0.531


prlow-cl
prheat

praer  && aerosol print
title
  contain lmr/1 test4 One-cell
    model pool fire
&& ------------ end of global input ---------------
&& cell 1 models the LTV, height = 30 ft (9.14 m)
&&                      volume = 2200 ft3 (62.3 m3)
&&     1 ft3=28.3168e-3 m3
cell=1
control
 nhtm=3  mxslab=10 numtbc=2  maxtbc=3
   jint=1  jpool=1           &&
eoi
geometry  62.3  9.14   && volume and height
atmos=2  1.2306e+5   293.  && STD: 20 C, 101.325 kPa
  n2=0.9075  o2=0.0925
  condense
&&  atmchem
&&  h-burn
&& ------------ structures ----------------------
struc
&& --------------- vessel wall --------------------
&& name   istr  ishape  nslab ibc tint chrl vufac bctr heit
 veswal1 wall cylinder 5      0  293.0 9.14 0.0 0.0 9.14 && 566.48 9.14 0.  0.   9.14
 1.524 1.534 1.544 1.555 1.565 1.575
 ss  ss  ss  ss  ss
&& --------------- cavity wall --------------------
 veswal2 wall cylinder 5      0  293.0 9.14 0.0 0.0 9.14 && 566.48 9.14 0.  0.   9.144
 1.524 1.534 1.544 1.555 1.565 1.575
 ss  ss  ss  ss  ss
&& --------------- vessel roof ---------------------
 vesroof  roof slab    5      0  293.0 9.14 0.0 0.0 6.82 && 566.48 9.14 0.  0.   6.82
  0.00  0.01  0.02  0.03  0.04 0.05
  ss  ss  ss  ss  ss
&&
rad-heat
    emsvt .8 .8 .8 .8   && nhtm+pool surface
    gaswal   7.4
eoi
ht-tran  on  on  on  on  on
low-cell
    geometry  0.55742  && floor area, 6 ft2


&& ---------------- stainless liner specification ---------------
    interm
      lay-nam = liner
      compos = 1 ss  28.52 && based on pan thickness of 0.25"x6ft2x503 lb/ft3 density

      temp=293.0
      physics
        ht-coef name=bas-mat var-x=time
          x=2 0.0 10000.0
          y=2 1.0e+1 1.0e+1
        eoi
      eoi
```

```
    eoi   &&  terminate the layer

     pool
        compos  1  nal   22.68 && 50 lbs
            temp  616.5   && 650 F
            physics
&&                  boil
&&
&& sodium pool fire -
              poolfire
                ratios  0.22  0.5  0.5 0.5   &&
              eoi

          eoi    &&   terminate physics
    eoi   &&  terminate the layer
        bc  293.0
eoi  &&  terminate cell 1 input
&& ---------------------- end cell 1 ------------------------



eof
```

Table B.2.4-4. Test 6 Input for CONTAIN-LMR

```
&&
&& ------------------------- one-cell-----------------------------
&&            contain lmr/1 test problem 1cell -Test6
&&
&& model na pool fire -
&& using one cell experiment in AI-AEC-13055
&&   Test 6 - oxide fraction of peroxide =
&&
&&            initial p=3.6 psig (2.4821e4 Pa)
&&            initial T=300 K, 80.33 F (assumed at STD)
&&            peak delP=0.41 psi (2827 Pa)
&&            gas T rise=13 F
&&
&&
&& ---------------------- global input ----------------------------
control ncells=1 ntitl=2 ntzone=4  nsectn=10  nac=4 eoi
material
compound h2 o2 co co2 h2ol h2ov fe n2 conc uo2 pu
  u mno mgo k2o nav nal uo2 puo2 cao ssox  ss fel
  na2o2 na2o naoh na
times 60000.0   0.
&& ---------- time zones ---------------
  5.0   10.0        10.
 10.0   20.0     1000.
 10.0   20.0     2000.
 10.0   50.0     3600.

&& ---------------------------------------
1.0   && cell timestep fraction

longedt 2

fast

thermo

aerosol
  deldif=1.0e-4  densty=300.
  diam1=1.0e-08   tgas2=3000.0   pgas2=10.0e+5
  na2o2=1.0e-6  0.531
  na2o=1.0e-6   0.531
  naoh=1.0e-6   0.531
  na  =1.0e-6   0.531
```

