TREND User Manual

Version 5.0 – December 2020

Cite as: Span, R.; Beckmüller, R.; Hielscher, S.; Jäger, A.; Mickoleit, E.; Neumann, T.; Pohl, S. M.; Semrau, B.; Thol, M. (2020):
TREND. Thermodynamic Reference and Engineering Data 5.0.
Lehrstuhl für Thermodynamik, Ruhr-Universität Bochum.

Acknowledgements

Contents

1 INTRODUCTION

License: This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

This software package has been developed as part of the research in thermodynamic property models of the thermodynamics group at the Ruhr University Bochum. It is written in FORTRAN 95, using the Intel Visual Fortran compiler XE (Version 12 or higher) for debugging and compilation. It is subject to ongoing development and will incorporate extended functionality in future versions. The thermodynamic properties are mainly calculated using highly accurate equations of state (EOS) explicit in the Helmholtz free energy. More information on the general structure of Helmholtz-type equations of state can be found in [1]. For mixtures, the non-ideal interaction of fluids is accounted for using mixture models in the way they were developed for the GERG-2004 EOS [2]. Other equation models, e.g., PR, SRK, or LKP, are also available.

We appreciate any comments, bug reports, or suggestions for improvement. Please contact us through: TREND@thermo.rub.de

2 SETUP AND INSTALLATION

The TREND software package includes the following files and folders:

Table 1. Folders and files of the TREND software package.

File/Folder	Comment
TREND.xlsx	Example spreadsheet including the available Excel functions. More details can be found in the TREND Excel Manual.
TREND.xlam	Excel Addin file, which can be implemented in MS Excel for the property calculation in Excel spreadsheets. This Addin calls the TREND.DLL.
TREND_Win32.DLL	Dynamic link library compiled for Win32 (also works in x64 Windows) platforms including the calculation routines for the thermodynamic property calculation.
TREND_x64.DLL	Dynamic link library compiled for Win 64 bit applications.
FLUIDS	FOLDER with .FLD files for all fluids that can be calculated by the software package. Each .FLD file contains the parameters of the Helmholtz EOS, transport properties, general fluid information, and the corresponding references.
BINARY_MIX_FILES	FOLDER with .MIX files containing binary interaction parameters for the calculation of mixtures and the corresponding references.
SRK	FOLDER containing the FOLDERS "BINARY_SRK" and "FLUIDS_SRK". These folders contain all necessary parameters for calculations with the Soave-Redlich-Kwong equation of state [13].
PR	FOLDER containing the FOLDERS "BINARY_PR" and "FLUIDS_PR". These folders contain all necessary parameters for calculations with the Peng Robinson equation of state [15].
LKP	FOLDER containing the FOLDERS "BINARY_LKP" and "FLUIDS_LKP". These folders contain all necessary parameters for calculations with the Lee-Kesler-Plöcker equation of state [14].
GEN_EQ	FOLDER containing the FOLDER "FLUIDS_GEN_EQ". This folder contains all necessary parameters for calculations with the generalized equations [1], [19], [20].
COSTALD	FOLDER containing all necessary parameters for calculations with the Costald EOS [16].
RKM	FOLDER containing all necessary parameters for calculations with the RKM [17] and ERKM [18] method.
GERG-2008	FOLDER containing the FLUIDS and BINARY_MIX_FILES folders for the full GERG-2008 model [3].
EOS-CG-2016	FOLDER containing the FLUIDS and BINARY_MIX_FILES folders for the original EOS_CG model. The EOS_CG model is a model for humid gases and CCS mixtures and was developed by Gernert and Span (2016) [4].
EOS-CG-2019	FOLDER containing the FLUIDS and BINARY_MIX_FILES folders for the extended EOS_CG model. The EOS_CG model is a model for humid gases and CCS mixtures and was developed by Gernert and Span (2016) [4] and supplemented with new components by Herrig (2018) [31].
EOS-LNG	FOLDER containing the FLUIDS and BINARY_MIX_FILES folders for the EOS-LNG according to Thol et al. (2018) [5].
PC-SAFT	Folder containing the pure-fluid parameters and mixing parameters k_{ij} for the PCP-SAFT model according to Gross and Sadowski (2001) [6], Gross (2005) [7], and Gross and Vrabec (2006) [8].

File/Folder	Comment
GE_MODELS	FOLDER containing parameters of excess Gibbs energy models (also called activity coefficient models) for the liquid phase. The FOLDER contains the two FOLDERS "cosmo-sac" and "unifac".
	The FOLDER "unifac" contains two folders, "psrk" and "helm_ge".
	The FOLDER "psrk" contains the parameter files for the Predictive Soave-Redlich-Kwong (PSRK) equation of state [24]. The volume parameters and the surface parameters of the subgroups are stored in the FILE "rk_qk.par". The main group interaction parameters are stored in the file "Interac.par". The interaction parameters are given in the form "nr. of main group 1, nr. of main group 2, a12, b12, c12, a21, b21, c21" All parameters have been taken from [25].
	The FOLDER "helm_ge" contains the UNIFAC parameters when used in combination with the multi-fluid mixture model in a theoretically based departure function [26]. The parameter files "rk_qk.par", "Interac.par", and the subgroup definition file "groups.par" contain the parameters of UNIFAC used in the PSRK, taken from [25].
	The FOLDER "cosmo-sac" contains parameters of the COSMO-SAC model [27] and the sigma-profiles [32-34]. These parameters are needed when using the combination of the multi-fluid mixture model with COSMO-SAC [28].
FORTRAN	FOLDER containing the FORTRAN 95 files with the source code that is compiled into the .DLL files.
MATLAB	FOLDER contains TREND.mlappinstall file to install the TREND application in MATLAB.
PYTHON	FOLDER contains python interface fluid.py and an example trend_interface.py.

The files and folder listed above can be placed anywhere on the hard drive, but must be in the same directory, e.g. D:\TREND\. The .DLL files can be placed either in the Windows system directory (by default: C:\WINDOWS\system32\ for 32bit Windows systems and C:\WINDOWS\SysWOW64\ for 64bit Windows systems) or in the directory of the application calling them.

3 USING THE TREND.DLL

3.1 THERMODYNAMIC PROPERTIES – TREND_EOS

The TREND.DLL incorporates a number of functions that can be called from an external application. All thermodynamic property functions (the first group of functions in Table 4) are called with a common calling sequence:

Property = TREND_EOS(CALCTYPE;INPUT;PROP1;PROP2,FLUIDS,MOLEFRACTIONS;EOSTYPE;MIXTYPE; PATH;UNIT;ERRORFLAG;TYPEHANDLE)

Table 2. Input parameters for the function TREND_EOS.

Variable	Type	Description
CALCTYPE	Character(12)	Defines the property, which is calculated; see Table 4
INPUT	Character(12)	Defines the types and combination of PROP1 and PROP2; see Table 3
PROP1	Double	Numerical value of property 1
PROP2	Double	Numerical value of property 2
FLUIDS	Character(30) dimension(30)	List of fluid names of the involved components. The names of the fluids must match the names of the fluid parameter files (.FLD) in the \FLUIDS folder. The maximum number of components in a mixture is 20
MOLFRACTIONS	Double dimension(30)	List of mole fractions of the involved components (Mole fractions $x < 10^{-14}$ will not be considered)
EOSTYPE	Integer dimension(30)	List of the equation type for each component; see Table 8
MIXTYPE	Integer	Defines the combining rule of the mixture, see Table 8
PATH[a]	Character(255)	Path to the directory, where the FLUIDS and BINARY_MIX_FILES folders are located. To use the integrated equations of the DLL, the path has to be named "HC".
UNIT	Character(20)	Defines the units of the input properties and the resulting properties. Available units are specific, molar, and reduced. Using molar and specific units, all input and output properties (including mole fraction) are given in the corresponding molar or specific SI units. Exceptions are: - Pressure-related units in MPa - Surface tension in 10^3 N/m
ERRORFLAG	Integer	Internal error variable
TYPEHANDLE	C_PTR	Pointer to memory address of the global data type gl. The value is 0 on first call and contains the memory address on subsequent calls.

[a]A new feature since Trend 3.0 is the direct implementation of not only the Helmholtz EOS and Helmholtz mixing rules into the DLL but also of the PR, SRK, and LKP, etc. If "HC" is used as PATH, no FLUIDS and BINARY_MIX_FILES folders will be needed. The parameters used to calculate a property for the specified substance are directly taken from the DLL. Note that this feature is not available for the mixtype 13 (COSMO-SAC) and 22 (PSRK).

Table 3: Input combinations of thermodynamic properties for the property functions

Input code	PROP1	PROP2	Comment
"TD" [1]	Temperature [K]	Density [mol/m^3]	
"TP" [1]	Temperature [K]	Pressure [MPa]	
"PH" [1]	Pressure [MPa]	Enthalpy [J/mol]	

Input code	PROP1	PROP2	Comment
"PS" [1]	Pressure [MPa]	Entropy [J/mol/K]	
"TQ"	Temperature [K]	Quality [-]	
"PQ"	Pressure [MPa]	Quality [-]	
"PVAP"	Pressure [MPa]	- [2]	Calculation of the saturated vapor at given pressure
"PLIQ"	Pressure [MPa]	- [2]	Calculation of the saturated liquid at given pressure
"TVAP"	Temperature [K]	- [2]	Calculation of the saturated vapor at given temperature
"TLIQ"	Temperature [K]	- [2]	Calculation of the saturated liquid at given temperature
"TSUBV+"	Temperature [K]	- [2]	Calculation of the resublimation curve at given temperature. Only possible for pure substances.
"TSUBS+"	Temperature [K]	- [2]	Calculation of the sublimation curve at given temperature. Only possible for pure substances.
"PSUBV+"	Pressure [MPa]	- [2]	Calculation of the resublimation curve at given pressure. Only possible for pure substances.
"PSUBS+"	Pressure [MPa]	- [2]	Calculation of the sublimation curve at given pressure. Only possible for pure substances.
"TMLTL+"	Temperature [K]	- [2]	Calculation of the freezing curve at given temperature. Only possible for pure substances.
"TMLTS+"	Temperature [K]	- [2]	Calculation of the melting curve at given temperature. Only possible for pure substances.
"PMLTL+"	Pressure [MPa]	- [2]	Calculation of the freezing curve at given pressure. Also implemented for mixtures.
"PMLTS+"	Pressure [MPa]	- [2]	Calculation of the melting curve at given pressure. Only possible for pure substances.

