An open-source thermodynamic software library

Documentation pages for v. 1.1.7

Tobias K. S. Ritschel, Jozsef Gaspar, Andrea Capolei, John Bagterp Jørgensen

Department of Applied Mathematics and Computer Science and Center for Energy Resources Engineering (CERE),
Technical University of Denmark, DK-2800 Kgs. Lyngby, Denmark

Abstract

This document contains documentation of Matlab and C routines in the thermodynamic software library Compute ThermoLib. For Matlab/Mex routines, this documentation is also available by help <name-of-routine>. For C routines, the documentation can also be found in the source code files. You may want to consult the official website of Compute ThermoLib for instructions on installation, interfacing with the DIPPR database and brief tutorials illustrating the use of the library functions for selected Matlab and C functions. In general, the Matlab and C interfaces are constructed to be similar which is also reflected by the documentation for each of the routines being close to identical. The key difference is that the C routines expect you as the user to allocate memory, both for the outputs and for an auxiliary array of memory. The C documentation shows how much memory should be allocated.

Keywords: Thermodynamic software library, documentation pages, Matlab, C
## Contents

1. LoadParams (Matlab) 5
2. MixRealHSV (Matlab/Mex) 7
3. MixRealVapHSV (Matlab/Mex) 9
4. MixRealLiqHSV (Matlab/Mex) 11
5. MixResPhHSV (Matlab/Mex) 13
6. MixFug (Matlab/Mex) 16
7. MixSolvePhEoS (Matlab/Mex) 18
8. MixIdHSV (Matlab/Mex) 20
9. MixIdVapHSV (Matlab/Mex) 22
10. MixIdLiqHSV (Matlab/Mex) 24
11. PureRealHSV (Matlab/Mex) 26
12. PureRealVapHSV (Matlab/Mex) 29
13. PureRealLiqHSV (Matlab/Mex) 31
14. PureResHSV (Matlab/Mex) 33
15. PureResPhHSV (Matlab/Mex) 36
16. PureFug (Matlab/Mex) 38
17. PureSolveEoS (Matlab/Mex) 40
18. PureSolvePhEoS (Matlab/Mex) 42
19. PureIdVapHSV (Matlab/Mex) 44
20. PureIdLiqHSV (Matlab/Mex) 46
21. MixParams (Matlab/Mex) 48
22. PureParams (Matlab/Mex) 50
23. SolvePolynomiumCardano (Matlab/Mex) 52
24. SolvePolynomiumNewton (Matlab/Mex) 53
25. IdGasHeatCap (Matlab/Mex) 55
<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>26</td>
<td>IdGasHeatCapInt (Matlab/Mex)</td>
</tr>
<tr>
<td>27</td>
<td>IdLiqVol (Matlab/Mex)</td>
</tr>
<tr>
<td>28</td>
<td>IdSatPres (Matlab/Mex)</td>
</tr>
<tr>
<td>29</td>
<td>LoadParams (C)</td>
</tr>
<tr>
<td>30</td>
<td>MixRealHSV (C)</td>
</tr>
<tr>
<td>31</td>
<td>MixRealVapHSV (C)</td>
</tr>
<tr>
<td>32</td>
<td>MixRealLiqHSV (C)</td>
</tr>
<tr>
<td>33</td>
<td>MixResPhHSV (C)</td>
</tr>
<tr>
<td>34</td>
<td>MixSolvePhEoS (C)</td>
</tr>
<tr>
<td>35</td>
<td>MixIdHSV (C)</td>
</tr>
<tr>
<td>36</td>
<td>MixIdVapHSV (C)</td>
</tr>
<tr>
<td>37</td>
<td>MixIdLiqHSV (C)</td>
</tr>
<tr>
<td>38</td>
<td>PureRealHSV (C)</td>
</tr>
<tr>
<td>39</td>
<td>PureRealVapHSV (C)</td>
</tr>
<tr>
<td>40</td>
<td>PureRealLiqHSV (C)</td>
</tr>
<tr>
<td>41</td>
<td>PureResHSV (C)</td>
</tr>
<tr>
<td>42</td>
<td>PureResPhHSV (C)</td>
</tr>
<tr>
<td>43</td>
<td>PureSolveEoS (C)</td>
</tr>
<tr>
<td>44</td>
<td>PureSolvePhEoS (C)</td>
</tr>
<tr>
<td>45</td>
<td>PureIdHSV (C)</td>
</tr>
<tr>
<td>46</td>
<td>PureIdVapHSV (C)</td>
</tr>
<tr>
<td>47</td>
<td>PureIdLiqHSV (C)</td>
</tr>
<tr>
<td>48</td>
<td>MixParams (C)</td>
</tr>
<tr>
<td>49</td>
<td>PureParams (C)</td>
</tr>
<tr>
<td>50</td>
<td>SolvePolynomiumNewton (C)</td>
</tr>
<tr>
<td>51</td>
<td>IdGasHeatCap (C)</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>52</td>
<td>IdGasHeatCapInt (C)</td>
</tr>
<tr>
<td>53</td>
<td>IdLiqVol (C)</td>
</tr>
<tr>
<td>54</td>
<td>IdSatPres (C)</td>
</tr>
</tbody>
</table>
1. LoadParams (Matlab)