```
prlow-cl
prheat

praer  && aerosol print
title
  contain lmr/1 test4 One-cell
    model pool fire
&& ------------ end of global input --------------
&& cell 1 models the LTV, height = 30 ft (9.14 m)
&&                      volume = 2200 ft3 (62.3 m3)
&&     1 ft3=28.3168e-3 m3
cell=1
control
 nhtm=3  mxslab=10 numtbc=2  maxtbc=3
   jint=1  jpool=1           &&
eoi
geometry  62.3  9.14   && volume and height
atmos=2  1.2588e+5  293.0  &&
  n2=0.98  o2=0.02
  condense
&&  atmchem
&&   h-burn
&& ------------- structures -----------------------
struc
&& ---------------- vessel wall --------------------
&& name   istr  ishape  nslab ibc tint chrl vufac bctr heit
 veswal1 wall cylinder 5      0  293. 9.14 0.0 0.0 9.14 && 566.48 9.14 0.  0.   9.14
 1.524 1.534 1.544 1.555 1.565 1.575
 ss  ss  ss  ss  ss
&& ---------------- cavity wall --------------------
 veswal2 wall cylinder 5      0  293. 9.14 0.0 0.0 9.14 && 566.48 9.14 0.  0.   9.144
 1.524 1.534 1.544 1.555 1.565 1.575
 ss  ss  ss  ss  ss
&& ---------------- vessel roof ---------------------
 vesroof  roof slab    5      0  293. 9.14 0.0 0.0 6.82 && 566.48 9.14 0.  0.   6.82
  0.00  0.01  0.02  0.03  0.04 0.05
  ss  ss  ss  ss  ss
&&
rad-heat
    emsvt .8 .8 .8 .8   && nhtm+pool surface
    gaswal   7.4
eoi
ht-tran  on  on  on  on  on
low-cell
    geometry  0.55742  && floor area, 6 ft2


&& ---------------- stainless liner specification ----------------
    interm
      lay-nam = liner
      compos = 1 ss  28.52 && based on pan thickness of 0.25"x6ft2x503 lb/ft3 density

      temp=293.
      physics
        ht-coef name=bas-mat var-x=time
          x=2 0.0 10000.0
          y=2 1.0e+1 1.0e+1
        eoi
      eoi


    eoi   &&  terminate the layer

     pool
        compos  1  nal   22.68 && 50 lbs
            temp  699.8   && 800 F
            physics
&&                boil
&&
```

```
&& sodium pool fire -
              poolfire
                ratios  1.0  0.5  0.5 0.5   &&
              eoi

           eoi    &&   terminate physics
     eoi   &&  terminate the layer
          bc  293.
eoi  &&  terminate cell 1 input
&& ---------------------- end cell 1 ------------------------