[1] The combination of PROP1 and PROP2 in the input code is arbitrary, e.g.: "TD" works as well as "DT" (the first letter always indicates PROP1, the second PROP2).
[2] PROP2 must be an arbitrary value larger than zero. It will be ignored by the routines.

For the input combinations "TD", "TP", "PH", and "PS" the property is calculated regardless of which phase the fluid or mixture is in at the given input parameters. Internally, the system is tested for stability. If the system is found unstable (meaning that more than one phase is present at the given input parameters), a flash calculation is called and the function property is calculated as overall property of all present phases, according to:

$$Z = Z_{vap}(x_{vap}, \rho_{vap}, T) \cdot q + Z_{liq}(x_{liq}, \rho_{liq}, T) \cdot (1 - q).$$

Z is the overall or bulk property, Z_{vap} the saturated vapor property, calculated at the vapor phase composition and density x_{vap} and ρ_{vap}, Z_{liq} the saturated liquid property, calculated at the liquid phase composition and density x_{liq} and ρ_{liq}, and q the molar vapor fraction. The densities and compositions of the corresponding phases are calculated by the flash algorithm.

Models for solid CO_2 [9], solid H_2O [10], and gas hydrates of the binary systems water + carbon dioxide, nitrogen, oxygen, argon, carbon monoxide, methane, ethane, or propane [11] have been

implemented. The gas hydrate models are based on the work published by Jäger et al. [12] and are published as journal article [35], [36]. **It is very important that the fluid files (folder "FLUIDS\" and "BINARY_MIX_FILES\") in the main folder are used when these solid phases shall be considered. Otherwise, the formation of these solids might be predicted wrong!!** Some special input codes for solids exist, see Table 3.

Using the input combinations as indicated in Table 3, the given limits of the equation of state (e.g., T_{min}, T_{max}, p_{max}, etc.) will be taken into account. For calculations outside the range of validity, a "&" has to be used at the end of the input code (e.g., "TD&" or "TP&").

When choosing the inputs "TP+", "PH+", or "PS+", solid phases and equilibria of up to three phases are considered. "TD+" is not implemented, yet. Note that for the inputs without an additional "+" only two-phase equilibria are considered, even if all phases are fluid phases.

Table 4: Available thermodynamic properties and the corresponding CALCTYPE.

Property	Symbol	CALCTYPE
Temperature	T	T
Density	ρ	D
Pressure	p	P
Internal energy	u	U
Enthalpy	h	H
Entropy	s	S
Gibbs free energy	$g = h - Ts$	G
Helmholtz free energy	$a = u - Ts$	A
Isobaric heat capacity	c_p	CP
Isochoric heat capacity	c_v	CV
Speed of sound	w_s	WS
Second Virial Coefficient	B	BVIR
Third Virial Coefficient	C	CVIR
Forth Virial Coefficient	D	DVIR
1st derivative of B w.r.t. temperature	dB/dT	DBDT
1st derivative of C w.r.t. temperature	dC/dT	DCDT
Second cross virial coeffcient	B_{12}	B12
Ideal gas isobaric heat capacity	c_p^o	CP0
Quality (molar vapor fraction)	q	Q
Compressibility factor	Z	Z
Numerical solution for $p(T,\rho)$	p_{numer}	PNUM
Numerical speed of sound solution $w(p,s)$	$w_{s,\,numer}$	WSNUM
Volume Expansivity	α_v	VEXP
Isothermal Compressibility	κ_T	COMPT
Isentropic Compressibility	κ_s	COMPS
Isothermal expansion coefficient	k_T	EXPT
Isentropic expansion coefficient	k_s	EXPS
Joule-Thomson coefficient	μ	JTCO
Isothermal throttling coefficient	δ_T	THROT
Grueneisen Coefficient	Γ	GRUEN
Phase Identification Parameter	Π	PIP
Density of spinodal for given T	$\rho_{Spin,vap/liq}$	DSPIN
First derivative of u w.r.t. v at const. T	$(\partial u/\partial v)_T$	DUDV
Riemann Scalar Curvature	$RIEM$	RIEM
Phasetype	PHT	PHASE

Property	Symbol	CALCTYPE
1st derivative of pressure wrt temperatur	$(\partial p/\partial T)_\rho$	DPDT
1st derivative of pressure wrt density	$(\partial p/\partial \rho)_T$	DPDD
1st derivative of density wrt temperature	$(\partial \rho/\partial T)_p$	DDDT
2nd derivative of pressure wrt temperatur	$(\partial^2 p/\partial T^2)_\rho$	D2PDT2
2nd derivative of pressure wrt density	$(\partial^2 p/\partial \rho^2)_T$	D2PDD2
2nd derivative of pressure wrt temperature and density	$(\partial^2 p/\partial \rho \partial T)$	D2PDDT
Fundamental derivative of gas dynamics	γ_{GD}	GAMMAGD
Residual pressure	p^r	PR
Residual Isochoric Heat Capacity	c_v^r	CVR
Residual internal energy	u^r	UR
Residual enthalpy	h^r	HR
Residual entropy	s^r	SR
Residual Chemical Potential (pure fluid)	μ^r	CPOTR
Reduced residual Helmholtz energy	α^r	A00
1st derivative of α^r w.r.t δ, multiplied with δ	$\delta \cdot \alpha^r_\delta$	A01
1st derivative of α^r w.r.t τ, multiplied with τ	$\tau \cdot \alpha^r_\tau$	A10
2nd derivative of α^r w.r.t δ and τ multiplied with $\delta*\tau$	$\delta\tau \cdot \alpha^r_{\delta\tau}$	A11
2nd derivative of α^r w.r.t δ, multiplied with δ^2	$\delta^2 \cdot \alpha^r_{\delta\delta}$	A02
2nd derivative of α^r w.r.t τ, multiplied with τ^2	$\tau^2 \cdot \alpha^r_{\tau\tau}$	A20
3rd derivative of α^r w.r.t τ, δ, and δ, multiplied with $\tau*\delta^2$	$\tau\delta^2 \cdot \alpha^r_{\tau\delta\delta}$	A12
3rd derivative of α^r w.r.t τ, τ, and δ, multiplied with $\delta*\tau^2$	$\tau^2\delta \cdot \alpha^r_{\tau\tau\delta}$	A21
3rd derivative of α^r w.r.t δ, multiplied with δ^3	$\delta^3 \cdot \alpha^r_{\delta\delta\delta}$	A03
3rd derivative of α^r w.r.t τ, multiplied with τ^3	$\tau^3 \cdot \alpha^r_{\tau\tau\tau}$	A30
Excess volume	v^E	VE
Excess enthalpy	h^E	HE
Excess Gibbs energy	g^E	GE
Surface tension	σ	ST
Dielectric constant	ε	DE
Dynamic viscosity	η	ETA
Thermal conductivity[1]	λ	TCX
Acentric factor	ω	AF
Density maximum of water and heavy water	$T((\partial \rho/\partial T)_p = 0)$	TDENSMAX
Boyle temperature	$T(B=0)$	TBOYLE
Joule-Thomson temperature	$T(T \cdot (dB/dT) - B = 0)$	TJT
Joule-inversion temperature	$T((dB/dT) = 0)$	TJINV

[1]Slight differences in the thermal conductivity compared to TREND 4.0 are attributable to updated viscosity models

3.2 FLUID/MIXTURE SPECIFICATIONS – TREND_SPEC_EOS

Other properties, which are independent of the state point (e.g., molar mass or critical parameters), are called as follows:

Property =
TREND_SPEC_EOS(FLUIDS;MOLEFRACTIONS;EOSTYPE;MIXTYPE;PATH;UNIT;LIMITS_TEXT;LIMITS_VALUES;ERRORFLAG;TYPEHANDLE)

Table 5. Input parameters for the function TREND_EOS.

Variable	Type	Description
FLUIDS	Character(30) dimension(30)	List of fluid names of the involved components. The names of the fluids must match the names of the fluid parameter files (.FLD) in the \FLUIDS folder – both in spelling and case! The maximum number of components in a mixture is 20
MOLFRACTIONS	Double dimension(30)	List of mole fractions of the involved components (Mole fractions $x < 10^{-14}$ will not be considered)
EOSTYPE	Integer dimension(30)	List of the equation type for each component; see Table 8
MIXTYPE	Integer	Defines the combining rule of the mixture, see Table 8
PATH	Character(255)	Path to the directory, where the FLUIDS and BINARY_MIX_FILES folders are located. To use the integrated equations of the DLL, the path has to be named "HC".
LIMITS_TEXT	Character(30) dimension(30)	Description of properties. The entries are arranged in the following order: 1) molar weight "MW" 2) triple-point temperature "Ttrip" 3) triple-point pressure "ptrip" 4) critical temperature "Tcrit" 5) critical pressure "pcrit" 6) critical density "Dcrit" 7) acentric factor "AF" 8) minimum temperature for range of validity "Tmin" 9) maximum temperature for range of validity "Tmax" 10) maximum pressure for range of validity "Pmax" 11) maximum density for range of validity "Dmax"
LIMITS_VALUES	Double array(30,31)	List the results for the properties defined in LIMITS_TEXT
ERRORFLAG	Integer	Internal error variable
TYPEHANDLE	C_PTR	Pointer to memory address of the global data type gl. The value is 0 on first call and contains the memory address on subsequent calls.

3.3 THERMODYNAMIC PROPERTIES – ALLPROP

The subroutine **ALLPROP** requires the following calling sequence:

ALLPROP(INPUT;PROP1;PROP2;FLUIDS;MOLFRACTIONS;EOSTYPE;MIXTYPE;PATH;UNIT; *T; D; P; U; H; S; G; A; CP; CV; WS; B; C; CP0; Q*)

All variables written in italic are output variables containing the respective property. All of these variables have to have the DOUBLE PRECISION type.

