NUREG/IA-0421

International Agreement Report

Improvements and Validation of the System Code TRACE for Lead and Lead-Alloy Cooled Fast Reactors Safety-Related Investigations

Prepared by:
Wadim Jaeger, Victor Hugo Sanchez Espinoza

Karlsruher Institute of Technology (KIT)
Institute for Neutron Physics and Reactor Technology (INR)
Hermann-von-Helmholtz-Platz 1
76344, Eggenstein-Leopoldshafen, Germany

A. Calvo, NRC Project Manager

Office of Nuclear Regulatory Research
U.S. Nuclear Regulatory Commission
Washington, DC 20555-0001

Manuscript Completed: January 2013
Date Published: February 2013

Prepared as part of
The Agreement on Research Participation and Technical Exchange
Under the Thermal-Hydraulic Code Applications and Maintenance Program (CAMP)

Published by
U.S. Nuclear Regulatory Commission

AVAILABILITY OF REFERENCE MATERIALS IN NRC PUBLICATIONS

NRC Reference Material

As of November 1999, you may electronically access NUREG-series publications and other NRC records at NRC's Public Electronic Reading Room at http://www.nrc.gov/reading-rm.html. Publicly released records include, to name a few, NUREG-series publications; *Federal Register* notices; applicant, licensee, and vendor documents and correspondence; NRC correspondence and internal memoranda; bulletins and information notices; inspection and investigative reports; licensee event reports; and Commission papers and their attachments.

NRC publications in the NUREG series, NRC regulations, and Title 10, "Energy," in the *Code of Federal Regulations* may also be purchased from one of these two sources.

1. The Superintendent of Documents
   U.S. Government Printing Office
   Mail Stop SSOP
   Washington, DC 20402–0001
   Internet: bookstore.gpo.gov
   Telephone: 202-512-1800
   Fax: 202-512-2250
2. The National Technical Information Service
   Springfield, VA 22161–0002
   www.ntis.gov
   1–800–553–6847 or, locally, 703–605–6000

A single copy of each NRC draft report for comment is available free, to the extent of supply, upon written request as follows:

Address: U.S. Nuclear Regulatory Commission
Office of Administration
Publications Branch
Washington, DC 20555-0001
E-mail: DISTRIBUTION.RESOURCE@NRC.GOV
Facsimile: 301–415–2289

Some publications in the NUREG series that are posted at NRC's Web site address http://www.nrc.gov/reading-rm/doc-collections/nuregs are updated periodically and may differ from the last printed version. Although references to material found on a Web site bear the date the material was accessed, the material available on the date cited may subsequently be removed from the site.

Non-NRC Reference Material

Documents available from public and special technical libraries include all open literature items, such as books, journal articles, transactions, *Federal Register* notices, Federal and State legislation, and congressional reports. Such documents as theses, dissertations, foreign reports and translations, and non-NRC conference proceedings may be purchased from their sponsoring organization.

Copies of industry codes and standards used in a substantive manner in the NRC regulatory process are maintained at—

The NRC Technical Library
Two White Flint North
11545 Rockville Pike
Rockville, MD 20852–2738

These standards are available in the library for reference use by the public. Codes and standards are usually copyrighted and may be purchased from the originating organization or, if they are American National Standards, from—

American National Standards Institute
11 West 42nd Street
New York, NY 10036–8002
www.ansi.org
212–642–4900

Legally binding regulatory requirements are stated only in laws; NRC regulations; licenses, including technical specifications; or orders, not in NUREG-series publications. The views expressed in contractor-prepared publications in this series are not necessarily those of the NRC.

The NUREG series comprises (1) technical and administrative reports and books prepared by the staff (NUREG-XXXX) or agency contractors (NUREG/CR-XXXX), (2) proceedings of conferences (NUREG/CP-XXXX), (3) reports resulting from international agreements (NUREG/IA-XXXX), (4) brochures (NUREG/BR-XXXX), and (5) compilations of legal decisions and orders of the Commission and Atomic and Safety Licensing Boards and of Directors' decisions under Section 2.206 of NRC's regulations (NUREG-0750).

NUREG/IA-0421

International Agreement Report

Improvements and Validation of the System Code TRACE for Lead and Lead-Alloy Cooled Fast Reactors Safety-Related Investigations

Prepared by:
Wadim Jaeger, Victor Hugo Sanchez Espinoza

Karlsruher Institute of Technology (KIT)
Institute for Neutron Physics and Reactor Technology (INR)
Hermann-von-Helmholtz-Platz 1
76344, Eggenstein-Leopoldshafen, Germany

A. Calvo, NRC Project Manager

Office of Nuclear Regulatory Research
U.S. Nuclear Regulatory Commission
Washington, DC 20555-0001

Manuscript Completed: January 2013
Date Published: February 2013

Prepared as part of
The Agreement on Research Participation and Technical Exchange
Under the Thermal-Hydraulic Code Applications and Maintenance Program (CAMP)

Published by
U.S. Nuclear Regulatory Commission

ABSTRACT

This report deals with the validation of the system code TRACE related to lead and lead-alloy cooled nuclear systems. This validation process made in necessary to revise routines of the TRACE source code in order to make lead as a coolant available, and to add functions for heat transfer in bundles cooled with liquid metals. Several experimental as well as theoretical benchmarks were selected to test, on the one hand side, the existing physical models related to liquid metals, and, on the other hand side, the new implemented functions and routines. The results for the different investigations are in good agreement to the experimental data or the other codes. The present analysis showed that the revision of the TRACE models was necessary and that the changes yield better results.

FOREWORD

This assessment report deals with the extensions of different models related to the applications of TRACE for the safety-related investigations of lead cooled fast reactor systems for transmutation of waste. In addition, the validation work performed to demonstrate the prediction capability of TRACE for the analysis of experimental set-ups which are relevant for sub-critical and critical lead cooled fast systems are described. This work is complementary to the code validation work performed for LWRs with the goal to extend the simulation scope of TRACE to other reactor systems.

CONTENTS

FIGURES

TABLES

EXECUTIVE SUMMARY

The scope of this paper is to present results related to the improvement und validation of the system code TRACE applied to lead and lead-bismuth cooled nuclear systems. Since TRACE is the current LWR reference code of the U.S.NRC for analysis of operational conditions as well as of design based accidents, the validation related to liquid metals was out of the focus.

Since nowadays, innovative nuclear energy systems will be designed and evaluated, it is necessary to provide a tool that is capable of predicting safety related parameters, which are consistent with theory and experiment. The tool of choice at INR was the best-estimate tool TRACE. Therefore, it was necessary to review the thermo physical properties and the heat transfer models for liquid metals.

The thermo physical properties of lead-bismuth were improved because of inconsistencies. Properties of pure lead and Diphyl THT (commonly used at accelerator driven systems as coolant on the secondary side) were implemented.

The heat transfer models of TRACE related to liquid metals covered only pipe flow. Since the heat transfer in bundle arrays will differ from the one in tubes, additional routines were implemented.

Four different experiments or benchmarks were selected to validate TRACE. These are:

- The LACANES benchmark deals with the characterization of the local pressure losses, due to form and frictional losses, at the HELIOS facility. Several benchmark participants calculated the losses measured during an isothermal forced convection case. The obtained results are in very good agreement to the experimental data.

- The second example is the XADS target. In this investigation, the natural convection capabilities were tested since the independent cooling system of the target employs no pumps at all. Since no experimental data are available, the results of TRACE were compared to the ones of other codes. Investigated scenarios are steady state, beam power switch-on transient, beam power interruption transient, unprotected loss of heat sink, and variation of geometry. The comparison to results of other goods showed a high agreement.

- The modeling of a heat exchanger at the CHEOPE facility, related to the international MEGAPIE project, was the task for the third example. Some experimental data were available. The special feature of that heat exchanger is its uncommon geometry, employing an annulus with spirals to force the fluid on a helical upward path. Prelimenary results showed that the standard TRACE version is not able to calculate the experimental findings. Errors were in the range of 20 - 50 %. Several Nusselt correlations were selected and implemented into the TRACE code. The evaluation of the results showed that the results could be removed and the error amounts now to a few percent.

- The last example was of theoretical nature and deals with the validation of the heat transfer to liquid metals in bundle arrays and lead environment. Since again no experimental data are available, a second program was used to evaluate the results of TRACE. The sub channel code MATRA was selected since that code already has lead

as coolant, and one can distinguish between pipe and bundle flow. The comparison showed that, on the one hand side, the implementation of a correlation was necessary to avoid unrequired conservatism. On the other hand side, it showed that the TRACE results are in good agreement to the results of MATRA. Hence, the thermal properties of lead and the heat transfer correlation for bundle flow were implemented correctly.

In general, the code changes made were all necessary, and the results for the different examples were all satisfying due to the good agreement with experimental data and the consistency compared to other codes.

ABBREVIATIONS

ATHLET	Analysis of Thermal Hydraulics of Leaks and transients
CAMP	Code Applications and Maintenance Program
CANDU	Canadian Deuterium Uranium
CHEOPE	Chemical Operational Transient
CFD	Computational Fluid Dynamics
DTHT	Diphyl THT
ENEA	Ente per le Nuove Technologie, L'energiae l'Ambiente
HELIOS	Heavy Eutectic liquid metal Loop for Integral test of Operability and Safety
IAEA	International Atomic Energy Agency
INR	Institut fuer Neutronenphysik und Reaktortechnik
IPPE	Institute of Physics and Power Engineering
KIT	Karlsruhe Institute of Technology
LACANES	Lead Alloy-Cooled Advanced Nuclear Energy Systems
LBE	Lead-Bismuth-Eutectics
LFR	Lead-cooled Fast Reactor
MATRA	Multichannel Analyzer for steady states and Transients in Rod Arrays
MEGAPIE	Megawatt Pilot Experiment
NEA	Nuclear Energy Agency
OECD	Organization for Economic Co-operation and Development
RELAP5	Reactor Excursion and Leak Analysis Program
SA	Sub-Assembly
SNU	Seoul National University
TRAC	Transient Reactor Analysis Code
TRACE	TRAC/RELAP Advanced Computational Engine
U.S. NRC	Unites States Nuclear Regulatory Commission
VDI	Verein Deutscher Ingenieure
XADS	Experimental Accelerator Driven System

NOMENCLATURE

Symbol	Name of quantity	Value	Unit
A	Flow area	-	m
cp	Specific heat	-	$J \cdot kg^{-1} \cdot K^{-1}$
d	Diameter	-	m
D	Diameter of the spiral	-	m
f	Friction factor	-	-
K	Form loss factor	-	-
k	Thermal conductivity	-	$W \cdot m^{-1} \cdot K^{-1}$
l	length	-	m
Nu	Nusselt number	-	-
p	Rod pitch	-	m
p	Pressure	-	Pa
Pe	Péclet number	-	-
Pr	Prandtl number	-	-
Re	Reynolds number	-	-
R	Radius	-	m
T	Temperature	-	T
v	Velocity	-	$m \cdot s^{-1}$
Δ	Wall roughness	-	m
Δp	Pressure drop	-	Pa
δ	Angel of turning (bend)	-	°
ε	Criterion of thermal similarity of the fuel rods	-	-
ε	Ratio of the projected grid cross section and the undisturbed flow area	-	-
η	Dynamic viscosity	-	Pa·s
ρ	Density	-	$kg \cdot m^{-3}$
σ	Surface tension	-	$N \cdot m^{-1}$

Subscript

boil	Boiling point
fr	Friction
grid	Grid spacer
j; j+1/2; j+1;…	Cell index position
loc	Local quantity
melt	Melting point
Re	Reynolds
total	Total value
u	Undisturbed
w	Wall
Δ	Wall roughness

1 INTRODUCTION

The Institute for Neutron Physics and Reactor Technology (INR) and the Karlsruhe Institute of Technology (KIT) takes part in the ongoing process of validating the best estimate codes such as RELAP5 and TRACE. At the INR, system codes will be used for design and safety analysis of liquid metal cooled reactors. In this report, the modifications of TRACE related to lead and lead-alloy cooled reactor systems are presented.

Within the Generation IV international Forum [1], six innovative nuclear reactor concepts have been proposed to meet the requirements of future nuclear power plants. One of the proposed concepts is the lead-cooled fast reactor, which can be operated with pure lead or with an eutectic mixture of lead and bismuth. The advantage of pure lead is the high availability but the high melting temperature (327°C) requires new materials. The advantage of lead-bismuth-eutectic is the low melting point (127°C). Nevertheless, the reserves of bismuth are limited, which makes it expensive. The production of Polonium-210 is also one major drawback of the use of lead-bismuth.

Due to the chemical inertance, lead and lead-bismuth can replace sodium as the major liquid metal coolant. Lead-bismuth is also the favorable option for accelerator driven system for burning nuclear waste like minor actinides.

All over the world, researchers are working on lead and lead-bismuth cooled systems. Hence, some experimental data are available, unfortunately only for lead bismuth. These experimental data can be used to validate system codes. Since the thermo-physical properties of lead and lead-bismuth are similar, and the heat transfer to these fluids follows the same rule, conclusions made for one coolant can easily be adopted for the other one.

At the INR, the system code TRACE is being validated regarding its applicability to fast reactor transient simulations. The original TRACE version contains already some heat transfer models for liquid metals (lead-bismuth, sodium). TRACE was also chosen because it is the reference thermal hydraulic code system of the U.S. NRC, still under development. The modular structure facilitates the changes in the code.

The following section, section 2, deals with the thermo-physical properties. At first, the existing properties of lead-bismuth were revised and updated. Based on the review, the thermo physical properties of pure lead and of synthetic Diphyl THT oil were implemented into TRACE.

Section 3 covers the improvements related to the heat transfer to liquid metal. Existing functions were improved but also new ones were implemented (distinction between pipe and bundle flow)

The efforts for the validation of the modifications presented in section 2 and 3 are part of the fourth section. Four different issues were selected for the validation of TRACE, starting with the validation of pressure drop modeling and testing of the natural convection mechanism. Later on, the heat transfer was the major topic of the validation process.

In the last section conclusions and outlook are given.

The presented results are an extraction of the PhD Thesis of W. Jaeger [2] and of conference papers [3] [4] [5]. Some parts have been taken entirely verbatim.

2 THERMO PHYSICAL PROPERTIES

2.1 Lead-bismuth eutectics

A review of the thermo physical properties of lead-bismuth (PbBi or LBE) revealed that some properties in the TRACE source code are imprecise. Hence, the improvement of these properties is necessary in view of the foreseen investigations.

To the main thermo physical properties of liquid metals belong melting and boiling point, density, thermal conductivity, dynamic viscosity, specific heat, surface tension and saturation pressure. In the following sub-sections, new functions for these properties will be derived from the open literature.

Values for the temperature dependent properties were taken out of the following references: Adamov and Orlov [6], Blaskett and Boxall [7], Hultgren et al. [8], IAEA [9], Kirillov et al. [10], Kutateladze et al. [11], Lyon [12], McLain and Martens [13], Morita et al. [14], Novakovic et al. [15] OECD/NEA [16], Ohno, Miyahara and Kurata [17], Petrazzini [18], and Tipton [19]. The functional fits of the OECD/NEA were used as validation for the new functions.

A complete overview of all these correlations for the properties will be given at the end of this section.

2.1.1 Melting and boiling point

The first parameters that were evaluated are the melting and the boiling point. Values of these properties are tabulated in Table 1.

Table 1. Values for the lead-bismuth melting and boiling point

Reference	Melting point [K]	Boiling point [K]
Adamov and Orlov [6]	396.75	1670.00
Blaskett and Boxall [7]	398.15	-
Hultgren et al. [8]	398.00	-
IAEA [9]	398.15	1670.00
Kutateladze et al. [11]	396.65	1670.00
McLain and Martens [13]	398.15	-
Tipton [19]	398.15	1670.00

The average values for the melting and the boiling point are:

$$T_{melt} = (397.75 \pm 0.61)K$$

$$T_{boil} = 1670K$$

2.1.2 Density

Four citations were found for evaluating the density. These references are listed in Table 2 and depicted in Figure 1 together with the resulting function, labeled as “New approximation”. One can see that the new approximation is super-imposed to the curve of the OECD/NEA. The curve of the original TRACE version, blue curve, is predicting higher values than the other two curves. The functions to describe the density for the original TRACE version and the new approximation are given below in Eq. 1 and Eq. 2 respectively.

Table 2. Citations and values for the density of lead-bismuth

Reference	Temperature range [K]	Number of data points
IAEA [9]	403 - 1073	8
Kutateladze et al. [11]	403 - 973	13
Lyon [12]	473 - 1273	5
McLain and Martens [13]	477 - 1255	8
OECD/NEA [16]	400 - 1300	functional fit

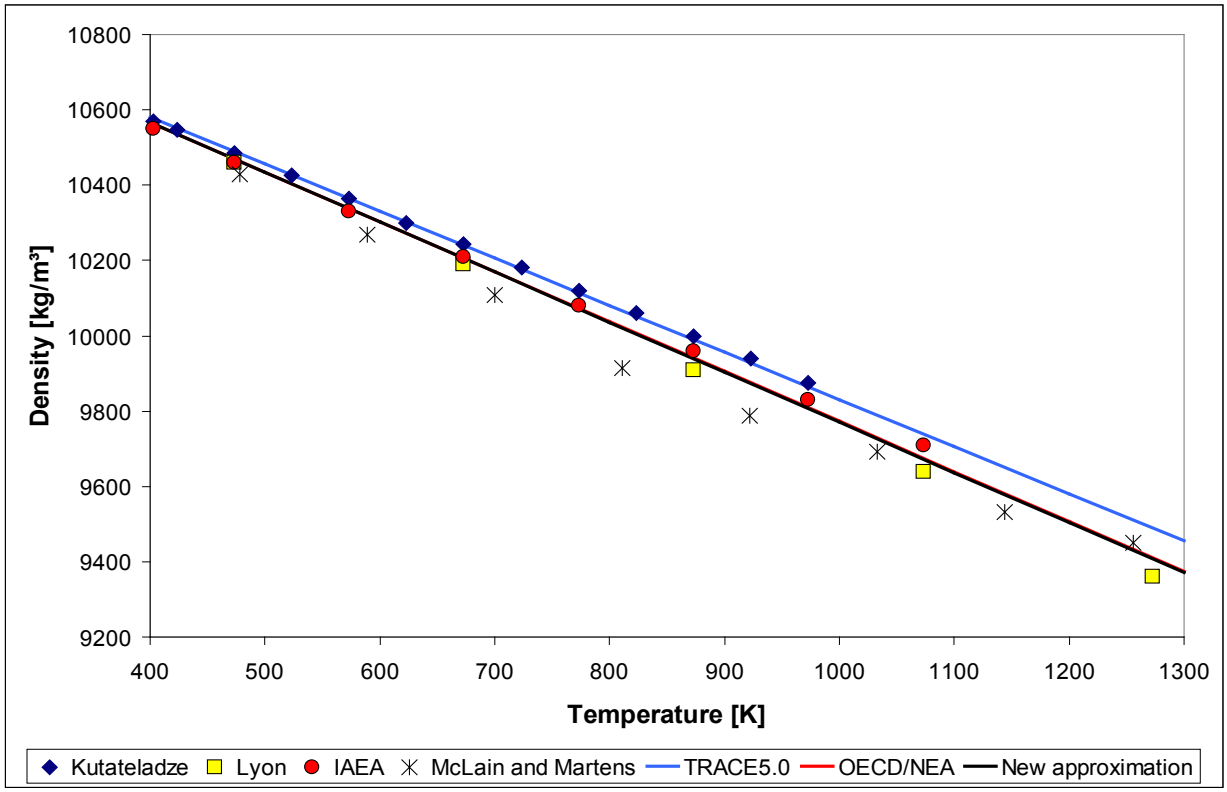

Figure 1. Lead-bismuth density versus temperature

$$\rho = -1.2514 \cdot (T - 398.5) + 10583 + \frac{dp}{d\rho} \qquad \left[\frac{kg}{m^3}\right] \qquad (1)$$

$$\rho = -1.3312 \cdot T + 11105 + \frac{dp}{d\rho} \qquad \left[\frac{kg}{m^3}\right] \tag{2}$$

The new approximation is valid between the melting point (approx. 398 K) and 1300 K.
With respect to the density change due to the pressure change, an additional term was implemented into the original TRACE density fit. This additional term has been maintained in the new fit. This term writes as follows:

$$\frac{dp}{d\rho} = \frac{1}{\left[1600 - 0.5 \cdot \left(T - T_{melt}\right)\right]} \tag{3}$$

Due to the high speed of sound, the influence of the pressure change can be neglected.

2.1.3 Thermal conductivity

Table 3 shows the list references taking into account to evaluate the thermal conductivity. One of the citations (Lyon) provides two different data sets. The first one is the usual temperature dependent thermal conductivity, whereas the second one is based on the electrical conductivity. The Wiedemann-Franz-Lorenz law couples the electrical and thermal conductivity.

Table 3. Citations and values for the thermal conductivity of lead-bismuth

Reference	Temperature range [K]	Number of data points
IAEA [9]	403 - 1073	8
Kirillov et al. [10]	417 - 627	6
Kutateladze et al. [11]	403 - 973	13
Lyon 1 [12]	433 - 593	5
Lyon 2 [12]	473 - 773	4
OECD/NEA [16]	400 - 1100	functional fit

The above mentioned data points and the new approximation are shown in Figure 2 . At lower temperatures, the difference between the original TRACE curve and the new correlation is in the range of 10 % or more. At higher temperatures, the difference diminishes.

The functional fits for the thermal conductivity are given in Eq. 4 and Eq. 5 where

$$k = 9.829 \cdot 10^{-3} \cdot T + 7.0473 \qquad \left[\frac{W}{m \cdot K}\right] \tag{4}$$

for TRACE, and

$$k = 1.181 \cdot 10^{-2} \cdot T + 5.3557 \qquad \left[\frac{W}{m \cdot K}\right] \tag{5}$$

for the new approximation, with a range of validity between the melting point and 1100 K.

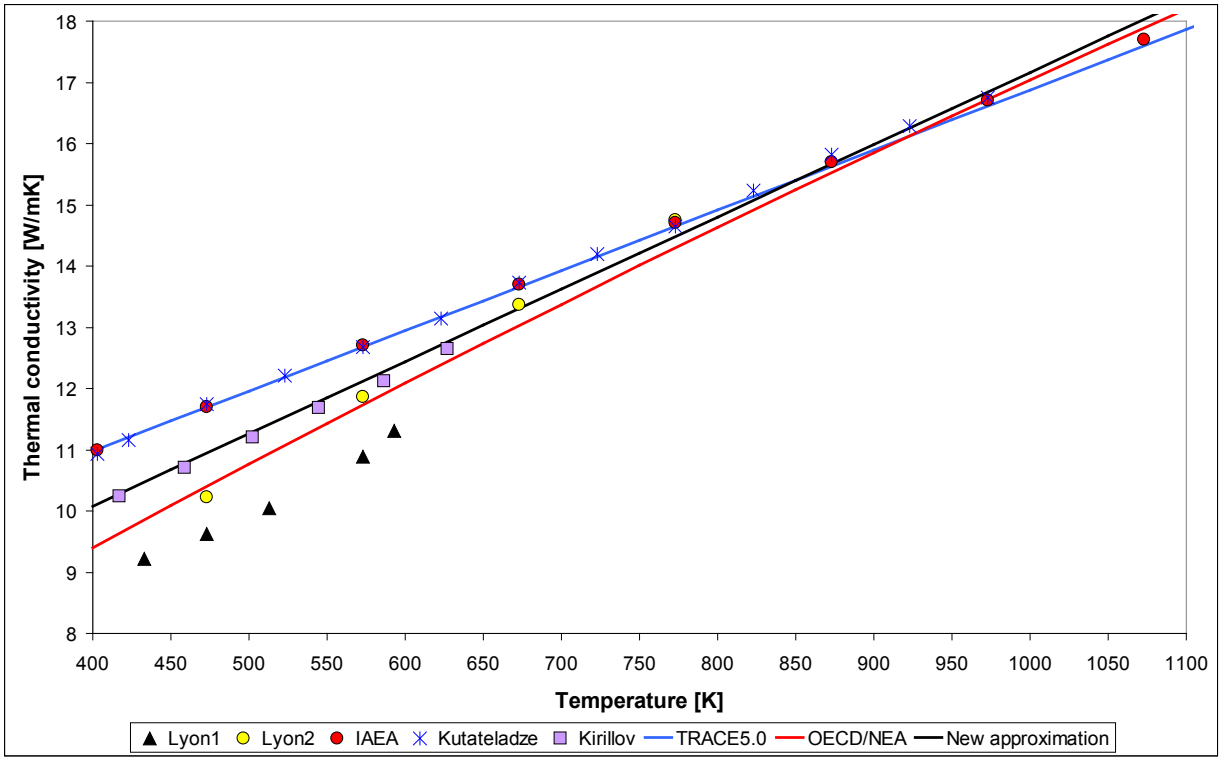

Figure 2. Lead-bismuth thermal conductivity versus temperature

2.1.4 Dynamic viscosity

For the dynamic viscosity, the same citations then for the density were used, see Table 4. These data points, together with the original TRACE function and the new approximation, are exhibited in Figure 3.

The curve of the OECD/NEA and the one for the new approximation are again close together and, it differs from the original TRACE fit. At lower temperatures, the trend of the original TRACE fit is slightly different but at higher temperatures, the difference is more pronounced.

Table 4. Citations and values for the dynamic viscosity of lead-bismuth

Reference	Temperature range [K]	Number of data points
IAEA [9]	403 - 1073	8
Kutateladze et al. [11]	403 - 973	13
Lyon [12]	605 - 873	5
McLain and Martens [13]	588 - 922	7
OECD/NEA [16]	601 - 1500	functional fit

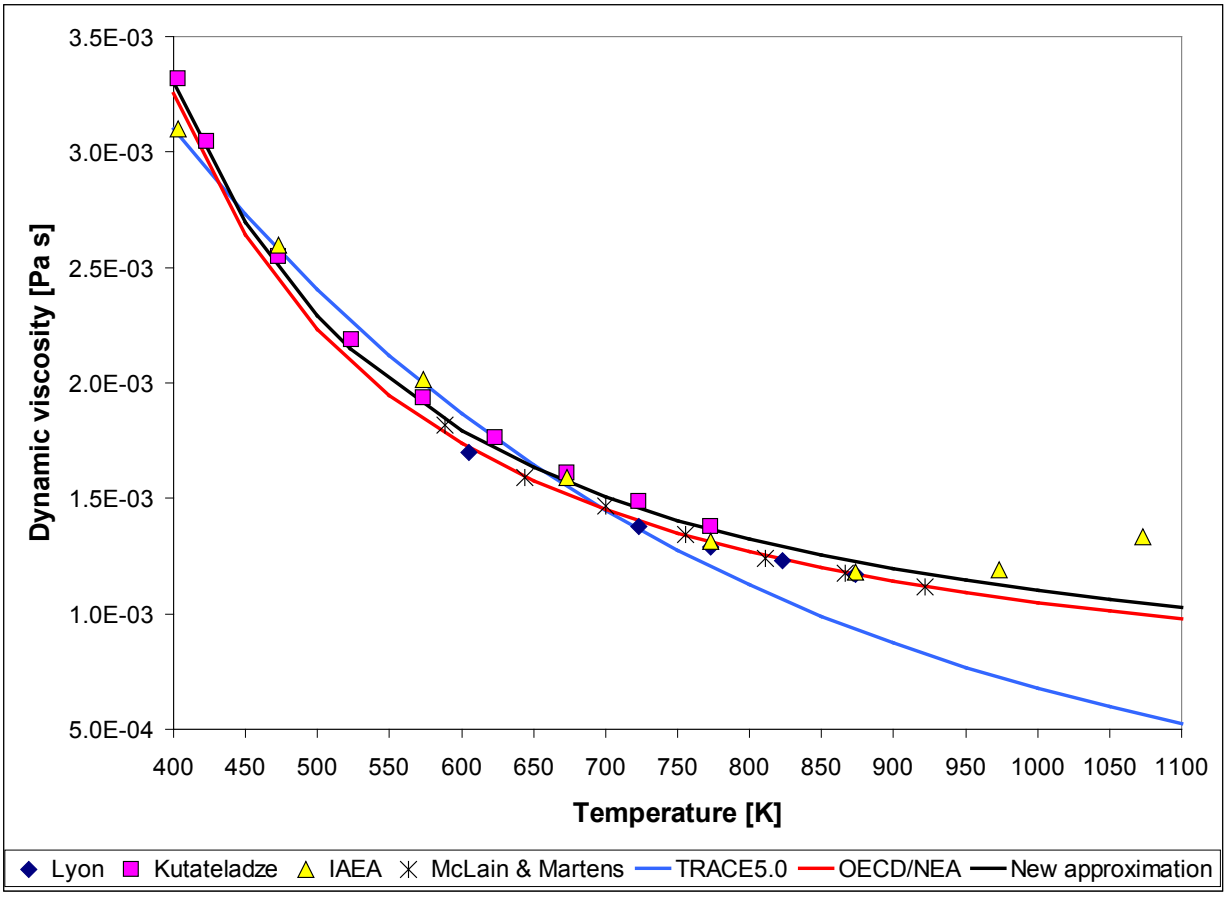

Figure 3. Lead-bismuth dynamic viscosity versus temperature

The functional fits for the original TRACE version (Eq. 6) and the new approximation (Eq. 7) are given below.

$$\eta = 7.685 \cdot 10^{-7} \cdot exp\left(-2.412 \cdot 10^{-3} \cdot T\right) \cdot \rho(T) \qquad [Pa \cdot s] \tag{6}$$

$$\eta = 5.293 \cdot 10^{-4} \cdot exp\left(\frac{732.28}{T}\right) \qquad [Pa \cdot s] \tag{7}$$

The new approximation is valid between the melting point and 1100 K.

2.1.5 Specific heat

Table 5 shows the citations for the specific heat, and Figure 4 shows the plotted data points as well as the functional fits.

Table 5. Citations and values for the specific heat of lead-bismuth

Reference	Temperature range [K]	Number of data points
Hultgren et al. [8]	400 - 1100	8
IAEA [9]	403 - 1073	8
Kutateladze [11]	403 - 1073	15
Lyon [12]	417 - 631	5
McLain and Martens [13]	400 - 723	2
OECD/NEA [16]	400 - 1100	functional fit

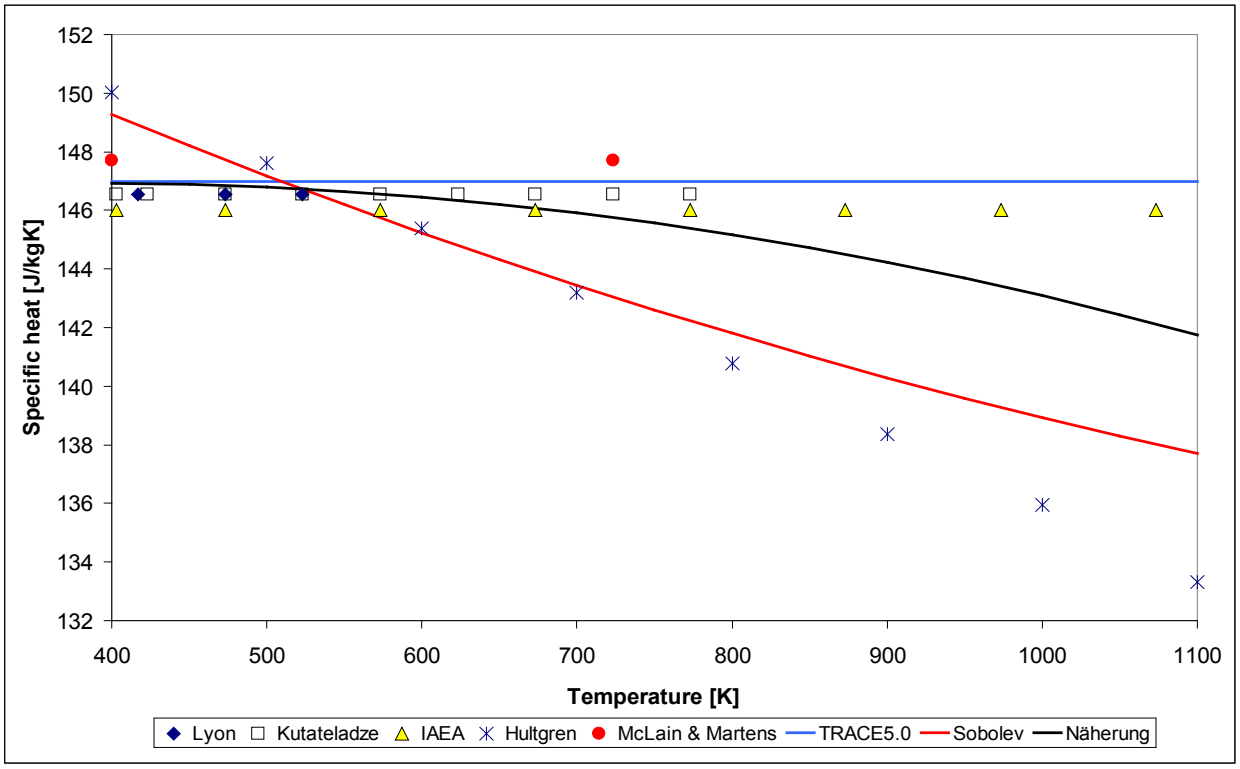

Figure 4. Lead-bismuth specific heat versus temperature

Again, differences between the original TRACE fit and the new approximation are visible. TRACE considers a constant value only (cp = 147 J/kg·K) for the whole temperature range. Due to the discrepancies between the several data sets, the new approximation follows a different trend than the OECD/NEA fit. Nevertheless, the constant value will be replaced by the new approximation, given in Eq. 8.

$$c_p = -9.97 \cdot 10^{-6} \cdot T^2 + 7.622 \cdot 10^{-3} \cdot T + 145 \qquad \left[\frac{N}{m}\right] \qquad (8)$$

The range of validity is given between the melting point and 1100 K

2.1.6 Surface tension

The references for the evaluation of the surface tension are tabulated in Table 6. Figure 5 shows the comparison of the original TRACE fit with the data point and the corresponding new approximation. It looks like that the TRACE fit is based on the Lyon reference. A look to the functional fit of the original TRACE versions (Eq. 9) strengthens this assumption. This fit is valid only between the two data points given by Lyon. For temperatures below 1073 K, the surface tension is constant to 0.367 N/m and for temperatures above 1273 K, it amounts 0.356 N/m.