Load thermodynamic parameters from the DIPPR database into array

SYNOPSIS:
params = LoadDIPPRParameters(comp)
params = LoadDIPPRParameters(comp, EoS)
params = LoadDIPPRParameters(comp, EoS, k)
params = LoadDIPPRParameters(comp, epsilon, sigma, Omega, Psi, m, k)

DESCRIPTION:
Loads thermodynamic and equation of state parameters for the specified
components into an array that is used by the thermodynamic routines
provided by this library.

REQUIRED PARAMETERS:
comp - List of component numbers (see <path-to-library>/matlab/data/DataName.m)

OPTIONAL PARAMETERS:
EoS - String specifying whether to use PR or SRK equation of state (case-insensitive)
epsilon - Equation of state parameter
sigma - Equation of state parameter
Omega - Equation of state parameter
Psi - Equation of state parameter
m(omega) - Polynomium evaluated at acentricity factors
k - Symmetric matrix containing the van der Waals mixing rule parameters

RETURNS:
params - Vector containing all parameters needed thermodynamic functions

DEPENDENCIES:
<path-to-library>/matlab/data/DIPPRdata.mat

See also
LoadParams
PureRealHSV MixRealHSV
PureRealVapHSV MixRealVapHSV
PureRealLiqHSV MixRealLiqHSV
PureFug MixFug
PureResHSV
PureResPhHSV MixResPhHSV
PureSolveEoS
PureSolvePhEoS MixSolvePhEoS
PureRealSatTemp
PureRealSatPres
PureEvalEoS MixEvalEoS
PureIdHSV MixIdHSV
PureIdVapHSV MixIdVapHSV
<table>
<thead>
<tr>
<th>PureIdLiqHSV</th>
<th>MixIdLiqHSV</th>
</tr>
</thead>
<tbody>
<tr>
<td>PureParams</td>
<td>MixParams</td>
</tr>
<tr>
<td>PureIdSatTemp</td>
<td>MixParams</td>
</tr>
<tr>
<td>IdGasHeatCap</td>
<td>MixParams</td>
</tr>
<tr>
<td>IdGasHeatCapInt</td>
<td>MixParams</td>
</tr>
<tr>
<td>IdLiqVol</td>
<td>MixParams</td>
</tr>
<tr>
<td>IdSatPres</td>
<td>MixParams</td>
</tr>
<tr>
<td>SolvePolynomialNewton</td>
<td>MixParams</td>
</tr>
</tbody>
</table>
2. MixRealHSV (Matlab/Mex)

Compute enthalpy, entropy and volume of a real liquid mixture

SYNOPSIS:

\[
\begin{align*}
[H_v, S_v, V_v, H_l, S_l, V_l, \ldots \\
\quad dH_v, dS_v, dV_v, dH_l, dS_l, dV_l, \ldots \\
\quad d^2H_v, d^2S_v, d^2V_v, d^2H_l, d^2S_l, d^2V_l] = \ldots
\end{align*}
\]

MixRealHSV(T, P, nv, nl, params, tol, itmax)

DESCRIPTION:

Computes enthalpy, entropy and volume of a real liquid mixture together with first and second order temperature and pressure derivatives. Derivatives are computed based on the number of output arguments.

The output is formatted such that for the first order derivatives

- Element 1: temperature derivative
- Element 2: pressure derivative
- Element 3+: composition derivatives

and for the symmetric second order derivatives

- Element (1, 1): temperature derivative
- Element (2, 1): temperature and pressure derivative
- Element (2, 2): pressure derivative
- Elements (3+, 1): temperature and composition derivatives
- Elements (3+, 2): pressure and composition derivatives
- Elements (3+, 3+): composition derivatives

REQUIRED PARAMETERS:

- T - Temperature [K]
- P - Temperature [Pa]
- nv - mole numbers in vapor phase [kmol]
- nl - mole numbers in liquid phase [kmol]
- params - Vector with various parameters obtained by calling LoadParams

OPTIONAL PARAMETERS:

- tol - Tolerance for Newton iterations
- itmax - Maximum number of Newton iterations

RETURNS:

- H_v - Enthalpy of vapor phase
- S_v - Entropy of vapor phase
- V_v - Volume of vapor phase
- H_l - Enthalpy of liquid phase
S1  - Entropy of liquid phase
Vl  - Volume of liquid phase
dHv - First order derivatives of vapor enthalpy
dSv - First order derivatives of vapor entropy
dVv - First order derivatives of vapor volume
dHl - First order derivatives of liquid enthalpy
dSl - First order derivatives of liquid entropy
dVl - First order derivatives of liquid volume
d2Hv - Second order derivatives of vapor enthalpy
d2Sv - Second order derivatives of vapor entropy
d2Vv - Second order derivatives of vapor volume
d2Hl - Second order derivatives of liquid enthalpy
d2Sl - Second order derivatives of liquid entropy
d2Vl - Second order derivatives of liquid volume

DEPENDENCIES:
MixIdVapHSV
MixResPhHSV

See also LoadParams
PureRealHSV          MixRealHSV
PureRealVapHSV       MixRealVapHSV
PureRealLiqHSV       MixRealLiqHSV
PureFug              MixFug
PureResHSV           MixResHSV
PureResPhHSV         MixResPhHSV
PureSolveEoS         MixSolveEoS
PureSolvePhEoS       MixSolvePhEoS
PureRealSatTemp      MixRealSatTemp
PureRealSatPres      MixRealSatPres
PureEvalEoS          MixEvalEoS
PureIdHSV            MixIdHSV
PureIdVapHSV         MixIdVapHSV
PureIdLiqHSV         MixIdLiqHSV
PureParams           MixParams
PureIdSatTemp        MixIdSatTemp
IdGasHeatCap         MixIdGasHeatCap
IdGasHeatCapInt      MixIdGasHeatCapInt
IdLiqVol             MixIdLiqVol
IdSatPres            MixIdSatPres
SolvePolynomialNewton
3. MixRealVapHSV (Matlab/Mex)

Compute enthalpy, entropy and volume of a real vapor mixture

SYNOPSIS:

\[
[H_v, S_v, V_v, \ldots ] \\
\text{d}H_v, \text{d}S_v, \text{d}V_v, \ldots \\
\text{d}^2H_v, \text{d}^2S_v, \text{d}^2V_v] = \ldots \\
\text{MixRealVapHSV}(T, P, nv, \text{params}, \text{tol}, \text{itmax})
\]

DESCRIPTION:
Computes enthalpy, entropy and volume of a real vapor mixture together with first and second order temperature and pressure derivatives. Derivatives are computed based on the number of output arguments.

The output is formatted such that for the first order derivatives

Element 1 : temperature derivative
Element 2 : pressure derivative
Element 3+: composition derivatives

and for the symmetric second order derivatives

Element (1, 1) : temperature derivative
Element (2, 1) : temperature and pressure derivative
Element (2, 2) : pressure derivative
Element (3+, 1) : temperature and composition derivatives
Element (3+, 2) : pressure and composition derivatives
Element (3+, 3+) : composition derivatives

REQUIRED PARAMETERS:
\begin{align*}
T & \quad \text{- Temperature [K]} \\
P & \quad \text{- Temperature [Pa]} \\
nv & \quad \text{- mole numbers in vapor phase [kmol]} \\
\text{params} & \quad \text{- Vector with various parameters obtained by calling LoadParams}
\end{align*}

OPTIONAL PARAMETERS:
\begin{align*}
\text{tol} & \quad \text{- Tolerance for Newton iterations} \\
\text{itmax} & \quad \text{- Maximum number of Newton iterations}
\end{align*}

RETURNS:
\begin{align*}
H_v & \quad \text{- Enthalpy of vapor phase} \\
S_v & \quad \text{- Entropy of vapor phase} \\
V_v & \quad \text{- Volume of vapor phase} \\
\text{d}H_v & \quad \text{- First order derivatives of vapor enthalpy} \\
\text{d}S_v & \quad \text{- First order derivatives of vapor entropy}
\end{align*}
First order derivatives of vapor volume
Second order derivatives of vapor enthalpy
Second order derivatives of vapor entropy
Second order derivatives of vapor volume

DEPENDENCIES:
MixIdVapHSV
MixResPhHSV

See also LoadParams
PureRealHSV MixRealHSV
PureRealVapHSV MixRealVapHSV
PureRealLiqHSV MixRealLiqHSV
PureFug MixFug
PureResHSV
PureResPhHSV MixResPhHSV
PureSolveEoS MixSolvePhEoS
PureRealSatTemp
PureRealSatPres
PureEvalEoS MixEvalEoS
PureIdHSV MixIdHSV
PureIdVapHSV MixIdVapHSV
PureIdLiqHSV MixIdLiqHSV
PureParams MixParams
PureIdSatTemp
IdGasHeatCap
IdGasHeatCapInt
IdLiqVol
IdSatPres
SolvePolynomialNewton
4. MixRealLiqHSV (Matlab/Mex)

Compute enthalpy, entropy and volume of a real liquid mixture

SYNOPSIS:

\[ [H_l, S_l, V_l, \ldots ] \]
\[ dH_l, dS_l, dV_l, \ldots \]
\[ d^2H_l, d^2S_l, d^2V_l ] = \ldots \]
\[ \text{MixRealLiqHSV}(T, P, n_l, \text{params}, \text{tol}, \text{itmax}) \]

DESCRIPTION:

Computes enthalpy, entropy and volume of a real liquid mixture together with first and second order temperature and pressure derivatives. Derivatives are computed based on the number of output arguments.