eof
```

Table B.2.4-5. Calculation Results for Test 4 [Beiriger 1973].

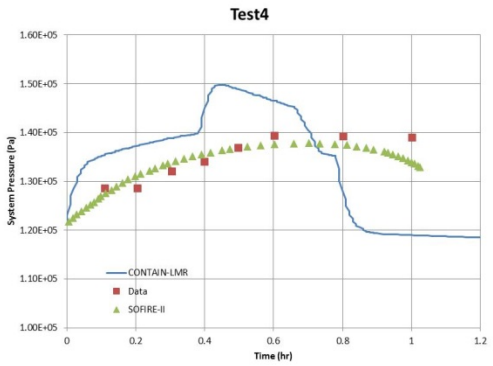
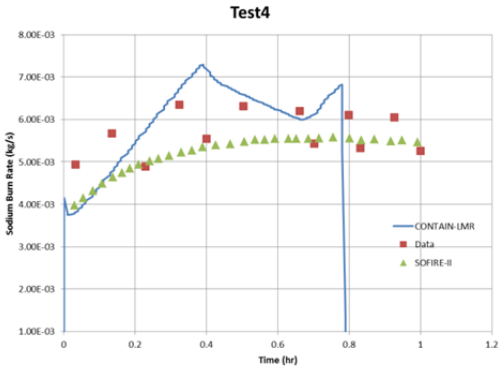
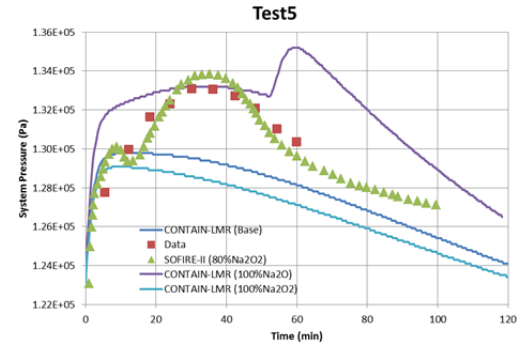
	CONTAIN-LMR seems to over-predict the pressures to about 0.7 hour before under-predict the pressures. In CONTAIN-LMR, the heat structures were modeled
	In terms of the burn rate, CONTAIN-LMR underestimates the values at the start, but it exceeds the data at about 0.4 hour then started to decrease closely with the data, but reaction dropped off quickly at about 0.8 hour, which is limited by the model that is subjected further analysis. The percentage of oxygen consumed predicted by CONTAIN-LMR is about 6.22, significantly lower than that of the experiment and SOFIRE-II code.

Table B.2.4-6. Calculation Results for Test 5 [Beiriger 1973].

	In this case, three CONTAIN-LMR runs were conducted and results were compared with the experimental data. The base case uses the measured fraction of the monoxide. The other two cases as a sensitivity study examine if only 100% of monoxide or peroxide. The results show that the base case falls between the two sensitivity cases. Assuming 100% monoxide yields a better pressure results, because of the reaction rate. Assuming 100% peroxide under-predicts the system pressure.

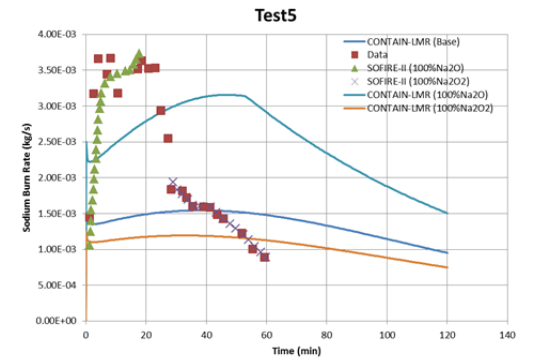
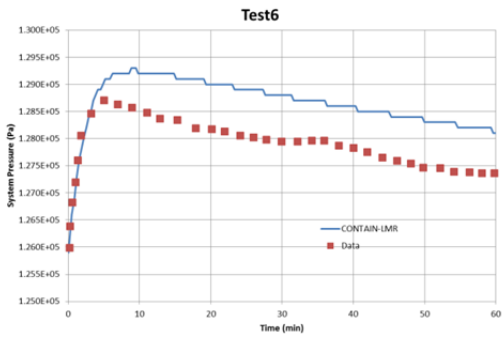
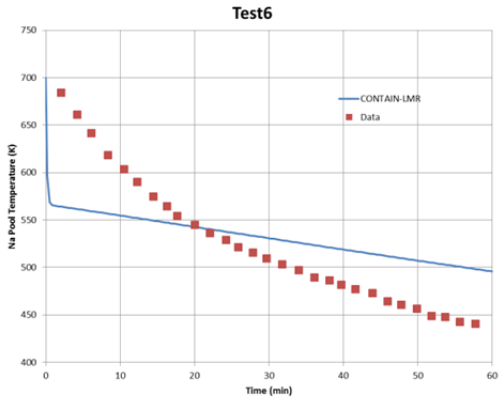
	In terms of the burn rate, CONTAIN-LMR calculations underestimate the rate for the first 20 minutes of the experiment. Then the calculations overestimate the results after that time. Assuming 100% monoxide for CONTAIN-LMR yields a better burn rate, than the CONTAIN-LMR base case and 100% peroxide case. The percentage of oxygen consumed was predicted by CONTAIN-LMR is about 5.3 for the base case, 5 for the peroxide case and 6 for monoxide case.

Table B.2.4-7. Calculation Results for Test 6 [Beiriger 1973].

	CONTAIN-LMR estimates the oxygen consumed is 0.52 percent where the data yields only 0.34 percent. In terms of prediction in system pressure, CONTAIN-LMR yields a slightly high over pressure trend.
	In terms of the sodium pool temperature, CONTAIN-LMR predicts the pool temperature decreases rapidly as it conducts to the heat the pan, which was assumed to have a 293 K. In the experiment, the pool temperature was measured after the spill, so the pan may have a similar temperature as the sodium pool. In terms of the slope of the temperature decreases, the experiment predicts deeper slope than that of CONTAIN-LMR. If the pan was insulated, the only mode of rapid temperature decrease is due to heat transfer to the structure, since the pressure of the experiment seems to be lower than CONTAIN-LMR. The gas temperature from the experiment after the fire was 308 K, where CONTAIN-LMR calculated the maximum gas temperature of 301 K. Additional sensitivity studies may be required to benchmark this test.

Intentionally Leave Blank

DISTRIBUTION

External Distribution

1 U.S. Department of Energy
Attn: Craig Welling
NE-74/GTN
1000 Independence Avenue SW
Washington, DC 20585

2 Argonne National Laboratory
Attn: Matt Bucknor (1)
Attn: Robert N. Hill (1)
Attn: Tanju Sofu (1)
9700 S. Cass Avenue
Argonne, IL 60439

1 Brookhaven National Laboratory
Attn: Robert Bari
Building 130
PO Box 5000
Upton, NY 11973-5000

2 Idaho National Laboratory
Attn: Brad J. Merrill (1)
P.O. Box 1625
Idaho Falls, ID 83415

Attn: Roald Wigeland (1)
Idaho Falls, ID 83425-3860

1 Oak Ridge National Laboratory
Attn: George Flanagan
PO Box 2008
MS 6165
Oak Ridge, Tennessee 37831-6165

Internal Distribution

1	MS0736	Susan Y. Pickering	06230
1	MS0748	Matthew R. Denman	06231
1	MS0748	Randall O. Gauntt	06232
1	MS0748	Larry H. Humphries	06232
1	MS0748	David L.Y. Louie	06232
1	MS0899	Technical Library	09536 (electronic copy)


Sandia National Laboratories