If the user knows that the specified pure fluid or mixture at given conditions (only temperature and density "TD" or temperature and pressure "TP" as input possible) is homogeneous, the subroutine **ALLPROP_HOM** can be used:

ALLPROP_HOM(INPUT;PROP1;PROP2;FLUIDS;MOLFRACTIONS;EOSTYPE;MIXTYPE;PATH;UNIT;PHASE_IND; *T; D; P; U; H; S; G; A; CP; CV; WS; B; C; CP0*)

No stability test or flash calculation is performed when calling this routine. Using temperature and density as input, all properties are calculated directly. Using temperature and pressure, the density is calculated with the density solver and the given value of PHASE_IND determines which density solution is taken, if multiple solutions exist. Following inputs are possible: PHASE_IND = 0: Solution with lower Gibbs energy, PHASE_IND = 1: prefer liquid solution, PHASE_IND = 2: prefer vapor solution.

3.4 THERMODYNAMIC PROPERTIES – GAMMATF_EOS

The subroutine **GAMMATF_EOS** calculates the thermodynamic factor according to Kooijman and Taylor [21]:

GAMMATF_EOS(INPUT;PROP1;PROP2,FLUIDS;MOLFRACTIONS;EOSTYPE;MIXTYPE;PATH;GAMMATF ;ERRORFLAG;TYPEHANDLE)

Input parameters are defined as described in Table 2. GAMMATF is the output matrix with 30x30 entries.

3.5 THERMODYNAMIC PROPERTIES –HENRY_EOS

The subroutine **HENRY_EOS** calculates the Henry's constants H_{12} and H_{21} according to the definitions:

$$H_{12} = \lim_{x_1' \to 0} \left(\frac{f_1'}{x_1'} \right)$$

and

$$H_{21} = \lim_{x_2' \to 0} \left(\frac{f_2'}{x_2'} \right).$$

The Henry's constant H_{12} describes the solution of component 1 in component 2 and H_{21} describes the solution of component 2 in component 1.

HENRY_EOS(TEMP;FLUIDS;MOLFRACTIONS;EOSTYPE;MIXTYPE;PATH;H12;H21;ERRORFLAG;TYPEHANDLE)

The input parameters FLUIDS, MOLFRACTIONS, EOSTYPE, MIXTYPE, PATH, ERRORFLAG, and TYPEHANDLE are defined as described in Table 2. TEMP is the (input) temperature in K and the variable is of type DOUBLE PRECISION. H12 and H21 are the output variables and are both of type DOUBLE PRECISION.

3.6 THERMODYNAMIC PROPERTIES – FUGCOEFF_EOS

The subroutine **FUGCOEFF_EOS** calculates the fugacity coefficients for pure fluids or for each component in a mixture:

FUGCOEFF_EOS(INPUT;PROP1;PROP2,FLUIDS;MOLFRACTIONS;EOSTYPE;MIXTYPE;PATH;FUGCOEFF, ;ERRORFLAG;TYPEHANDLE)

Input parameters are defined as described in Table 2. FUGCOEFF is the output matrix with 30x30 entries.

3.7 THERMODYNAMIC PROPERTIES – ANCILLARY EQUATIONS

Ancillary equations are NOT fundamental equations of state!

The ancillary equations are used to generate starting values for the iteration of saturation properties with equations of state. With lower accuracy requirements, they enable an estimation of saturation properties with reduced computing time.

The function **ANC_EQ** calculates saturation properties for pure fluids based on ancillary equations:

Property = ANC_EQ(CALCTYPE;INPUT;PROP1;PROP2,FLUIDS,MOLEFRACTIONS;EOSTYPE;MIXTYPE; PATH;UNIT;ERRORFLAG;TYPEHANDLE)

Input parameters are identical to TREND_EOS and are defined as described in Table 2. Depending on the searched property, the temperature or the pressure is required for the calculation of the saturation state. The function ANC_EQ checks if the required parameters is defined in the input variables and calculates the corresponding searched property.

Table 6: Available thermodynamic properties and the corresponding CALCTYPE for ANC_EQ.

Property	Symbol	CALCTYPE
Melting temperature	T_{melt}	TMELT
Melting pressure	p_{melt}	PMELT
Sublimation temperature	T_{sub}	TSUB
Sublimation pressure	p_{sub}	PSUB
Saturated liquid density	d_{liq}	DLIQ
Saturated vapor density	d_{vap}	DVAP
Vapor pressure	p_{vap}	PVAP

3.8 FLUID SPECIFICATIONS – CAS_FROM_FLUID

The subroutine **CAS_FROM_FLUID** translates the fluid name to the CAS number. If a mixture is defined in FLUIDS, only the CAS number of the first entry is given in the resulting variable CASNR.

CAS_FROM_FLUID (FLUIDS;EOSTYPE;MIXTYPE;PATH;CASNR;ERRORFLAG;TYPEHANDLE)

Input parameters are defined as described in Table 2.

3.9 FLUID INFORMATIONS – LITERATURE_REF

This function works only for models based on the Helmholtz energy (EQTYPE = 1).

The results of the algorithm depend on the property model, which is developed for every substance and (binary) mixture. The literature reference of the property model can be obtained by using the subroutine Literature_ref. The following arguments are mandatory

LITERATURE_REF(FLUIDS, EOSTYPE, MIXTYPE, PROP, PATH, REF_OUT, ERRORFLAG, TYPEHANDLE)

Table 7. Input parameters for the literature-references subroutine.

Variable	Type	Description

fluids	Character(30) dimension(30)	List of fluid names of the involved components. The names of the fluids must match the names of the fluid parameter files (.FLD) in the \FLUIDS folder The maximum number of components in a mixture is 20
EOSTYPE	double	List of the equation type for each component; see Table 8
MIXTYPE	integer	Defines the combining rule of the mixture, see Table 8
prop	Character(10)	Specifies for which property the literature reference shall be obtained. Implemented properties for literature references: 'p','d','t','u','h','s','g','a','cp','cv','ws','bvir','cvir','z','de','tcx','stn', 'eta','mlt','sbl'
path	Character(255)	Path to the directory, where the FLUIDS and BINARY_MIX_FILES folders are located. To use the integrated equations of the DLL, the path has to be named "HC".
ref_out	Character(1000)	Contains literature reference
Errorflag	Integer	Internal error variable
typehandle	C_PTR	Pointer to memory address of the global data type gl. The value is 0 on first call and contains the memory address on subsequent calls.

If only one fluid is specified, the literature reference for the given property and the underlying property model is written to ref_out.

If two fluids are specified, the combination or binary specific mixing rule is written to ref_out.

There will be no output for more than two fluids, because the mixing rules are binary-specific. For a ternary mixture, the three possible binary mixtures have to be created and forwarded to the subroutine in order to obtain all mixing rules.

3.10 THERMODYNAMIC PROPERTIES – PROPERTY UNIT

The subroutine **PROPUNIT** defines the unit of the corresponding calculated property depending on the input parameters "UNIT" and "CALCTYPE".

PROPUNIT (UNIT;CALCTYPE;UNITRESULT;ERRORFLAG)

Input parameters are defined as described in Table 2.

4 USING TREND IN MATLAB®

Users who want to use TREND in MATLAB can use the new available MATLAB application. To make Trend available in MATLAB, just run the *TREND.mlappinstall* file. Then TREND should be available under the apps in MATLAB. Before the first calculation the path to the fluid files must be changed. For the use of TREND in your own MALTAB implementations, the defined fluid class can be seen in the installation path "C:\Users\username\Documents\MATLAB\Add-Ons\Apps\TREND\code". This path also contains all necessary files for using TREND in MATLAB. The TREND application is only available and tested starting with the MALTAB 2018a version on 64-bit machines. By now the following functions are available with the use of a fluid class object:

- [<double> result, <int> error] = TREND_EOS(<double> prop1, <double> prop2, <int> error)
- [<char*30,30> text, <double,30> limits, <int> error] = TREND_SPEC_EOS(<int> error)
- [<figure> fig <int> error] = PTDIAG_OUT(<int> env_p_in,<double> press, <double> temp)
- [<figure> fig, <int> error] = PTXDIAG_OUT
- [<figure> fig, <int> error] = TERNARY_DIAG

5 USING TREND IN PYTHON

For TREND 5.0 a new python interface is available, which is like the MATLAB interface. To get started we recommend using Visual Studio Code and the newest python interpreter. To use TREND in python, make sure you have installed the "ctypes" package. In the python folder you find an example file with the name "trend_interface.py". To run this file, you will need to specify your path to the TREND dll and fluids directory. Currently only the TREND_EOS function is available via the fluid class.

6 EOSTYPE

The input parameter EOSTYPE gives the user the choice to calculate thermodynamic properties using different types of equations of state. It is also (partly) possible to calculate mixtures using different types of EOS for each component in the mixture. The mixing rule needs to be specified with the variable MIXTYPE.

EOSTYPE is an integer of dimension 30. It has to be passed with as many entries as there are components in the mixture, and the rest of the array should be filled with zeros. All equations used in TREND are transformed to the Helmholtz energy first. Thus, different equation types can be mixed using Helmholtz mixing rules. This does not work the other way around! Some examples are given in Table 8.

Table 8: Valid Input combinations for EOSTYPE. The MIXTYPE has to be chosen according to the EOSTYPE.