Table 6. Citations and values for the surface tension of lead-bismuth

Reference	Temperature range [K]	Number of data points
IAEA [9]	403 - 1073	8
Lyon [12]	1073 - 1273	2
Novakovic et al. [15]	623 - 773	8
OECD/NEA [16]	400 - 1300	functional fit

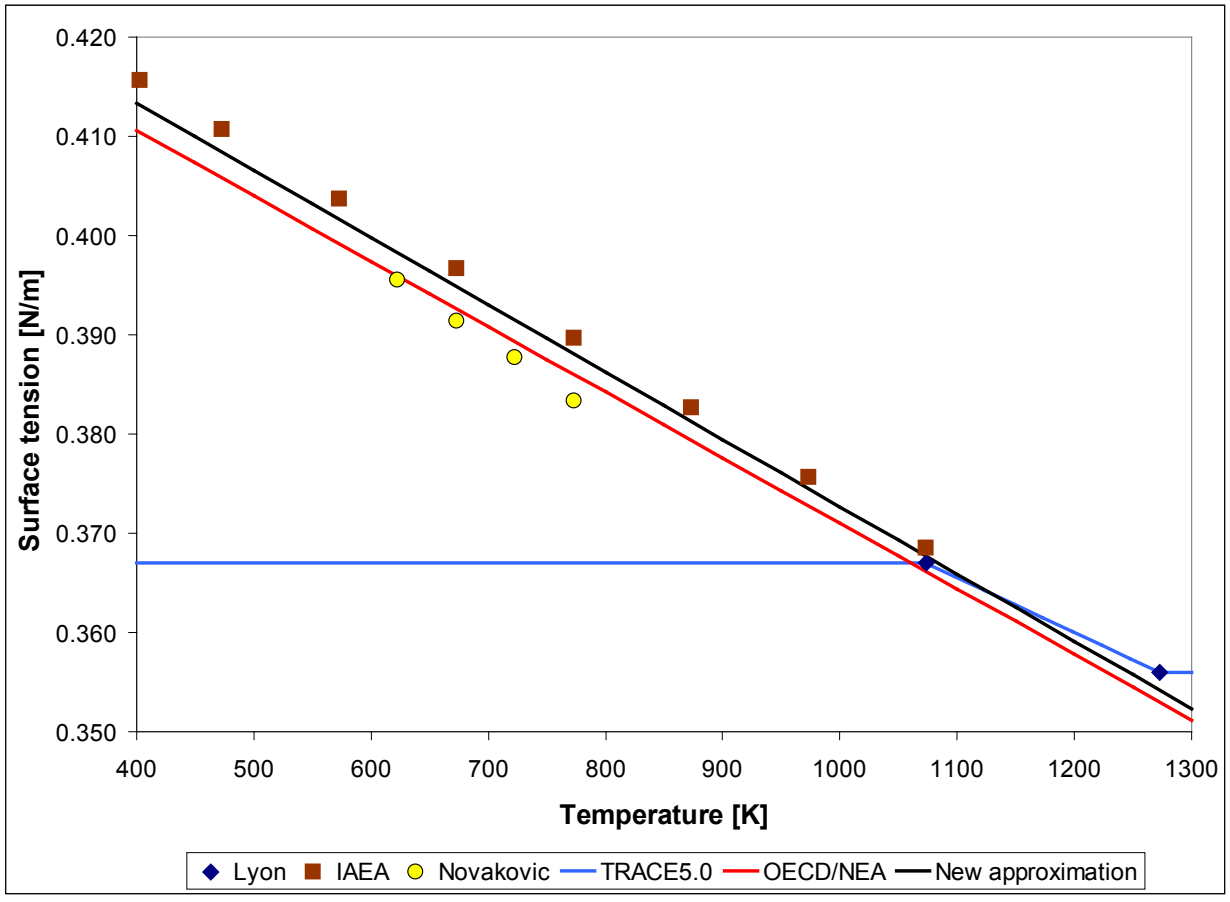

Figure 5. Lead-bismuth surface tension versus temperature

$$\sigma = 0.367 + \frac{(0.356 - 0.367)}{(1273 - 1073)} \cdot (1273 - T) \qquad \left[\frac{N}{m}\right] \tag{9}$$

The new approximation fit is written as follows,

$$\sigma = -6.78 \cdot 10^{-5} \cdot T + 0.405 \qquad \left[\frac{N}{m}\right] \tag{10}$$

and it is valid between the melting point and 1300 K

2.1.7 Saturation pressure

For the saturation pressure, four references have been identified. Three citations (Table 7) contain no data points but temperature dependent correlations.

Table 7. Citations for the saturation pressure of lead-bismuth

Reference	Temperature range [K]	Number of data points
Morita et al. [14]	500 - 2100	functional fit
Ohno, Miyahara and Kurata [17]	823 - 1023	functional fit
OECD/NEA [16]	500 - 2100	functional fit
Petrazzini [18]	500 - 2100	functional fit

For selected temperatures, the functional fits of the three citations were used to produce temperature dependent data points. These new data points were used to develop a new approximation, to replace the one of the original TRACE version.
Figure Figure 6 shows these new generated data points and the fits of TRACE, OECD/NEA and the new approximation.
The agreement between the data points and the approximations is quite well but the logarithmic scale conceals the differences. At higher temperatures, the discrepancy between the original TRACE function (Eq. 11) and the new one (Eq. 12) is relatively large,

$$p = 1.8954 \cdot 10^{9} \cdot exp\left(\frac{-22670}{T}\right) \qquad [Pa] \tag{11}$$

$$p = 1.4552 \cdot 10^{9} \cdot exp\left(\frac{-22849}{T}\right) \qquad [Pa] \tag{12}$$

with a range of validity given between 500 and 2100 K.

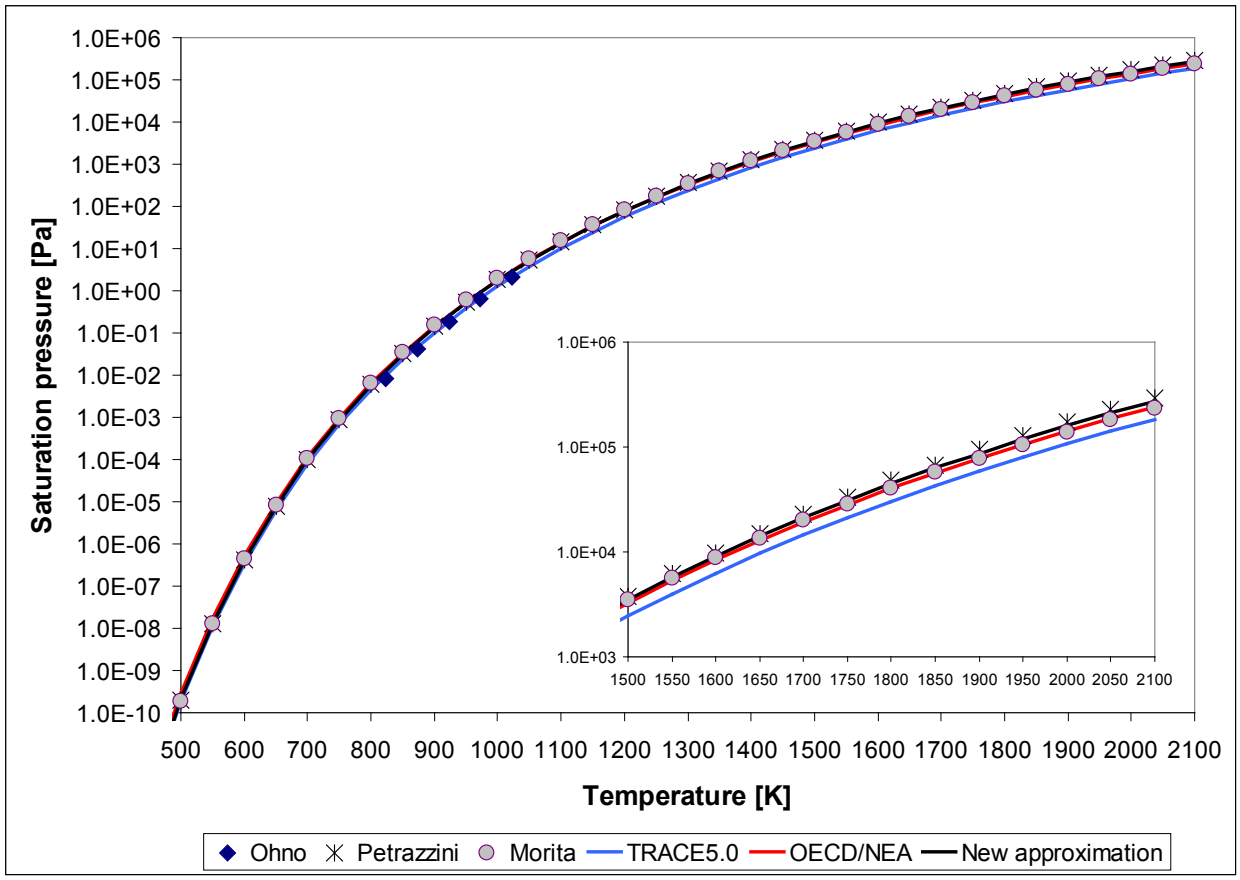

Figure 6. Lead-bismuth saturation pressure versus temperature

2.2 Lead

The thermo-physical properties of lead were implemented into TRACE to investigate lead cooled systems. As for lead-bismuth, the properties will be described in the next sub sections. The used references are listed here: Adamov and Orlov [6], Blaskett and Boxall [7], Brandes and Brook [20], Gurvich [21], Hofman [22], Holman [23], Hultgren et al. [8], IAEA [9], Iida and Guthrie [24], Jauch [25], Kanda and Dominique [in Thurnay], Kirillov et al. [10], Knacke, Kubaschewski, and Hesselmann [26], Kutateladze et al. [11], Lyon [12], OECD/NEA [16], Stull and Sinke [27], and Thurnay [28].

2.2.1 Melting and boiling point

For the melting and boiling point, several citations are used to define the values, see Table 8.

Table 8. Citations and values for the melting and boiling point of lead

Reference	Melting point [K]	Boiling point [K]
Adamov and Orlov [6]	600.600	-
Blaskett and Boxall [7]	600.550	1998.15
Brandes and Brook [20]	600.612	2023.15
Gurvich [21]	600.650	2019.00
Hofman [22]	600.450	2013.15
Hultgren et al. [8]	600.600	-
IAEA [9]	600.550	2018.15
Knacke, Kubaschewski and Hesselmann [26]	600.652	2020.00
Kutateladze et al. [11]	600.550	2013.15
Thurnay [28]	600.650	2021.84
Tipton [19]	600.550	2013.15

$$T_{melt} = (600.58 \pm 0.06)K$$

$$T_{boil} = (2015.53 \pm 7.12)K$$

2.2.2 Density

The citations for the temperature-dependent lead density are gathered in Table 9, and the data sets are shown in Figure 7.

Table 9. Citations and values for the density of lead

Reference	Temperature range [K]	Number of data points
Adamov and Orlov [6]	610 - 1200	9
IAEA [9]	603 - 1073	6
Kutateladze et al. [11]	673 - 1073	9
Lyon [12]	673 - 1273	5
OECD/NEA [16]	601 - 1300	functional fit
Thurnay [28]	601 - 1300	functional fit

$$\rho = -1.2272 \cdot T + 11408 + \frac{dp}{d\rho} \qquad \left[\frac{kg}{m^3}\right] \tag{13}$$

For the pressure dependent term, a speed of sound of 1790 m/s has been assumed. Hence, Eq. 3 writes now as follows

$$\frac{dp}{d\rho} = \frac{1}{\left[1790 - 0.5 \cdot \left(T - T_{melt}\right)\right]} \tag{14}$$

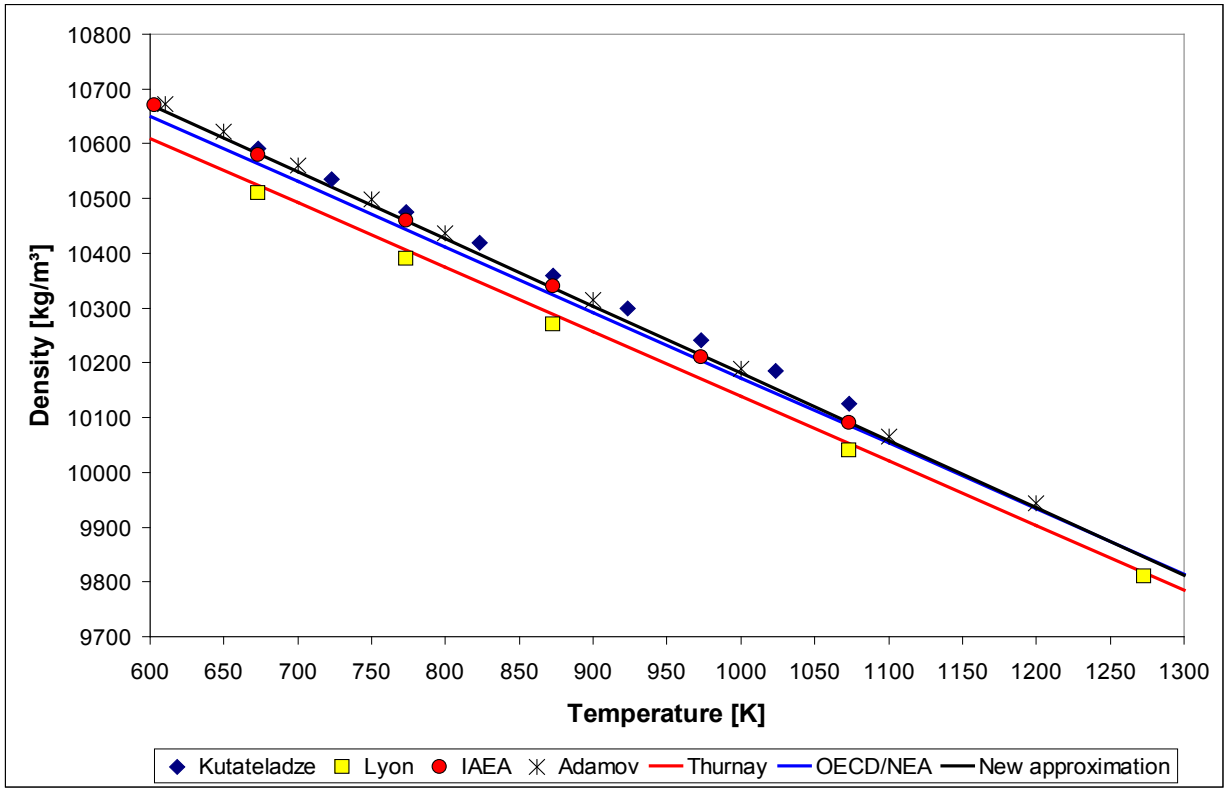

Figure 7. Lead density versus temperature

2.2.3 Thermal conductivity

For the thermal conductivity of lead, the electrical conductivity was again used for some of the references and it was transferred via the Wiedemann-Franz-Lorenz. The data points are again depicted in a diagram see Figure 8.

One can see there, that for older citations the difference between the measured thermal conductivity and the one calculated from the electrical conductivity is very pronounced (Hofman and Lyon). Whereas for later citations (Brandes and Brook), the two options are almost identical.

Table 10. Citations and values for the thermal conductivity of lead

Reference	Temperature range [K]	Number of data points

Reference	Temperature range [K]	Number of data points
Adamov and Orlov [6]	673 - 973	2
Brandes and Brook1 [20]	600 - 873	4
Brandes and Brook2 [20]	600 - 1273	6
Hofman1 [22]	601 - 873	4
Hofman2 [22]	603 - 1129	8
Holman [23]	644 - 977	2
IAEA [9]	603 - 1073	6
Iida and Guthrie [24]	601 - 871	6
Kirillov et al. [10]	601 - 1273	6
Kutateladze et al. [11]	673 - 1073	9
Lyon1 [12]	603 - 973	5
Lyon2 [12]	601 - 1273	5
Thurnay [28]	601 - 1073	6
OECD/NEA [16]	601 - 1300	functional fit

Following approximation will be implemented into TRACE:

$$k = 0.0101 \cdot T + 9.9855 \qquad \left[\frac{W}{m \cdot K}\right] \tag{15}$$

and it is valid between the melting point and 1300 K.

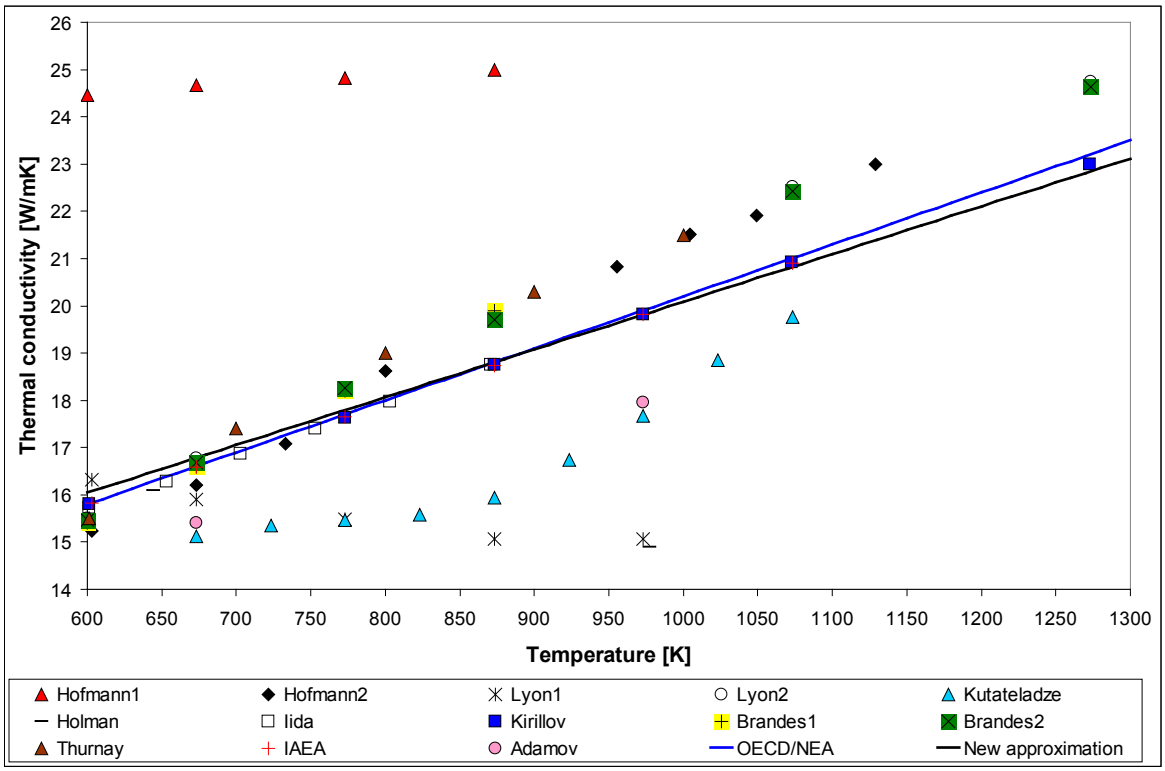

Figure 8. Lead thermal conductivity versus temperature

2.2.4 Dynamic viscosity

The citations reviewed for the dyanmic viscosity are summarized in Table 12 and the corresponding data points are displayed in Figure 9.

Table 11. Citations and values for the dynamic viscosity of lead

Reference	Temperature range [K]	Number of data points
IAEA [9]	603 - 1073	9
Kanda and Dominique [28]	602 - 773	8
Kutateladze et al. [11]	673 - 1073	9
Lyon [12]	714 - 1117	5
Thurnay [28]	601 - 800	functional fit
OECD/NEA [16]	601 - 1500	functional fit

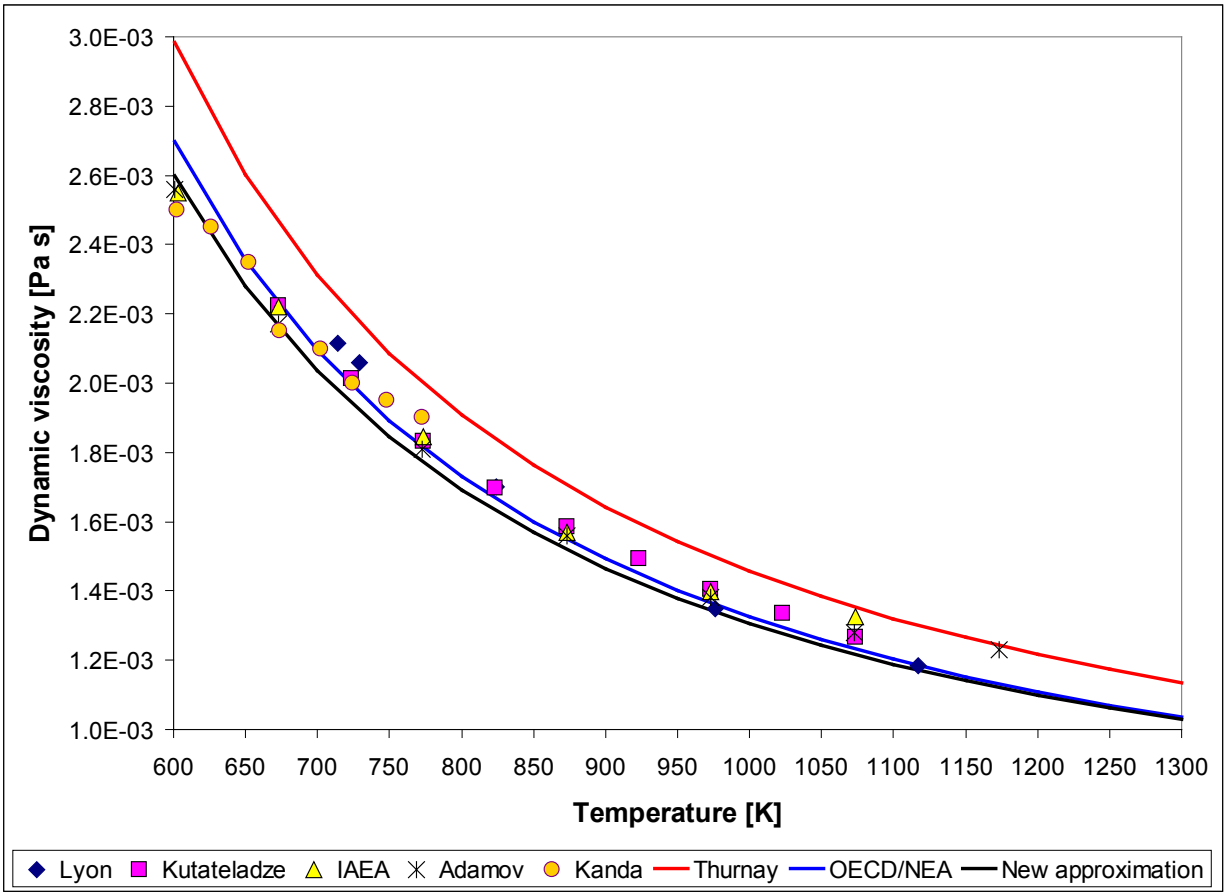

Figure 9. Lead dynamic viscosity versus temperature

One can clearly see that the data points and the new approximation are in good agreement. The fit of Thurnay over predicts the dynamic viscosity by about 10 % compared to the new approximation. The fit of the OECD/NEA is in good agreement to the new approximation, which is given below and it is valid between the melting point and 1300 K.

$$\eta = 4.636 \cdot 10^{-4} \cdot exp\left(\frac{1036}{T}\right) \qquad [Pa \cdot s] \tag{16}$$

2.2.5 Specific heat

Ten sets of data points were found in the open literature for the evaluation the dynamic viscosity of lead. As shown in Table 12, values for temperatures of up to 2000 K are available. One functional fit is even valid until 3600 K.

Table 12. Citations and values for the specific heat of lead

Reference	Temperature range [K]	Number of data points
Adamov and Orlov [6]	601 - 1000	5
Brandes and Brook [20]	601 - 873	4
Holman [23]	644 - 977	2
Hultgren et al. [8]	601 - 2000	14
IAEA [9]	673 - 1073	5
Knacke, Kubaschewski and Hesselmann [26]	601 - 2000	14
Kutateladze et al. [11]	673 - 1073	9
Lyon [12]	601 - 773	3
Stull and Sinke [27]	700 - 2000	14
Thurnay [28]	601 - 2021	16
Gurvich [21]	601 - 3600	functional fit
OECD/NEA [16]	601 - 2000	functional fit

The plotted data points in Figure 10 show the discrepancies among them. However, the new approximation is in good agreement to other functional fits. At lower temperatures, the new approximation over predicts the specific heat, compared to the other fits, but at higher temperatures (1300 K), the new fit is almost identical to the one of Gurvich. The one of the OECD/NEA fit are always below the other two.

The new approximation, valid between the melting point and 2000 K, is expressed as follows:

$$c_p = 1.133 \cdot 10^{-6} \cdot T^2 - 0.03859 \cdot T + 170 \qquad \left[\frac{J}{kg \cdot K}\right] \tag{17}$$

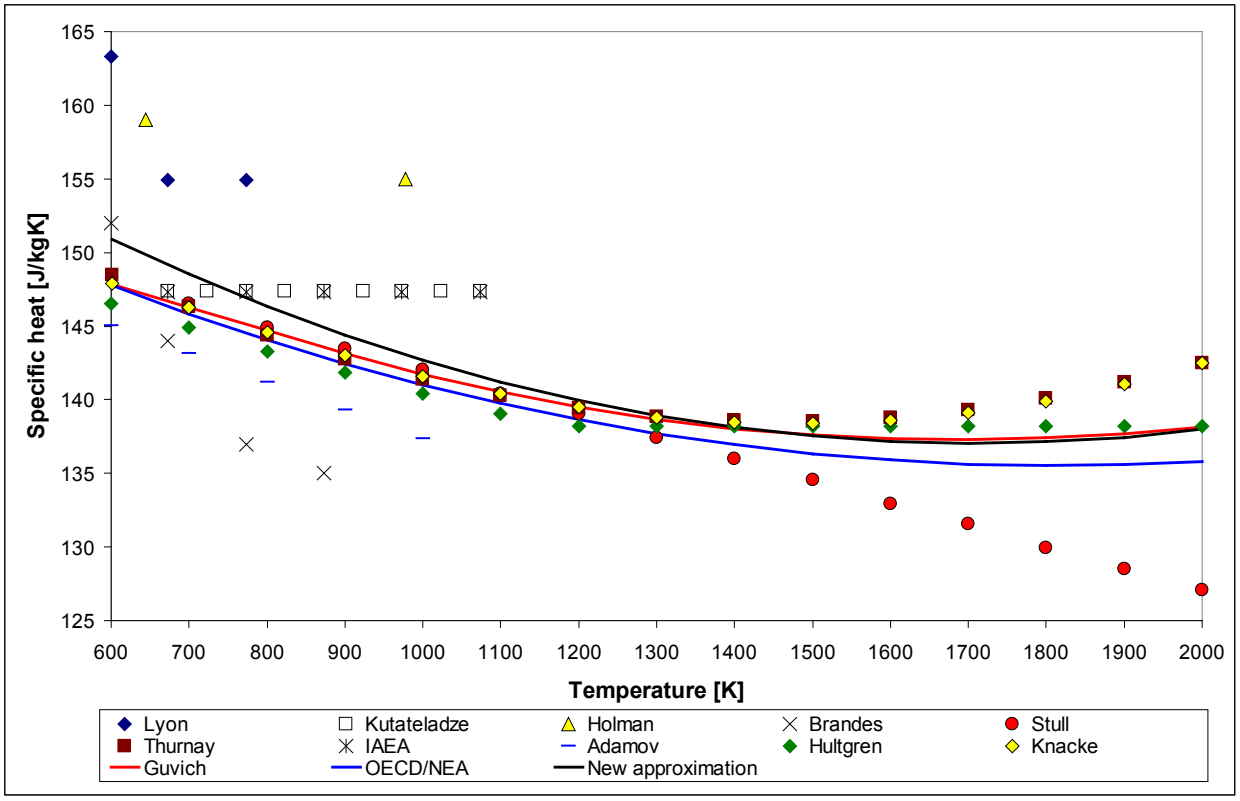

Figure 10. Lead specific heat versus temperature

2.2.6 Surface tension

The citations for the surface tension can be found in Table 13, and the graphical representation is given in Figure 11.

The new approximation is nicely enveloped by the functional fits of the OECD/NEA and Thurnay, and the agreement with measured data is good.