The output is formatted such that for the first order derivatives

Element 1: temperature derivative
Element 2: pressure derivative
Element 3+: composition derivatives

and for the symmetric second order derivatives

Element (1, 1): temperature derivative
Element (2, 1): temperature and pressure derivative
Element (2, 2): pressure derivative
Elements (3+, 1): temperature and composition derivatives
Elements (3+, 2): pressure and composition derivatives
Elements (3+, 3+): composition derivatives

REQUIRED PARAMETERS:

\text{T} - Temperature \([\text{K}]\)
\text{P} - Temperature \([\text{Pa}]\)
\text{n}_l - mole numbers in liquid phase \([\text{kmol}]\)
\text{params} - Vector with various parameters obtained by calling \text{LoadParams}

OPTIONAL PARAMETERS:

\text{tol} - Tolerance for Newton iterations
\text{itmax} - Maximum number of Newton iterations

RETURNS:

\text{H}_l - Enthalpy of liquid phase
\text{S}_l - Entropy of liquid phase
\text{V}_l - Volume of liquid phase
\text{dH}_l - First order derivatives of liquid enthalpy
\text{dS}_l - First order derivatives of liquid entropy
dVl - First order derivatives of liquid volume

d2Hl - Second order derivatives of liquid enthalpy

d2Sl - Second order derivatives of liquid entropy

d2Vl - Second order derivatives of liquid volume

DEPENDENCIES:
MixIdVapHSV
MixResPhHSV

See also LoadParams
PureRealHSV MixRealHSV
PureRealVapHSV MixRealVapHSV
PureRealLiqHSV MixRealLiqHSV
PureFug MixFug
PureResHSV
PureResPhHSV MixResPhHSV
PureSolveEoS MixSolvePhEoS
PureRealSatTemp
PureRealSatPres
PureEvalEoS MixEvalEoS
PureIdHSV MixIdHSV
PureIdVapHSV MixIdVapHSV
PureIdLiqHSV MixIdLiqHSV
PureParams MixParams
PureIdSatTemp
IdGasHeatCap
IdGasHeatCapInt
IdLiqVol
IdSatPres
SolvePolynomiunNewton
5. MixResPhHSV (Matlab/Mex)

Compute volume and residual enthalpy and entropy of phase

SYNOPSIS:
[h, s, v,
 dh, ds, dv,
 d2h, d2s, d2v] = ...
MixResPhHSV(T, P, n, phase, params, tol, itmax)

DESCRIPTION:
Computes volume and residual enthalpy and entropy of a real mixture together with
first and second order temperature and pressure derivatives. Derivatives
are computed based on the number of output arguments. The enthalpy and
entropy are obtained from a cubic equation of state

\[ h^R(T, P) = RT (Z - 1) + 1/((\epsilon - \sigma) b_m)(T \frac{d a_m}{dT} - a_m(T))f(Z, B) \]
\[ s^R(T, P) = R \ln(Z - 1) + 1/((\epsilon - \sigma) b_m) \frac{d a_m}{dT} f(Z, B) \]

where

\[ f(Z, B) = \ln\left( \frac{Z + \epsilon B}{Z + \sigma B} \right) \]

The volume is obtained by solution of a cubic equation of state
and the quadratic van der Waals mixing rules

\[ a_m(T, n) = \sum_{i=1}^{N_C} \sum_{j=1}^{N_C} x_i x_j a_{ij}(T) \]
\[ b_m(n) = \sum_{i=1}^{N_C} x_i b_{ij} \]
\[ a_{ij}(T) = (1 - k_{ij}) \sqrt{a_i(T) a_j(T)} \]
\[ x_i = n_i / \sum_{j=1}^{N_C} n_j \]
\[ a_i(T) = \alpha(Tr, \omega) \Psi (RT_c)^2/P_c \]
\[ b_i = \Omega RT_c/P_c \]
\[ \alpha(Tr, \omega) = (1 + m(\omega)\sqrt{1 - Tr})^2 \]
\[ m(\omega) = m_0 + m_1 \omega + m_2 \omega^2 \]
\[ Tr = T/T_c \]

The output is formatted such that for the first order derivatives

Element 1: temperature derivative
Element 2: pressure derivative
Element 3+: composition derivatives

and for the symmetric second order derivatives

Element (1, 1): temperature derivative
Element (2, 1): temperature and pressure derivative
Element (2, 2): pressure derivative
Elements (3+, 1): temperature and composition derivatives
Elements (3+, 2): pressure and composition derivatives
Elements (3+, 3+): composition derivatives