Mixture	EOSTYPE	MIXTYPE	Calculation
Methane;Ethane;Propane	1;1;1	1	Helmholtz-EOS used for all pure fluids, as indicated by the first three entries. GERG-2004[b] mixing rules used
Methane;Ethane;Propane	1;1;1	12	Helmholtz EOS used for all pure fluids. Theoretically based departure function with UNIFAC (PSRK parameters) is used [26]
Methane;Ethane;Propane	1;1;1	13	Helmholtz EOS used for all pure fluids. Theoretically based departure function with COSMO-SAC is used [28]
Methane;Ethane;Propane	2;2;2	2	SRK[a] used for all pure fluids, SRK[c] mixing rules used (quadratic mixing rule for a with combining rule $a_{ij} = \left(a_i \cdot a_j\right)^{0.5}(1 - k_{ij})$ and linear mixing rule for b)
Methane;Ethane;Propane	2;2;2	21	SRK[a] used for all pure fluids, SRK mixing rules used (quadratic mixing rule for a with combining rule $a_{ij} = \left(a_i \cdot a_j\right)^{0.5}(1 - k_{ij})$ and quadratic mixing rule for b with $b_{ij} = \left(b_i \cdot b_j\right)^{0.5}(1 - l_{ij})$)
Methane;Ethane;Propane	2;2;2	22	PSRK mixing rules used, see [24][25]
Methane;Ethane	3;3	3	PR[e] used for all pure fluids, PR[e] mixing rules used
Methane;Ethane	3;3	31	PR[e] used for all pure fluids, PR mixing rules used (quadratic mixing rule for a with combining rule $a_{ij} = \left(a_i \cdot a_j\right)^{0.5}(1 - k_{ij})$ and quadratic mixing rule for b with $b_{ij} = \left(b_i \cdot b_j\right)^{0.5}(1 - l_{ij})$)
Methane;Ethane;Propane	4;4;4	4	LKP[d] used for all pure fluids, LKP[d] mixing rules used
Methane	51	51	Generalized equations[i] used for all pure fluids; mixtures are calculated according to the Lorentz-Berthelot rule.
Methane;Ethane	52;52	52	
Methane;Ethane;Propane	53;53;53	53	
Methane;Ethane	6;6	6	PCP-SAFT equations of state[j]
Methane;Ethane;Propane	7;7;7	7	AGA8 equation of state[k]

Mixture	EOSTYPE	MIXTYPE	Calculation
Methane;Ethane;Propane	8;8;8	8	RKM[g] method used for mixtures; only valid for saturated liquid densities
Methane;Ethane;Propane	81;81;81	81	ERKM[h] method used for mixtures; only valid for saturated liquid and homogeneous liquid densities
Methane;Ethane;Propane	9;9;9	9	COSTALD[f] EOS used for pure fluids and mixtures; only valid for saturated liquid and homogeneous liquid densities
Water;NaCl	1;15	1	Aqueous Solutions containing NaCl
Methane;Ethane;Propane	1;2;1		SRK[a] used for Ethane, Helmholtz-EOS used for Methane and Propane.

[a] SRK: cubic EOS by Redlich, Kwong and Soave, see Ref. [13]

[b] The quadratic mixing rules as given in the GERG-2004 monograph [2] are employed to calculate mixture reducing parameters. They have two adjustable binary interaction parameters, which are set to unity for unknown mixtures. These mixing rules can also be applied to cubic EOS by first integrating these EOS to obtain the Helmholtz free energy. For a large number of fluids binary parameters are available. These are given in the folder BINARY_MIX_FILES.

[c] The mixing rules introduced by Soave [13] are used. The binary interaction parameter is read from the BIN_SRK.mix file which lies in the BINARY_SRK folder. If no parameter for a binary mixture is given, k_{ij} is set to the standard value (unity).

[d] LKP: EOS by Lee, Kessler and Plöcker, see Ref. [14]

[e] PR: EOS by Peng and Robinson, see Ref. [15]; same mixing rules as for the SRK are used

[f] COSTALD: EOS by Thomson et al., see Ref. [16]

[g] RKM: method by McCarty, see Ref. [17]

[h] ERKM: method by Tietz et al., see Ref. [18]

[i] Generalized equations: EOSTYPE 51 - EOS by Alexandrov et al. (2013), see Ref. [19]
EOSTYPE 52 - EOS by Span (2000), see Ref. [1]
EOSTYPE 53 - EOS by Sun & Ely (2005), see Ref. [20]

[j] PCP-SAFT equation of state according to Gross, Sadowski, and Vrabec [6], [7], [8]. The model parameters are listed in the fluid file: "Trend 5.0\PC-SAFT\pcsaftfluids.fld" There are different parameter sets available from different publications. The one, which is used, is marked with a "#". The default parameters are arbitrarily chosen and no recommendations from the TREND developers.

[k] AGA8: mixing rules given in the AGA Report No.8 Part 1 [29][18]

7 FLASH CALCULATION

The calculation of a two-phase or three-phase flash is carried out by the subroutine FLASH3. This subroutine has the following calling sequence:

FLASH3(input, prop1, prop2, fluids, moles, eos_indicator, mix_indicator, path, unit, phasetype, phase_text, x_phase, prop_phase, prop_overall, lnfug_phase, chempot_phase, phasefrac, prop_name_unit, errorflag, gl_handle)

Additional to the input parameters according to Table 2 minus the parameter CALCTYPE the FLASH subroutine returns the following parameters to the calling application (Table 9):

Table 9: Return parameters of the FLASH subroutine.

Parameter	Type	Comment
PHASETYPE	Double Precision, dimension(5)	Indicates the phases that are in equilibrium: phasetype(1) = first phase in equilibrium phasetype(2) = second phase in equilibrium phasetype(3) = third phase in equilibrium Indicators: 1 - vapor phase 2 - "lighter" liquid phase 3 - "heavier" liquid phase 4 - pure solid phase 5 - gas hydrate phase
PHASE_TEXT	Character(4), dimension(5)	Contains the abbreviation of the determined phases VAP = vapor, LIQ1 = "lighter" liquid phase, LIQ2 = "heavier" liquid phase, SOL = pure solid phase, HYD = gas hydrate phase
X_PHASE	Double Precision dimension (30,5)	Molar composition of the phases in equilibrium First index: mole fraction of the component in the mixture Second index: indicates the phase, see phasetype
PROP_PHASE	Double Precision dimension (30,5)	Properties of the phases. First index is the property: T, p, ρ, u, h, s, g, a, c_p, c_v, w, hydration number, small cage occupancy, large cage occupancy Second index: indicates the phase, see phasetype
PROP_OVERALL	Double Precision dimension (30)	Overall properties of the mixture: T, p, ρ, u, h, s, g, a, c_p, c_v, w
LNFUG_PHASE	Double Precision, dimension (30,5)	Logarithm of the fugacities in MPa of all components in the different phases. First index: component Second index: indicates the phase, see phasetype
CHEMPOT_PHASE	Double Precision, dimension (30,5)	Chemical potential of all components in the different phases. First index: component Second index: indicates the phase, see phasetype
PHASEFRAC	Double Precision, dimension (5)	Molar phase fraction of the phase. For phase indicator explanation, see phasetype
PROP_NAME_UNIT	Character(30), dimension(37,2)	Contains the name of the calculated property and the unit of the respective property depending on the given unit (molar/specific).

In the subroutine FLASH a stability analysis determines if the given mixture is stable or if it decomposes into two or three phases. If the mixture is found to be unstable, the mole fractions of the split phases are calculated subsequently.

As this is an iterative procedure which can fail sometimes, an additional routine was implemented that allows the user to specify start values for the split phases. In this subroutine, only equilibria of fluid phases are considered. The properties have to be specified on a molar basis. If the properties are known on a specific basis, they have to be converted with the molar mass (TREND_SPEC_EOS) first. Composition(s) can be converted with the COMP function, see section 11. The subroutine has the following calling sequence:

FLASH_EXPERT(input, prop1, prop2, fluids, moles, EOS_indicator, MIX_indicator, path, xliq_in, xvap_in, rholiq_est, rhovap_est, iPhase_try, xliq, xvap, propliq, propvap, vapfrac, errorflag, gl_handle)

Additionally, to the parameters described in Table 2 and Table 9, initial estimates for some variables depending on the chosen INPUT combination have to be passed to the routine. These additional parameters are described in Table 10. In case of a dew or bubble point calculation with given temperature or pressure (INPUT: "TVAP", "TLIQ", "PVAP", "PLIQ") the initial estimate for the pressure in case of given temperature or the temperature in case of given pressure has to be passed to the routine in the variable PROP2.

Table 10: Additional parameters of the FLASH_EXPERT subroutine.

Parameter	Type	Comment
XLIQ	Double Precision, dimension (30)	Molar composition of the (heavier) liquid phase.
XVAP	Double Precision, dimension (30)	Molar composition of the vapor (or lighter liquid) phase.
PROPLIQ	Double Precision, dimension (30)	Properties of the saturated (heavier) liquid phase. Property order: T, p, ρ, u, h, s, g, a, c_p, c_v, w
PROPVAP	Double Precision, dimension (30)	Properties of the saturated vapor (or lighter liquid) phase. Property order: T, p, ρ, u, h, s, g, a, c_p, c_v, w
Q	Double Precision	Molar vapor fraction.
XLIQIN	Double Precision, dimension (30)	Initial estimate for the molar composition of the (heavier) liquid phase. This input is mandatory.
XVAPIN	Double Precision, dimension (30)	Initial estimate for the molar composition of the vapor or (lighter) liquid phase. This input is mandatory.
RHOLIQEST	Double Precision	Initial estimate for the density of the (heavier) liquid phase in mol/m³. This input is optional. If no estimate is available, this parameter has to be set to "0".
RHOVAPEST	Double Precision	Initial estimate for the density of the vapor or (lighter) liquid phase in mol/m³. This input is optional. If no estimate is available, this parameter has to be set to "0".
IPHASETRY	Integer	Guess which phases are present in equilibrium 0: No guess, let the algorithm decide 4: Two liquid phases (LLE) 5: A vapor and a liquid phase (VLE) The numbers 1, 2, 3, and 6 are also possible inputs to the routine. The effect of these inputs is

Parameter	Type	Comment
		explained in more detail in the example excel spreadsheet.

The user can also manually specify the initial estimates for a three-phase equilibrium of fluid phases. The subroutine that performs a three-phase equilibrium calculation with given initial estimates has the following form:

FLASH_NC_3P_EXPERT(input, prop1, prop2, fluids, moles, EOS_indicator, MIX_indicator, path, xliq1_in, xliq2_in, xvap_in, rholiq1_est, rholiq2_est, rhovap_est, xliq1, xliq2, xvap, propliq1, propliq2, propvap, Phasefrac, errorflag, gl_handle)

The additional input parameters of this routine are summarized in Table 11, the additional output parameters in Table 12. According to Gibbs' Phase rule $F = N - P + 2$, where F denotes the degree of freedom, N the number of components and P the number of phases, the three-phase area for binary mixtures is a line in a pT-plot and for multicomponent mixtures a region. Hence, some additional input combinations are needed. All possible inputs are listed in Table 13.