Table 13. Citations and values for the surface tension of lead

Reference	Temperature range [K]	Number of data points
Adamov and Orlov [6]	601 - 1023	10
Brandes and Brook [20]	601 - 1023	10
IAEA [9]	603 - 1073	6
Jauch [25]	601 - 1300	15
Lyon [12]	623 - 773	4
Thurnay [28]	601 - 1200	functional fit
OECD/NEA [16]	601 - 5400	functional fit

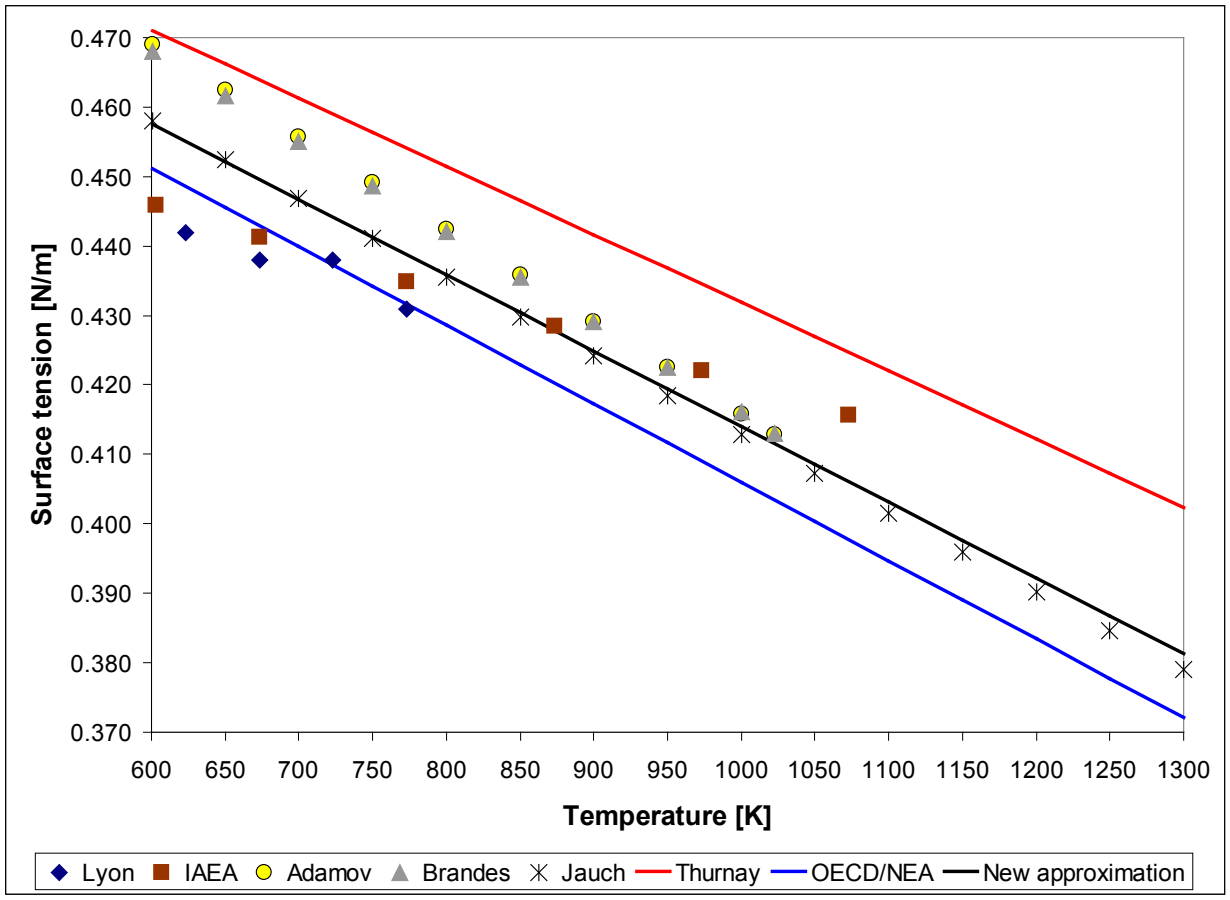

Figure 11. Lead surface tension versus temperature

The function for the surface tension writes as follows, and is valid between the melting point and 1300 K

$$\sigma = -1.09 \cdot 10^{-4} \cdot T + 0.523 \qquad \left[\frac{N}{m}\right] \tag{18}$$

2.2.7 Saturation pressure

The last parameter is again the saturation pressure. Four data sets and two fit were found (see Table 14) to derive a functional fit. The data sets are shown in Figure 12 together with the functional fits of Weast [25], the OECD/NEA, and with the new approximation.

The agreement among the data sets and the functional fits is good. At lower temperatures, the fit of the OECD/NEA and the new approximation are slightly apart from each other. The new approximation is valid between the melting point and 2000 K and it is given as follows:

$$p = 6.718 \cdot 10^{9} \cdot exp\left(\frac{-22237}{T}\right) \qquad [Pa] \tag{19}$$

Table 14. Citations and values for the saturation pressure of lead

Reference	Temperature range [K]	Number of data points
Adamov and Orlov [6]	601 - 1200	6
Brandes and Brook [20]	601 - 2050	30
Hultgren et al. [8]	601 - 2100	17
Lyon [12]	1260 - 1884	5
OECD/NEA [16]	798 - 1598	functional fit
Weast	601 - 2000	functional fit

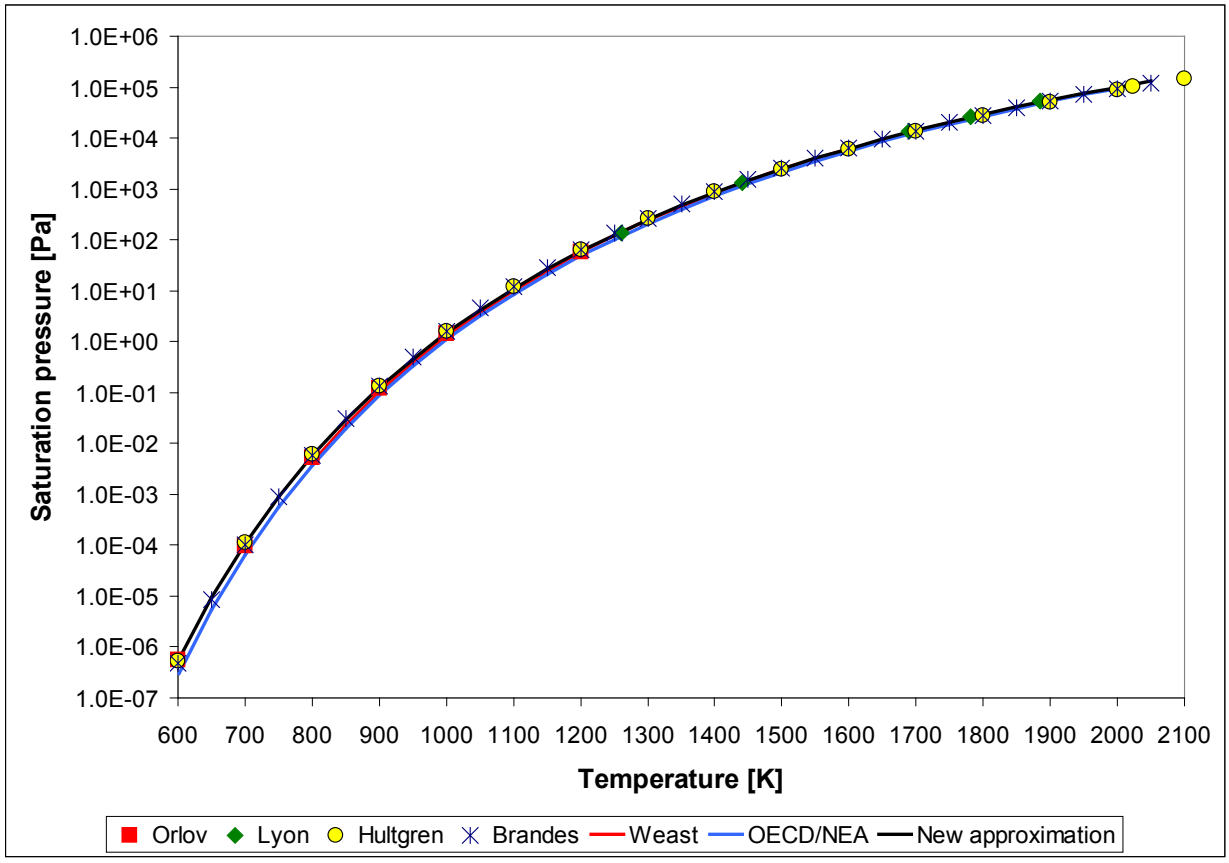

Figure 12. Lead saturation pressure versus temperature

2.3 Diphyl THT

Since Diphyl THT is used in lead/lead-bismuth experimental facilities as intermediate coolant, it is necessary also to implement the thermo-physical properties of DTHT in the TRACE so that these experiments can be used for code validation purposes. Diphyl is a registered trademark of the LANXESS Deutschland GmbH.

This coolant, Diphyl THT (DTHT), is an organic heat transfer medium used in the temperature range of 273 - 520 K. The thermo-physical properties of this coolant are given below. Two

references were used to evaluate these properties, namely the VDI Waermeatlas [29] and Chen [30]. The density is defined by the relation:

$$\rho = -0.6495 \cdot T + 1194 \qquad \left[\frac{kg}{m^3}\right] \tag{20}$$

Thermal conductivity is given by the following equation:

$$k = -2.544 \cdot 10^{-5} \cdot T + 0.1167 \qquad \left[\frac{W}{m \cdot K}\right] \tag{21}$$

Dynamic viscosity is expressed as follows:

$$\eta = 2.4163 \cdot 10^{-3} \cdot exp\left(\frac{1.194 \cdot 10^{6}}{T^2} - \frac{3.037 \cdot 10^{3}}{T}\right) \qquad [Pa \cdot s] \tag{22}$$

The specific heat is defined by the equation:

$$c_p = 3.4757 \cdot T + 513.84 \qquad \left[\frac{N}{m}\right] \tag{23}$$

The surface tension is given as follows:

$$\sigma = 9.22 \cdot 10^{-8} \cdot (T - 273.15)^2 + 1.451 \cdot 10^{-4} \cdot (T - 273.15) + 0.0462 \qquad \left[\frac{N}{m}\right] \tag{24}$$

The saturation pressure can be expressed as:

$$p = 4.0884 \cdot 10^{10} \cdot exp\left(\frac{-8084.4}{T}\right) \qquad [Pa] \tag{25}$$

These properties in dependence of the temperature are shown in the Figure 13 to Figure 18.

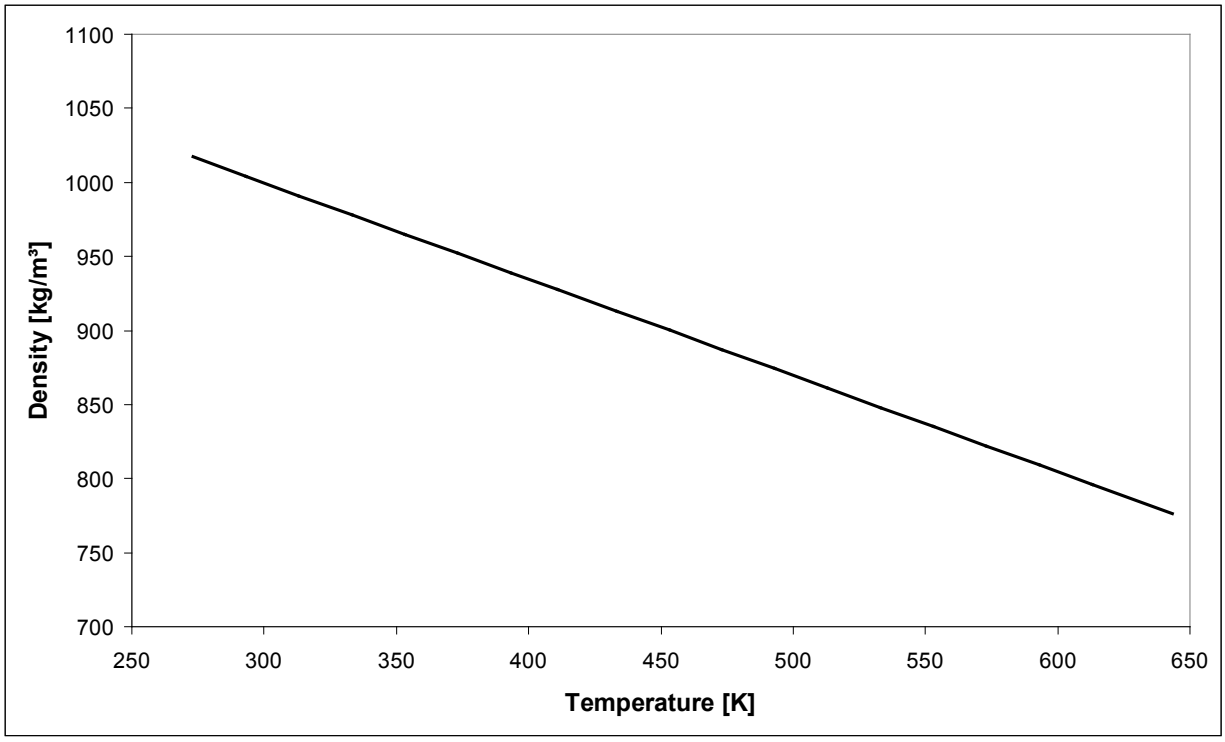

Figure 13. Diphyl THT density versus temperature

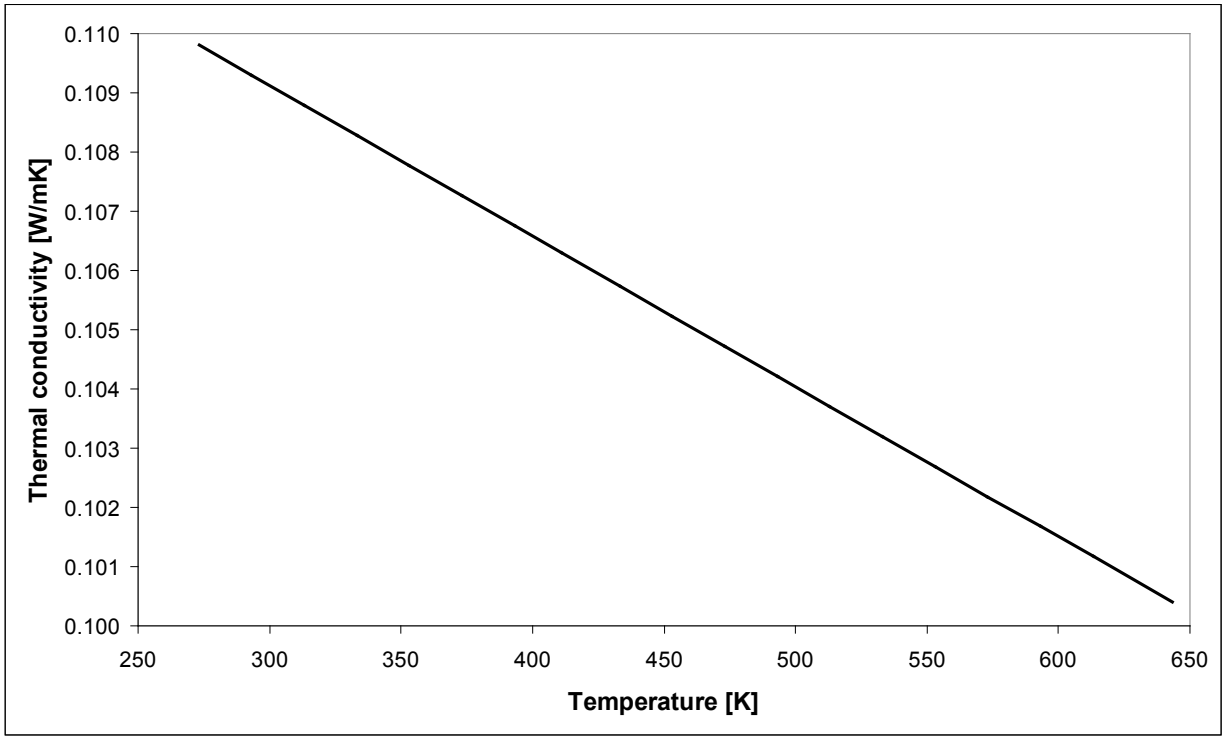

Figure 14. Diphyl THT thermal conductivity versus temperature

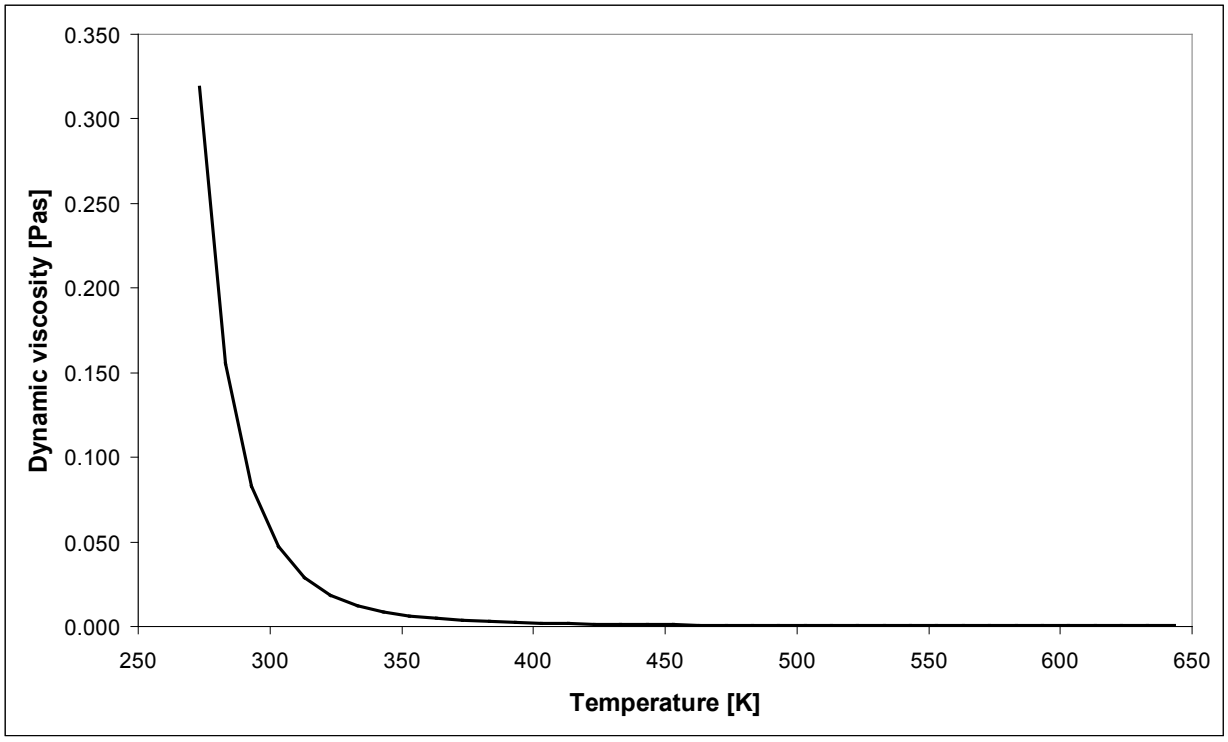

Figure 15. Diphyl THT dynamic viscosity versus temperature

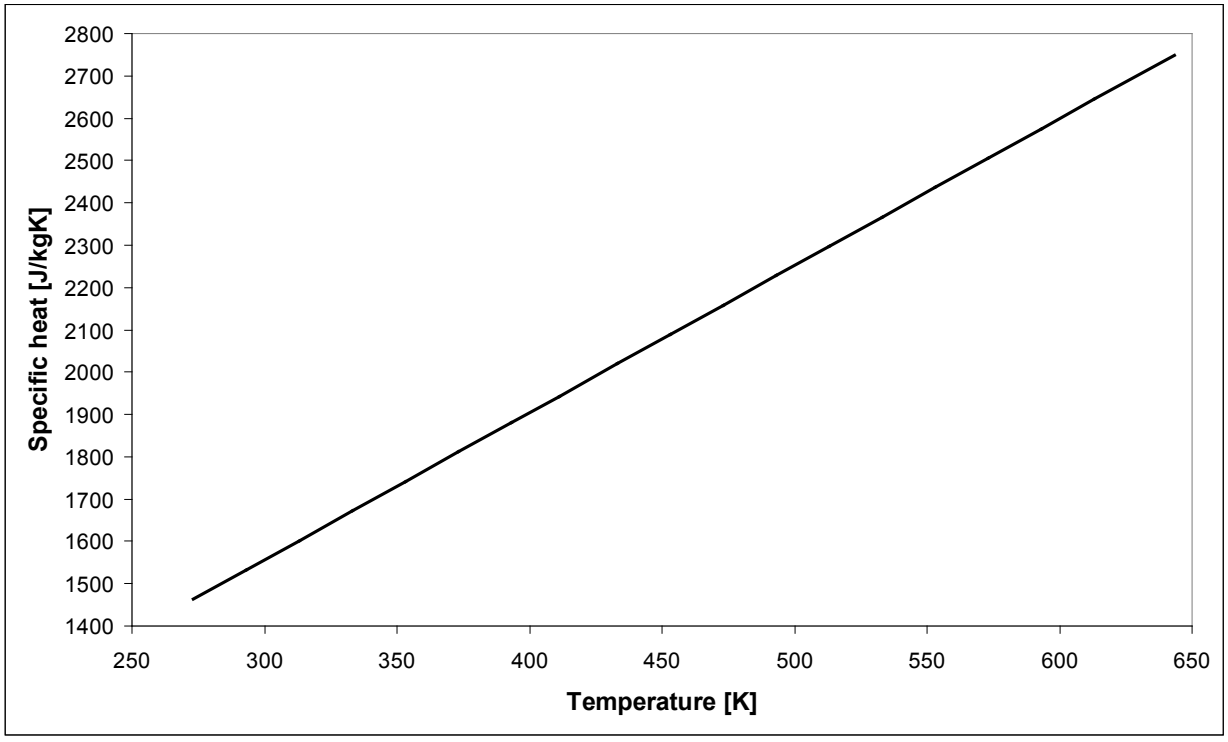

Figure 16. Diphyl THT specific heat versus temperature

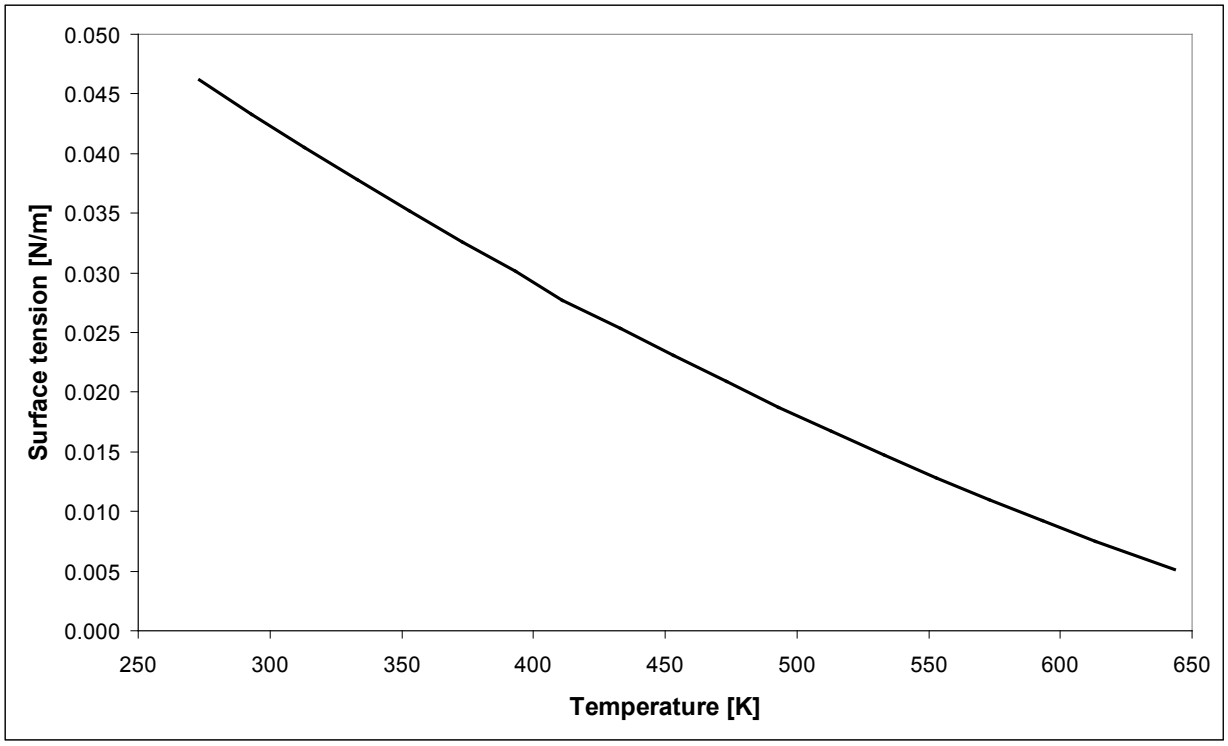

Figure 17. Diphyl THT surface tension versus temperature

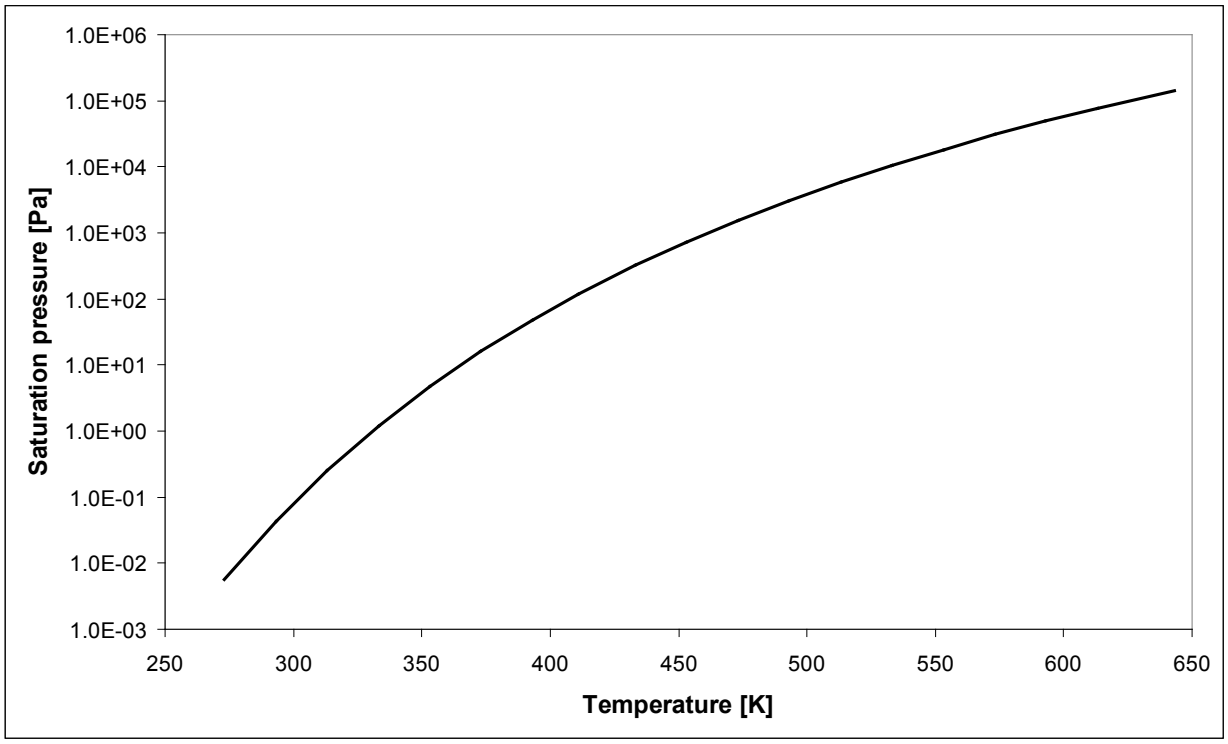

Figure 18. Diphyl THT saturation pressure versus temperature

3 HEAT TRANSFER MODELS

3.1 Heat transfer in pipes to fluids with Prandtl Numbers << 1

The TRACE correlation writes as follow.

$$Nu = 4.8 + 0.025 \cdot Pe^{0.8} \tag{26}$$

A review of the heat transfer models of TRACE for fluids with very low Prandtl numbers revealed that the used correlation could not be found in a reference the author of this report is aware of. Neither the TRACE theory manual [31] nor the TRACE source codes give hinds where the correlation was taken from. Hence, a review of literature were performed to identify a correlation which is commonly used and validated.

The most common used correlation [32], [33] is the one of Seban and Shimazaki [34] and writes as follow:

$$Nu = 5 + 0.025 \cdot Pe^{0.8} \tag{27}$$

Indeed, the difference is quite small but for reason of thoroughness, Eq. 27 will replace the original correlation

3.2 Heat transfer in bundles to fluids with Prandtl Numbers << 1

Unfortunately, TRACE does not distinguish between pipe and bundle flow. Whereas for other fluids, TRACE has both options available. This has been achieved by connecting the hydraulic component to the inner side of a heat structure, which represents a pipe with internal heat source (channel/bundle), or to the outer side, which represents a pipe with heated wall. The same kind of code structure was used to distinguish between pipe and bundle flow of liquid metals.

Since the actual structure of TRACE does not allow differing between quadratic and hexagonal arrays, a general correlation needs to be implemented.

Investigations of Pfrang and Struwe [32], and Mikityuk [33] could identify a correlation of Ushakov et al. [35] as the most suitable one. This correlation is given below

$$Nu = Nu_{lam} + \frac{3.67}{90 \cdot \left(\frac{p}{d}\right)^2} \cdot Pe^{\left[0.56+0.19\cdot\left(\frac{p}{d}\right)-0.1\cdot\left(\frac{p}{d}\right)^{-80}\right]} \cdot \left\{1 - \frac{1}{\frac{1}{6}\cdot\left[\left(\frac{p}{d}\right)^{30} - 1\right] + \sqrt{1.15 + 1.24 \cdot \varepsilon}}\right\} \tag{28}$$

with

$$Nu_{lam} = \left[7.88 \cdot \left(\frac{p}{d}\right) - 6.3 \cdot \left(\frac{p}{d}\right)^{-17 \cdot \left(\frac{p}{d}\right) \cdot \left(\frac{p}{d} - 0.81\right)} \right] \cdot \left[1 - \frac{3.6 \cdot \left(\frac{p}{d}\right)}{\left(\frac{p}{d}\right)^{20} \cdot \left(1 + 2.5 \cdot \varepsilon^{0.86}\right) + 3.2} \right] \quad (29)$$

This correlation can be applied for Peclet numbers between 1 and 4000, p/d ratios between 1 and 2, and for ε between 0.01 and infinity. This factor is the approximate criterion of thermal similarity of the fuel rods, depending on the geometry and the thermal conductivity of the pin.

A simplified version of Eq. 28 is given below, with the condition that the p/d ration should be greater than 1.3.

$$Nu = 7.55 \cdot \left(\frac{p}{d}\right) - 20 \cdot \left(\frac{p}{d}\right)^{-13} + \left(\frac{3.67}{90}\right) \cdot \left(\frac{p}{d}\right)^{-2} \cdot Pe^{\left[0.56 + 0.19 \cdot \left(\frac{p}{d}\right)\right]} \quad (30)$$

As soon as one inserts the given p/d ratio, the correlation will be comparable to Eq. 27.

3.3 Heat transfer in helical coils to fluids with Prandtl Numbers ≈ 1

The post-test analysis of experiments with lead-alloy coolants made it necessary to implement the thermo physical properties of Diphyl THT, described in subsection 2.3. Usually one can apply the correlation for forced convective pipe flow, given below, since Diphyl THT has a Prandtl number in between 10^0 - 10^2.

$$Nu = \frac{f \cdot (Re - 1000) \cdot Pr}{\left[1 - 12.7 \cdot \sqrt{f} \cdot \left(Pr^{0.66} - 1\right)\right]} \cdot \left(\frac{Pr}{Pr_w}\right)^{0.11} \quad (31)$$

where

$$f = \frac{0.5}{\left[1.58 \cdot ln(Re) - 3.28\right]^2} \quad (32)$$

The term $(Pr/Pr_w)^{0.11}$ gave respect to the temperature gradient between the bulk and the wall. A detailed check of the TRACE module Reflood.f90 revealed that this factor is wrongly implemented. Actually, TRACE uses the bulk Prandtl number also for the wall Prandtl number. Hence, this factor will be always unity. This deficiency was removed by re-writing the function VarLiqPropEffct.

Since this correlation is valid only for straight pipes but the Diphyl THT will be used in heat exchangers with helical upward flow paths (details in the next section), a new correlation needs to be identified taking into account the geometrical peculiarities of the flow path.

Three correlations were selected and subsequently implemented into the TRACE source code. The results will be given in the next section.
The first correlation is a correlation taken from Gnielinski [36].

$$Nu = \frac{\frac{f}{2} \cdot Re \cdot Pr}{\left[1 - 12.7 \cdot \sqrt{\frac{f}{2}} \cdot \left(Pr^{0.66} - 1\right)\right]} \cdot \left(\frac{Pr}{Pr_w}\right)^{0.14} \tag{33}$$

This correlation is similar to the standard one, Eq. 31, since it is originated from the same research team. The main difference to the standard one is the usage of different friction factors. Equation 33 employs friction factors of Ito [37], which are derived from experiment with coiled tubes, and are given next.

$$f = \begin{cases} \dfrac{16}{Re} & Re \le 13 \cdot \left(\dfrac{D}{d}\right)^{-0.5} \\ \dfrac{344 \cdot \left(\dfrac{D}{d}\right)^{-0.5}}{\left\{1.56 + log_{10}\left[Re \cdot \left(\dfrac{D}{d}\right)^{-0.5}\right]\right\}^{5.73}} & 13 \cdot \left(\dfrac{D}{d}\right)^{-0.5} \le Re \le 200 \cdot \left[1 + 13.2 \cdot \left(\dfrac{D}{d}\right)^{-0.5}\right] \\ 0.076 \cdot Re^{-0.25} + 0.0075 \cdot \left(\dfrac{D}{d}\right)^{-0.5} & Re \ge 15000 \end{cases} \tag{34}$$

One can see that these friction factors use a D/d ratio. That ratio gives respect to the diameter of the spiral or helix and d is the diameter of the coiled tube.