REQUIRED PARAMETERS:
T - Temperature [K]
P - Temperature [Pa]
n - mole numbers [kmol]
phase - 0: vapor, 1: liquid
params - Vector with various parameters obtained by calling
LoadParams

OPTIONAL PARAMETERS:
tol - Tolerance for Newton iterations
itmax - Maximum number of Newton iterations

RETURNS:
h - Enthalpy
s - Entropy
v - Volume
dh - First order derivatives of enthalpy
ds - First order derivatives of entropy
dv - First order derivatives of volume
d2h - Second order derivatives of enthalpy
d2s - Second order derivatives of entropy
d2v - Second order derivatives of volume

DEPENDENCIES:
MixParams
MixSolveEoS

See also LoadParams
PureRealHSV MixRealHSV
PureRealVapHSV MixRealVapHSV
PureRealLiqHSV MixRealLiqHSV
PureFug MixFug
PureResHSV
PureResPhHSV MixResPhHSV
PureSolveEoS MixSolvePhEoS
PureRealSatTemp
PureRealSatPres
PureEvalEoS MixEvalEoS
PureIdHSV MixIdHSV
PureIdVapHSV MixIdVapHSV
PureIdLiqHSV MixIdLiqHSV
PureParams MixParams
I4
PureIdSatTemp
IdGasHeatCap
IdGasHeatCapInt
IdLiqVol
IdSatPres
SolvePolynomiumNewton
6. MixFug (Matlab/Mex)

Compute logarithmic fugacity coefficients of a real mixture

SYNOPSIS:

\[ \text{[lnphi, dlnphi, d2lnphi]} = \text{MixFug}(T, P, n, \text{phase}, \text{params}, \text{tol}, \text{itmax}) \]

DESCRIPTION:

Computes fugacities of a mixture together with first and second order temperature, pressure and composition derivatives. Derivatives are computed based on the number of output arguments. The fugacities are obtained from a cubic equation of state

\[
\ln \phi_i(T, P) = (Z - 1) \frac{b_i}{b_{mix}} - \ln(Z - B) - \frac{1}{(\epsilon - \sigma)} \frac{1}{(RT) b_{mix}} \sum_{j=1}^{N_C} x_j a_{ij}(T) - \text{amix}(T, n) \frac{b_i}{b_{mix}} f(Z, B)
\]

where

\[
f(Z, B) = \ln\left( \frac{Z + \epsilon B}{Z + \sigma B} \right)
\]

The output is formatted such that for the first order derivatives

Element (i, 1) : temperature derivative of \( \ln \phi_i \)
Element (i, 2) : pressure derivative of \( \ln \phi_i \)
Element (i, 3+) : composition derivatives of \( \ln \phi_i \)

and for the symmetric second order derivatives

Element (1, 1, i) : temperature derivative of \( \ln \phi_i \)
Element (2, 1, i) : temperature and pressure derivative of \( \ln \phi_i \)
Element (2, 2, i) : pressure derivative of \( \ln \phi_i \)
Elements (3+, 1, i) : temperature and composition derivatives of \( \ln \phi_i \)
Elements (3+, 2, i) : pressure and composition derivatives of \( \ln \phi_i \)
Elements (3+, 3+, i) : composition derivatives of \( \ln \phi_i \)

REQUIRED PARAMETERS:

- \( T \) - Temperature [K]
- \( P \) - Temperature [Pa]
- \( n \) - mole numbers [kmol]
- phase - 0: vapor, 1: liquid
- params - Vector with various parameters obtained by calling LoadParams

OPTIONAL PARAMETERS:

- \( \text{tol} \) - Tolerance for Newton iterations
- \( \text{itmax} \) - Maximum number of Newton iterations
RETURNS:

*lnphi* - Logarithmic fugacity coefficient
*dlnphi* - First order derivatives
*d2lnphi* - Second order derivatives

DEPENDENCIES:

*MixSolveEoS*

See also *LoadParams*

**PureRealHSV**  **MixRealHSV**
**PureRealVapHSV**  **MixRealVapHSV**
**PureRealLiqHSV**  **MixRealLiqHSV**
**PureFug**  **MixFug**
**PureResHSV**  **MixResHSV**
**PureResPhHSV**  **MixResPhHSV**
**PureSolveEoS**  **MixSolveEoS**
**PureSolvePhEoS**  **MixSolvePhEoS**
**PureRealSatTemp**
**PureRealSatPres**
**PureEvalEoS**  **MixEvalEoS**
**PureIdHSV**  **MixIdHSV**
**PureIdVapHSV**  **MixIdVapHSV**
**PureIdLiqHSV**  **MixIdLiqHSV**
**PureParams**  **MixParams**
**PureIdSatTemp**
**IdGasHeatCap**
**IdGasHeatCapInt**
**IdLiqVol**
**IdSatPres**
**SolvePolynomiumNewton**
7. MixSolvePhEoS (Matlab/Mex)