Table 11: Additional input parameters of the FLASH_NC_3P_EXPERT subroutine.

Parameter	Type	Comment
XLIQ1IN	Double Precision Array, length 30	Initial estimate for the molar composition of the (lighter) liquid phase. This input is mandatory.
XLIQ2IN	Double Precision Array, length 30	Initial estimate for the molar composition of the (heavier) liquid phase. This input is mandatory.
XVAPIN	Double Precision Array, length 30	Initial estimate for the molar composition of the vapor phase. This input is mandatory.
RHOLIQ1EST	Double Precision	Initial estimate for the density of the (lighter) liquid phase in mol/m³. This input is optional. If no estimate is available, this parameter has to be set to "0".
RHOLIQ2EST	Double Precision	Initial estimate for the density of the (heavier) liquid phase in mol/m³. This input is optional. If no estimate is available, this parameter has to be set to "0".
RHOVAPEST	Double Precision	Initial estimate for the density of the vapor phase in mol/m³. This input is optional. If no estimate is available, this parameter has to be set to "0".

Table 12: Additional parameters of the FLASH_NC_3P_EXPERT subroutine.

Parameter	Type	Comment
XLIQ1	Double Precision Array, length 30	Molar composition of the (lighter) liquid phase.
XLIQ2	Double Precision Array, length 30	Molar composition of the (heavier) liquid phase.
XVAP	Double Precision Array, length 30	Molar composition of the vapor phase.
PROPLIQ1	Double Precision Array, length 30	Properties of the saturated (lighter) liquid phase. Property order: $T, p, \rho, u, h, s, g, a, c_p, c_v, w$

Parameter	Type	Comment
PROPLIQ2	Double Precision Array, length 30	Properties of the saturated (heavier) liquid phase. Property order: $T, p, \rho, u, h, s, g, a, c_p, c_v, w$
PROPVAP	Double Precision Array, length 30	Properties of the saturated vapor phase. Property order: $T, p, \rho, u, h, s, g, a, c_p, c_v, w$
Phasefrac	Double Precision Array, length 3	Molar phase fractions for all phases: Phasefrac(1): molar amount of (lighter) liquid / overall amount of substance. Phasefrac(2): molar amount of (heavier) liquid / overall amount of substance. Phasefrac(3): molar amount of vapor / overall amount of substance.

Table 13: Input combinations of thermodynamic properties for the property functions

Input code	PROP1	PROP2	Comment
"TP" [1]	Temperature [K]	Pressure [MPa]	
"PVAP"	Pressure [MPa]	Temperature [K] [2]	Calculate the temperature at given pressure, where Phasefrac(1) = 0 (see Table 12). This is the temperature at which in a two phase equilibrium of vapor and (heavy) liquid the first amount of the (lighter) liquid phase is formed.
"PLIQ"	Pressure [MPa]	Temperature [K] [2]	Calculate the temperature at given pressure, where Phasefrac(3) = 0 (see Table 12). This is the temperature at which in a two phase equilibrium of (light) liquid and (heavy) liquid the first amount of the vapor phase is formed.
"PL2"	Pressure [MPa]	Temperature [K] [2]	Calculate the temperature at given pressure, where Phasefrac(2) = 0 (see Table 12). This is the temperature at which in a three phase equilibrium of vapor and (light) liquid the first amount of the (heavier) liquid phase is formed.
"TVAP"	Temperature [K]	Pressure [MPa] [2]	Calculate the pressure at given temperature, where Phasefrac(1) = 0 (see Table 12). This is the temperature at which in a two phase equilibrium of vapor and (heavy) liquid the first amount of the (lighter) liquid phase is formed.
"TLIQ"	Temperature [K]	Pressure [MPa] [2]	Calculate the pressure at given temperature, where Phasefrac(3) = 0 (see Table 12). This is the temperature at which in a two phase equilibrium of (light) liquid and (heavy) liquid the first amount of the vapor phase is formed.
"TL2"	Temperature [K]	Pressure [MPa] [2]	Calculate the temperature at given pressure, where Phasefrac(2) = 0 (see Table 12). This is the temperature at which in a three phase equilibrium of vapor and (light)

Input code	PROP1	PROP2	Comment
			liquid the first amount of the (heavier) liquid phase is formed.

[1] The combination of PROP1 and PROP2 in the input code is arbitrary, e.g.: "TP" works as well as "PT" (the first letter always indicates PROP1, the second PROP2).

[2] PROP2 is needed as initial estimate for this input combination

8 PHASE ENVELOPE CALCULATION

The subroutine PTDIAG_OUT calculates points on the phase envelope (bubble and dew points at constant composition) for a given mixture. In case a pure fluid is passed to the routine, it returns the vapor pressure curve. Beside the temperature and pressure for each point, the routine returns the densities of the original and the incipient phase, the composition of the incipient phase and a point ID number. The subroutine has the following calling sequence:

PTDIAG_OUT(ENV_PV;FLUIDS;MOLFRACTIONS;EOSTYPE;MIXTYPE;PATH;P_SPEC;T_SPEC;
            T_PTS;P_PTS;RHOLIQ_PTS;RHOVAP_PTS;POINTID;X_PTS;FILEOUT;ERRORFLAG;
            TYPEHANDLE)

Additionally to the input parameters FLUIDS, MOLFRACTIONS, EOSTYPE, MIXTYPE, PATH and TYPEHANDLE which are already described in section 3.1, the input parameters for this routine are:

Table 14: Additional input parameters for the routine PTDIAG_OUT

Parameter	Type	Comment
ENV_PV	Integer	Toggle to set whether pressure (1) or volume-based (2) phase envelope calculation is used.
T_SPEC	Double Precision	Specified temperature [K] for which a point on the phase envelope is calculated. If no temperature is specified this parameter has to be set to zero!
P_SPEC	Double Precision	Specified pressure [MPa] for which a point on the phase envelope is calculated. If no temperature is specified this parameter has to be set to zero!
FILEOUT	String, length 255	This parameter is only used in the programming process and has no meaning or influence in the release version.

The return parameters of the routine PTDIAG_OUT are:

Table 15: Output parameters of the routine PTDIAG_OUT

Parameter	Type	Comment
T_PTS	Double Precision Array, length 400	Array with the temperatures of the calculated saturation points.
P_PTS	Double Precision Array, length 400	Array with the pressures of the calculated saturation points.
RHOLIQ_PTS	Double Precision Array, length 400	Array with the densities of the incipient phase of the calculated saturation points.
RHOVAP_PTS	Double Precision Array, length 400	Array with the densities of the original phase of the calculated saturation points.
POINTID	Integer Array, length 400	Array with the point IDs for each saturation point. The point IDs are: 0: Normal point on the phase envelope 1: Critical point (interpolated) 2: Maximum temperature (cricondentherm) 3: Maximum pressure (cricondenbar) 4: Specified temperature 5: Specified pressure
X_PTS	Double Precision Array, length {400,30}	Array with the composition of the incipient phase of the calculated saturation points.
ERRORFLAG	Integer	Error code

9   *P-X* DIAGRAM AND *T-X* DIAGRAM CALCULATION

The subroutine PTXDIAGEOS calculates bubble and dew lines for a given binary mixture in a *p-x* diagram or a *T-x* diagram. If INPUT is set to "TlLIQ" or "TVAP" a *p-x* diagram is plotted at constant temperature. To calculate a *T-x* diagram at constant pressure INPUT has to be set to "PLIQ" or "PVAP". The input parameters are INPUT, PROP, FLUIDS, MOLFRACTIONS, EOSTYPE, MIXTYPE, PATH, FILEOUT and TYPEHANDLE. They are already described in Table 2 and 10. The subroutine has the following calling sequence:

> PTXDIAGEOS
> (INPUT;PROP;FLUIDS;EOSTYPE;MIXTYPE;PATH;P_PTS;T_PTS;X_PTS;RHOVAP_PTS;RHOLIQ_
> PTS;POINTS;FILEOUT;ERRORFLAG; TYPEHANDLE)

The returned parameters are the number of calculated points, temperature and pressure as well as the compositions and densities for each phase. The following table summarizes the output parameters, their type, and description.

Table 12: Output parameters of the routine PTDIAG_OUT

Parameter	Type	Comment
T_PTS	Double Precision Array, length 300	Array with the temperatures of the calculated saturation points.
P_PTS	Double Precision Array, length 300	Array with the pressures of the calculated saturation points.
RHOLIQ_PTS	Double Precision Array, length 300	Array with the densities of the incipient phase of the calculated saturation points.
RHOVAP_PTS	Double Precision Array, length 300	Array with the densities of the original phase of the calculated saturation points.
POINTS	Integer	Number of calculated points
X_PTS	Double Precision Array, length {300,4}	Array with the composition of the incipient phase of the calculated saturation points.
ERRORFLAG	Integer	Error code

10 TERNARY DIAGRAM CALCULATION

The subroutine TERNARY_DIAG_OUT calculates bubble and dew lines for a given ternary mixture in a ternary phase diagram. No INPUT is needed because the phase diagram is always calculated at constant temperature PROP1 and constant pressure PROP2. The input parameters are PROP, FLUIDS, MOLFRACTIONS, EOSTYPE, MIXTYPE, PATH, FILEOUT and TYPEHANDLE. They are already described in in Table 2 and 10. The subroutine has the following calling sequence:

TERNARY_DIAG_OUT(PROP1;PROP2;FLUIDS;EOSTYPE;MIXTYPE;PATH;X_PTS;RHOVAP_PTS;RHOLIQ_PTS;POINTS;ERRORFLAG; TYPEHANDLE)

The returned parameters are the number of calculated points, the compositions, and densities for each phase. So far, the routine covers three independent two phase regions starting at each of the binary mixtures. The following table summarizes the return parameters, their type, and description.