The second correlation is a correlation developed by Leung, Milenkovich and Class [38] by modeling the flow path with CFX10 to derive are new Nusselt correlation.

$$Nu = 0.013 \cdot Re^{0.87} \cdot Pr^{0.47} \tag{35}$$

The third and last correlation could be found in the work of Rogers and Mayhew [39] and is similar to the Dittus-Boelter correlation, as shown in the next equation

$$Nu = 0.023 \cdot Re^{0.85} \cdot Pr^{0.4} \cdot \left(\frac{d}{D}\right)^{0.1} \tag{36}$$

In addition to the correlations, the selection scheme for the heat transfer correlation was changed in order to guarantee that TRACE picks the right correlation, see the appendix with the code modifications.

4 TRACE VALIDATION

4.1 The LACANES Benchmark

The OECD/NEA organized a benchmark dealing with the thermal hydraulic safety issues of lead alloy-cooled advanced nuclear energy systems (LACANES) [40]. The experimental data for the benchmark are obtained from the HELIOS [41] facility (see Figure 19) at the Seoul National University (SNU). After characterizing the experimental loop, results for isothermal steady state forced convection scenarios were obtained in sense of pressure drops along the flow path. The benchmark participants did the post-test analysis of this scenario without knowing the experimental pressure losses. This contains a case with a low mass flow rate and one with a high one.

The task for the participants was to provide detailed information about the friction losses and form losses for the components, and at certain positions of the loop. Thus, the different physical models, related to pressure drop, of the codes the participants used, could be compared.

The total pressure drop can be calculated via the following correlation.

$$\Delta p_{total} = \frac{1}{2} \cdot \rho \cdot \sum \left[v_{local}^2 \cdot \left(f \cdot \frac{l}{d} + K \right) \right] \tag{37}$$

To calculate the total pressure drop, one has to provide information about the geometrical and thermal hydraulic dependent losses (e.g., influence of mass flow rate, change in flow direction/area). For the friction losses, the Churchill correlation [42] is implemented in TRACE (see Eq.38)

$$f = 8 \cdot \left[\left(\frac{8}{Re} \right)^{12} + \frac{1}{(A+B)^{3/2}} \right]^{1/12} \tag{38}$$

where

$$A = \left\{ 2.457 \cdot ln \left[\left(\frac{7}{Re} \right)^{0.9} + 0.27 \cdot \left(\frac{\Delta}{d} \right) \right] \right\}^{16} \tag{39}$$

and

$$B = \left(\frac{37530}{Re} \right)^{16} \tag{40}$$

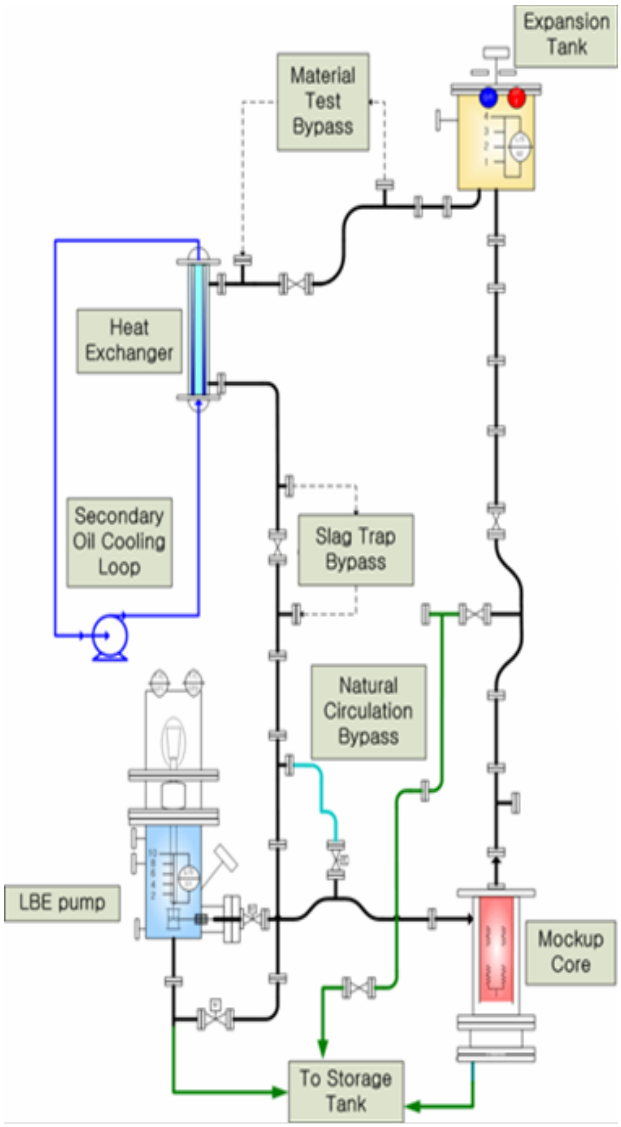

Figure 19. Schematic diagram of the HELIOS facility

Concerning the form loss coefficients TRACE has an option to include losses due to sudden flow are changes. The corresponding correlations are given below,

$$K = \left(1 - \frac{A_j}{A_{j+1}}\right) \tag{41}$$

for an sudden expansion, and

$$K = 0.5 - 0.7 \cdot \left(\frac{A_{j+1}}{A_j}\right) + 0.2 \cdot \left(\frac{A_{j+1}}{A_j}\right)^2 \tag{42}$$

for an sudden contraction.

Since the HELIOS loop facilitates not only sudden expansions and contractions but other geometrical changes (e.g., bends, tees, valves), additional K factors needs to be provided. Some correlations for K factors will be given hear.
The form loss related to an orifice was calculated according to a correlation found in Idelchik [43].

$$K = K' \cdot \left(1 - \frac{A_{j+1/2}}{A_j}\right) + \left(1 - \frac{A_{j+1/2}}{A_{j+1}}\right)^2 + \tau \cdot \sqrt{1 - \frac{A_{j+1/2}}{A_j}} \cdot \left(1 - \frac{A_{j+1/2}}{A_{j+1}}\right) + K_{fr} \qquad (43)$$

The same reference also provides a correlation for K factors in bends, as shown below

$$K = k_{\Delta} \cdot k_{Re} \cdot K_{loc} + K_{fr} \qquad (44)$$

where

$$K_{loc} = A_1 \cdot B_1 \cdot C_1 \qquad (45)$$

and

$$K_{fr} = 0.0175 \cdot \frac{R_0}{d} \cdot \delta \cdot f \qquad (46)$$

The parameters k_{Δ}, k_{Re}, K_{fric}, K', A_1, B_1, C_1, τ and f depend on the geometry (Δ/d) and the thermal hydraulic condition(Re), and can be found in Idelchik [43].

A correlation for spacer grids, Rehme [44], was used since the heat exchanger and the mock up core employ them.

$$K = C \cdot \varepsilon \qquad (47)$$

$$C = 3.5 + \left(\frac{73.5}{Re^{0.264}}\right) + \left(\frac{2.79 \cdot 10^{10}}{Re^{2.79}}\right) \qquad (48)$$

$$\varepsilon = \frac{A_{grid}}{A_u} \qquad (49)$$

C denotes the form loss across the spacer and ε is the ratio of the projected spacer grid cross section and the undisturbed flow area

For the definition of the used flow areas in the above displayed correlations, please refer to Figure 20.

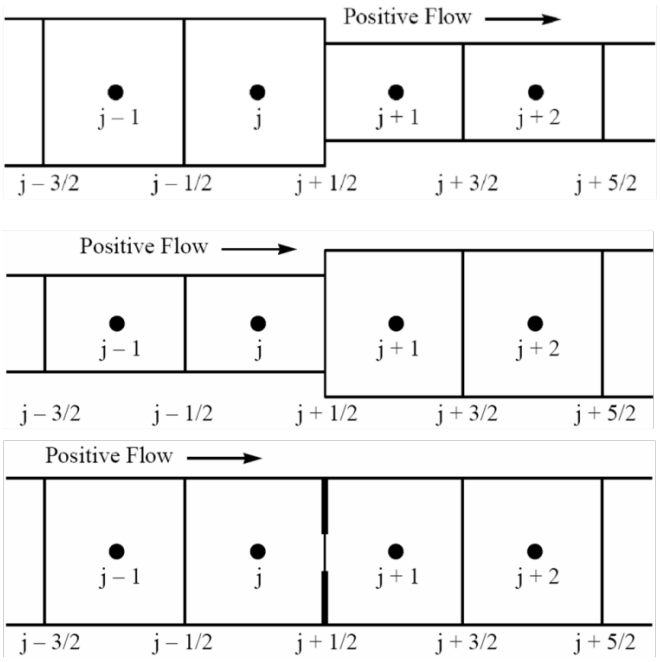

Figure 20. Definition of index specific flow areas

Values for the K factors in tees were taken out of the VDI Waermeatlas [29]. For reasons of reproducibility of the results, a table containing all calculated K factors will be provided for both cases.

Table 15. K factors for various components at high or low mass flow rate

Component		Low mass flow rate (3.27 kg/s)		High mass flow rate (13.57 kg/s)	
		Ref. vel. (m/s)	K	Ref. vel. (m/s)	K
Bend	45°	0.1632	0.2289	0.6774	0.1593
	90°	0.1632	0.3174	0.6774	0.2223
Tee	Straight	0.1632	0.0000	0.6774	0.0000
	Branch	0.1632	2.0000	0.6774	2.0000
Expansion tank	Inlet	0.1632	0.9993	0.6774	0.9993
	Outlet	0.1632	0.4818	0.6774	0.4818
Core	Inlet	0.0324	0.3780	0.1342	0.3780
	Outlet	0.2216	0.4690	0.9196	0.4690
	Spacer	0.2216	6.5502	0.9196	5.2825
Heat exchanger	Inlet	0.1632	0.9500	0.6774	0.9500
	Outlet	0.1632	0.9500	0.6774	0.9500
	Spacer	0.0292	12.4820	0.1210	9.7602
Orifice		0.1429	7.4015	0.5932	7.3826
Valve		0.2918	0.9740	1.2125	0.9740

The TRACE model of the HELIOS loop is given in Figure 21

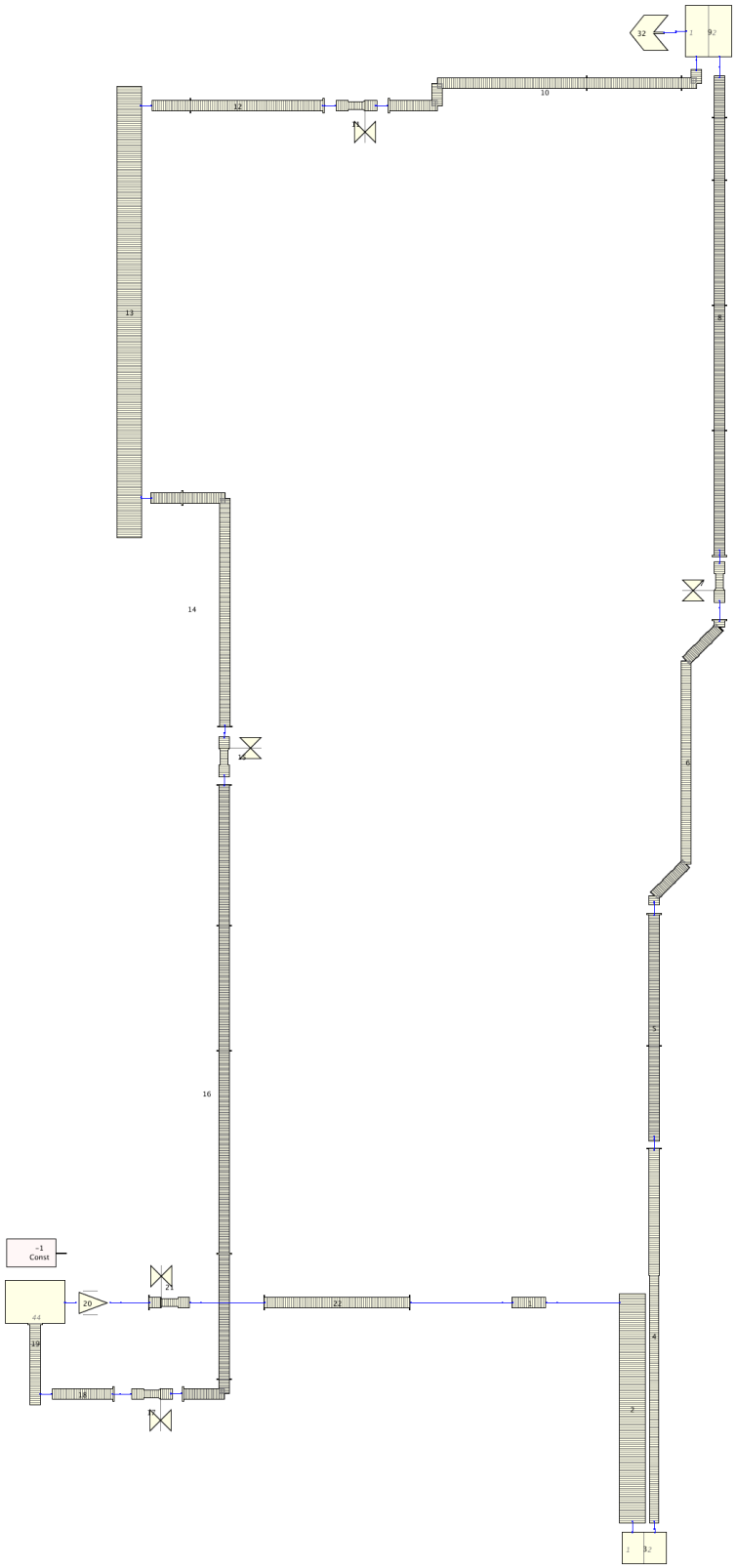

Figure 21. TRACE model of the HELIOS facility

This model uses 23 components, 16 PIPEs, 1 BREAK, 5 VALVEs and 1 PUMP with a total number of 2530 cells. This number is a result of the relative short cell lengths of 10-15 mm. In order to avoid strong variations of the cell length, they have been adjusted to be on the same order than the one for the gaskets (4.5 mm)

The following two diagrams (Figure 22 and Figure 23) show the comparison of the experimental data with the code calculations of the different benchmark participants. Along the loop, values for the pressure drop were calculated and summed over the accumulated length. The accumulation starts at the inlet to the mock up core.

For the low mass flow rate case one can see that the relative, as well as the absolute, error is large compared to the results. This is related to the measurement accuracy at low mass flow rates. This might be the reason why to experimental points have positive value. However, one can also see that the results of the INR/KIT fit very well to the remaining two measurement points. Also the curves of ENEA and of the SNU are close to the one of the INR/KIT.

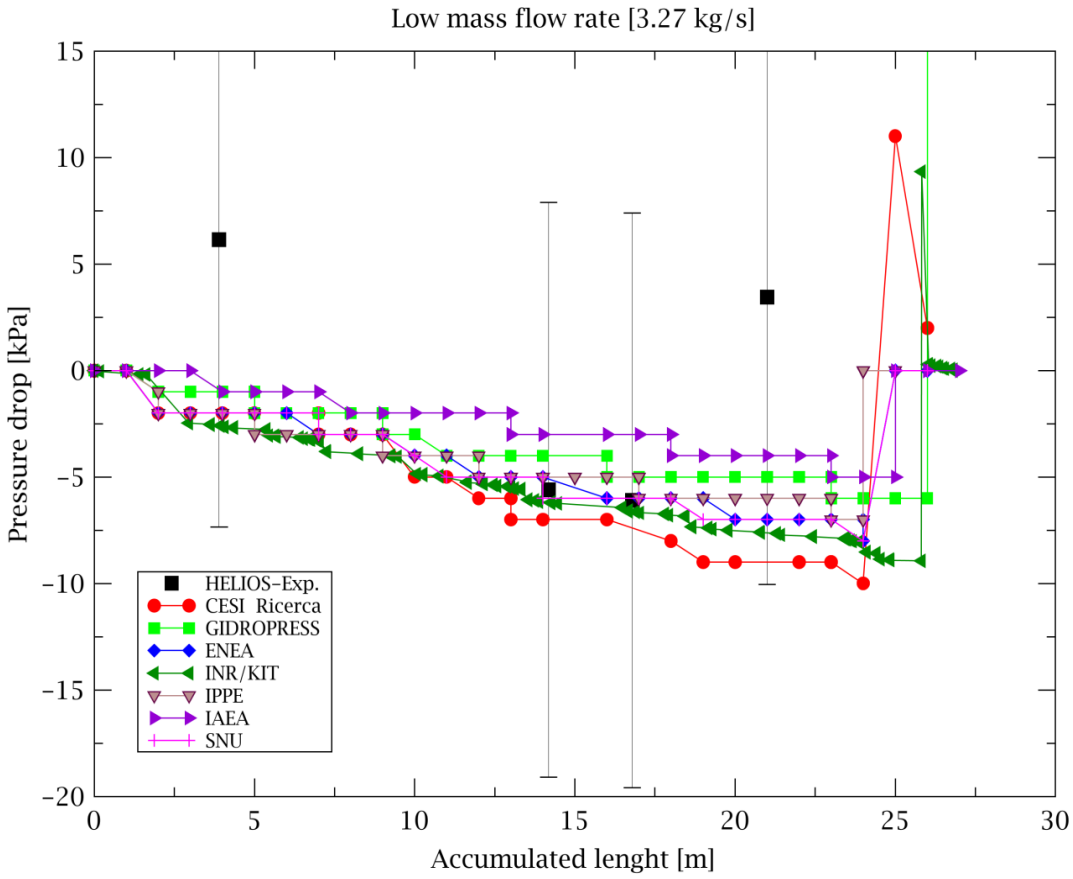

Figure 22. Results for low mass flow rate case

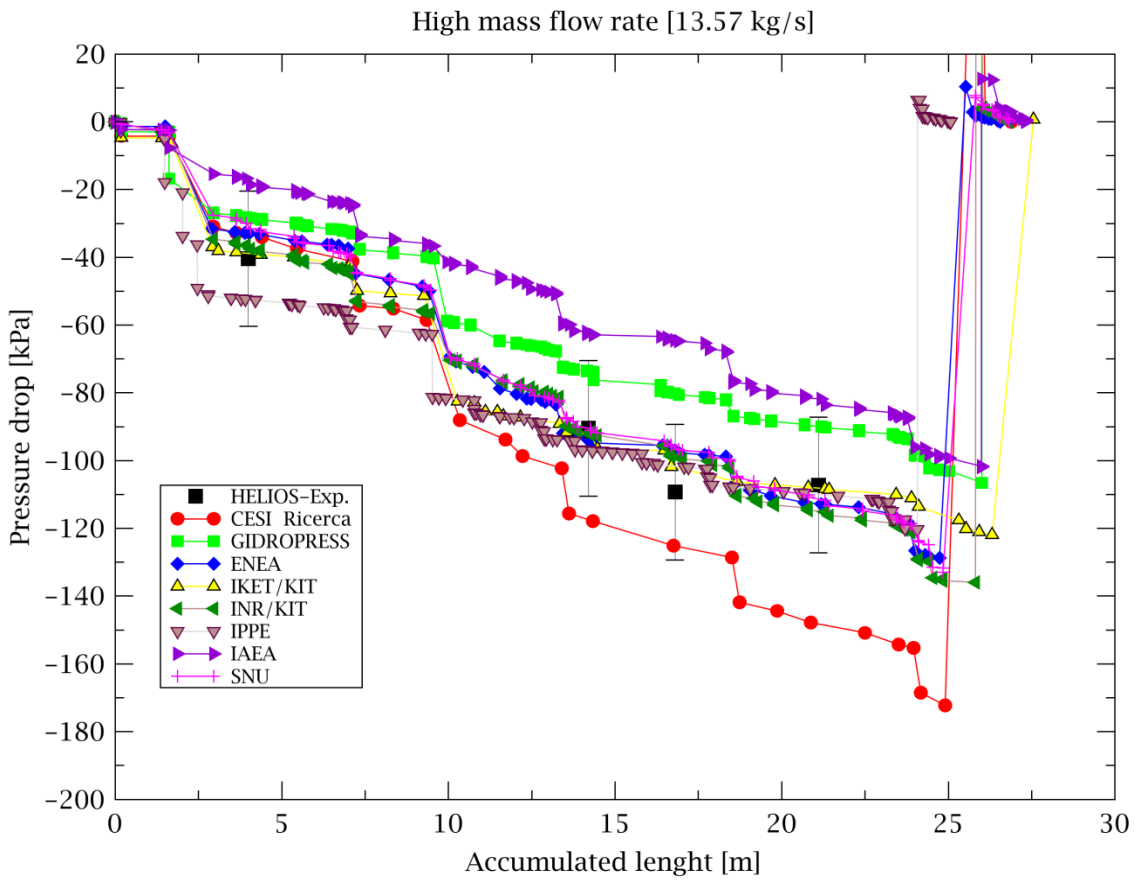

Figure 23. Results for the high mass flow rate case

At high mass flow rates the relative error becomes smaller due to the higher pressure drops. Again, the results of the INR/KIT are in very good agreement to the experimental results.

The reason for the discrepancies among the benchmark participants is probably related to, on the one hand side, the different physical models (e.g., friction factor, K factors), and, on the other hand side, the modeling itself. There are various correlations around for calculating the K factor in bends, orifices, spacer grids, etc. Hence, it is obvious that differences will arise.

This investigation showed that the physical models used in TRACE are in good agreement to theory and experiments. It can also be concluded that the additional correlations for the K factor, which the authors selected, are appropriate for the presented study.

TRACE is able to calculate the pressure drop with high accuracy in a lead-alloy environment. This is necessary since the next subsection deals with natural convection cases where the evaluation of the pressure drop is highly important.

4.2 XADS

The experimental accelerator driven system (XADS) is a proposed concept for transmutation of radioactive waste in form of minor actinides. Since the XADS [45] is the theoretical concept, no experimental data are available. Hence, in order to investigate the system and to validate TRACE, a code-to-code comparison was done.
For this analysis, only the target section will be investigated. The target section is located in the center of the sub critical core. A proton beam is guided to the target window and during a spallation reaction in the window neutrons will be released into the sub critical zone. Due to the high thermal loads of the window, a separate cooling circuit is required.

Figure 24 shows on the left hand side the schematic diagram of the target section. The center tube is evacuated and reserved for the proton beam. At the bottom of the tube (colored in red) the beam hits the window, neutrons will be released, and the produced heat taken by liquid lead-bismuth. It will transfer the heat to the heat exchanger, operated with Diphyl THT.

The diagram in the center of Figure 24 shows the break up into components. The blues dyed components indicating the primary, red dyed ones the secondary, and green dyed ones the tertiary (the lead-bismuth pool of the whole system).

The left hand side of Figure 24 shows then the TRACE model of the target section. To allow heat transfer between the three circuits, heat structures were added to the model. The heat produced in the target window is estimated to be 2.62 MW.

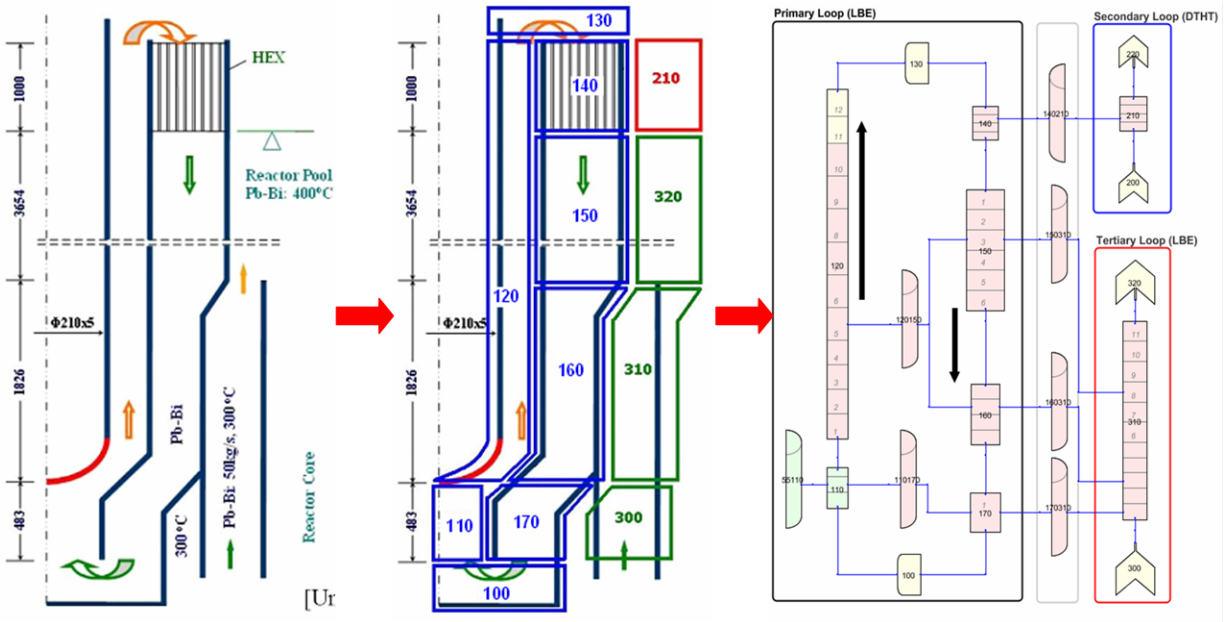

Figure 24. Drawing of the XADS target (left), component break up (center), and TRACE model (right)

Several different scenarios were selected, among them also transient scenarios like unprotected loss of heat sink or beam power interruption. The results of TRACE were compared to results of other codes like RELAP5, ATHLET, HETRAF and HERETA [30], [46].

The first investigation is the prediction of the steady state condition. The proton beam is off and the whole system is in thermal equilibrium since the lead-bismuth of the target cooling circuit has the same temperature like the whole reactor pool. Hence, the secondary side is off and no heat will be removed. Then, the secondary side will be switched on with a Diphyl THT temperature of 170°C and a mass flow rate of about 145 Kg/s, which will be kept constant during all transients if nothing else is mentioned. The response of the system is shown in Figure 25. One can see the mass flow rate and temperature development beneath the target window. At the beginning, the mass flow rate is zero, and the temperature has the values of the pool. After the secondary side becomes available, heat will be removed from the system. Due to the temperature gradient, the lead-bismuth starts to circulate with increasing mass flow rates. The temperature difference between primary and secondary side reduces and the mass flow rate of the primary side reduces, too. After a certain time, steady state has reached at a lead-bismuth mass flow rate of about 75 kg/s and a temperature is in the range of 200°C.

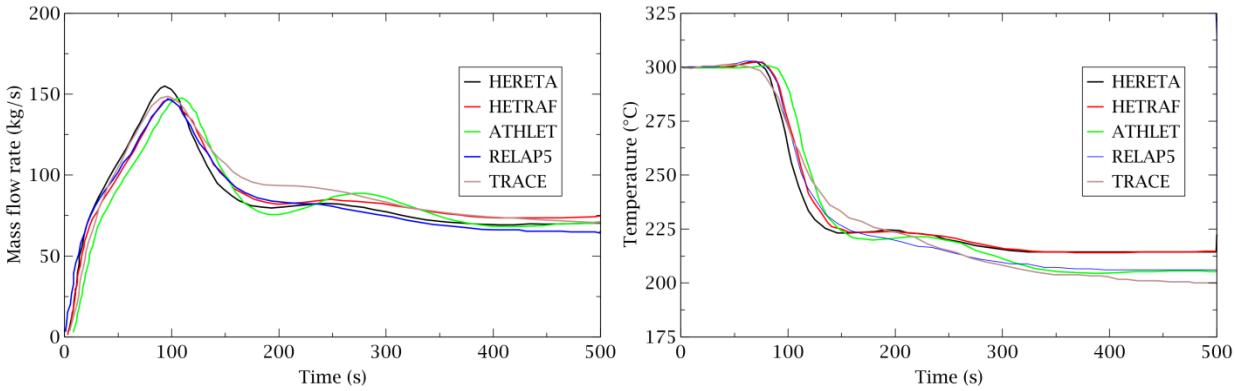

Figure 25. Mass flow rate (left) and temperature (right) versus time for steady state condition

The predictions by all codes are almost identical. They all have pronounced peak of the mass flow rate and the same behavior over time. Discrepancies can arise due to differences in the physical models (e.g., natural convection, thermo physical properties) and in the modeling of the target section. The results for mass flow rate and temperature below the window will serve as the reference condition for the following transient analysis.

Now, the proton beam will be switched and the reaction of the system will be evaluated. Exactly after running 100 s in steady state condition, 2.62 MW of heat will be released to the system. The feedback for the mass flow rate and the temperature are given in Figure 26.

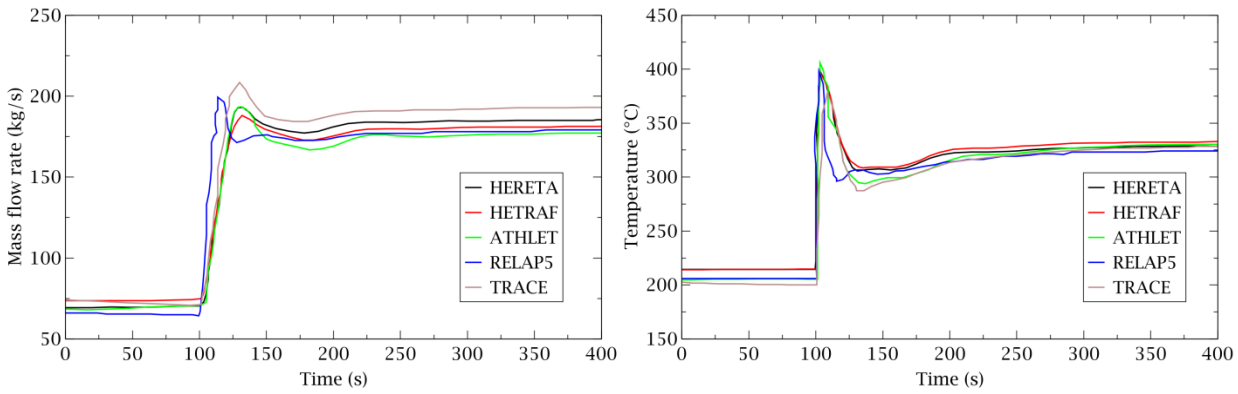

Figure 26. Mass flow rate (left) and temperature (right) versus time for beam power switch-on transient

Immediately after the beam is switched on, the temperature jumps from about 200°C to values of around 400°C, whereas the temperature of the metallic target window will be higher. The mass flow rate follows that trend from approximately 75 kg/s to 200 kg/s. After roughly 200 s, the system reaches a new steady state condition at 180 kg/s and 325°C.

Since the sudden release of the heat courses the jump in the temperature, which might be of disadvantages for the window, a slow increase of the power was proposed. Thus, the temperature peak should be avoided. Such scenario was adopted for the next analysis. Over a period of 200 s, the power of the beam was increased linear until 100% were reached. Hence, the thermal load of the window could be released.

Figure 27 shows now the mass flow rate and the temperature right below the window versus the time for power ramp scenario. At the end, the mass flow rate and the temperature are identical to the ones with the power jump. However, in between one can clearly see that the temperature, as well as the mass flow rate, peak was avoided and the trend is similar to the one of the power profile.