Solve cubic equation of state for compressibility factor

SYNOPSIS:

\[
\begin{align*}
[Z, \ldots \\
dZT, dZP, dZn, \ldots \\
d2ZT, d2ZP, d2ZTP, d2Zn, d2ZPn, d2Zn] & = \ldots \\
\text{MixSolvePhEoS}(T, P, n, \text{phase}, \text{params}, \text{tol}, \text{itmax})
\end{align*}
\]

DESCRIPTION:
Solves a cubic equation of state and the quadratic van der Waals mixing rules

\[
P = \frac{RT}{V - bm} - \frac{am(T)}{((V + \epsilon bm)(V + \sigma bm))}
\]

for the compressibility factor. Other functions are

\[
\begin{align*}
am(T, n) &= \sum_{i=1}^{N_C} \sum_{j=1}^{N_C} x_i x_j a_{ij}(T) \\
bm(n) &= \sum_{i=1}^{N_C} x_i b_{ij} \\
aij(T) &= (1 - k_{ij}) \sqrt{a_i(T) a_j(T)} \\
x_i &= \frac{n_i}{\sum_{j=1}^{N_C} n_j} \\
a_i(T) &= \alpha(Tr, \omega) \Psi \frac{(RTc)^2}{Pc} \\
b_i &= \Omega \frac{RTc}{Pc} \\
alpha(Tr, \omega) &= (1 + m(\omega) \sqrt{1 - Tr})^2 \\
m(\omega) &= m_0 + m_1 \omega + m_2 \omega^2 \\
Tr &= \frac{T}{Tc}
\end{align*}
\]

REQUIRED PARAMETERS:

\begin{align*}
T & \quad - \text{Temperature [K]} \\
P & \quad - \text{Pressure [Pa]} \\
n & \quad - \text{mole numbers [kmol]} \\
\text{phase} & \quad - 0: \text{vapor}, 1: \text{liquid} \\
\text{params} & \quad - \text{Vector with various parameters obtained by calling LoadParams}
\end{align*}

OPTIONAL PARAMETERS:

\begin{align*}
\text{tol} & \quad - \text{Tolerance for Newton iterations} \\
\text{itmax} & \quad - \text{Maximum number of Newton iterations}
\end{align*}

RETURNS:

\begin{align*}
Z & \quad - \text{Compressibility factor} \\
dZT & \quad - \text{First temperature derivative} \\
dZP & \quad - \text{First pressure derivative} \\
dZn & \quad - \text{First composition derivatives} \\
d2ZT & \quad - \text{Second temperature derivative} \\
d2ZP & \quad - \text{Second pressure derivative} \\
d2ZTP & \quad - \text{Second temperature and pressure derivative}
\end{align*}
d2ZTn - Second temperature and composition derivatives

d2ZPn - Second pressure and composition derivatives

d2Zn  - Second composition derivatives

DEPENDENCIES:
MixParams
SolvePolynomiumNewton

See also LoadParams
PureRealHSV       MixRealHSV
PureRealVapHSV    MixRealVapHSV
PureRealLiqHSV    MixRealLiqHSV
PureFug           MixFug
PureResHSV        MixResHSV
PureResPhHSV      MixResPhHSV
PureSolveEoS      MixSolvePhEoS
PureRealSatTemp   MixRealSatTemp
PureRealSatPres   MixRealSatPres
PureEvalEoS       MixEvalEoS
PureIdHSV         MixIdHSV
PureIdVapHSV      MixIdVapHSV
PureIdLiqHSV      MixIdLiqHSV
PureParams        MixParams
PureIdSatTemp     MixIdSatTemp
IdGasHeatCap
IdGasHeatCapInt
IdLiqVol
IdSatPres
SolvePolynomiumNewton
8. MixIdHSV (Matlab/Mex)

Compute vapor and liquid enthalpy, entropy and volume of an ideal mixture

SYNOPSIS:

\[ [H_v, S_v, V_v, H_l, S_l, V_l, \ldots] \]
\[ dH_v, dS_v, dV_v, dH_l, dS_l, dV_l, \ldots \]
\[ d^2H_v, d^2S_v, d^2V_v, d^2H_l, d^2S_l, d^2V_l] = \ldots \]

MixIdHSV(T, P, nv, nl, params)

DESCRIPTION:
Computes vapor and liquid enthalpy, entropy and volume of an ideal mixture together with first and second order temperature and pressure derivatives. Derivatives are computed based on the number of output arguments.