Table 16: Output parameters of the routine TERNARY_DIAG_OUT

Parameter	Type	Comment
X_PTS	Double Precision Array, length (300,18)	Array with the composition of the incipient phase of the calculated saturation points.
RHOLIQ_PTS	Double Precision Array, length (300,3)	Array with the densities of the incipient phase of the calculated saturation points.
RHOVAP_PTS	Double Precision Array, length (300,3)	Array with the densities of the original phase of the calculated saturation points.
POINTS	Integer	Number of calculated points
ERRORFLAG	Integer	Error code

11 COMPOSITION CONVERSION

The subroutine COMP converts the composition of a mixture between a molar or a specific basis. The subroutine has the following calling sequence:

COMP(FLUIDS; MOLES, EOSTYPE; MIXTYPE; PATH; UNIT; COMPOSITON; ERRORFLAG; TYPEHANDLE)

Please note, that in the array MOLES the input composition must be given, which can be either molar or specific. Furthermore, the given input array must be fully molar or fully specific. A mixed input is not supported. In the following table the Parameters given to the COMP-routine are summarized.

Table 17: Parameters for the subroutine COMP

Parameter	Type	Comment
FLUIDS	Character(30) dimension(30)	List of fluid names of the involved components. The names of the fluids must match the names of the fluid parameter files (.FLD) in the \FLUIDS folder – both in spelling and case! The maximum number of components in a mixture is 20.
MOLES	Double dimension(30)	List of mole fractions of the involved components. (Mole fractions $x < 10^{-14}$ will not be considered)
EOSTYPE	Integer dimension(30)	List of the equation type for each components; see Table 8.
MIXTYPE	Integer	Defines the combining rule of the mixture, see Table 8.
PATH	Character(255)	Path to the directory, where the FLUIDS and BINARY_MIX_FILES folders are located. To use the integrated equations of the DLL, the path has to be named "HC".
UNIT	Character, length(20)	Either 'molar' or 'specific' for the conversion of the input vector moles.
COMPOSITION	Double dimension(30)	Array with the converted fractions (result)
ERRORFLAG	Integer	Internal error variable.
TYPEHANDLE	C_PTR	Pointer to memory address of the global data type gl. The value is 0 on first call and contains the memory address on subsequent calls.

12 CALCULATIONS WITH SEAWATER

The model implemented for seawater is based on the IAPWS Release for Seawater [22]. This model is the most accurate standard property model for seawater available and recommended by IAWPS. Further information can be found in the IAWPS Release and the paper by Feistel [22], [23].

In TREND, seawater is handled as a pseudo pure fluid. Only the 'tp' input is implemented for seawater or seawater mixtures. For using the model in TREND, the input for the fluids has to be set to 'seawater', which has to be followed by 'salinity'. Otherwise, an error will be returned. Equation type for salinity is the same as for the other components. If only seawater is present, the fraction of seawater has to be set to 1. The salinity, which is defined as mass fraction of salt in the liquid seawater, must always be given as mass fraction of salt in seawater. The seawater model is also implemented for mixtures with other fluid phases. At the moment, only the Mixtype 1 is supported (Helmholtz mixtures). In mixtures, the sum of water and salt has to be given as "seawater" followed by the salinity (mass fraction) of salt in liquid seawater. For the proper calculation of mixtures with seawater present, the seawater has to be given as first entry in the fluid list, followed by salinity. For mixtures, only fluid phases are supported with the input 'tp'.

The mixture model with seawater present is a predictive model with no adjusted parameters. Especially, when seawater is present in a mixture, the limited range of validity (see Table 18) of the equation for pure seawater [22],[23] has to be followed for reliable calculations. Otherwise, unreasonable results are returned. Furthermore, it is not recommended to use other routines than the DLL calls given below when properties for seawater or a mixture with seawater are calculated. To apply the source code for calculations with seawater as a present fluid, it requires comprehensive knowledge of thermodynamics of mixtures and the implementation of the model in TREND. A detailed description of the model will be published at a later point of time. Since the routines for calculations with seawater are highly iterative, errors in these iterations might lead to an unreasonable prediction of liquid-liquid equilibrium at single state points when vapor-liquid equilibria for seawater mixtures should be present.

Seawater is supported in the following routines:

TRENDEOS, FLASH3, TRENDSPEC

The range of validity as given in the IAWPS release [22] allows the calculations in the following areas:

Table 18: different ranges of validity for the IAWPS seawater EOS:

$0 \leq S \leq 0.12 \text{ kg kg}^{-1}$;	$261 \text{ K} \leq T \leq 353 \text{ K}$;	$0.101325 \text{ MPa} \leq p \leq 100 \text{ MPa}$
$0 \leq S \leq 0.042 \text{ kg kg}^{-1}$;	$T_f \leq T \leq 313 \text{ K}$;	$p^{\text{vap}} \leq p \leq 0.101325 \text{ MPa}$
$0 \leq S \leq 0.12 \text{ kg kg}^{-1}$;	$T_f \leq T \leq 353 \text{ K}$;	$p = 0.101325 \text{ MPa}$

S is the absolute Salinity, T_f is the freezing temperature and p^{vap} is the vapor pressure.

13 NaCl-Brine

Since TREND 5.0 calculations for thermodynamic properties for NaCl-brine are implemented. The model is based on Pitzer's equations [30]. The parameters within this model were refitted to a comprehensive dataset to allow for the description of the water-part with the IAPWS-95 standard.

Similar to the seawater implementation, the brine is handled as pseudo-pure component. Next to the mass or mol fraction of the brine (1 for a water+NaCl mixture), the molality m (mol_{NaCl}/kg_{H2O}) of NaCl has to be given as input. The molality always needs to be given as moles of NaCl per kg of water in the solution (mol_{NaCl}/kg_{H2O}). At the moment, only properties for a NaCl-brine are implemented, calculations for other salts or mixtures with other components like CO_2 are not supported in this version. Possible input combinations are: 'tp', 'tp+', and 'pmltl+ ' for the calculation of freezing temperatures of the brine. The fluids always have to be given as 'water' and 'nacl' as shown in the example below. In contrast to the other components, the Eq_type for the salt has to be set to 15, while the Mix Type has to remain 1. Up to now, the brine model is only available in combination with Helmholtz equations of state for the solvent water.

Calculations are possible from the freezing point up to T = 373.15 K, pressures up to p = 50 MPa and molalities up to m = 6 mol_{NaCl}/kg_{H2O}.

The brine model is supported in the functions FLASH3 and TRENDEOS for the properties shown in the provided FLASH example of the Excel Spreadsheet.

Example for the input for the calculation of thermodynamic properties; e.g. the specific density for a 2 m NaCl brine with TREND_EOS:

Component	Composition	Eq-Type
water	1	1
nacl	2	15

D = TREND_EOS('d';'tp+';298.15;0.1,{'water', 'nacl'},{1,2};[1,15];1;'HC';'specific';ERRORFLAG;TYPE)

14 SOME NOTES ON COMPILATION, COMPILERS AND CALLING CONVENTIONS

The Win32 DLL provided with this software was compiled using the Intel Visual Fortran Compiler XE, Version 2015.0.6.285. At the time of this release, compilation was tested using the following compilers:

- Intel Parallel Studio XE 2015 Update 6 Composer Edition

Passing strings and arrays to functions or subroutines of a DLL requires some consideration, since it often goes along with communication of different programming languages. For instance, in FORTRAN a string is defined as a variable with a defined length, while in C a string does not have a defined length, the end of the string is here marked with a certain character type. The exchange of strings (and also arrays) between different languages can be performed using a number of different standards, which have to be set in the compiler options.

In case of the TREND.dll the calling conventions were set to "CVF", with the string length argument passing after each individual string. This means, that when strings are passed to the DLL (e.g. the fluid names), an additional parameter (type "Long") has to be passed that indicates the length of the string expected by the DLL.

Example: The double precision FORTRAN function func1 in the DLL "ABC.DLL" with two parameters

```
double        precision        function        func1(MyStrng,        MyDble)
        character(35)::MyStrng
        double                                              precision::MyDble
        func1                           =                                      …
end function func1
```

is called by VBA (the EXCEL programming interface):

```
Public Declare Function func1 Lib "ABC.DLL" (ByVal MyStrng As String, ByVal Len As Long, ByRef dble1
As Double) As Double
'...
Dim returnval As Double
Dim varDble As Double
Dim varStrng As Double
Dim StringLength as Long = 35

returnval = func1(varStrng, StringLength, varDble)
'...
```

This is the most convenient calling convention when using EXCEL to call the DLL. For other applications this calling convention might not be optimal. In such a case the source code provided with this software needs to be recompiled with the desired specifications.

15 REFERENCES

[1] Span, R.: *Multiparameter Equations of State*, Springer, 2000, ISBN: 3-540-67311-3.

[2] Kunz, O., Klimeck, R., Wagner, W., Jaeschke, M.: *The GERG-2004 Wide-Range Equation of state for Natural Gases and Other Mixtures*, Fortsch.-Ber. VDI, Reihe 6, Nr. 557, VDI, Düsseldorf, 2007.

[3] Kunz, O.; Wagner, W. (2012): *The GERG-2008 Wide-Range Equation of State for Natural Gases and Other Mixtures: An Expansion of GERG-2004*. J. Chem. Eng. Data 57(11), 3032 - 3091.

[4] Gernert, J; Span, R. (2016): *EOS–CG: A Helmholtz Energy Mixture Model for Humid Gases and CCS Mixtures*. J. Chem. Thermodyn., 93, 274-293.

[5] Thol, M.; Richter, M.; May, E. F.; Lemmon, E. W.; Span, R. (2019): *EOS-LNG: A Fundamental Equation of State for the Calculation of Thermodynamic Properties of Liquefied Natural Gases*. Accepted at J. Phys. Chem. Ref. Data

[6] Gross, J.; Sadowski, G. (2001): *Perturbed-Chain SAFT: An Equation of State Based on a Perturbation Theory for Chain* Molecules. Ind. Eng. Chem. Res. 40(4), 1244 - 1260.

[7] Gross, J. (2005): *An Equation-of-State Contribution for Polar Components: Quadrupolar Molecules*. AIChE J. 51(9), 2556 - 2568.

[8] Gross, J.; Vrabec, J. (2006): *An Equation-of-State Contribution for Polar Components: Dipolar Molecules*. AIChE J. 52(3), 1194 –1204.