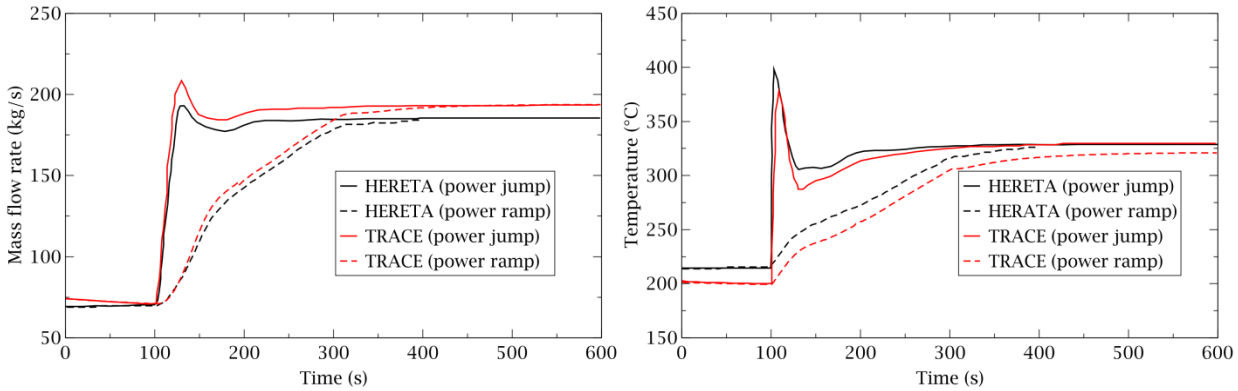

Figure 27. Mass flow rate (left) and temperature (right) versus time for beam power switch-on transient (power ramp)

The next transient is the unprotected loss of heat sink. In this case, the secondary side will be switched off while the beam stays online. Thus, the temperature in the primary loop will rise either until the beam will be switched off or until the system will fail. In the present case, the temperature of the beam window should not exceed 525°C.

As one can see on the right hand side in Figure 28, the temperature will rise almost linearly. The mass flow rate on the other side will drop. Since no heat will be removed, the temperature difference between bottom (hot) and top (cold) will decrease. After a while, the mass flow rate will be almost stable. One can also see that after roughly 200 seconds after the transient was initiated, the lead-bismuth temperature will reach 500°C and hence the temperature of the window will close to the limit value. Thus, within the first 200 s after the secondary side becomes unavailable the proton beam needs to be shut down.

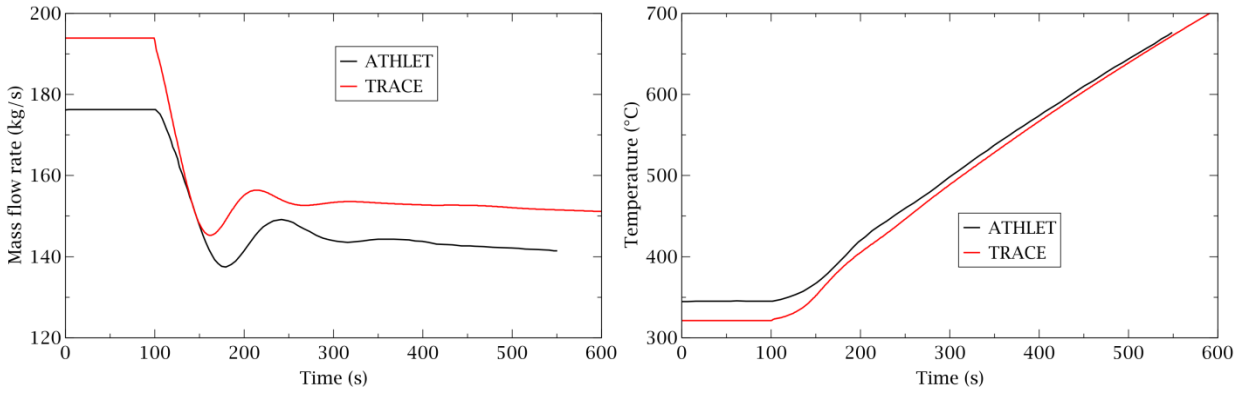

Figure 28. Mass flow rate (left) and temperature (right) versus time for unprotected loss of heat sink

Beam power interruptions are one think to consider when working with accelerators. Figure 29 shows for different interruption times the response of the system.

The upper two figures show the mass flow rate (left side) and the temperature (right) for an interruption time of three seconds. One can see that the drop of the mass flow rate and temperature is rather small. For higher interruption times, the drops are more pronounced. For interruption times greater than 30 s, the system response is similar to the beam power switch on transient. The relative short time period of 30 s are enough that the system reaches steady state after the proton beam has switched off. Hence, in case the proton beam will not get online within 30 s, the switch on should fallow the linear power ramp, mentioned earlier.

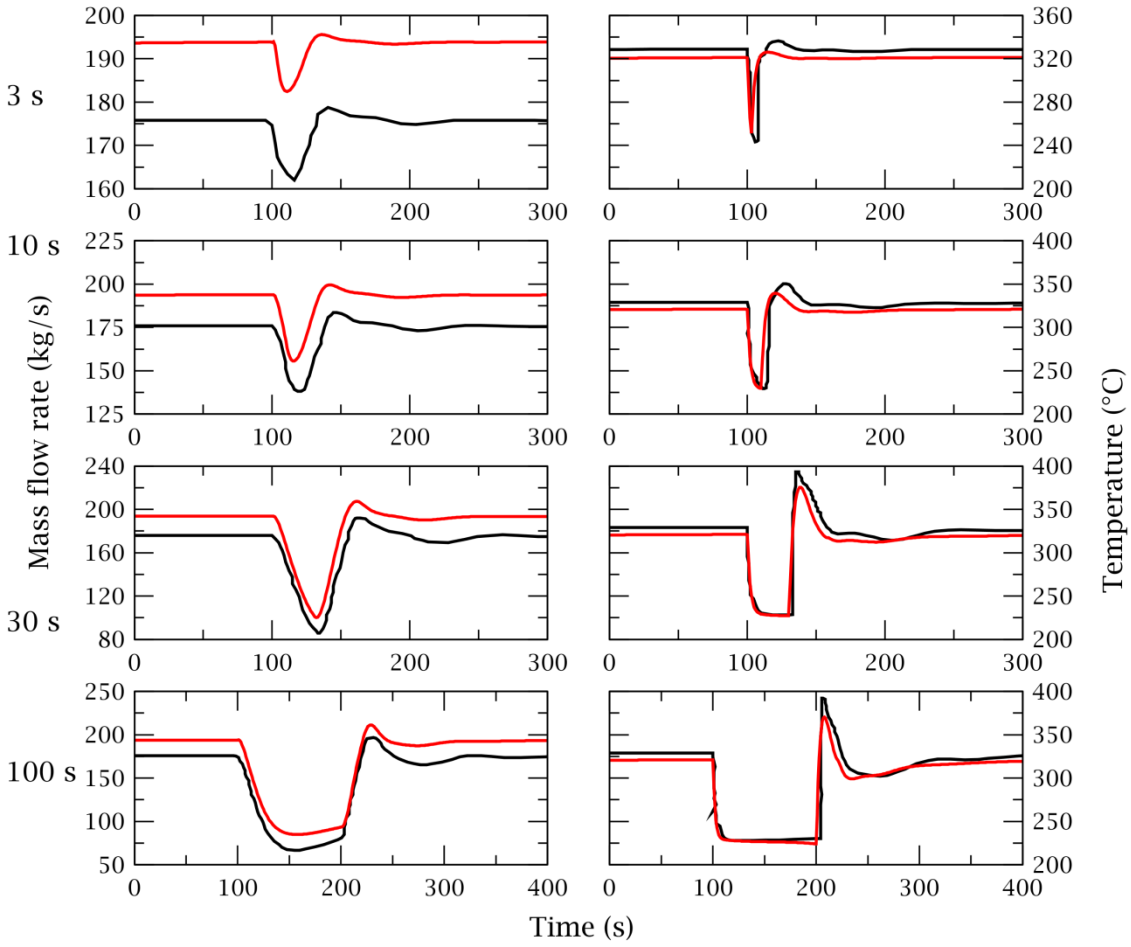

Figure 29. Mass flow rate (left) and temperature (right) versus time for beam power interruption transients. TRACE curves are red, and ATHLET curves are black dyed

The last investigation of this subsection is dedicated to the variation of the geometry of the target loop. The diameter of the funnel below the target window (refer to component 110 in Figure 24) was increased but also decreased. The consequences for the mass flow rate and temperature are shown in the figure below.

The original diameter of the funnel is 140 mm. In the first case, the diameter was reduced to 120 mm. This caused the mass flow rate to decrease and, as a consequence, to increase the temperature. The opposite behavior took place for the increase of the funnel diameter. Due to the change of the flow area, the hydraulic resistance will change, too.

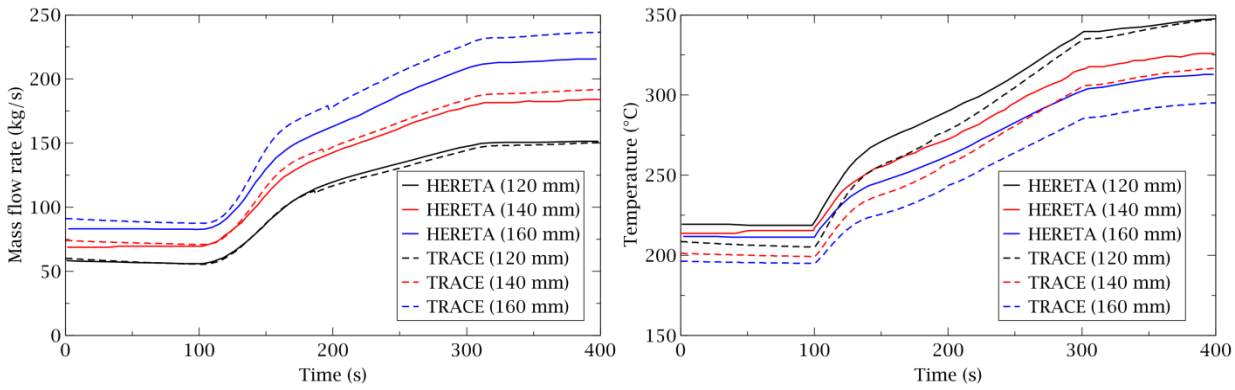

Figure 30. Mass flow rate (left) and temperature (right) versus time for different funnel diameters

4.3 CHEOPE/MEGAPIE

In this subsection, the post-test analysis of the experiments performed at the CHEOPE facility will be presented. At the CHEOPE facility, a mockup of a heat exchanger related to the MEGAPIE (Megawatt Pilot Experiment) project [47] was built to investigate the heat transfer from liquid lead-bismuth to Diphyl THT. The special feature of this heat exchanger is the geometrical layout. This heat exchanger is a triple annulus. In the center annulus, Diphyl THT will flow down, will be redirected to flow upwards trough the center annulus, and exits the heat exchanger. The liquid lead-bismuth flows down the outer annulus to transfer the heat to the Diphyl THT via counter current heat exchange.

Figure 31 shows the heat exchanger at the CHEOPE facility. On the right hand side, one can see the spirals in the center annulus, which force the Diphyl THT on a helical upward path. Some important key parameters of the heat exchanger are tabulated in Table 16. Experimental results are given in Table 17.

Table 16. Key parameter of the CHEOPE heat exchanger

Parameter (Unit)	Quantity
Temperature, inlet LBE/DTHT (K)	623.15/413.15
Volume flow, LBE/DTHT (m³)	0.00033/0.00083
Heat exchanger height (m)	1.2
Spiral diameter (m)	0.00150
Inner annulus outer diameter (m)	0.047
DTHT gap width (m)	0.0021
Wall thickness (m)	0.0015
LBE gap width (m)	0.00425
Spiral pitch (°)	30

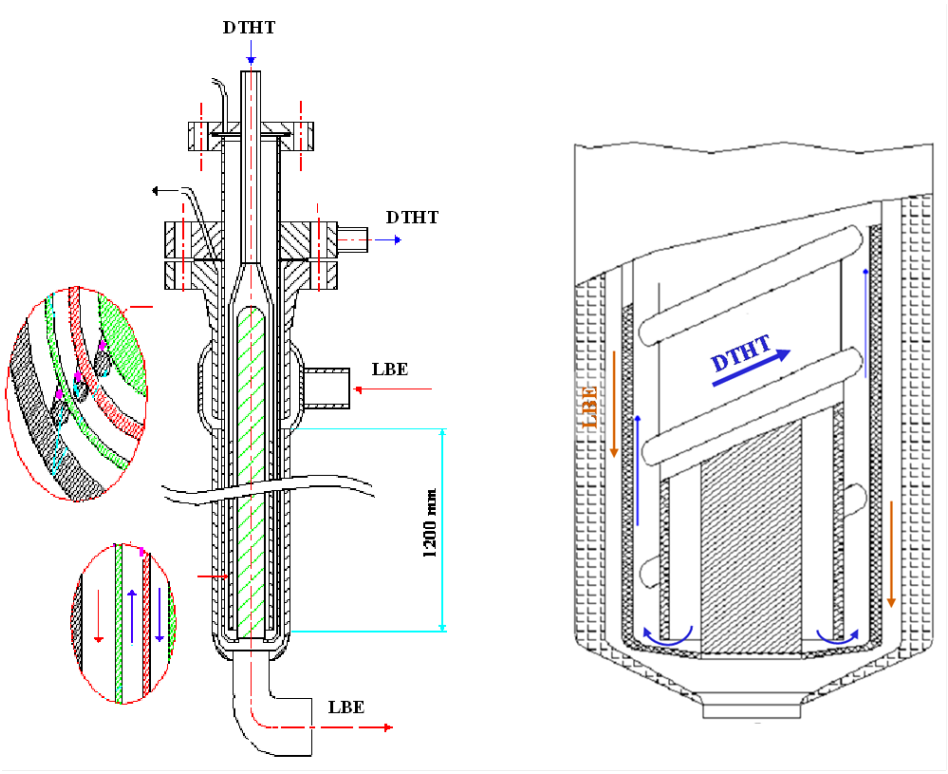

Figure 31. Drawing of the heat exchanger (left) and definition of the flow paths (right)

Table 17. Input and output data for selected experiments

Parameter (unit)	Experiment 1	Experiment 2	Experiment 3
Power (W)	27430	21590	10410
Volume flow LBE (m³/s)	$1.47 \cdot 10^{-4}$	$1.55 \cdot 10^{-4}$	$1.59 \cdot 10^{-4}$
Volume flow DTHT (m³/s)	$5.63 \cdot 10^{-4}$	$5.56 \cdot 10^{-4}$	$2.24 \cdot 10^{-4}$
Temperature, inlet LBE (K)	579.05	537.25	499.95
Temperature, inlet DTHT (K)	410.15	409.35	425.-15
Temperature, outlet LBE (K)	455.95	445.85	459.95
Temperature, outlet DTHT (K)	436.65	430.79	446.45

The task for the following investigations was to use the power, the volume flow rates and the inlet temperatures as boundary conditions, and to calculate the outlet temperature and the amount of heat that will be transferred. The results can be found in Table 18.

The results for the three correlations are labeled as TRACE-Nu1 (Gnielinski [36]), TRACE-Nu2 (Leung, Milenkovich and Class [38]), and TRACE-Nu3 (Rogers and Mayhew [39]). The results for the original TRACE version are labeled as TRACE V5.0.

One can clearly see that the original TRACE version was not able to calculate the values of the experiment. The standard correlation under predicts the heat transfer. Hence, with the given boundary conditions of inlet temperature and mass flow rate, TRACE cannot transfer the total amount of heat. For the first two experiments, the relative error related to the transferred power is between -15 and -20 %, whereas for experiment three, the error amounts to -50 %.

Table 18. Comparison of experimental and calculated results for the three experiments

	Properties (Unit)	Experiment	TRACE V5.0	TRACE-Nu1	TRACE-Nu2	TRACE-Nu3
Experiment 1	T_{in} LBE/DTHT (K)	579.05/410.15	579.05/410.15	579.05/410.15	579.05/410.15	579.05/410.15
	T_{out} LBE/DTHT (K)	455.95/436.65	476.33/433.00	456.11/437.15	457.43/436.88	459.94/436.37
	ΔT LBE/DTHT (K)	123.10/26.50	102.72/22.85	122.94/27.00	121.62/26.73	119.11/26.22
	Error ΔT (%)	-/-	-16.56/-13.77	-0.13/1.89	-1.20/0.87	-0.96/-3.58
	P LBE/DTHT (W)	27430/27430	23154/22804	27995/27298	27709/27005	27168/26447
	Error Power (%)	-/-	-15.59/-16.86	2.06/-0.48	1.02/-1.55	-0.96/-3.58
Experiment 2	T_{in} LBE/DTHT (K)	537.25/409.35	537.25/409.35	537.25/409.35	537.25/409.35	537.25/409.35
	T_{out} LBE/DTHT (K)	445.85/430.25	464.02/426.90	447.85/430.67	448.97/430.22	451.01/429.78
	ΔT LBE/DTHT (K)	91.40/20.90	73.23/17.55	89.40/21.32	88.28/20.87	86.24/20.43
	Error ΔT (%)	-/-	-19.88/-16.03	-2.19/2.01	-3.14/-0.14	-5.65/-3.25
	P LBE/DTHT (W)	21590/21590	17814/17251	21713/21063	21246/20799	20790/20318
	Error Power (%)	-/-	-17.49/20.10	0.57/-2.44	-1.59/-3.67	-3.71/-5.89
Experiment 3	T_{in} LBE/DTHT (K)	499.95/425.10	499.95/425.10	499.95/425.10	499.95/425.10	499.95/425.10
	T_{out} LBE/DTHT (K)	459.95/446.45	478.18/438.59	461.61/447.55	464.86/445.84	465.59/445.44
	ΔT LBE/DTHT (K)	40.00/21.30	21.77/13.49	38.34/22.45	35.09/20.74	34.36/20.34
	Error ΔT (%)	-/-	-45.58/-36.81	-4.15/5.15	-12.28/-2.86	-14.10/-4.73
	P LBE/DTHT (W)	10410/10410	5572/5285	9344/9309	8620/8520	8451/8343
	Error Power (%)	-/-	-46.48/-49.23	-10.24/-10.57	-17.20/-18.16	-18.82/-19.86

With all three presented Nusselt correlation for heat transfer in helical paths, the results could be improved considerably. For experiment 1, the error is below 4%, for experiment 2, the error is below 6%, and for experiment 3, the error could be reduced below 20%.
A graphical representation of the results is given in Figure 32 and Figure 33. Again, the clear improvement can be seen.

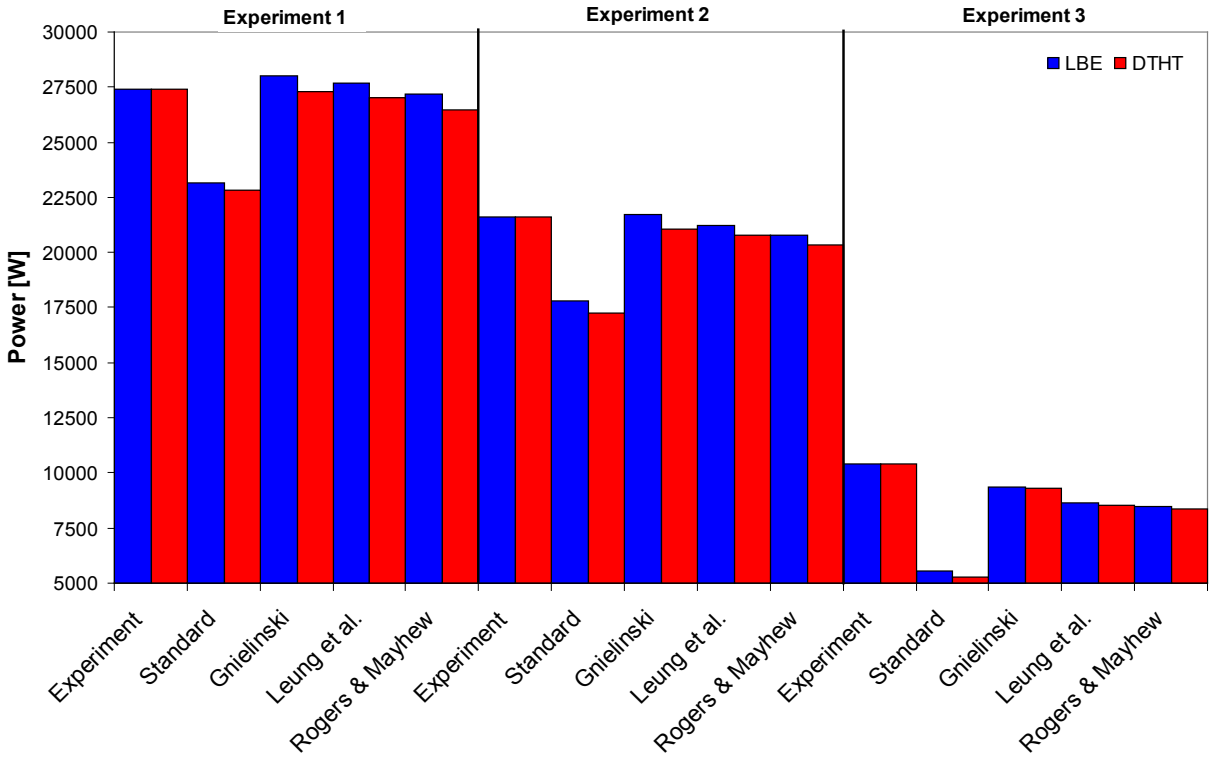

Figure 32. Comparison of the transferred/received power for the three experiments with different Nusselt correlations

This investigation showed that the TRACE model reaching their limits if applied to non-standard geometries like the presented one. Up to now, such a geometry is rather seldom used but new innovative systems follow new approaches and hence more complicated geometries will be used to guarantee a better performance.

Three different Nusselt correlations for heat transfer in helical flow paths were selected and implemented into TRACE. The one that improved the results most is the one of Gnielinski, but the two other correlations are also quite useful. The authors recommend the usage of the Gnielinski correlation for this but also for other applications with helical flow paths. The correlation of Leung, Milenkovich and Class is only valid for the presented heat exchanger geometry. Hence, other geometrical layouts ask for a revised correlation. Compared to the results obtained with the correlation of Rogers and Mayhew, the results of the Gnielinski correlation are closer to the experiment

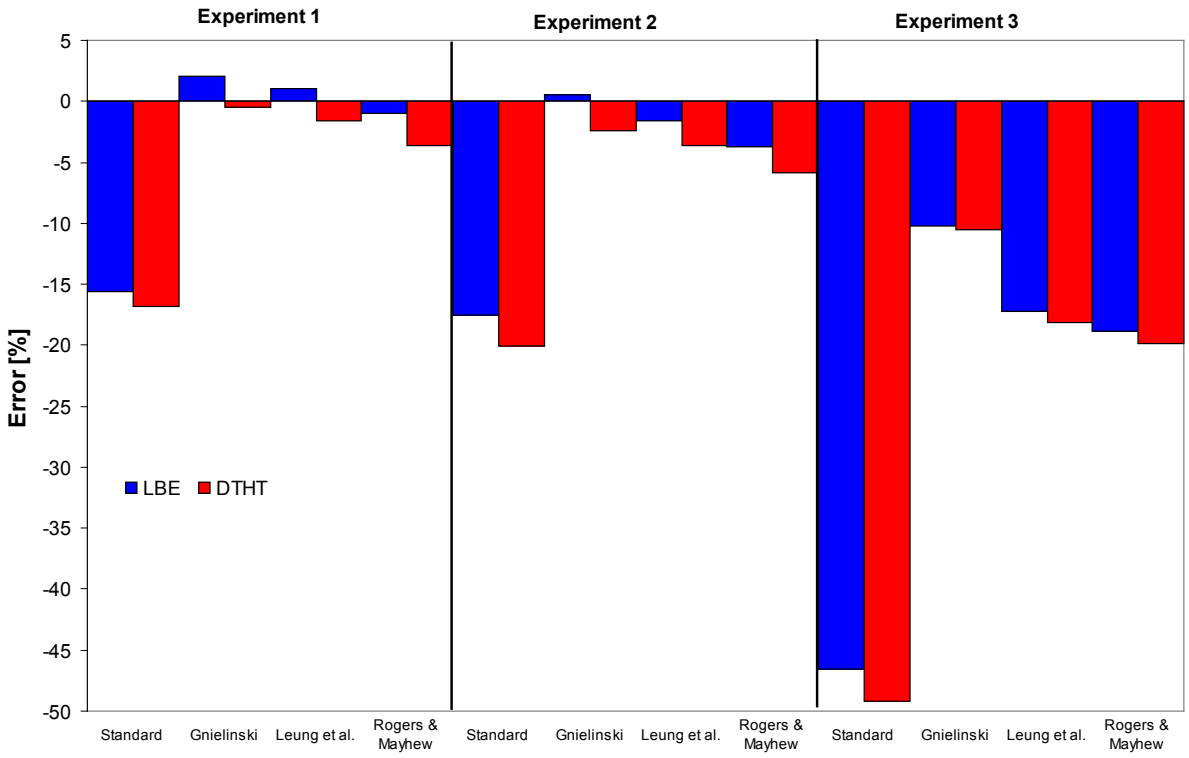

Figure 33. Comparison of the error related to the transferred/received power for the three experiments with different Nusselt correlations

4.4 LFR Sub-assembly

The investigation is related to the validation of the thermal properties of lead, and to the validation of the heat transfer in bundles. Since no experimental data were found in open literature, another code, which has lead as coolant option and is well validated, was selected for reasons of code-to-code comparisons. This code is the sub channel program MATRA [48]. This code also employs correlations for heat transfer in bundle arrays.

A representative sub-assembly of a LFR was modeled with MATRA. The results of MATRA for the hottest channel serve as boundary conditions for TRACE (e.g., inlet temperature, mass flow rate). The results for the MATRA channel and the TRACE channel were compared. A sub-assembly of a LFR is given in Figure 34. Due to symmetry, only 1/8 of the SA was modeled with MATRA.

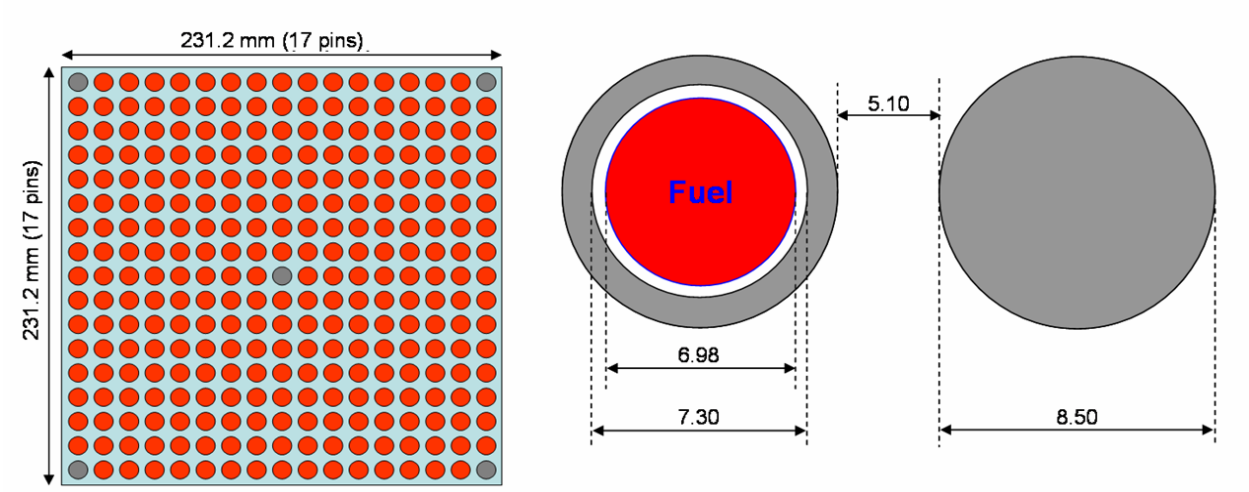

Figure 34. LFR sub-assembly

This SA has a 17x17 matrix with five solid rods at the corners and in the center. The MATRA and TRACE model can be found in the following figure. In total, 43 fuel rods, two steel rods and 45 cooling channels were modeled with MATRA. Channel no. 21 and rod no. 21 turned out to be the hottest ones. The characteristics of the channel and the rod were fed into TRACE.

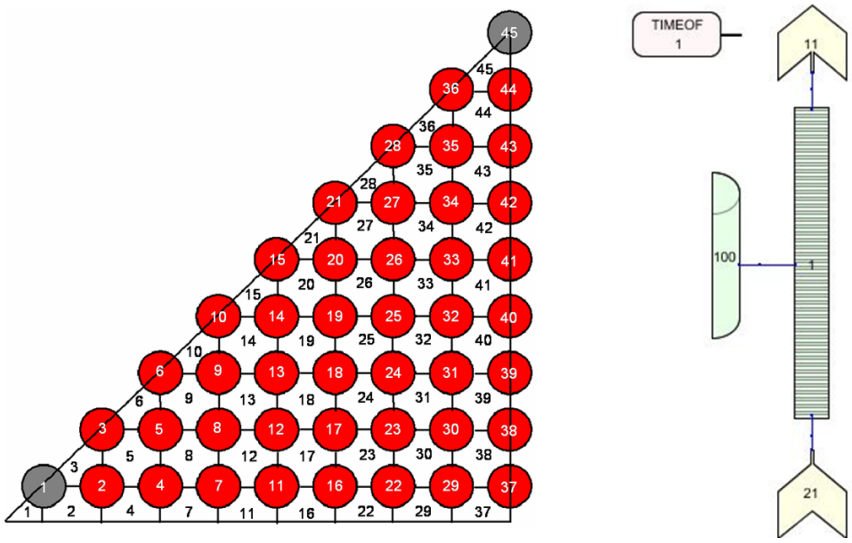

Figure 35. MATRA and TRACE model of the LFR SA

In a first step, only the thermo physical properties were implemented into TRACE. The heat transfer models remained unchanged. In the second step, also the heat transfer routines were improved. The results of the comparison of the cladding temperatures can be found in Figure 36.

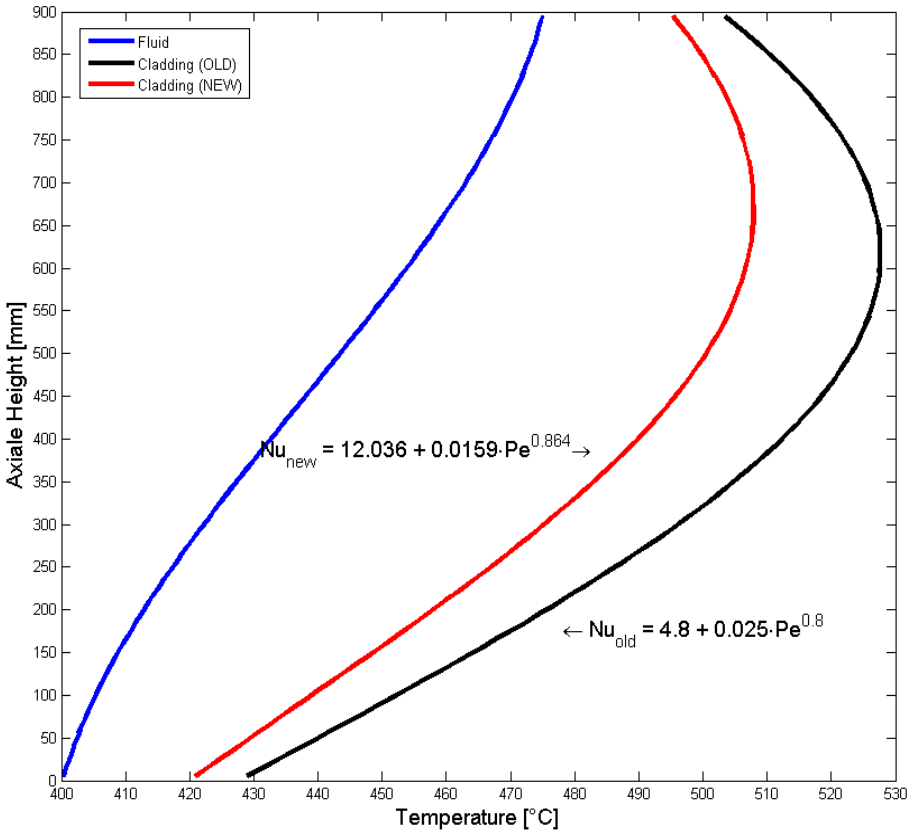

Figure 36. Cladding temperature calculated with the standard and the improved heat transfer model

One can see there the fluid temperature, which is of course identical for both cases, and the cladding temperatures. The black curve show results obtained with the standard correlation and the red one the one obtained with the new correlation, given in Eq. 30. One only needs to enter the p/d ratio in Eq. 30. The calculated cladding temperature could be reduced by up to 20 K.

With the implementation of a new heat transfer model unnecessary conservatism could be reduced.