[9] Jäger, A.; Span, R. (2012): *Equation of State for Solid Carbon Dioxide Based on the Gibbs Free Energy*, J. Chem. Eng. Data, 57(2), 590 - 597.

[10] Feistel, R.; Wagner, W. (2006): *A New Equation of State for H_2O Ice Ih*. J. Phys. Chem. Ref. Data, 35(2), 1021 - 1047.

[11] Jäger, A.: *Complex Phase Equilibria of Gas Hydrates and other Solid and Fluid Phases Modeled With Highly Accurate Equations of State*. PhD Thesis. Ruhr-Universität Bochum. Thermodynamics, 2015.

[12] Jäger, A.; Vinš, V.; Gernert, J.; Span, R.; Hrubý, J. (2013): *Phase Equilibria with Hydrate Formation in H_2O+CO_2 Mixtures Modeled With Reference Equations of State*. Fluid Phase Equilib., 338, 100 - 113.

[13] Soave, G. (1972): *Equilibrium Constants from a Modified Redlich-Kwong Equation of State*. Chem. Eng. Sci. 27, 1197 - 1203.

[14] Plöcker, U.; Knapp, H.; Prausnitz, J. (1978): *Calculation of High-Pressure Vapor-Liquid Equilibria from Corresponding-States Correlation with Emphasis on Asymmetric Mixtures,* Ind. Eng. Chem. Process. Des. Dev., 17, 324 - 332.

[15] Peng, D.-Y.; Robinson, D. B. (1976): *A New Two-Constant Equation of State,* Ind. Eng. Chem. Fundam., 15(1), 59 - 64.

[16] Thomson, G. H.; Brobst, K. R.; Hankinson, R. W. (1982): *An Improved Correlation for Densities of Compressed Liquids and Liquid Mixtures*. AIChE J., 28(4), 671 - 676.

[17] McCarty, R. D. (1982): *Mathematical models for the prediction of liquefied-natural-gas densities*. J. Chem. Thermodynamics 14(9), 837-854.

[18] Tietz, C.; Richter, M.; Kleinrahm, R.; Span, R. (2017): *Enhancement of the Revised Klosek and McKinley Method for Density Calculations of Liquefied Natural Gas (LNG) over the Temperature Range from (100 to 135) K at Pressures up to 10 MPa*. Fuel Process. Technol., 165, 19 - 26.

[19] Alexandrov, I.; Gerasimov, A.; Grigor'ev, B. (2013): *Generalized Fundamental Equation of State for the Normal Alkanes (C5-C50)*. Int. J. Thermophys., 34(10), 1865 - 1905.

[20] Sun, L.; Ely, J. F. (2005): *A Corresponding States Model for Generalized Engineering Equations of State*. Int. J. Thermophys., 26(3), 705 - 728.

[21] Kooijman, H. A.; Taylor, R. (1991): *Composition Derivatives of Activity Coefficient Models*. Chem. Eng. Comm. 102, 87 - 106.

[22] International Association for the Properties of Water and Steam (2008), IAPWS R13-08, *Release on the IAPWS Formulation 2008 for the Thermodynamic Properties of Seawater*.

[23] Feistel R. (2008), *A Gibbs Function for Seawater Thermodynamics for −6 to 80°C and Salinity up to 120 g kg^{-1}*, Deep Sea Res. Part Oceanogr. Res. Pap. 55, 1639 - 1671. doi:10.1016/j.dsr.2008.07.004.

[24] Holderbaum, T.; Gmehling, J.: *PSRK: A Group Contribution Equation of State Based on UNIFAC*. Fluid Phase Equilib., 70:251–265, 1991.

[25] Horstmann, S.; Jabłoniec, A.; Krafczyk, J.; Fischer, K.; Gmehling, J.: *PSRK group contribution equation of state: comprehensive revision and extension IV, including critical constants and α-function parameters for 1000 components*. Fluid Phase Equilib., 227(2):157–164, January 2005.

[26] Jäger, A.; Bell, I.H.; Breitkopf, C.*: A theoretically based departure function for multi-fluid mixture models.* Fluid Phase Equilibria 469, 56-69, 2018

[27] Lin, S.; Sandler, S.I.: *A Priori Equilibrium Prediction from a Segment Contribution Solvation Model*. Ind. Eng. Chem. Res. 41, 899-913, 2002.

[28] Jäger, A.; Mickoleit, E.; Breitkopf, C.: *A combination of multi-fluid mixture models with COSMO-SAC*. Fluid Phase Equilibria 476, 147-156, 2018.

[29] AGA Transmission Measurement Committee, Compressibility and Supercompressibility for Natural Gas and Other Hydrocarbon Gases, AGA Transmission Measurement Com. Rep. No. 8 (Arlington, Virginia, 1985).

[30] Pitzer, K.S.; ed., Activity coefficients in electrolyte solutions, 2nd ed, CRC Press, 1991

[31] Herrig, S. (2018): *New Helmholtz-Energy Equations of State for Pure Fluids and CCS-Relevant Mixtures*

[32] Mullins, E.; Oldland, R.; Liu, Y. A.; Wang, S.; Sandler, S. I.; Chen, C.-C.; Zwolak, M.; Seavey, K. C. Sigma-Profile Database for Using COSMO-Based Thermodynamic Methods. Ind. Eng. Chem. Res. 2006, 45 (12), 4389–4415. https://doi.org/10.1021/ie060370h.

[33] Bell, I.H.; Mickoleit, E.; Hsieh, C.-M.; Lin, S.-T.; Vrabec, J.; Breitkopf, C.; Jäger, A. (2020): A Benchmark Open-Source Implementation of COSMO-SAC, J. Chem. Theory Comput. 16(4), 2635–2646

[34] Hsieh, C.-M.; Sandler, S. I.; Lin, S.-T.: Improvements of COSMO-SAC for Vapor–Liquid and Liquid–Liquid Equilibrium Predictions. Fluid Phase Equilibria 2010, 297, 90–97.

[35] Hielscher, S.; Vinš, V.; Jäger, A.; Hruby, J.; Breitkopf, C.; Span, R.(2018): A new approach to model mixed hydrates, Fluid Phase Equilibria. 459 170-185

[36] Hielscher, S.; Semrau, B.; Jäger, A.; Vinš, V.; Breitkopf, C.; Hrubý, J.; Span, R. (2019): Modification of a model for mixed hydrates to represent double cage occupancy, Fluid Phase Equilibria. 490 48–60.

16 ERROR CODES

If an error occurs during the calculation of any property, a code will be printed, which is a negative integer number. The possible errors with an explanation are given in Table 19.

For an easier identification of the error, an error function is available:

EXPLANATION=ERRORTEXT(errornumber)

The input parameter is the errornumber according to the first column of Table 19. The return value is of the type character(255).

Table 19. Error codes with explanation

Error	Comment
-1111	MIX: Wrong input(s) to a routine. These are internal errors, for example negative temperatures, wrong iFlash values etc. are caught
-1234	PURE: PhaseDet, p(rho_it,T_it)
-1235	PURE: Temperature iteration T(p,h) failed
-1444	Function Fug_DryIce: FUGCOPURE_CALC: Exponent > 700. Argument to exponential function too big in dry ice fugacity calculation
-2111	PHASEDET: Subroutine RachRiche error during generation of startvalues
-2211	Subroutine PHASEDET_PURE: Flash calculation did not converge
-2212	Subroutine MAXWELL: Flash calculation with MAXWELL did not converge
-2222	FLASH: Flash calculation did not converge
-2223	Function TSUB_EQ: SUBLIMATION TEMPERATURE iteration did not converge (ancillary equation)
-2224	PHASEDET_SOL: MELTING TEMPERATURE iteration did not converge (ancillary equation)
-2225	PHASEDET_SOL: SUBLIMATION PRESSURE calculation failed
-2226	PHASEDET_SOL: MELTING PRESSURE calculation failed
-2230	PHASEDET_SOL: MELTING LINE: Given pressure not valid
-2231	PHASEDET_SOL: MELTING LINE: Given temperature not valid
-2232	PHASEDET_SOL: SUBLIMATION LINE: Given pressure not valid
-2233	PHASEDET_SOL: SUBLIMATION LINE: Given temperature not valid
-2408	Internal Error in Flash_Pure_PhaseBoundary (found same density roots)
-2908	cubic, LKP or generalized equation: No cp0 parameters set
-3333	MIX: Step size reduction error in flash algorithm (step size becomes too small)
-4321	MIX: Flash algorithm converged to unreasonable temperatures
-4322	Subroutine FLASH_PHASEBOUNDARY_CALC: MIX flash algorithm converged to unreasonable pressures
-4323	MIX: Flash algorithm converged to two phases with the same compositions
-4401	Subroutine FLASH_PHASEBOUNDARY: MIX p,x(liq)-flash failed: no bubble point found
-4402	Subroutine FLASH_PHASEBOUNDARY: MIX p,x(vap)-flash failed: no dew point found
-4403	Subroutine FLASH_PHASEBOUNDARY: MIX t,x(liq)-flash failed: no bubble point found
-4404	Subroutine FLASH_PHASEBOUNDARY: MIX t,x(vap)-flash failed: no dew point found
-4405	MIX: ph-flash internal error
-4406	Subroutine PHASEDET_PS: MIX ps-flash internal error
-4407	MIX: td-flash internal error
-4408	CRITICAL POINT FOR MIXTURES: internal error