The third step was the comparison of results of MATRA and of the improved TRACE version. One issue when working with liquid lead is the problem with corrosion and oxidation of the structural and cladding materials. Wherefore, different oxide layer thicknesses were considered at the surface of the cladding. The oxide layer prevents the corrosion of the cladding material but since it has a low thermal conductivity, new safety related challenges arise. The next two figures show the comparison of MATRA and TRACE results for different oxide layer thicknesses with different thermal conductivities. Based on a chosen temperature limit of 550°C, an oxide layer of 30 µm with k_{oxide} = 1.0 W/m·K and 15 µm with k_{oxide} = 0.5 W/m·K are enough to cause an violation.

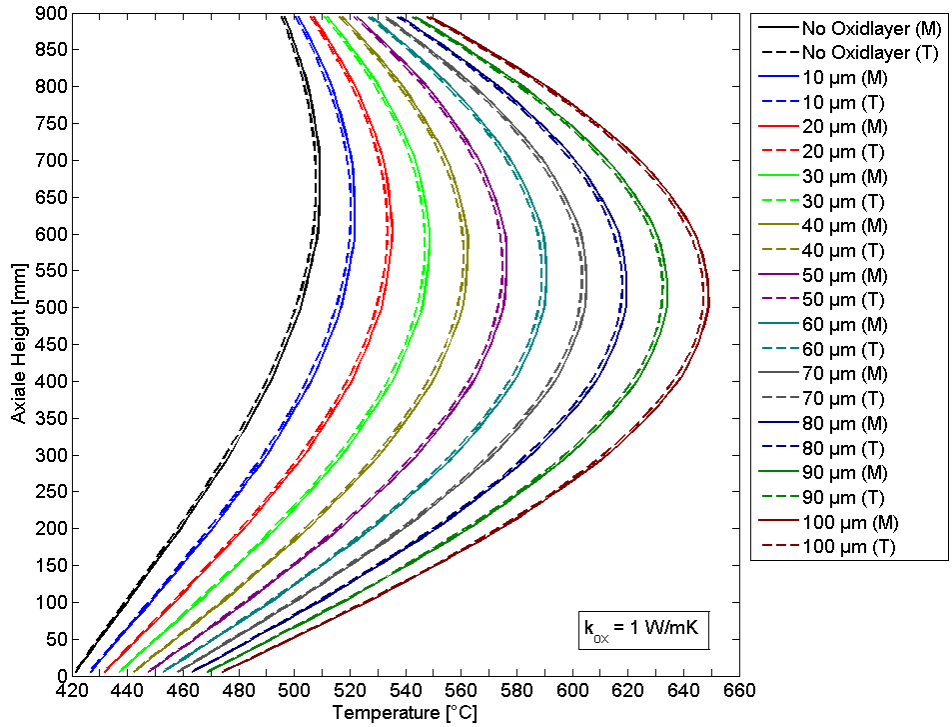

Figure 37. Cladding temperatures with different oxide layer thicknesses with high thermal conductivity

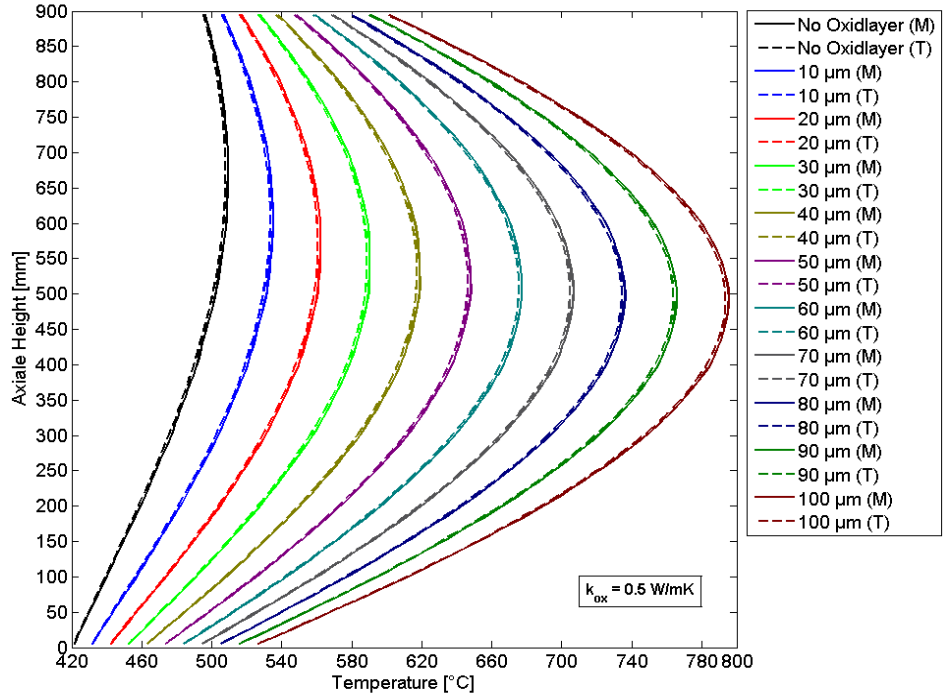

Figure 38. Cladding temperatures with different oxide layer thicknesses with low thermal conductivity

5 CONCLUSION

The present report contains the results gained during the validation effort related to lead and lead-bismuth cooled systems. Two new fluids, pure lead and Diphyl THT, have been implemented and one fluid, PbBi or LBE, was improved. In addition, the heat transfer models related to liquid metals have been extendent in order to distinguish between tube and bundle flow. Four different experiments/benchmarks/theoretical problems have been investigated for the purpose of validation.

Section 4 of this report collects the validation efforts concerning lead and lead-alloy cooled systems. The first example dealt with the characterization of an isothermal LBE loop regarding its pressure losses due to friction and form losses within the LACANES benchmark. The thermo physical properties of LBE were reviewed and updated. The TRACE results were compared to experimental data and to results of other benchmark participants. The second example, the XADS benchmark, was used to investigate TRACE ability to simulate conditions were natural circulation occurs. After the implementation of Diphyl THT as a new coolant, several operational (steady-state, beam power switch-on) and transient scenarios (beam power interruption, loss of heat sink) were performed with the modified TRACE version. Due to missing experimental data, the TRACE results were compared to those of other codes. Innovative reactor concepts employ component designs which strongly differ from LWR like designs (e.g., heat exchangers). Therefore, the operational behavior regarding the heat transfer was analyzed in the third section. The standard heat transfer correlation for forced convection appeared to be inappropriate for the analysis of this HX. Hence, several correlations developed for systems with a similar design like the present were implemented in TRACE. The results of TRACE were compared to the experimental data. The last example of chapter 3 is focused on the analysis of an LFR sub-assembly. Therefore, lead properties were implemented and the heat transfer routines were enhanced to handle bundle flow. The selected SA design was first investigated with the sub-channel code MATRA and later with the modified TRACE version. The results for cladding and fuel temperature are similar for both codes. This investigation was also used to identify the influence of an oxide layer on the cladding temperature. It was shown that for the selected parameter combinations, an oxide layer of 15 μm (k_{oxide} = 0.5 W/m·K) and 30 μm (k_{oxide} = 1 W/m·K), respectively causing a violation of the selected 550°C cladding temperature limit.

6 REFERENCES

[1] GIF-IV, "Generation IV International Formum - Annual Report," 2007.

[2] W. Jaeger, "Validation and application of the system code TRACE for safety related investigations of innovative nuclear energy systems," PhD thesis, Technical University of Dresden, 2012.

[3] W. Jaeger, V. Sánchez Espinoza and B. Feng, "Analyses of a XADS target with the system code TRACE," in *International Youth Nuclear Congress*, Interlaken, Switzerland, 2008.

[4] W. Jaeger and V. Sánchez Espinoza, "Influence of oxide layers on the cladding in a liquid lead environment: A comparison between MATRA and TRACE," in *International Conference on Reactor Physics*, Interlaken, Switzerland, 2008.

[5] W. Jaeger and V. Sánchez Espinoza, "Analyses of a LBE-Diphyl THT heat exchanger with the system code TRACE," in *International Conference on Nuclear Engineering*, Orlando, USA, 2008.

[6] E. Adamov and V. Orlov, Naturallysafe lead-cooled fast reactor for large-scale nuclear power, E. Adamov and V. Orlov, Eds., Moscow: Dollezhal RDIPE, 2001.

[7] D. Blaskett and D. Boxall, Lead and its alloys, New York: Ellis Horwood, 1990.

[8] R. Hultgren, P. Desai, D. Hawkins, M. Gleiser and K. Kelley, Selected values of the thermodynamic properties of binary alloys, Metals Park: American Society of Metals, 1973.

[9] IAEA, Comparative assessment of thermophysical and thermohydraulic characteristics of lead, lead-bismuth and sodium coolants for fast reactors, Vienna: International Atomic Energy Agency, IAEA-TECDOC-1289, 2002.

[10] P. Kirillov, V. Subbotin and P. Ushakov, Liquid metals, Washington, D.C.: NASA Technical Translation, NASA TT F-522, 1969.

[11] S. Kutateladze, V. Borishanskii, I. Novikov and O. Fedynskii, Liquid-metal heat transfer, New York: Consultants Bureau, Inc., 1959.

[12] R. Lyon, "Heat transfer at high heat fluxes in confined spaces," PhD thesis, University of Michigan, Ann Arbor, 1949.

[13] S. McLain and J. Martens, Reactor handbook - Volume 4: Engineering, S. McLain and J. Martens, Eds., New York: Interscience Publisher, Inc., 1964.

[14] K. Morita, "Thermophysical properties of lead-bismuth eutectic alloys for use in reactor safety analysis," Technical Report, NEA Nuclear Science Committee, Paris, 2004.

[15] R. Novakovic, E. Ricci, D. Giuranno and F. Gnecco, "Surface properties of Bi-Pb liquid alloys," *Surface Science,* vol. 515, pp. 377-389, 2002.

[16] OECD/NEA, Handbook on lead-bismuth eutectic alloy and lead properties, materials compatibility,thermal-hydraulics and technologies, Paris: OECD/NEA, 2007.

[17] S. Ohno, S. Miyahara and Y. Kurata, "Experimental investigation of lead-bismuth evaporation behavior," *Journal of Nuclear Science and Technology,* vol. 42, no. 7, pp. 593-599, 2005.

[18] M. Petrazzini, "Lead bismuth eutectic physical properties and thermodynamic tables," Technical Report ANSALDO Nucleare, Genoa, 2002.

[19] C. Tipton, Reactor handbook - Volume 1: materials, 2. ed., C. Tipton, Ed., New York: Interscience Publisher, Inc., 1960.

[20] E. Brandes and G. Brook, Smithells metals reference book, 7 ed., E. Brandes and G. Brook, Eds., New York: Butterworth Heineman Ltd., 1992.

[21] L. Gurvich, Thermodynamic Properties of Individual Substances - Volume 2, New York: Hemsiphere Publishing Corp., 1991.

[22] W. Hofman, Lead and lead alloys, Berlin: Springer-Verlag, 1970.

[23] J. Holman, Heat transfer, New York: McGraw-Hill, Inc., 1981.

[24] T. Iida and R. Guthrie, The physical properties of liquid metals, Oxford: Clarendon Press, 1988.

[25] U. Jauch, "Thermophysical properties in the system Li-Pb," Kernforschungszentrum Karlsruhe, KfK-4144, Karlsruhe, 1986.

[26] O. Knacke, O. Kubaschewski and K. Hesselmann, Thermochemical properties of inorganic substances, 2. ed., Berlin: Springer-Verlag, 1991.

[27] D. Stull and G. Sinke, Thermodynamic properties of the elements, D. Stull and G. Sinke, Eds., Washington, D.C.: American Chemical Society, 1956.

[28] K. Thurnay, "Thermal properties of the lead," Kernforschungszentrum Karlsruhe, Primary Report, Karlsruhe, 1993.

[29] VDI, VDI Warmeatls, 9. ed., Berlin: Springer Verlag, 2002.

[30] H. Chen and X. Cheng, "The ATHLET-MF code and its application to heavy liquid metal cooled systems," Forschungszentrum Karlsruhe, FZKA 7165, Karlsruhe, 2005.

[31] USNRC, "TRACE V5.0 Theory manual," US NRC, Washington, D.C., 2010.

[32] W. Pfrang and D. Struwe, "Assessment of correlations for heat transfer to the coolant for heavy liquid metal cooled core designs," Forschungszentrum Karlsruhe, FZKA 7352, Karlsruhe, 2007.

[33] K. Mikityuk, "Heat transfer to liquid metal: review of data and correlations for tube and bundles," *Nuclear Engineering and Design,* vol. 239, pp. 680-687, 2009.

[34] R. Seban and T. Shimazaki, "Heat transfer to a fluid flowing turbulently in a smooth pipe with walls at constant temperature," California Univ., Berkeley, Inst. of Engineering Research, AD-495380, 1949.

[35] P. Ushakov, A. Zhukov and N. Matyukhin, "Heat transfer to liquid metals in regular arrays of fuel elements," *High Temperature,* vol. 15, pp. 868-873, 1977.

[36] V. Gnielinski, Helically coiled tubes of circular cross section, Hemisphere Publishing Corp., 1987.

[37] H. Ito, "Friction factors for turbulent flow in curved pipes," *Journal of basic engineering, Transactions of the ASME,* vol. D81, pp. 123-134, 1959.

[38] W. Leung, R. Milenkovic and A. Class, "A computational study of the heat transfer characteristics of a spiral cooling pin," in *15th International Conference on Nuclear Engineering*, Nagoya, Japan, 2007.

[39] G. Rogers and Y. Mayhew, "Heat transfer and pressure loss in helically coiled tubes with turbulent flow," *International Journal of Heat and Mass transfer,* vol. 7, pp. 1207-1216, 1964.

[40] OECD/NEA, "Benchmarking of thermal-hydraulic loop models for lead-alloy cooled advanced nuclear energy systems (LACANES) - Task guideline for phase I: Characerization of HELIOS," OECD/NEA, NEA/NSC/DOC(2007)14Rev1, Paris, 2007.

[41] S. Jeong, C. Bahn, S. Chang, Y. oh, W. Nam, K. Ryu, H. Nam, J. Lim, N. Lee and I. Hwang, "Operation experience of LBE loop: HELIOS," in *International Conference on Advances in Nuclear Power Plants*, Reno, USA, 2006.

[42] S. Churchill, "Friction-factor equation spans all fluid-flow regimes," *Chemical Engineering,* vol. 11, pp. 91-92, 1977.

[43] I. Idelchik, Handbook on hydraulic resistance, 2. ed., Washington, D.C.: Hemisphere publishing Corp., 1986.

[44] K. Rehme, "Pressure drop correlations for fuel element spacers," *Nuclear Technology,* vol. 17, pp. 15-23, 1973.

[45] P. Richard, "Technical specifications, missions of XADS, recommendations for the main characteristics," CEA, DER/SERI/LCDI-02/4002, 2002.

[46] N. Tak, H. Neitzel, H. Chen and X. Cheng, "Thermal hydraulic analyses of window target unit for LBE cooled XADS," Forschungszentrum Karlsruhe, FZKA 7060, Karlsruhe, 2004.

[47] G. Bauer, M. Salvatores and G. Heusner, "MEGAPIE, a 1 MW pilot experiment for a liquid metall spallation target," *Journal of Nuclear Materials,* vol. 296, pp. 17-33, 2001.

[48] KAERI, "Development of the subchannel analysis code MATRA," Korean Atomic Energy Research Institute, KAERI/TR-1033/98, 1998.

APPENDIX A. TRACE SOURCE CODE CHANGES

With the help of this appendix the reader is able to reproduce the TRACE results presented in this thesis (only if one is in possession of the original source code). Table [X] at the end of this section gives an overview of all TRACE moduls, sub-routines or functions which have been updated or implemented. The names in the parenthesis indicate the module there the routine or function is located.

The lines of code, which have been changed or implemented, are shown afterwards. The first part of the code lines are for the thermo physical properties of DTHT, Pb and PbBi (LBE). The second part is related to the changes regarding the heat transfer models.

A.1 THERMOPHYSICAL PROPERTIES

A.1.1 COMMON ROUTINES

Modul EosDataM

```
INTEGER(sik), PARAMETER :: maxeos = 10 ! Maximum number of different fluids allowed
                                       ! in a TRACE model.
!
      INTEGER(sik), PARAMETER :: eosH2o = 1
      INTEGER(sik), PARAMETER :: eosD2o = 2
      INTEGER(sik), PARAMETER :: eosHe = 3
      INTEGER(sik), PARAMETER :: eosNa = 4
      INTEGER(sik), PARAMETER :: eosPbBi = 5
      INTEGER(sik), PARAMETER :: eosPb = 6
      INTEGER(sik), PARAMETER :: eosDTHT = 7
      INTEGER(sik), PARAMETER :: eosIAPWS = 8
      INTEGER(sik), PARAMETER :: eosN2 = 9
      INTEGER(sik), PARAMETER :: eosAir = 10
      CHARACTER(LEN=5), DIMENSION(maxeos) :: eosNames = (/ 'H2O  ', 'D2O  ', 'He   &
     &', 'Na  ', 'PbBi ', 'Pb  ', 'DTHT ',  'STH2O ', 'N2  ', 'Air  ' /)
      CHARACTER(LEN=5), DIMENSION(maxeos) :: eosIcNames = (/ 'h2o  ', 'd2o  ', 'he &
     &', 'na  ', 'pbbi ', 'Pb  ', 'dtht ', 'sth2o ', 'n2  ', 'air  ' /)

!     Molecular weight of Pb.
      REAL(sdk), PARAMETER :: PbMWt = 207.2_sdk
!
!     Molecular weight of PbBi.
      REAL(sdk), PARAMETER :: pbbiMWt = 0.44_sdk * 207.2_sdk + 0.56_sdk * 208.9808_sdk
!
!     Molecular weight of DTHT.
      REAL(sdk), PARAMETER :: dthtMWt = 236.4_sdk
```

Modul EosInitM

```
PRIVATE :: SetEoH, SetEoD, SetEoHe, SetEoNa, SetEoPbBi
PRIVATE :: SetEoPb, SetEoDTHT, SetEoIAPWS
PRIVATE :: SetEoN2, SetEoAir

  CALL SetEoH
  DO i = 1, nActEos
    SELECT CASE (actFluids(i))
      CASE (eosH2o)
        CYCLE
      CASE (eosD2o)
        CALL SetEoD()
      CASE (eosHe)
        CALL SetEoHe()
      CASE (eosNa)
        CALL SetEoNa()
      CASE (eosPbBi)
        CALL SetEoPbBi()
      CASE (eosPb)
        CALL SetEoPb()
      CASE (eosDTHT)
        CALL SetEoDTHT()
      CASE (eosIAPWS)
        CALL SetEoIAPWS()
      CASE (eosN2)
        CALL SetEoN2()
      CASE (eosAir)
        CALL SetEoAir()
      CASE DEFAULT
        CALL error(1, '*seteos* Bad eos flag.')
    END SELECT
  END DO
```

Modul EosM

```
PRIVATE :: Set3DEosPtrs, Set1DEosPtrs
PRIVATE :: ThermH, ThermD, ThermHe, ThermNa, ThermPbBi, ThermPb, ThermDTHT
PRIVATE :: ThermIAPWS, ThermN2, ThermAir
PRIVATE :: RhoLiHe, RhoLiNa, RhoLiPbBi, RhoLiPb, RhoLiDTHT, RhoLiN2, RhoLiAir
PRIVATE :: SatDeH, SatDeD, SatDeHe, SatDeNa, SatDePbBi, SatDePb, SatDeDTHT
PRIVATE :: SatDeIAPWS, SatDeN2, SatDeAir
PRIVATE :: SatPrH, SatPrD, SatPrHe, SatPrNa, SatPrPbBi, SatPrPb, SatPrDTHT
PRIVATE :: SatPrIAPWS, SatPrN2, SatPrAir
PRIVATE :: FPropH, FPropD, FPropHe, FPropNa, FPropPbBi, FPropPb, FPropDTHT
PRIVATE :: FPropIAPWS, FPropN2, FPropAir
PRIVATE :: SatTmH, SatTmD, SatTmHe, SatTmNa, SatTmPbBi, SatTmPb, SatTmDTHT
PRIVATE :: SatTmIAPWS, SatTmN2, SatTmAir
PRIVATE :: ViscLHe, ViscLNa, ViscLPbBi, ViscLPb, ViscLDTHT, ViscLN2, ViscLAir
```

```
    PRIVATE :: ViscVH, ViscVD, ViscVNa, ViscVPbBi, ViscVPb, ViscVDTHT, ViscVIAPWS
    PRIVATE :: ViscVN2, ViscVAir
    PRIVATE :: HeVH, HeVD, HeVHe, HeVNa, HeVPbBi, HeVPb, HeVDTHT, HeVIAPWS,
    PRIVATE :: HeVN2, HeVAir
    PRIVATE :: ThcLHe, ThcLNa, ThcLPbBi, ThcLPb, ThcLDTHT, ThcLN2, ThcLAir
    PRIVATE :: ThcVH, ThcVD, ThcVHe, ThcVNa, ThcVPbBi, ThcVPb, ThcVDTHT,
    PRIVATE :: ThcVN2, ThcVAir
    PRIVATE :: SigmaHe, SigmaNa, SigmaPbBi, SigmaPb, SigmaDTHT, SigmaN2, SigmaAir
    PRIVATE :: CpllHe, CpllNa, CpllPbBi, CpllPb, CpllDTHT, CpllN2, CpllAir
    PRIVATE :: CpvvH, CpvvD, CpvvHe, CpvvNa, CpvvPbBi, CpvvPb, CpvvDTHT
    PRIVATE :: CpvvIAPWS, CpvvN2, CpvvAir
    PRIVATE :: NullEosPtrs
!
    PUBLIC :: RhoLiIAPWS, energyLiIAPWS, ViscLIAPWS, CpllIAPWS, ThcLIAPWS
          ! needed for wall properties in RefloodM.f90 module
    PUBLIC :: RhoLiH, ViscLH, ThcLH, CpllH, RhoLiD, ViscLD, ThcLD, CpllD

      SELECT CASE (Eos)
        CASE (eosH2o)
          CALL ThermH(iBeg, iEnd, ccoTmp)
        CASE (eosD2o)
          CALL ThermD(iBeg, iEnd, ccoTmp)
        CASE (eosHe)
          CALL ThermHe(iBeg, iEnd, ccoTmp)
        CASE (eosNa)
          CALL ThermNa(iBeg, iEnd, ccoTmp)
        CASE (eosPbBi)
          CALL ThermPbBi(iBeg, iEnd, ccoTmp)
        CASE (eosPb)
          CALL ThermPB(iBeg, iEnd, ccoTmp)
        CASE (eosDTHT)
          CALL ThermDTHT(iBeg, iEnd, ccoTmp)
        CASE (eosIAPWS)
          CALL ThermIAPWS(iBeg, iEnd, ccoTmp)
        CASE (eosN2)
          CALL ThermN2(iBeg, iEnd, ccoTmp)
        CASE (eosAir)
          CALL ThermAir(iBeg, iEnd, ccoTmp)
        CASE DEFAULT
          CALL EosErr("Thermo", Eos)
      END SELECT

      SELECT CASE (Eos)
        CASE (eosH2o)
          CALL ThermH(1, 1, ccoTmp)
        CASE (eosD2o)
          CALL ThermD(1, 1, ccoTmp)
        CASE (eosNa)
          CALL ThermNa(1, 1, ccoTmp)
        CASE (eosHe)
```

```
      CALL ThermHe(1, 1, ccoTmp)
    CASE (eosPbBi)
      CALL ThermPbBi(1, 1, ccoTmp)
    CASE (eosPb)
      CALL ThermPB(1, 1, ccoTmp)
    CASE (eosDTHT)
      CALL ThermDTHT(1, 1, ccoTmp)
    CASE (eosIAPWS)
      CALL ThermIAPWS(1, 1, ccoTmp)
    CASE (eosN2)
      CALL ThermN2(1, 1, ccoTmp)
    CASE (eosAir)
      CALL ThermAir(1, 1, ccoTmp)
    CASE DEFAULT
      CALL EosErr("ThermS", Eos)
  END SELECT

  SELECT CASE (Eos)
    CASE (eosH2o)
      CALL RhoLiH(p, tl, rhol, drldp, drldt)
    CASE (eosD2o)
      CALL RhoLiD(p, tl, rhol, drldp, drldt)
    CASE (eosHe)
      CALL RhoLiHe(p, tl, rhol, drldp, drldt)
    CASE (eosNa)
      CALL RhoLiNa(p, tl, rhol, drldp, drldt)
    CASE (eosPbBi)
      CALL RhoLiPbBi(p, tl, rhol, drldp, drldt)
    CASE (eosPb)
      CALL RhoLiPb(p, tl, rhol, drldp, drldt)
    CASE (eosDTHT)
      CALL RhoLiDTHT(p, tl, rhol, drldp, drldt)
    CASE (eosIAPWS)
      CALL RhoLiIAPWS(p, tl, rhol, drldp, drldt)
    CASE (eosN2)
      CALL RhoLiN2(p, tl, rhol, drldp, drldt)
    CASE (eosAir)
      CALL RhoLiAir(p, tl, rhol, drldp, drldt)
    CASE DEFAULT
      CALL EosErr("RhoLiq", Eos)
  END SELECT

  SELECT CASE (Eos)
    CASE (eosH2o)
      SatDer = SatDeH(pres, temp)
    CASE (eosD2o)
      SatDer = SatDeD(pres, temp)
    CASE (eosHe)
      SatDer = SatDeHe()
    CASE (eosNa)
```

```
      SatDer = SatDeNa(pres)
    CASE (eosPbBi)
      SatDer = SatDePbBi(pres)
    CASE (eosPb)
      SatDer = SatDePb(pres)
    CASE (eosDTHT)
      SatDer = SatDeDTHT(pres)
    CASE (eosIAPWS)
      SatDer = SatDeIAPWS(pres, temp)
    CASE (eosN2)
      SatDer = SatDeN2()
    CASE (eosAir)
      SatDer = SatDeAir()
    CASE DEFAULT
      CALL EosErr('SatDer', Eos)
  END SELECT

  SELECT CASE (Eos)
    CASE (eosH2o)
      SatPrS = SatPrH(temp)
    CASE (eosD2o)
      SatPrS = SatPrD(temp)
    CASE (eosHe)
      SatPrS = SatPrHe(temp)
    CASE (eosNa)
      SatPrS = SatPrNa(temp)
    CASE (eosPbBi)
      SatPrS = SatPrPbBi(temp)
    CASE (eosPb)
      SatPrS = SatPrPb(temp)
    CASE (eosDTHT)
      SatPrS = SatPrDTHT(temp)
    CASE (eosIAPWS)
      IF (PRESENT(err)) THEN
        SatPrS = SatPrIAPWS(temp, err)
      ELSE
        SatPrS = SatPrIAPWS(temp)
      END IF
    CASE (eosN2)
      SatPrS = SatPrN2(temp)
    CASE (eosAir)
      SatPrS = SatPrAir(temp)
    CASE DEFAULT
      CALL EosErr("SatPrS", Eos)
  END SELECT

  SELECT CASE (Eos)
    CASE (eosH2o)
      SatTmP = SatTmH(pres)
    CASE (eosD2o)
```

```
      SatTmP = SatTmD(pres)
    CASE (eosHe)
      SatTmP = SatTmHe(pres)
    CASE (eosNa)
      SatTmP = SatTmNa(pres)
    CASE (eosPbBi)
      SatTmP = SatTmPbBi(pres)
    CASE (eosPb)
      SatTmP = SatTmPb(pres)
    CASE (eosDTHT)
      SatTmP = SatTmDTHT(pres)
    CASE (eosIAPWS)
      IF (PRESENT(err)) THEN
        SatTmP = SatTmIAPWS(pres, err)
      ELSE
        SatTmP = SatTmIAPWS(pres)
      END IF
    CASE (eosN2)
      SatTmP = SatTmN2(pres)
    CASE (eosAir)
      SatTmP = SatTmAir(pres)
    CASE DEFAULT
      CALL EosErr("SatTmP", Eos)
  END SELECT

  SELECT CASE (Eos)
    CASE (eosH2o)
     ViscL = ViscLH(h, p)
    CASE (eosD2o)
      ViscL = ViscLD(tl)
    CASE (eosHe)
      ViscL = ViscLHe()
    CASE (eosNa)
      ViscL = ViscLNa(tl)
    CASE (eosPbBi)
      ViscL = ViscLPbBi(tl)
   CASE (eosPb)
      ViscL = ViscLPb(tl)
    CASE (eosDTHT)
      ViscL = ViscLDTHT(tl)
    CASE (eosIAPWS)
     ViscL = ViscLIAPWS(rol, tl)
    CASE DEFAULT
      CALL EosErr("ViscL", Eos)
  END SELECT

    CASE (eosH2o)
      ViscV = ViscVH(p, rov, tv, pa, .FALSE., idx, xg)
    CASE (eosD2o)
      ViscV = ViscVD(p, rov, tv, pa, idx, xg)
```

```
  CASE (eosHe)
    ViscV = ViscVHe(tv, p, pa, xg)
  CASE (eosNa)
    ViscV = ViscVNa(tv, p, pa, xg)
  CASE (eosPbBi)
    ViscV = ViscVPbBi(tv, p, pa, xg)
  CASE (eosPb)
    ViscV = ViscVPb(tv, p, pa, xg)
  CASE (eosDTHT)
    ViscV = ViscVDTHT(tv, p, pa, xg)
  CASE (eosIAPWS)
    ViscV = ViscVIAPWS(p, rov, tv, pa, idx, xg)
  CASE (eosN2)
    ViscV = ViscVN2(tv, p, pa, xg)
  CASE (eosAir)
    ViscV = ViscVAir(tv, p, pa, xg)
  CASE DEFAULT
    CALL EosErr("ViscV", Eos)
END SELECT

SELECT CASE (Eos)
  CASE (eosH2o)
    HeV = HeVH(temp)
  CASE (eosD2o)
    HeV = HeVD(temp)
  CASE (eosHe)
    HeV = HeVHe(temp)
  CASE (eosNa)
    HeV = HeVNa(temp)
  CASE (eosPbBi)
    HeV = HeVPbBi(temp)
  CASE (eosPb)
    HeV = HeVPb(temp)
  CASE (eosDTHT)
    HeV = HeVDTHT(temp)
  CASE (eosIAPWS)
    HeV = HeVIAPWS(temp)
  CASE (eosN2)
    HeV = HeVN2(temp)
  CASE (eosAir)
    HeV = HeVAir(temp)
END SELECT

SELECT CASE (Eos)
  CASE (eosH2o)
    CALL FPropH(p, el, rol, rov,tv, tsat, hfg, cpl, cpv, visl, visv, cl,&
   &cv, sig, nc, pa, hls, hvs, idx, xg)
  CASE (eosD2o)
    CALL FPropD(p, el, rol, rov, tl, tv, tsat, hfg, cpl, cpv, visl, visv,&
   &cl, cv, sig, nc, pa, hls, hvs, idx, xg)
```

```
    CASE (eosHe)
      CALL FPropHe(p, pa, el, rol, tv, tsat, hfg, cpl, cpv, visl, visv, cl,&
     &cv, sig, nc, hls, hvs, xg)
    CASE (eosNa)
      CALL FPropNa(p, tl, tv, tsat, hfg, cpl, cpv, visl, visv, cl, cv, sig,&
     &nc, pa, hls, hvs, xg)
    CASE (eosPbBi)
      CALL FPropPbBi(p, tl, tv, tsat, hfg, cpl, cpv, visl, visv, cl, cv, sig,&
     &nc, pa, hls, hvs, xg)
    CASE (eosPb)
      CALL FPropPb(p, tl, tv, tsat, hfg, cpl, cpv, visl, visv, cl, cv, sig,&
     &nc, pa, hls, hvs, xg)
    CASE (eosDTHT)
      CALL FPropDTHT(p, tl, tv, tsat, hfg, cpl, cpv, visl, visv, cl, cv, sig,&
     &nc, pa, hls, hvs, xg)
    CASE (eosIAPWS)
      CALL FPropIAPWS(p, el, rol, rov, tl, tv, tsat, hfg, cpl, cpv, visl,&
     &visv, cl, cv, sig, nc, pa, hls, hvs, idx, xg)
    CASE (eosN2)
      CALL FPropN2(p, pa, el, rol, tv, tsat, hfg, cpl, cpv, visl, visv, cl,&
     &cv, sig, nc, hls, hvs, xg)
    CASE (eosAir)
      CALL FPropAir(p, pa, el, rol, tv, tsat, hfg, cpl, cpv, visl, visv, cl,&
     &cv, sig, nc, hls, hvs, xg)
    CASE DEFAULT
      CALL EosErr("FProp", Eos)
  END SELECT
END SUBROUTINE FProp

  SELECT CASE (Eos)
    CASE (eosH2o)
      ThcL = ThcLH(h)
    CASE (eosD2o)
      ThcL = ThcLD(h)
    CASE (eosHe)
      ThcL = ThcLHe()
    CASE (eosNa)
      ThcL = ThcLNa(tl)
    CASE (eosPbBi)
      ThcL = ThcLPbBi(tl)
    CASE (eosPb)
      ThcL = ThcLPb(tl)
    CASE (eosDTHT)
      ThcL =ThcLDTHT(tl)
    CASE (eosIAPWS)
      ThcL = ThcLIAPWS(tl, rol)
    CASE (eosN2)
      ThcL = ThcLN2()
    CASE (eosAir)
      ThcL = ThcLAir()
```

```
  CASE DEFAULT
    CALL EosErr("ThcL", Eos)
END SELECT

SELECT CASE (Eos)
  CASE (eosH2o)
    ThcV = ThcVH(p, rov, tv, pa, idx, xg)
  CASE (eosD2o)
    ThcV = ThcVD(p, rov, tv, pa, idx, xg)
  CASE (eosHe)
    ThcV = ThcVHe(tv, p, pa, xg)
  CASE (eosNa)
    ThcV = ThcVNa(tv, p, pa, xg)
  CASE (eosPbBi)
    ThcV = ThcVPbBi(tv, p, pa, xg)
  CASE (eosPb)
    ThcV = ThcVPb(tv, p, pa, xg)
  CASE (eosDTHT)
    ThcV = ThcVDTHT(tv, p, pa, xg)
  CASE (eosIAPWS)
    ThcV = ThcVIAPWS(p, rov, tv, pa, idx, xg)
  CASE (eosN2)
    ThcV = ThcVN2(tv, p, pa, xg)
  CASE (eosAir)
    ThcV = ThcVAir(tv, p, pa, xg)
  CASE DEFAULT
    CALL EosErr("ThcV", Eos)
END SELECT

SELECT CASE (Eos)
  CASE (eosH2o)
    CpIl = CpIlH(h, p)
  CASE (eosD2o)
    CpIl = CpIlD(h, p)
  CASE (eosHe)
    CpIl = CpIlHe(h, p)
  CASE (eosNa)
    CpIl = CpIlNa(tl)
  CASE (eosPbBi)
    CpIl = CpIlPbBi(tl)
  CASE (eosPb)
    CpIl = CpIlPb(tl)
  CASE (eosDTHT)
    CpIl = CpIlDTHT(tl)
  CASE (eosIAPWS)
    CpIl = CpIlIAPWS(tl, p)
  CASE (eosN2)
    CpIl = CpIlN2(h, p)
  CASE (eosAir)
    CpIl = CpIlAir(h, p)
```

```
      CASE DEFAULT
        CALL EosErr("Cpll", Eos)
    END SELECT

    SELECT CASE (Eos)
      CASE (eosH2o)
        Cpvv = CpvvH(t, p, pa, idx, xg)
      CASE (eosD2o)
        Cpvv = CpvvD(t, p, pa, idx, xg)
      CASE (eosHe)
        Cpvv = CpvvHe(t, p, pa, xg)
      CASE (eosNa)
        Cpvv = CpvvNa(t, p, pa, xg)
      CASE (eosPbBi)
        Cpvv = CpvvPbBi(t, p, pa, xg)
      CASE (eosPb)
        Cpvv = CpvvPb(t, p, pa, xg)
      CASE (eosDTHT)
        Cpvv = CpvvDTHT(t, p, pa, xg)
      CASE (eosIAPWS)
        Cpvv = CpvvIAPWS(t, p, pa, idx, xg)
      CASE (eosN2)
        Cpvv = CpvvN2(t, p, pa, xg)
      CASE (eosAir)
        Cpvv = CpvvAir(t, p, pa, xg)
      CASE DEFAULT
        CALL EosErr("Cpvv", Eos)
    END SELECT
```

Modul PreInputM

```
 CALL value(crdout, iclout, lennum, idum, it, fval)
 IF (it.NE.1).AND.(it.NE.2).AND.(crdout(iclout:iclout+1).NE.'lu'))THEN
   IF((crdout(iclout:iclout+2).NE.'H2O').AND.(crdout(iclout:iclout+2).NE.'D2O')&
  &.AND.(crdout(iclout:iclout+1).NE.'He').AND.(crdout(iclout:iclout+1).NE.'Na')&
  &.AND.(crdout(iclout:iclout+3).NE.'Pb').AND.(crdout(iclout:iclout+3).NE.'Pb&
  &Bi').AND.(crdout(iclout:iclout+3).NE.'DTHT').AND.(crdout(iclout:iclout+1).NE.'N2&
  &').AND.(crdout(iclout:iclout+2).NE.'Air').AND.(crdout(iclout:iclout+3).NE.'TR&
  &UE').AND.(crdout(iclout:iclout+4).NE.'FALSE'))THEN
     IF((crdout(iclout:iclout+2).NE.'h2o').AND.(crdout(iclout:iclout+2).NE.'d2o')&
    &.AND.(crdout(iclout:iclout+1).NE.'he').AND.(crdout(iclout:iclout+1).NE.'na')&
    &.AND.(crdout(iclout:iclout+3).NE.'pb').AND.(crdout(iclout:iclout+3).NE.'pb&
    &bi').AND.(crdout(iclout:iclout+3).NE.'dtht').AND.(crdout(iclout:iclout+1)&
    &.NE.'n2').AND.(crdout(iclout:iclout+2).NE.'air').AND.(crdout(iclout:i$\&$
    &clout+3).NE.'true').AND.(crdout(iclout:iclout+4).NE.'false'))THEN
       GO TO 125
```

A.1.2 DIPHYL THT

Modul EosInitM

```
      SUBROUTINE SetEoDTHT
!
        USE Eos, ONLY: HeV
        USE EosData
        USE EosNCGData, ONLY: SetCeoslpNCG
        !USE GlobalDat, ONLY: gasCon
!
        IMPLICIT NONE
!
        REAL(sdk) vapmol
!
!     vapmol     = molecular weight of vapor
!
        aeos14(eosDTHT) = 1.0d-05
        ceos1(eosDTHT) = 117.8_sdk
        ceos2(eosDTHT) = 0.223_sdk
        ceos3(eosDTHT) = 255.2_sdk
!
!       Non-condensable gas is defined by iGas.
        CALL SetCeoslpNCG(iGas, eosDTHT)
!
        vapmol = dthtMWt
!
        ceoslp(1, eosDTHT) = 9923.19_sdk != 0.05223*67222.1 for DTHT
                             !(see function SatTmDTHT).
        ceoslp(2, eosDTHT) = 10.163_sdk != 8.487+log10(133.32),
                            !where 133.32 Pa = 1 mm of Hg.
        ceoslp(3, eosDTHT) = LOG10(EXP(1.0_sdk)) * ceoslp(1, eosDTHT)! = 1524.8123
        ceoslp(5, eosDTHT) = 273.0_sdk
        ceoslp(7, eosDTHT) = 1750.0_sdk
        ceoslp(9, eosDTHT) = 1194.0_sdk
        ceoslp(10, eosDTHT) = HeV(ceoslp(5, eosDTHT), eosDTHT)
        ceoslp(11, eosDTHT) = 100000.0_sdk
        ceoslp(12, eosDTHT) = gasCon / vapmol
        ceoslp(14, eosDTHT) = 0.65141_sdk
        ceoslp(15, eosDTHT) = 0.0_sdk
        ceoslp(16, eosDTHT) = 1.3_sdk
        ceoslp(20, eosDTHT) = 9.056466d4
        ceoslp(21, eosDTHT) = 273.0_sdk
        ceoslp(24, eosDTHT) = ceoslp(7, eosDTHT)
        ceoslp(29, eosDTHT) = 273.0_sdk
        ceoslp(30, eosDTHT) = 1000.0_sdk
        ceoslp(31, eosDTHT) = 100.0d6
        ceoslp(32, eosDTHT) = 273.0_sdk
        ceoslp(33, eosDTHT) = 1773.15_sdk
        ceoslp(34, eosDTHT) = 273.0_sdk
        ceoslp(35, eosDTHT) = 1773.15_sdk
        ceoslp(36, eosDTHT) = 1773.15_sdk
```

```
      ceoslp(37, eosDTHT) = 1.0d7
      ceoslp(38, eosDTHT) = 773.15_sdk
      ceoslp(39, eosDTHT) = 139.69971285053d5
      ceoslp(40, eosDTHT) = 609.62462615967_sdk
!      ceoslp(28) = (ceoslp(12)-ceoslp(25))/ceoslp(12)
!
      ceoslp(4, eosDTHT) = ceoslp(12, eosDTHT) / (ceoslp(16, eosDTHT)-1.0_sdk)
      ceoslp(23, eosDTHT) = ceoslp(16, eosDTHT) * ceoslp(4, eosDTHT)
      ceoslp(8, eosDTHT) = - 611.2_sdk * 0.0010002_sdk + ceoslp(7, eosDTHT)&
     & * (ceoslp(5, eosDTHT)-ceoslp(29, eosDTHT))
      ceoslp(26, eosDTHT) = ceoslp(24, eosDTHT) * (ceoslp(5, eosDTHT)-&
     &ceoslp(29, eosDTHT))
      ceoslp(27, eosDTHT) = ceoslp(26, eosDTHT) + ceoslp(10, eosDTHT)
      ceoslp(6, eosDTHT) = ceoslp(27, eosDTHT) - ceoslp(12, eosDTHT)&
     & * ceoslp(5, eosDTHT)
!
    END SUBROUTINE SetEoDTHT
```

Modul EosM

```
    SUBROUTINE ThermDTHT(jstart, jstop, ccoThermDTHT)
!
      USE Io
      USE EosNCGData
      USE Global, ONLY: cco
      USE GlobalDat, ONLY: iftp, numberOfNCGases
      USE Flt
!
      IMPLICIT NONE
!
!    subroutine ThermDTHT evaluates the thermodynamic properties of DTHT
!
!    input variables
!      1. p      pressure
!      2. tl     liquid temperature
!      3. tv     vapor temperature
!      4. pa     partial pressure of the noncondensible
!
!    output variables
!      1. el     liquid internal energy
!      2. ev     vapor (steam and noncondensable mixture) internal
!                energy
!      3. tsat   saturation temperature corresponding to the total
!                pressure
!      4. rol    liquid density
!      5. rov    vapor (steam and noncondensable mixture) density
!      6. rova   density of the noncondensable
!      7. tssn   saturation temperature corresponding to the steam
!                partial pressure
!      8. eva    internal energy of the noncondensable
```

```
!       9. dtsdp  derivative of tsat wrt pressure
!      10. deldp  derivative of el wrt pressure
!      11. devdp  derivative of steam internal energy wrt steam
!              partial pressure
!      12. deldt  derivative of el wrt tl
!      13. devdt  derivative of steam internal energy wrt tv
!      14. drolp  derivative of rol wrt pressure
!      15. drovp  derivative of steam density wrt steam partial
!              pressure
!      16. drolt  derivative of rol wrt tl
!      17. drovt  derivative of steam density wrt tv
!      18. hvst   saturated steam enthalpy (psteam,tssn)
!      19. hlst   saturated liquid enthalpy (p,tssn)
!      20. dhvsp  derivative of hvst wrt steam partial pressure
!      21. dhlsp  derivative of hlst wrt pressure
!      22. dtssp  derivative of tssn wrt steam partial pressure
!      23. devat  derivative of eva wrt tv
!      24. devap  derivative of eva wrt pa
!      25. drvap  derivative of rova wrt pa
!      26. drvat  derivative of rova wrt tv
!
       INTEGER(sik), INTENT(IN) :: jstart, jstop
       INTEGER(sik), INTENT(IN), OPTIONAL :: ccoThermDTHT
       INTEGER(sik) :: ie, j, idx, n
!
       REAL(sdk) pt, tl2, tv1, pg, ps, rolst, rrolst, elsat, delsat, drlsdt, drlsdp
       REAL(sdk) rGasNC, rGasDTHT
       REAL(sdk), DIMENSION(numberOfNCGases) :: dRGasMixturedXgn, eNCGas
!
       ie = eosDTHT
       phaseChange = .FALSE.
       IF (PRESENT(ccoThermDTHT)) THEN
         idx = ccoThermDTHT
       ELSE
         idx = cco
       END IF
       DO j = jstart, jstop
!
         pt = p(j)
         pa(j) = 0.0_sdk
         pg = 0.0_sdk
         IF (p(j) < ceoslp(30, ie) .OR. p(j) > ceoslp(31, ie)) THEN
           pt = MIN(ceoslp(31, ie), MAX(ceoslp(30, ie), p(j)))
           IF (iftp /= 1) THEN
             CALL error(2, '*thermDTHT* pressure limit exceeded')
             WRITE (imout, 18) genTab(idx)%num, j
18             FORMAT (15 x, 'component', i4, ', ', 'cell', i4)
           END IF
         END IF
!
```

```
!     calculate saturation properties
!
          ps = pt - pg
          ps = MAX(ps, ceoslp(30, ie))
          tsat(j) = SatTmDTHT(pt)
          dtsdp(j) = SatDeDTHT(pt)
          tl2 = tl(j)
          tv1 = tv(j)
          tssn(j) = tsat(j)
          dtssp(j) = dtsdp(j)
!
          tl2 = MIN(ceoslp(33, ie), MAX(tl2, ceoslp(32, ie)))
          tv1 = MIN(ceoslp(35, ie), MAX(tv1, ceoslp(34, ie)))
!
!     calculate liquid properties
!
!       1. internal energy and its derivatives
!
          el(j) = ceoslp(7, ie) * (tl2-ceoslp(5, ie)) + ceoslp(8, ie)
          deldt(j) = ceoslp(7, ie)
          deldp(j) = 0.0_sdk
          elsat = ceoslp(7, ie) * (tssn(j)-ceoslp(5, ie)) + ceoslp(8, ie)
          delsat = ceoslp(7, ie)
!
          CALL RhoLiDTHT(pt, tssn(j), rolst, drlsdp, drlsdt)
!
          rrolst = 1.0_sdk / rolst
          hlst(j) = elsat + pt * rrolst
          dhlsp(j) = delsat * dtsdp(j) + rrolst - pt * rrolst * rrolst * &
         &(drlsdt*dtsdp(j)+drlsdp)
!
!       2. density and its derivatives
!
          CALL RhoLiDTHT(pt, tl2, rol(j), drolp(j), drolt(j))
!
!     calculate vapor properties
!
!     -----internal energy, enthalpy, and their derivatives
          hvst(j) = ceoslp(16, ie) * ceoslp(23, ie) * (tv1-ceoslp(26, ie))&
         & + ceoslp(6, ie)
          dhvsp(j) = 0.0_sdk
!
!     calculate noncondensable gas properties
!
          IF (iGas > 10_sik) THEN
            rGasNC = RGasMixture(xgn(1:numberOfNCGases, j), dRGasMixturedXgn)
            eva(j) = EnergyMixture(xgn(1:numberOfNCGases, j), tv(j), eNCGas)
            devat(j) = CvMixture(xgn(1:numberOfNCGases, j), tv(j))
          devap(j) = 0.0_sdk
            drvap(j) = 1.0_sdk / (rGasNC * tv(j))
```

```
          rova(j) = drvap(j) * pa(j)
            drvat(j) =-rGasNC * rova(j) * drvap(j)
            DO n = 1, numberOfNCGases
              drvax(n, j) =-pa(j) * dRGasMixturedXgn(n)/ (rGasNC**2 * tv(j))
              devax(n, j) = eNCGas(n) - eNCGas(1)
            END DO
          ELSE
            rGasNC = ncGasProperties(iGas)%rGas
            eva(j) = EnergyNCGas(ncGasProperties(iGas)%cvFit, tv1)
            devat(j) = CvFit(ncGasProperties(iGas)%cvFit, tv1)
            devap(j) = 0.0_sdk
            drvap(j) = 1.0_sdk / (rGasNC*tv1)
            rova(j) = drvap(j) * pa(j)
            drvat(j) =-rGasNC * rova(j) * drvap(j)
          END IF
!
!         Calculate non-condensable vapor properties.
          rGasDTHT = rGasNC
          ev(j) = eva(j)
          devdt(j) = devap(j)
          devdp(j) = 0.0_sdk
          drovp(j) = drvap(j)
          rov(j) = drvap(j) * ps
          drovt(j) = drvat(j)
!
!    calculate air-steam mixture properties
!
          ev(j) = (ev(j)*rov(j)+eva(j)*rova(j))
          rov(j) = rov(j) + rova(j)
          ev(j) = ev(j) / rov(j)
          p(j) = pt
          tl(j) = tl2
          tv(j) = tv1
!
!   Calculate cpLiq and cpVap because everything is available now to calculate it.
!          cpLiq(j) = deldt(j) - p(j) * drolt(j) / (rol(j)**2)
!          cpVap(j) = CpvvDTHT(tv(j), p(j), pa(j), xgn(1:, j))
        END DO
!
     END SUBROUTINE ThermDTHT

    SUBROUTINE RhoLiDTHT(p, tl, rhol, drldp, drldt)
    IMPLICIT NONE
!
!    subroutine RhoLiDTHT evaluates the density of DTHT liquid and its
!    derivatives with respect to total pressure and liquid temper-
!    ature as a function of total pressure and liquid temperature.
!
!         total pressure              p      in (pa)
!         liquid temperature          tl     in (k)
```

```
!       liquid density            rol    in (kg/m**3)
!       drol/dp                  drldp  in (kg/m**3/pa)
!       drol/dt                  drldt  in (kg/m**3/k)
!
        REAL(sdk), INTENT(IN) :: p, tl
        REAL(sdk), INTENT(OUT) :: rhol, drldp, drldt
!
        drldt = - 0.6495_sdk
        drldp = 1.0e-6_sdk
        rhol = ceoslp(9, eosDTHT) + drldt * (tl) + drldp * (p)
!
    END SUBROUTINE RhoLiDTHT

    REAL(sdkx) FUNCTION SatDeDTHT(pres)
    IMPLICIT NONE

!   function SatDeDTHT evaluates the derivative of the DTHT saturation
!   temperature with respect to total pressure as a function
!   of the saturation pressure.
!
!       pres                  pressure (pa)
!       SatDeDTHT                 dtsat/dp in (k/pa)
!
        REAL(sdk), INTENT(IN) :: pres
!
        SatDeDTHT = ceoslp(3, eosDTHT) / (pres*(LOG10(pres)-ceoslp(2, eosDTHT))**2)
!
    END FUNCTION SatDeDTHT

   REAL(sdkx) FUNCTION SatPrDTHT(temp)
!
   IMPLICIT NONE
!
!   function SatPrDTHT evaluates the DTHT saturation pressure
!   as a function of the saturation temperature
!
!       saturation temperature      temp   in (k)
!       saturation pressure        SatPrDTHT in (pa)
!
        REAL(sdk), INTENT(IN) :: temp
!
        SatPrDTHT = 10.0_sdk ** (ceoslp(2, eosDTHT)-ceoslp(1, eosDTHT)/temp)
        SatPrDTHT = MAX(SatPrDTHT, ceoslp(30, eosDTHT))
!
    END FUNCTION SatPrDTHT

    REAL(sdkx) FUNCTION SatTmDTHT(pres)
!
    IMPLICIT NONE
!
```

```
!     function SatTmDTHT evaluates the DTHT saturation temperature
!     as a function of the saturation pressure.
!     Use correlation log10(pres) = -0.05223*A/T + B
!     where A = 67,222.1 and B = 8.48666 + log10(133.32) for DTHT
!     133.32 Pa per mm of Hg.
!
!        saturation pressure        pres      in (pa)
!        saturation temperature     SatTmPb in (k)
!
        REAL(sdk), INTENT(IN) :: pres
!
        SatTmDTHT = ceoslp(1, eosDTHT) / (ceoslp(2, eosDTHT)-LOG10(pres))
!
      END FUNCTION SatTmDTHT

      REAL(sdkx) FUNCTION ViscLDTHT(tl)
!
      IMPLICIT NONE
!
!     function ViscLDTHT evaluates the DTHT liquid dynamic viscosity
!
!        liquid temperature         tl     in (K)
!
        REAL(sdk), INTENT(IN) :: tl
!
        ViscLDTHT = 2.4163e-03_sdk * (EXP((1.1938e+06_sdk / tl**2) &
       &- (3.0369e+03_sdk / tl)))
!
      END FUNCTION ViscLDTHT

      REAL(sdkx) FUNCTION ViscVDTHT(tv, p, pa, xg)
!
      USE GlobalDat
      IMPLICIT NONE
!
!     function ViscVDTHT evaluates the DTHT vapor dynamic viscosity
!
!        vapor viscosity            ViscVDTHT  in (pa*s)
!
        REAL(sdk), INTENT(IN) :: p, pa, tv
        REAL(sdk), DIMENSION(:), INTENT(IN) :: xg
        REAL(sdk) :: f
!
        ViscVDTHT = 2.0e-04_sdk
        IF (pa > paSign) THEN
          f = MIN(1.0_sdk, pa/p)
          ViscVDTHT = (1.0_sdk-f) * ViscVDTHT + f * ViscNC(tv, xg)
        END IF
!
      END FUNCTION ViscVDTHT
```

```
      REAL(sdkx) FUNCTION HeVDTHT(temp)
!
      IMPLICIT NONE
!
!     function HeVDTHT calculates the heat of evaporation of DTHT liquid
!     as a function of liquid temperature for low pressures
!
!        liquid temperature          temp   in (k)
!        heat of evaporation         HeVDTHT    in (j/kg)
!
        REAL(sdk), INTENT(IN) :: temp
!
        HeVDTHT = HeVH(temp)
!
      END FUNCTION HeVDTHT

      SUBROUTINE FPropDTHT(p, tl, tv, tsat, hfg, cpl, cpv, visl, visv, cl, cv,&
     &sig, nc, pa, hls, hvs, xg)
!
      USE GlobalDat, ONLY: numberOfNCGases
!
      IMPLICIT NONE
!
!     subroutine FPropDTHT evaluates the DTHT fluid enthalpy,
!     heat of vaporization, specific heat, viscosity,
!     thermal conductivity, and surface tension
!
        INTEGER(sik), INTENT(IN) :: nc
        INTEGER(sik) i
!
        REAL(sdk), DIMENSION(:), INTENT(IN) :: p, pa, tl, tv, hvs, hls, tsat
        REAL(sdk), DIMENSION(:), INTENT(OUT) :: hfg, cpl, cpv, visl, visv, cl, cv, sig
        REAL(sdk), DIMENSION(:, :), POINTER :: xg
        REAL(sdk), DIMENSION(:), POINTER :: xgi
!
        DO i = 1, nc
          IF (iGas > 10_sik) THEN
            xgi => xg(1:numberOfNCGases, i)
          ELSE
            xgi => xg(1:, i)
          END IF
!
          hfg(i) = hvs(i) - hls(i)
!
          cpl(i) = CpllDTHT(tl(i))
          cpv(i) = CpvvDTHT(tv(i), p(i), pa(i), xgi)
!
          visl(i) = ViscLDTHT(tl(i))
          visv(i) = ViscVDTHT(tv(i), p(i), pa(i), xgi)
```

```
!
        cl(i) = ThcLDTHT(tl(i))
        cv(i) = ThcVDTHT(tv(i), p(i), pa(i), xgi)
!
        sig(i) = SigmaDTHT(tsat(i))
!
      END DO
!
    END SUBROUTINE FPropDTHT

    REAL(sdkx) FUNCTION ThcLDTHT(tl)
!
    IMPLICIT NONE
!
!   function ThcLDTHT evaluates the DTHT liquid thermal conductivity.
!
!       liquid temperature          tl     in (k)
!       thermal conductivity        ThcLDTHT   in (w/m/k)
!
      REAL(sdk), INTENT(IN) :: tl
!
!
      ThcLDTHT = 0.1167_sdk - 2.544e-05_sdk * tl
!
    END FUNCTION ThcLDTHT

    REAL(sdkx) FUNCTION ThcVDTHT(tv, p, pa, xg)
!
    IMPLICIT NONE
!
!   function ThcVDTHT evaluates the vapor thermal conductivity
!
!     thermal conductivity ThcVDTHT in (w/m k)
!
      REAL(sdk), INTENT(IN) :: tv, p, pa
      REAL(sdk), DIMENSION(:), INTENT(IN) :: xg
!
      IF (pa > paSign) THEN
        ThcVDTHT = ((p - pa)  + pa * ThcNCG(tv, xg, eosDTHT)) / p
      ELSE
      ThcVDTHT = 1.0_sdk
      END IF
!
    END FUNCTION ThcVDTHT

    REAL(sdkx) FUNCTION SigmaDTHT(tl)
!
    IMPLICIT NONE
!
!   surface tension of DTHT.
```

```
!
!     liquid temperature        tl       in K
!     surface tension           SigmaDTHT  in N/m
!
      REAL(sdk), INTENT(IN) :: tl
!
    SigmaDTHT = (9.22e-8_sdk * (tl - 273.15)**2) - (1.451e-4_sdk * (tl - 273.15)) &
   &+ 0.0462_sdk
!
    END FUNCTION SigmaDTHT

    REAL(sdkx) FUNCTION CpllDTHT(tl)
!
    IMPLICIT NONE
!
!   function CpllDTHT evaluates the specific heat of DTHT liquid
!
!       liquid temperature          tl         in (K)
!       liquid specific heat        CpllDTHT   in (j/kg/k)
!
      REAL(sdk), INTENT(IN) :: tl
!
!
      CpllDTHT = 3.4757_sdk * tl + 513.84_sdk
!
    END FUNCTION CpllDTHT

    REAL(sdkx) FUNCTION CpvvDTHT(tv, p, pa, xg)
!
      IMPLICIT NONE
!
!   function CpvvDTHT evaluates the specific heat of DTHT vapor
!   as a function of vapor temperature, total pressure, and
!   noncondensable-gas pressure
!
!     vapor specific heat         CpvvDTHT  in (j/kg/k)
!
      REAL(sdk), INTENT(IN) :: tv, p, pa
      REAL(sdk), DIMENSION(:), INTENT(IN) :: xg
!
      IF (pa > paSign) THEN
        CpvvDTHT = ((p - pa) * ceoslp(22, eosDTHT) + pa * &
       &CpvNCG(tv, xg, eosDTHT)) / p
      ELSE
        CpvvDTHT = ceoslp(22, eosDTHT)
      END IF
!
    END FUNCTION CpvvDTHT
```

A.1.3 LEAD

Modul EosInitM

```
\begin{lstlisting}[firstnumber=832]
      SUBROUTINE SetEoPb
!
        USE Eos, ONLY: HeV
        USE EosData
        USE EosNCGData, ONLY: SetCeoslpNCG
        !USE GlobalDat, ONLY: gasCon
!
        IMPLICIT NONE
!
        REAL(sdk) vapmol
!
!     subroutine SetEoPb initializes the Pb equation-of-state constants
!
!     vapmol     = molecular weight of vapor
!
        aeos14(eosPb) = 1.0d-05
        ceos1(eosPb) = 117.8_sdk
        ceos2(eosPb) = 0.223_sdk
        ceos3(eosPb) = 255.2_sdk
!
!       Non-condensable gas is defined by iGas.
        CALL SetCeoslpNCG(iGas, eosPb)
!
        vapmol = pbbiMWt
!
        ceoslp(1, eosPb) = 9657.41_sdk ! = 0.05223*184,901 for Pb
                           !(see function SatTmPb).
        ceoslp(2, eosPb) = 9.827_sdk ! = 7.702+log10(133.32),
                          !where 133.32 Pa = 1 mm of Hg.
        ceoslp(3, eosPb) = LOG10(EXP(1.0_sdk)) * ceoslp(1, eosPb)! = 4194.16
        ceoslp(5, eosPb) = 600.6_sdk
        ceoslp(7, eosPb) = 147.3_sdk
        ceoslp(9, eosPb) = 11408.0_sdk
        ceoslp(10, eosPb) = HeV(ceoslp(5, eosPb), eosPb)
        ceoslp(11, eosPb) = 100000.0_sdk
        ceoslp(12, eosPb) = gasCon / vapmol
        ceoslp(14, eosPb) = 0.65141_sdk
        ceoslp(15, eosPb) = 0.0_sdk
        ceoslp(16, eosPb) = 1.3_sdk
        ceoslp(20, eosPb) = 9.056466d4
        ceoslp(21, eosPb) = 600.6_sdk
        ceoslp(24, eosPb) = ceoslp(7, eosPb)
        ceoslp(29, eosPb) = 600.6_sdk
        ceoslp(30, eosPb) = 1000.0_sdk
        ceoslp(31, eosPb) = 450.0d6
```

```
      ceoslp(32, eosPb) = 600.6_sdk
      ceoslp(33, eosPb) = 2000.0_sdk
      ceoslp(34, eosPb) = 600.6_sdk
      ceoslp(35, eosPb) = 2000.0_sdk
      ceoslp(36, eosPb) = 1000.0_sdk
      ceoslp(37, eosPb) = 544.6d5
      ceoslp(38, eosPb) = 5400.0_sdk
      ceoslp(39, eosPb) = 139.69971285053d5
      ceoslp(40, eosPb) = 609.62462615967_sdk
!     ceoslp(28) = (ceoslp(12)-ceoslp(25))/ceoslp(12)
!
      ceoslp(4, eosPb) = ceoslp(12, eosPb) / (ceoslp(16, eosPb)-1.0_sdk)
      ceoslp(23, eosPb) = ceoslp(16, eosPb) * ceoslp(4, eosPb)
      ceoslp(8, eosPb) = - 611.2_sdk * 0.0010002_sdk + ceoslp(7, eosPb) * &
     &(ceoslp(5, eosPb)-ceoslp(29, eosPb))
      ceoslp(26, eosPb) = ceoslp(24, eosPb) * (ceoslp(5, eosPb)-ceoslp(29, eosPb))
      ceoslp(27, eosPb) = ceoslp(26, eosPb) + ceoslp(10, eosPb)
      ceoslp(6, eosPb) = ceoslp(27, eosPb) - ceoslp(12, eosPb) * ceoslp(5, eosPb)
!
    END SUBROUTINE SetEoPb
```