Error	Comment
-4409	Calculation of Henry constant not possible because temperature is lower than triple point temperature or higher than critical temperature
-4444	MIX: LUDECOMP failed to invert matrix
-5211	Surface tension not implemented for mixtures
-5222	Surface tension not defined in the homogeneous region
-5223	Property for mixtures not implemented
-5234	No model for this property and fluid available
-5235	Dielectric constant models cannot be mixed with model DE2
-5242	Error in viscosity calculation
-5243	Existing model for the viscosity not implemented yet
-5244	Existing model for the thermal conductivity not implemented yet
-5245	No viscosity model exists, which is needed for the calculation of the critical enhancement of the thermal conductivity
-5246	VS7 model is treated as VS9 model. Rename it to VS7 and adjust the format in the FLD file.
-5247	Literature reference only available for pure fluids and binary mixtures
-5248	Property only available for mixtures
-5249	Property only available for binary mixtures
-5250	Quality not implemented as input parameter for mixtures
-5251	This diagram is not defined for the specified number of fluids
-5300	ECS: invalid reference fluid
-5301	ECS: internal error
-5302	ECS: iteration failed
-5400	Internal error while iterating the melting temperature
-5501	Creation of four points at low pressures failed: generation of starting point failed. Stability analysis failed.
-5502	Phase envelope: generation of start values failed
-5503	Phase envelope: generation of start values failed
-5504	Successive substitution routine failed
-5507	Subroutine PHASENV: MIX: phase envelope calculation failed (step reduction becomes too large)
-5508	Subroutine PHASENV: MIX: phase envelope calculation failed
-5509	Subroutine PHASENV_pbased or PHASENV_vbased: MIX: phase envelope calculation failed
-5511	p_min for "normal" phase env. OR p_max for open phase env. reached OR exit if the specified point is found and the routine is called from the phase boundary flash algorithm OR one of these variables does not change any more: T, p, rho_vap
-5520	Subroutine PHASENV: MIX: too many points found -> exceeds the vector storage capacity
-5530	Subroutine pxdiag: T is larger than tc(1) and tc(2)
-5531	Subroutine ptxdiag_out: prop is below 0.5*min(tminfluid) or 0.5*min(pminfluid) respectively
-5555	Subroutine SATPLOT: MIX: Phase envelope calculation failed
-5566	Subroutine SYSOFFEQS_PT: MIX: Internal error in pt-flash
-5660	Density maximum not implemented for mixtures or wrong input code (only TP valid)
-5661	Iteration of density maximum failed
-5662	Iteration of temperature failed
-5666	Function DSPIN_EOS: Spinodals: No valid input code, only TLIQ and TVAP possible
-5667	Function DSPIN_CALC: Spinodals: Change in curvature between minimum and phase boundary (unreasonable result)

Error	Comment
-5668	Function DSPIN_EOS: Spinodales not implemented for mixtures
-5669	Function DSPIN_EOS: Error when calculating the density of the spinodal
-5700	Function PROP_EOS: Valid input combination for PNUMER_EOS is "TD" only
-5777	Costald EOS: Different EOS types cannot be combined here
-5778	Function PROP_EOS: Costald EOS: No valid input code, only TLIQ or TP possible or wrong property (only density possible)
-5888	Function VSATTAIT: Costald EOS: Temperature out of valid temperature range
-5997	Function PROP_EOS: RKM EOS: No valid input code, only TLIQ possible or wrong property (only density possible)
-5998	Function PROP_EOS: ERKM EOS: No valid input code, only TLIQ or TP possible or wrong property (only density possible)
-5999	Function v_RKM: Revised Klosek-McKinley EOS: Temperature and or pressure out of valid range
-6000	Subroutine REF_CALC: No reference state set
-6001	Acentric factor for generalized negative => calculation not possible
-6600	No strict equation boundaries available
-6666	Property in two-phase region undefined
-6667	Property in homogeneous region undefined
-7000	UNCERTAINTIES: Estimation of uncertainties only possible for pure fluids
-7001	UNCERTAINTIES: Uncty File not found
-7002	UNCERTAINTIES: Estimation of uncertainties only possible for Helmholtz EOS (see uncty file)
-7003	UNCERTAINTIES: estimation in 2phase region not yet implemented
-7777	MIX: Calculation of vapor fugacities failed
-7778	MIX: Calculation of liquid fugacities failed
-7779	MIX: Calculation of solid fugacities failed
-7877	DLL could not be opened in Excel
-7878	SETUP: Fluid name or path wrong
-7879	SETUP Error: Opening fluid file, does not exist in CAS list
-7880	SETUP Error: End of fluid-file is reached during read
-7881	SETUP Error: Opening .MIX file
-7882	SETUP Error: End of .MIX-file is reached during read
-7883	SETUP Error: Opening atcoeff file
-7884	SETUP Error: CAS list was not found
-7885	SETUP Error: Opening SRK file
-7886	SETUP Error: Opening PR file
-7887	SETUP Error: Opening LKP file
-7888	SETUP Error: Opening BIN SRK or PR file
-7889	SETUP Error: Opening BIN LKP file
-7890	SETUP Error: CAS-ID not found
-7891	SETUP Error: Opening COSTALD file
-7892	SETUP Error: Only Helmholtz EOS, PR, SRK, LKP and the corresponding mixing rules are hardcoded at the moment
-7893	SETUP Error: Opening RKM files
-7894	SETUP Error: Read error in srkfluids.fld
-7895	SETUP Error: Read error in bin_srk.mix or bin_pr.mix
-7896	SETUP Error: Read error in rk_qk.par

Error	Comment
-7897	SETUP Error: Read error in Interac.par
-7898	SETUP Error: COSMO-SAC
-7899	COSMO-SAC_CALC Error
-7999	Wrong equation format (e.g. ECS)
-8000	Internal error Helmholtz_reader (sub_part found twice)
-8001	Internal error: Read error in Costald.fld
-8002	RM: Fluid components do not match the 8 possible components of the RKM
-8877	PURE: Density iteration failed
-8878	Fluid not found
-8879	gE-Model: Fluid or file not found
-8880	Costald fluid not found
-8881	Costald parameter file not found
-8888	Density iteration failed
-8889	find_crit_tpd: iteration of critical point failed
-9000	SETUP Error: Parameter file for generalized EOS not found
-9876	NAN in FNRDERIVSMIX - fugcoef_pure, SRK, dlog(<0)
-9901	Calculation not possible for pure fluids
-9902	Calculation not possible for mixtures
-9903	Calculation not possible for pure substances or binary mixtures
-9904	Calculation not possible for solid phase of chosen substance
-9907	Pressure <= lower pressure limit of the corresponding equation
-9908	Pressure >= lower pressure limit of the corresponding equation
-9909	Temperature <= lower temperature limit of the corresponding equation
-9910	Temperature >= upper temperature limit of the corresponding equation
-9911	Temperature input <= 0
-9912	Temperature <= Tminfluid
-9913	Temperature >= Tmaxfluid
-9914	Density >= Rhomaxfluid
-9915	p >= pmelt --> solid phase
-9916	Input parameters out of range (viscosity of water)
-9917	ALLPROP_HOM: phase_ind out of range (phase_ind is an input parameter that specifies the phase which is tried)
-9918	Temperature <= Tmin of transport equations
-9919	Temperature >= Tmax of transport equations
-9920	Density >= rhomax of transport equations
-9922	Density input <= 0
-9932	Pressure input >= Pmaxfluid
-9933	Pressure input <= 0
-9935	Property not available
-9936	Input Property out of range (seawater or brines); see manual for range of validity
-9942	Entropy input out of range
-9941	Enthalpy input out of range
-9943	T rho flash, density cannot be found in the given range of pmin to pmax
-9950	Error in composition convertion

Error	Comment
-9951	One mole fraction exceeds 0 < x < 1
-9952	Sum of the mole fractions is not 1
-9953	Number of fluids =/ number of mole fractions =/ number of eqtypes
-9954	Error in composition entry
-9955	Wrong Input (combination)
-9956	AGA8: Wrong Input, only TD is valid
-9957	EOS_Indicator does not exist or wrong mixtype
-9958	Cubic EOS cannot be mixed with Helmholtz EOS
-9959	Error in EOS_Indicator entry, wrong format
-9960	Error in fluid entry (e.g. fluidname contains blanks)
-9961	Option only valid for Helmholtz, PR, SRK, and LKP
-9971	Negative radical in calculation of speed of sound
-9981	Call of Flash_pure with Temp <= ttp or Temp >= tc
-9982	Call of Flash_pure with press <= ptp or press >= pc
-9994	AGA8: Only valid in gas phase
-9995	AGA8: A fnr-function is needed for the calculation, which is not provided by the AGA8
-9996	AGA8: Fluid components do not match the 21 possible components of the AGA8
-9997	AGA8: Density out of range (valid only up to 50% of critical density of mixture)
-9998	AGA8: Temperature out of range (valid between 143.15K and 453.15K)
-9999	AGA8: Pressure out of range (valid up to 280MPa)
-12800	Property not implemented for seawater or brines
-12900	Property not provided / implemented for solid equation
-12901	Input combination not implemented for solids
-12902	No solid model available for chosen substances
-12910	Input combination not implemented for seawater or brines
-14444	Wrong value in solidtype(1). Either internal error, or the solid former does not exist
-15566	Internal error in PhaseDet_sol
-15567	Internal error in PhaseDet_ph_ps_sol
-15570	Internal error in Flash_PhaseBoundary_sol
-15571	Internal error in ptflash_sol_NC_4P
-15572	Internal error in ptflash_sol_NC_2P
-18867	Pressure iteration for solid failed
-19900	Prediction of solids only possible in combination with Helmholtz EOS
-19912	Temperature <= tmin of solid equation
-19914	Density >= rhomax for solid
-19915	p >= pmelt -->
-19932	Pressure >= pmax for solid equation
-19941	Enthalpy or entropy input out of range
-44444	PURE: Density iteration failed (rho_calc)
-101010	Internal error in Phasedet_sol for mixtures
-111111	Solid function not activated
-121212	Error while generating start values for VLE pure, press or Temp <= 0
-121213	Invalid iFlash entry in VLE pure
-123456	Error in molar specific handling (only "molar", "specific", or "reduced" allowed)

Error	Comment
-123457	Prop_unit subroutine: Invalid Property
-898964	Flash_Pure_PhaseBoundary_calc: Density iteration failed
-898965	Flash_Pure_PhaseBoundary_calc did not converge in specified number of iterations
-898966	Flash_Pure_PhaseBoundary_calc: safety caution for numerical effects (see oil, Pvap is 1.d-10 --> very small steps)
-898967	Flash_Pure_PhaseBoundary_calc: wrong solution (wrong slope)
-898968	Flash_Pure_PhaseBoundary_calc: pure fluid in two phase region, density cannot be determined