Modul EosM

```
    SUBROUTINE ThermPb(jstart, jstop, ccoThermPb)
!
      USE Io
      USE EosNCGData
      USE Global, ONLY: cco
      USE GlobalDat, ONLY: iftp, numberOfNCGases
      USE Flt
!
      IMPLICIT NONE
!
!     subroutine ThermPb evaluates the thermodynamic properties of Pb
!
!     input variables
!       1. p      pressure
!       2. tl     liquid temperature
!       3. tv     vapor temperature
!       4. pa     partial pressure of the noncondensible
!
!     output variables
!       1. el     liquid internal energy
!       2. ev     vapor (steam and noncondensable mixture) internal
!                 energy
!       3. tsat   saturation temperature corresponding to the total
!                 pressure
!       4. rol    liquid density
!       5. rov    vapor (steam and noncondensable mixture) density
!       6. rova   density of the noncondensable
```

```
!       7. tssn   saturation temperature corresponding to the steam
!                 partial pressure
!       8. eva    internal energy of the noncondensable
!       9. dtsdp  derivative of tsat wrt pressure
!      10. deldp  derivative of el wrt pressure
!      11. devdp  derivative of steam internal energy wrt steam
!                 partial pressure
!      12. deldt  derivative of el wrt tl
!      13. devdt  derivative of steam internal energy wrt tv
!      14. drolp  derivative of rol wrt pressure
!      15. drovp  derivative of steam density wrt steam partial
!                 pressure
!      16. drolt  derivative of rol wrt tl
!      17. drovt  derivative of steam density wrt tv
!      18. hvst   saturated steam enthalpy (psteam,tssn)
!      19. hlst   saturated liquid enthalpy (p,tssn)
!      20. dhvsp  derivative of hvst wrt steam partial pressure
!      21. dhlsp  derivative of hlst wrt pressure
!      22. dtssp  derivative of tssn wrt steam partial pressure
!      23. devat  derivative of eva wrt tv
!      24. devap  derivative of eva wrt pa
!      25. drvap  derivative of rova wrt pa
!      26. drvat  derivative of rova wrt tv
!
       INTEGER(sik), INTENT(IN) :: jstart, jstop
       INTEGER(sik), INTENT(IN), OPTIONAL :: ccoThermPb
       INTEGER(sik) :: ie, j, idx, n
!
       REAL(sdk) pt, tl2, tv1, pg, ps, rolst, rrolst, elsat, delsat, drlsdt, drlsdp
       REAL(sdk) rGasNC, rGasPb
       REAL(sdk), DIMENSION(numberOfNCGases) :: dRGasMixturedXgn, eNCGas
!
       ie = eosPb
       phaseChange = .FALSE.
       IF (PRESENT(ccoThermPb)) THEN
         idx = ccoThermPb
       ELSE
         idx = cco
       END IF
       DO j = jstart, jstop
!
         pt = p(j)
         pa(j) = 0.0_sdk
         pg = 0.0_sdk
         IF (p(j) < ceoslp(30, ie) .OR. p(j) > ceoslp(31, ie)) THEN
           pt = MIN(ceoslp(31, ie), MAX(ceoslp(30, ie), p(j)))
           IF (iftp /= 1) THEN
             CALL error(2, '*thermPb* pressure limit exceeded')
             WRITE (imout, 18) genTab(idx)%num, j
18             FORMAT (15 x, 'component', i4, ', ', 'cell', i4)
```

```
          END IF
        END IF
!
!   calculate saturation properties
!
        ps = pt - pg
        ps = MAX(ps, ceoslp(30, ie))
        tsat(j) = SatTmPb(pt)
        dtsdp(j) = SatDePb(pt)
        tl2 = tl(j)
        tv1 = tv(j)
        tssn(j) = tsat(j)
        dtssp(j) = dtsdp(j)
!
        tl2 = MIN(ceoslp(33, ie), MAX(tl2, ceoslp(32, ie)))
        tv1 = MIN(ceoslp(35, ie), MAX(tv1, ceoslp(34, ie)))
!
!   calculate liquid properties
!
!     1. internal energy and its derivatives
!
        el(j) = ceoslp(7, ie) * (tl2-ceoslp(5, ie)) + ceoslp(8, ie)
        deldt(j) = ceoslp(7, ie)
        deldp(j) = 0.0_sdk
        elsat = ceoslp(7, ie) * (tssn(j)-ceoslp(5, ie)) + ceoslp(8, ie)
        delsat = ceoslp(7, ie)
!
        CALL RhoLiPb(pt, tssn(j), rolst, drlsdp, drlsdt)
!
        rrolst = 1.0_sdk / rolst
        hlst(j) = elsat + pt * rrolst
        dhlsp(j) = delsat * dtsdp(j) + rrolst - pt * rrolst * rrolst * &
       &(drlsdt*dtsdp(j)+drlsdp)
!
!     2. density and its derivatives
!
        CALL RhoLiPb(pt, tl2, rol(j), drolp(j), drolt(j))
!
!   calculate vapor properties
!
!   -----internal energy, enthalpy, and their derivatives
        hvst(j) = ceoslp(16, ie) * ceoslp(23, ie) * (tv1-ceoslp(26, ie))&
       & + ceoslp(6, ie)
        dhvsp(j) = 0.0_sdk
!
!   calculate noncondensable gas properties
!
        IF (iGas > 10_sik) THEN
          rGasNC = RGasMixture(xgn(1:numberOfNCGases, j), dRGasMixturedXgn)
          eva(j) = EnergyMixture(xgn(1:numberOfNCGases, j), tv(j), eNCGas)
```

```
            devat(j) = CvMixture(xgn(1:numberOfNCGases, j), tv(j))
          devap(j) = 0.0_sdk
            drvap(j) = 1.0_sdk / (rGasNC * tv(j))
          rova(j) = drvap(j) * pa(j)
            drvat(j) =-rGasNC * rova(j) * drvap(j)
            DO n = 1, numberOfNCGases
              drvax(n, j) =-pa(j) * dRGasMixturedXgn(n)/ (rGasNC**2 * tv(j))
              devax(n, j) = eNCGas(n) - eNCGas(1)
            END DO
          ELSE
            rGasNC = ncGasProperties(iGas)%rGas
            eva(j) = EnergyNCGas(ncGasProperties(iGas)%cvFit, tv1)
            devat(j) = CvFit(ncGasProperties(iGas)%cvFit, tv1)
            devap(j) = 0.0_sdk
            drvap(j) = 1.0_sdk / (rGasNC*tv1)
            rova(j) = drvap(j) * pa(j)
            drvat(j) =-rGasNC * rova(j) * drvap(j)
          END IF
!
!         Calculate non-condensable vapor properties.
          rGasPb = rGasNC
          ev(j) = eva(j)
          devdt(j) = devap(j)
          devdp(j) = 0.0_sdk
          drovp(j) = drvap(j)
          rov(j) = drvap(j) * ps
          drovt(j) = drvat(j)
!
!     calculate air-steam mixture properties
!
          ev(j) = (ev(j)*rov(j)+eva(j)*rova(j))
          rov(j) = rov(j) + rova(j)
          ev(j) = ev(j) / rov(j)
          p(j) = pt
          tl(j) = tl2
          tv(j) = tv1
!
!    Calculate cpLiq and cpVap because everything is available now to calculate it.
!           cpLiq(j) = deldt(j) - p(j) * drolt(j) / (rol(j)**2)
!           cpVap(j) = CpvvPb(tv(j), p(j), pa(j), xgn(1:, j))
        END DO
!
      END SUBROUTINE ThermPb

      SUBROUTINE RhoLiPb(p, tl, rhol, drldp, drldt)
!
        IMPLICIT NONE
!
!     subroutine RhoLiPb evaluates the density of Pb liquid and its
!     derivatives with respect to total pressure and liquid temper-
```

```
!     ature as a function of total pressure and liquid temperature.
!
!     drldp is based on a sound speed of 1790 m/s.
!
!        total pressure             p      in (pa)
!        liquid temperature         tl     in (k)
!        liquid density             rol    in (kg/m**3)
!        drol/dp                    drldp  in (kg/m**3/pa)
!        drol/dt                    drldt  in (kg/m**3/k)
!
        REAL(sdk), INTENT(IN) :: p, tl
        REAL(sdk), INTENT(OUT) :: rhol, drldp, drldt
!
        drldt = - 1.2272_sdk
        drldp = 1 / ((1790 - (0.5 * (tl - 600.6)))**2)
        rhol = ceoslp(9, eosPb) + drldt * (tl) + drldp * (p)
!
      END SUBROUTINE RhoLiPb

      REAL(sdkx) FUNCTION SatDePb(pres)
!
        IMPLICIT NONE
!
!     function SatDePb evaluates the derivative of the Pb saturation
!     temperature with respect to total pressure as a function
!     of the saturation pressure.
!
!        pres                    pressure (pa)
!        SatDePb                 dtsat/dp in (k/pa)
!
        REAL(sdk), INTENT(IN) :: pres
!
        SatDePb = ceoslp(3, eosPb) / (pres*(LOG10(pres)-ceoslp(2, eosPb))**2)
!
      END FUNCTION SatDePb

      REAL(sdkx) FUNCTION SatPrPb(temp)
!
        IMPLICIT NONE
!
!     function SatPrPb evaluates the Pb saturation pressure
!     as a function of the saturation temperature
!
!        saturation temperature     temp   in (k)
!        saturation pressure        SatPrPb in (pa)
!
        REAL(sdk), INTENT(IN) :: temp
!
        SatPrPb = 10.0_sdk ** (ceoslp(2, eosPb)-ceoslp(1, eosPb)/temp)
        SatPrPb = MAX(SatPrPb, ceoslp(30, eosPb))
```

```
!
    END FUNCTION SatPrPb



    REAL(sdkx) FUNCTION SatTmPb(pres)
!
      IMPLICIT NONE
!
!   function SatTmPb evaluates the Pb saturation temperature
!   as a function of the saturation pressure.
!   Boiling Point of Pb = 1740 C.
!   Use correlation log10(pres) = -0.05223*A/T + B
!   where A = 184,901.52 and B = 7.70232 + log10(133.32) for Pb
!   133.32 Pa per mm of Hg.
!
!       saturation pressure        pres      in (pa)
!       saturation temperature     SatTmPb in (k)
!
      REAL(sdk), INTENT(IN) :: pres
!
      SatTmPb = ceoslp(1, eosPb) / (ceoslp(2, eosPb)-LOG10(pres))
!
    END FUNCTION SatTmPb

    REAL(sdkx) FUNCTION ViscLPb(tl)
!
      IMPLICIT NONE
!
!   function ViscLPb evaluates the Pb liquid dynamic viscosity
!
!       liquid temperature         tl     in (K)
!
      REAL(sdk), INTENT(IN) :: tl
!
      ViscLPb = 4.636e-4_sdk * EXP(1036.0_sdk / tl)
!
    END FUNCTION ViscLPb

    REAL(sdkx) FUNCTION ViscVPb(tv, p, pa, xg)
!
      USE GlobalDat
      IMPLICIT NONE

!   function ViscVPb evaluates the Pb vapor dynamic viscosity
!
!       vapor viscosity            ViscVPb  in (pa*s)
!
      REAL(sdk), INTENT(IN) :: p, pa, tv
      REAL(sdk), DIMENSION(:), INTENT(IN) :: xg
```

```
      REAL(sdk) :: f
!
      ViscVPb = 2.0e-04_sdk
      IF (pa > paSign) THEN
        f = MIN(1.0_sdk, pa/p)
        ViscVPb = (1.0_sdk-f) * ViscVPb + f * ViscNC(tv, xg)
      END IF
!
    END FUNCTION ViscVPb

    REAL(sdkx) FUNCTION HeVPb(temp)
!
      IMPLICIT NONE
!
!   function HeVPb calculates the heat of evaporation of Pb liquid
!   as a function of liquid temperature for low pressures
!
!       liquid temperature          temp   in (k)
!       heat of evaporation         HeVPb    in (j/kg)
!
      REAL(sdk), INTENT(IN) :: temp
!
      HeVPb = HeVH(temp)
!
    END FUNCTION HeVPb

   SUBROUTINE FPropPb(p, tl, tv, tsat, hfg, cpl, cpv, visl, visv, cl, cv, sig,&
  &nc, pa, hls, hvs, xg)
!
      USE GlobalDat, ONLY: numberOfNCGases
!
      IMPLICIT NONE
!
!   subroutine FPropPb evaluates the Pb fluid enthalpy,
!   heat of vaporization, specific heat, viscosity,
!   thermal conductivity, and surface tension
!
      INTEGER(sik), INTENT(IN) :: nc
      INTEGER(sik) i
!
      REAL(sdk), DIMENSION(:), INTENT(IN) :: p, pa, tl, tv, hvs, hls, tsat
      REAL(sdk), DIMENSION(:), INTENT(OUT) :: hfg, cpl, cpv, visl, visv, cl, cv, sig
      REAL(sdk), DIMENSION(:, :), POINTER :: xg
      REAL(sdk), DIMENSION(:), POINTER :: xgi
!
      DO i = 1, nc
        IF (iGas > 10_sik) THEN
          xgi => xg(1:numberOfNCGases, i)
        ELSE
          xgi => xg(1:, i)
```

```
        END IF
!
        hfg(i) = hvs(i) - hls(i)
!
        cpl(i) = CpllPb(tl(i))
        cpv(i) = CpvvPb(tv(i), p(i), pa(i), xgi)
!
        visl(i) = ViscLPb(tl(i))
        visv(i) = ViscVPb(tv(i), p(i), pa(i), xgi)
!
        cl(i) = ThcLPb(tl(i))
        cv(i) = ThcVPb(tv(i), p(i), pa(i), xgi)
!
        sig(i) = SigmaPb(tsat(i))
!
      END DO
!
    END SUBROUTINE FPropPb

    REAL(sdkx) FUNCTION ThcLPb(tl)
!
      IMPLICIT NONE
!
!   function ThcLPb evaluates the Pb liquid thermal conductivity.
!
!       liquid temperature          tl     in (k)
!       thermal conductivity        ThcLPb   in (w/m/k)
!
      REAL(sdk), INTENT(IN) :: tl
!
      ThcLPb = 9.9855_sdk + 0.0101_sdk * tl
!
    END FUNCTION ThcLPb

    REAL(sdkx) FUNCTION ThcVPb(tv, p, pa, xg)
!
      IMPLICIT NONE
!
!   function ThcVPb evaluates the vapor thermal conductivity
!
!     thermal conductivity ThcVPb in (w/m k)
!
      REAL(sdk), INTENT(IN) :: tv, p, pa
      REAL(sdk), DIMENSION(:), INTENT(IN) :: xg
!
      IF (pa > paSign) THEN
        ThcVPb = ((p - pa)  + pa * ThcNCG(tv, xg, eosPb)) / p
      ELSE
      ThcVPb = 1.0_sdk
      END IF
```

```
!
      END FUNCTION ThcVPb



      REAL(sdkx) FUNCTION SigmaPb(tl)
!
        IMPLICIT NONE
!
!     surface tension of Pb.
!
!     liquid temperature        tl       in K
!     surface tension           SigmaPb  in N/m
!
        REAL(sdk), INTENT(IN) :: tl
!
      SigmaPb = 0.523_sdk - 1.09E-4_sdk * tl
!
      END FUNCTION SigmaPb

      REAL(sdkx) FUNCTION CpllPb(tl)
!
       IMPLICIT NONE
!
!     function CpllPb evaluates the specific heat of Pb liquid
!
!        liquid temperature          tl         in (K)
!        liquid specific heat        CpllPb   in (j/kg/k)
!
        REAL(sdk), INTENT(IN) :: tl
!
        CpllPb = 1.133E-06_sdk * tl ** 2 - 3.859E-02_sdk * tl + 170.00_sdk
!
      END FUNCTION CpllPb

      REAL(sdkx) FUNCTION CpvvPb(tv, p, pa, xg)
!
        IMPLICIT NONE
!
!     function CpvvPb evaluates the specific heat of Pb vapor
!     as a function of vapor temperature, total pressure, and
!     noncondensable-gas pressure
!
!        vapor specific heat         CpvvPb  in (j/kg/k)
!
        REAL(sdk), INTENT(IN) :: tv, p, pa
        REAL(sdk), DIMENSION(:), INTENT(IN) :: xg
!
        IF (pa > paSign) THEN
          CpvvPb = ((p - pa) * ceoslp(22, eosPb) + pa * CpvNCG(tv, xg, eosPb)) / p
```

```
      ELSE
        CpvvPb = ceoslp(22, eosPb)
      END IF
!
    END FUNCTION CpvvPb
```

A.1.4 LEAD-BISMUTH-EUTECTICS

Modul EosInitM

```
   SUBROUTINE SetEoPbBi
!
      USE Eos, ONLY: HeV
      USE EosData
      USE EosNCGData, ONLY: SetCeoslpNCG
      !USE GlobalDat, ONLY: gasCon
!
      IMPLICIT NONE
!
      REAL(sdk) vapmol
!
!   subroutine SetEoPbBi initializes the PbBi equation-of-state constants
!
!   vapmol     = molecular weight of vapor
!
      aeos14(eosPbBi) = 1.0d-05
      ceos1(eosPbBi) = 117.8_sdk
      ceos2(eosPbBi) = 0.223_sdk
      ceos3(eosPbBi) = 255.2_sdk
!
!     Non-condensable gas is defined by iGas.
      CALL SetCeoslpNCG(iGas, eosPbBi)
!
      vapmol = lbeMWt
!
      ceoslp(1, eosPbBi) = 9923.19_sdk ! = 0.05223*189,990 for PbBi
                           !(see function SatTmPbBi).
      ceoslp(2, eosPbBi) = 10.163_sdk ! = 8.038 + log10(133.32),
                           !where 133.32 Pa = 1 mm of Hg.
      ceoslp(3, eosPbBi) = LOG10(EXP(1.0_sdk)) * ceoslp(1, eosPbBi)! = 4309.59
      ceoslp(5, eosPbBi) = 398.5_sdk
      ceoslp(7, eosPbBi) = 145.1_sdk
      ceoslp(9, eosPbBi) = 11105.0_sdk
      ceoslp(10, eosPbBi) = HeV(ceoslp(5, eosPbBi), eosPbBi)
      ceoslp(11, eosPbBi) = 100000.0_sdk
      ceoslp(12, eosPbBi) = gasCon / vapmol
      ceoslp(14, eosPbBi) = 0.65141_sdk
      ceoslp(15, eosPbBi) = 0.0_sdk
      ceoslp(16, eosPbBi) = 1.3_sdk
```

```
        ceoslp(20, eosPbBi) = 9.056466d4
        ceoslp(21, eosPbBi) = 398.5_sdk
        ceoslp(24, eosPbBi) = ceoslp(7, eosPbBi)
        ceoslp(29, eosPbBi) = 398.5_sdk
        ceoslp(30, eosPbBi) = 1000.0_sdk
        ceoslp(31, eosPbBi) = 450.0d5
        ceoslp(32, eosPbBi) = 398.5_sdk
        ceoslp(33, eosPbBi) = 3000.0_sdk
        ceoslp(34, eosPbBi) = 398.5_sdk
        ceoslp(35, eosPbBi) = 3000.0_sdk
        ceoslp(36, eosPbBi) = 1000.0_sdk
        ceoslp(37, eosPbBi) = 221.2d5
        ceoslp(38, eosPbBi) = 647.3_sdk
        ceoslp(39, eosPbBi) = 139.69971285053d5
        ceoslp(40, eosPbBi) = 609.62462615967_sdk
!     ceoslp(28) = (ceoslp(12)-ceoslp(25))/ceoslp(12)
!
        ceoslp(4, eosPbBi) = ceoslp(12, eosPbBi) / (ceoslp(16, eosPbBi)-1.0_sdk)
        ceoslp(23, eosPbBi) = ceoslp(16, eosPbBi) * ceoslp(4, eosPbBi)
        ceoslp(8, eosPbBi) = - 611.2_sdk * 0.0010002_sdk + ceoslp(7, eosPbBi) * &
       &(ceoslp(5, eosPbBi)-ceoslp(29, eosPbBi))
        ceoslp(26, eosPbBi) = ceoslp(24, eosPbBi) * (ceoslp(5, eosPbBi)-&
       &ceoslp(29, eosPbBi))
        ceoslp(27, eosPbBi) = ceoslp(26, eosPbBi) + ceoslp(10, eosPbBi)
        ceoslp(6, eosPbBi) = ceoslp(27, eosPbBi) - ceoslp(12, eosPbBi) &
       &* ceoslp(5, eosPbBi)
!
     END SUBROUTINE SetEoPbBi
```

Modul EosM

```
    SUBROUTINE RhoLiPbBi(p, tl, rhol, drldp, drldt)
       IMPLICIT NONE
!
!    subroutine RhoLiPbBi evaluates the density of PbBi liquid and its
!     derivatives with respect to total pressure and liquid temper-
!     ature as a function of total pressure and liquid temperature.
!
!   drldp is based on a sound speed of 1600 m/s.
!   drldt is an estimate from available data IPPE (Russia) Report.
!
!        total pressure             p      in (pa)
!        liquid temperature         tl     in (k)
!        liquid density             rol    in (kg/m**3)
!        drol/dp                   drldp  in (kg/m**3/pa)
!        drol/dt                   drldt  in (kg/m**3/k)
!
       REAL(sdk), INTENT(IN) :: p, tl
       REAL(sdk), INTENT(OUT) :: rhol, drldp, drldt
!
```

```
      drldt = - 1.3312_sdk
      drldp = 1 / ((1600 - (0.5 * (tl - 398)))**2)
      rhol = ceoslp(9, eosPbBi) + drldt * (tl) + drldp * (p)
!
    END SUBROUTINE RhoLiPbBi

    REAL(sdkx) FUNCTION ViscLPbBi(tl)
!
      IMPLICIT NONE
!
!   function ViscLPbBi evaluates the PbBi liquid dynamic viscosity
!
!       liquid temperature          tl     in (K)
!
      REAL(sdk), INTENT(IN) :: tl

      ViscLPbBi = 5.293e-4_sdk * EXP(732.3_sdk / tl)
!
    END FUNCTION ViscLPbBi

    REAL(sdkx) FUNCTION ThcLPbBi(tl)
!
      IMPLICIT NONE
!
!   function ThcLPbBi evaluates the PbBi liquid thermal conductivity.
!
!       liquid temperature          tl     in (k)
!       thermal conductivity        ThcLPbBi   in (w/m/k)
!
      REAL(sdk), INTENT(IN) :: tl
!
!
      ThcLPbBi = 5.3557_sdk + 1.181E-02_sdk * tl
!
    END FUNCTION ThcLPbBi

    REAL(sdkx) FUNCTION SigmaPbBi(tl)
!
      IMPLICIT NONE
!
!   surface tension of PbBi.
!
      REAL(sdk), INTENT(IN) :: tl
!
!    Correlation taken from Wadim Jaeger, PhD canditate, FZK/IRS
!
    SigmaPbBi = 0.405_sdk - 6.78E-5_sdk * tl
!
    END FUNCTION SigmaPbBi
```

```
    REAL(sdkx) FUNCTION CpllPbBi(tl)
!
      IMPLICIT NONE
!
!   function CpllPbBi evaluates the specific heat of PbBi liquid
!
!       liquid temperature         tl        in (K)
!       liquid specific heat       CpllPbBi  in (j/kg/k)
!
      REAL(sdk), INTENT(IN) :: tl
!
      CpllPbBi = 145.0_sdk - 9.97E-06_sdk * tl ** 2 + 7.622E-03_sdk * tl
!
    END FUNCTION CpllPbBi
```

A.2 HEAT TRANSFER

A.2.1 COMMON ROUTINES

Modul RefloodM

```
    PRIVATE :: LiqMetalNu       ! Get Nu-number for liquid metal HT.
    PRIVATE :: LiqMetalNuBundle  ! Get Nu-number for liquid metal HT in bundle array.
    PRIVATE :: DTHTNu           ! Get Nu-number for Diphyl THT.

    REAL(sdk) FUNCTION GetLiqHTC(tWallLocal)

      USE Eos, ONLY : eosLBE, eosPbBi, eosNa, eosDTHT

!   Purpose: to obtain wall heat transfer coefficient for liquid phase
!         in forced convection.  The models used below are appropriate
!         for single-phase liquid as well as bubbly-slug flow.  However,
!         an adjustment to the characteristic diameter is needed for
!         the case of annular flow.
!
      Implicit NONE
      REAL(sdk), INTENT(IN) :: tWallLocal  ! Wall temperature
      REAL(sdk) :: Nu
      REAL(sdk) :: NuLaminar
      REAL(sdk) :: NuTurbulent
      REAL(sdk) :: NuNC
      REAL(sdk) :: Re
      REAL(sdk) :: Pr
      REAL(sdk) :: NussSRCSelect

!     Single-Phase liquid Reynolds and Prandtl No.

      Re = ABS(gliq) * hydroDiam / visl
```

```
      Re = MAX(one, Re)
      Pr = cpl * visl / cndctvtyLiq
!
!     Apply two-phase correction to Reynolds No., limited to avoid
!     unrealistically high values.

      Re = Re / MAX(0.2_sdk, one - alpha)
!
      IF (EosType == eosLBE .OR. EosType == eosPbBi .OR. EosType == eosNa) THEN
         IF (isTube) THEN !Liquid metal single phase HT.
            Re = MAX (Re, 1.0e-20_sdk)
            Nu = LiqMetalNu(Re, Pr)
         ELSE  !Rod Bundle Geometry
            Nu = LiqMetalNuBundle(Re,Pr)
         END IF
      END IF
      IF (EosType == eosDTHT) THEN
         Nu = DTHTNu(Re, Pr)
      END IF
      IF (EosType .ne. eosDTHT .AND. EosType &
     &.ne. eosLBE .AND. EosType .ne. eosPbBi .AND. EosType .ne. eosNa) THEN
        IF (isTube) THEN
          NuLaminar = NuForcedConvLamTube()
          NuTurbulent = Gnielinski(Re, Pr)
          NuNC = NuNCLiq(Pr, tWallLocal)
        ELSE  !Rod Bundle Geometry
          NuLaminar = NuForcedConvLamRod(Re, Pr)
          NuTurbulent = NuForcedConvTurbRod(Re, Pr)
          NuNC = NuNCLiqRod(Pr, tWallLocal)
        END IF
          NuTurbulent = VarLiqPropEffct(Pr, tWallLocal) * NuTurbulent
          Nu = MAX( NuLaminar, NuTurbulent, NuNC)
      END IF

      GetLiqHTC = Nu * cndctvtyLiq / hydroDiam

      RETURN
    END FUNCTION GetLiqHTC
```

A.2.2 DIPHYL THT

Modul RefloodM

```
    REAL(sdk) FUNCTION DTHTNu(Re,Pr)

!   Gnielinski correlation for helical coiled tubes

      REAL(sdk), INTENT(in) :: re  ! Reynolds number
      REAL(sdk), INTENT(in) :: pr  ! Prandtl number
```

```
      REAL(sdk) :: f            ! Coefficient
      REAL(sdk) :: R             ! radius of curvature
      REAL(sdk) :: D             ! Diameter

      R = 0.0235_sdk
      D = 2.0_sdk * R

!     CALL SetLiqProps(visl, cpl, cl, h, ro)
!     CALL SetLiqProps(vislw, cplw, clw, hw, row)

      IF (Re <= (13.5_sdk * ((D / hydroDiam)**(-0.5_sdk)))) THEN

        f = 16.0_sdk / Re
        dTHTNu  = (((f / 2.0_sdk) * Re * Pr ) / (1.0_sdk + ((12.7_sdk * &
       &((f / 2.0_sdk)**0.5_sdk)) * ((Pr**0.667_sdk) - 1.0_sdk)))) * 1.03_sdk

      ELSE IF (Re >= (13.5_sdk * ((D / hydroDiam)**(-0.5_sdk))) .AND. &
      &Re <= 15000.0_sdk) THEN

        f = (344.0_sdk * ((D / hydroDiam)**(-0.5_sdk))) / ((1.56_sdk + &
       &LOG10(Re * 0.285_sdk))**5.73_sdk)
        dTHTNu  = (((f / 2.0_sdk) * Re * Pr ) / (1.0_sdk + ((12.7_sdk * &
        &((f / 2.0_sdk)**0.5_sdk)) * ((Pr**0.667_sdk) - 1.0_sdk)))) * 1.03_sdk

      ELSE

        f = (0.076_sdk * Re**(-0.25_sdk)) + (0.0075_sdk * ((D / &
       &hydroDiam)**(-0.5_sdk)))
        dTHTNu  = (((f / 2.0_sdk) * Re * Pr ) / (1.0_sdk + ((12.7_sdk * &
       &((f / 2.0_sdk)**0.5_sdk)) * ((Pr**0.667_sdk) - 1.0_sdk)))) * 1.03_sdk

      END IF
!
    END FUNCTION DTHTNu
```

A.2.3 LIQUID METALS

Modul RefloodM

```
      USE Eos, ONLY : eosPbBi, eosLBE

      ELSE IF(EosType == eosPbBi .OR. EosType == eosLBE) THEN

        whtReg = forcedConvLiq
        hlWall = GetLiqHTC(tWall)

    REAL(sdk) FUNCTION LiqMetalNu(Re,Pr)
!
!   Purpose: to obtain wall heat transfer coefficient for liquid metals
```

```
!
      USE HTPar, ONLY: remHiCondLM, remLowCondLM, liqMetalCondNu

      Implicit NONE
!
      REAL(sdk), INTENT(in) :: re  ! Reynolds number
      REAL(sdk), INTENT(in) :: pr  ! Prandtl number
      REAL(sdk) :: LiqMetalNuLow, LiqMetalNuHi
!
      IF (re > remHiCondLM) THEN
!
!     turbulent forced convection htc for liquid metal.
        liqMetalNu = 5.0_sdk + 0.025_sdk * (re*pr) ** 0.8_sdk
!
      ELSE IF (re < remLowCondLM) THEN
!
!     conduction htc for liquid metal across a thin gap.
        liqMetalNu = liqMetalCondNu
!
      ELSE
!
!     Transition for liquid metal HTC.
        liqMetalNuHi = 5.0_sdk + 0.025_sdk * (re*pr) ** 0.8_sdk
        liqMetalNuLow = liqMetalCondNu
        liqMetalNu = liqMetalNuLow + (liqMetalNuHi-liqMetalNuLow) * &
       &(re-remLowCondLM) / (remHiCondLM-remLowCondLM)
!
      END IF
!
      RETURN
!
    END FUNCTION LiqMetalNu

    REAL(sdk) FUNCTION LiqMetalNuBundle(Re,Pr)
!
!     Purpose: to obtain wall heat transfer coefficient for liquid metals
!
      REAL(sdk), INTENT(in) :: re  ! Reynolds number
      REAL(sdk), INTENT(in) :: pr  ! Prandtl number
!
      liqMetalNuBundle = 7.55_sdk * 1.3238_sdk - 20.0_sdk * (1.3238_sdk ** &
     &(-13.0_sdk)) + (3.67_sdk/90.0_sdk) * (1.3238_sdk ** (-2.0_sdk)) * &
     &(re*pr) ** (0.56_sdk + 0.19_sdk * 1.3238_sdk)
!
    END FUNCTION LiqMetalNuBundle
```

NRC FORM 335
(9-2004)
NRCMD 3.7

U.S. NUCLEAR REGULATORY COMMISSION

1. REPORT NUMBER
(Assigned by NRC, Add Vol., Supp., Rev., and Addendum Numbers, if any.)
NUREG/IA-0421

BIBLIOGRAPHIC DATA SHEET

(See instructions on the reverse)

2. TITLE AND SUBTITLE
Improvements and Validation of the System Code TRACE for Lead and Lead-Alloy Cooled Fast Reactors Safety-Related Investigations

3. DATE REPORT PUBLISHED
MONTH: February
YEAR: 2013

4. FIN OR GRANT NUMBER

5. AUTHOR(S)
Wadim Jaeger, Victor Hugo Sanchez Espinoza

6. TYPE OF REPORT
Technical

7. PERIOD COVERED *(Inclusive Dates)*

8. PERFORMING ORGANIZATION - NAME AND ADDRESS *(If NRC, provide Division, Office or Region, U.S. Nuclear Regulatory Commission, and mailing address; if contractor, provide name and mailing address.)*
Karlsruher Institute of Technology (KIT)
Institute for Neutron Physics and Reactor Technology (INR)
Hermann-von-Helmholtz-Platz 1
76344, Eggenstein-Leopoldshafen, Germany

9. SPONSORING ORGANIZATION - NAME AND ADDRESS *(If NRC, type "Same as above"; if contractor, provide NRC Division, Office or Region, U.S. Nuclear Regulatory Commission, and mailing address.)*
Division of Systems Analysis
Office of Nuclear Regulatory Research
U.S. Nuclear Regulatory Commission
Washington, DC 20555-0001

10. SUPPLEMENTARY NOTES
A. Calvo, NRC Project Manager

11. ABSTRACT *(200 words or less)*
This report deals with the validation of the system code TRACE related to lead and lead-alloy cooled nuclear systems. This validation process made in necessary to revise routines of the TRACE source code in order to make lead as a coolant available, and to ad functions for heat transfer in bundles cooled with liquid metals. Several experimental as well as theoretical benchmarks were selected to test, on the one hand side, the existing physical models related to liquid metals, and, on the other hand side, the new implemented functions and routines. The results for the different investigations are in good agreement to the experimental data or the other codes. The present analysis showed that the revision of the TRACE models was necessary and that the changes yield better results.

12. KEY WORDS/DESCRIPTORS *(List words or phrases that will assist researchers in locating the report.)*
Karlsruher Institute of Technology (KIT)
Thermal-hydraulic
TRACE
LWRs
Safety-related investigations of lead cooled fast reactor systems for transmutation of waste
Sub-critical and critical lead cooled fast systems
LACANES benchmark
XADS target
CHEOPE facility
MEGAPIE project

13. AVAILABILITY STATEMENT
unlimited

14. SECURITY CLASSIFICATION
(This Page)
unclassified
(This Report)
unclassified

15. NUMBER OF PAGES

16. PRICE

Federal Recycling Program

UNITED STATES
NUCLEAR REGULATORY COMMISSION
WASHINGTON, DC 20555-0001

OFFICIAL BUSINESS

NUREG/IA-0421

Improvements and Validation of the System Code TRACE for Lead and Lead-Alloy Cooled Fast Reactors Safety-Related Investigations

February 2013