The output is formatted such that for the first order derivatives

Element 1: temperature derivative
Element 2: pressure derivative
Element 3+: composition derivatives

and for the symmetric second order derivatives

Element (1, 1): temperature derivative
Element (2, 1): temperature and pressure derivative
Element (2, 2): pressure derivative
Elements (3+, 1): temperature and composition derivatives
Elements (3+, 2): pressure and composition derivatives
Elements (3+, 3+): composition derivatives

REQUIRED PARAMETERS:
T - Temperature [K]
P - Temperature [Pa]
nv - mole numbers in vapor phase [kmol]
nl - mole numbers in liquid phase [kmol]
params - Vector with various parameters obtained by calling LoadParams

RETURNS:
Hv - Enthalpy of vapor phase
Sv - Entropy of vapor phase
Vv - Volume of vapor phase
Hl - Enthalpy of liquid phase
Sl - Entropy of liquid phase
Vl - Volume of liquid phase
dHv - First order derivatives of vapor enthalpy
dSv - First order derivatives of vapor entropy
\[ \text{dVv} \quad \text{First order derivatives of vapor volume} \\
\text{dHl} \quad \text{First order derivatives of liquid enthalpy} \\
\text{dSl} \quad \text{First order derivatives of liquid entropy} \\
\text{dVl} \quad \text{First order derivatives of liquid volume} \\
\text{d2Hv} \quad \text{Second order derivatives of vapor enthalpy} \\
\text{d2Sv} \quad \text{Second order derivatives of vapor entropy} \\
\text{d2Vv} \quad \text{Second order derivatives of vapor volume} \\
\text{d2Hl} \quad \text{Second order derivatives of liquid enthalpy} \\
\text{d2Sl} \quad \text{Second order derivatives of liquid entropy} \\
\text{d2Vl} \quad \text{Second order derivatives of liquid volume} \\
\]

**DEPENDENCIES:**
- \text{IdGasHeatCap}
- \text{IdGasHeatCapInt}
- \text{IdLiqVol}
- \text{IdSatPres}
- \text{See also LoadParams}
- \text{PureRealHSV}
- \text{MixRealHSV}
- \text{PureRealVapHSV}
- \text{MixRealVapHSV}
- \text{PureRealLiqHSV}
- \text{MixRealLiqHSV}
- \text{PureFug}
- \text{MixFug}
- \text{PureResHSV}
- \text{MixResHSV}
- \text{PureSolveEoS}
- \text{MixSolveEoS}
- \text{PureRealSatTemp}
- \text{PureRealSatPres}
- \text{PureEvalEoS}
- \text{MixEvalEoS}
- \text{PureIdHSV}
- \text{MixIdHSV}
- \text{PureIdVapHSV}
- \text{MixIdVapHSV}
- \text{PureIdLiqHSV}
- \text{MixIdLiqHSV}
- \text{PureParams}
- \text{MixParams}
- \text{PureIdSatTemp}
- \text{IdGasHeatCap}
- \text{IdGasHeatCapInt}
- \text{IdLiqVol}
- \text{IdSatPres}
- \text{SolvePolynomiumNewton}
9. MixIdVapHSV (Matlab/Mex)

Compute enthalpy, entropy and volume of an ideal vapor mixture

SYNOPSIS:
[Hv, Sv, Vv, ...
   dHv, dSv, dVv, ...
   d2Hv, d2Sv, d2Vv] = ...
MixIdVapHSV(T, P, nv, params)

DESCRIPTION:
Computes enthalpy, entropy and volume of an ideal vapor mixture based on the ideal gas law and DIPPR correlations. First and second order temperature and pressure derivatives are computed based on the number of output arguments.

The output is formatted such that for the first order derivatives

   Element 1 : temperature derivative
   Element 2 : pressure derivative
   Element 3+: composition derivatives

and for the symmetric second order derivatives

   Element (1, 1) : temperature derivative
   Element (2, 1) : temperature and pressure derivative
   Element (2, 2) : pressure derivative
   Elements (3+, 1) : temperature and composition derivatives
   Elements (3+, 2) : pressure and composition derivatives
   Elements (3+, 3+): composition derivatives

REQUIRED PARAMETERS:
 T    - Temperature [K]
 P    - Temperature [Pa]
 nv   - mole numbers in vapor phase [kmol]
 params - Vector with various parameters obtained by calling LoadParams

RETURNS:
 Hv   - Enthalpy of vapor phase
 Sv   - Entropy of vapor phase
 Vv   - Volume of vapor phase
 dHv  - First order derivatives of vapor enthalpy
 dSv  - First order derivatives of vapor entropy
 dVv  - First order derivatives of vapor volume
 d2Hv - Second order derivatives of vapor enthalpy
 d2Sv - Second order derivatives of vapor entropy
d2Vv  - Second order derivatives of vapor volume

DEPENDENCIES:

IdGasHeatCap
IdGasHeatCapInt

See also LoadParams

PureRealHSV   MixRealHSV
PureRealVapHSV MixRealVapHSV
PureRealLiqHSV MixRealLiqHSV
PureFug       MixFug
PureResHSV    MixResHSV
PureResPhHSV  MixResPhHSV
PureSolveEoS  MixSolveEoS
PureRealSatTemp MixRealSatTemp
PureRealSatPres MixRealSatPres
PureEvalEoS   MixEvalEoS
PureIdHSV     MixIdHSV
PureIdVapHSV  MixIdVapHSV
PureIdLiqHSV  MixIdLiqHSV
PureParams    MixParams
PureIdSatTemp MixIdSatTemp
IdGasHeatCap  MixIdGasHeatCap
IdGasHeatCapInt MixIdGasHeatCapInt
IdLiqVol      MixIdLiqVol
IdSatPres     MixIdSatPres
SolvePolynomiunNewton
10. MixIdLiqHSV (Matlab/Mex)

Compute enthalpy, entropy and volume of an ideal liquid mixture

SYNOPSIS:
[Hl, Sl, Vl, ...
  dHl, dSl, dVl, ...
  d2Hl, d2Sl, d2Vl] = ...
MixIdLiqHSV(T, P, nl, params)

DESCRIPTION:
Computes enthalpy, entropy and volume of an ideal liquid mixture together
with first and second order temperature and pressure derivatives.
Derivatives are computed based on the number of output arguments.

The output is formatted such that for the first order derivatives

   Element 1 : temperature derivative
   Element 2 : pressure derivative
   Element 3+: composition derivatives

and for the symmetric second order derivatives

   Element (1, 1) : temperature derivative
   Element (2, 1) : temperature and pressure derivative
   Element (2, 2) : pressure derivative
   Elements (3+, 1) : temperature and composition derivatives
   Elements (3+, 2) : pressure and composition derivatives
   Elements (3+, 3+) : composition derivatives

REQUIRED PARAMETERS:
T      - Temperature [K]
P      - Temperature [Pa]
nl     - mole numbers in liquid phase [kmol]
params - Vector with various parameters obtained by calling
         LoadParams

RETURNS:
Hl     - Enthalpy of liquid phase
Sl     - Entropy of liquid phase
Vl     - Volume of liquid phase
dHl    - First order derivatives of liquid enthalpy
dSl    - First order derivatives of liquid entropy
dVl    - First order derivatives of liquid volume
d2Hl   - Second order derivatives of liquid enthalpy
d2Sl   - Second order derivatives of liquid entropy
d2Vl   - Second order derivatives of liquid volume
DEPENDENCIES:
MixIdVapHSV
IdLiqVol
IdSatPres

See also LoadParams
PureRealHSV MixRealHSV
PureRealVapHSV MixRealVapHSV
PureRealLiqHSV MixRealLiqHSV
PureFug MixFug
PureResHSV MixResHSV
PureResPhHSV MixResPhHSV
PureSolveEoS MixSolveEoS
PureSolvePhEoS MixSolvePhEoS
PureRealSatTemp MixRealSatTemp
PureEvalEoS MixEvalEoS
PureIdHSV MixIdHSV
PureIdVapHSV MixIdVapHSV
PureIdLiqHSV MixIdLiqHSV
PureParams MixParams
PureIdSatTemp MixIdSatTemp
IdGasHeatCap MixIdGasHeatCap
IdGasHeatCapInt MixIdGasHeatCapInt
IdLiqVol MixIdLiqVol
IdSatPres MixIdSatPres
SolvePolynomiumNewton MixSolvePolynomiumNewton
11. PureRealHSV (Matlab/Mex)

Compute pure component vapor and liquid enthalpy, entropy and volume

SYNOPSIS:
[hv, sv, vv, hl, sl, vl, ...]
   dhvT, dsvT, dvvT, dhvP, dsvP, dvvP, ...
   dh1T, dsl1T, dh1P, dsl1P, dvlP, ...
   d2hvT, d2svT, d2vvT, d2hvP, d2svP, d2vvP, ...
   d2hvTP, d2svTP, d2vvTP, ...
   d2h1T, d2s1T, d2v1T, d2h1P, d2s1P, d2v1P, ...
   d2h1TP, d2s1TP, d2v1TP] = ...
   PureRealHSV(T, P, params, tol, itmax)

DESCRIPTION:
Computes vapor and liquid enthalpy, entropy and volume of real pure components
together with first and second order temperature and pressure derivatives.
Derivatives are computed based on the number of output arguments.

The enthalpy and entropy are computed from ideal and residual
properties while the volume is obtained as the solution to a cubic
equation of state

REQUIRED PARAMETERS:
T    - Temperature [K]
P    - Temperature [Pa]
params - Vector with various parameters obtained by calling
         LoadParams

OPTIONAL PARAMETERS:
tol  - Tolerance for Newton iterations
itmax - Maximum number of Newton iterations

RETURNS:
hv   - Molar enthalpy of each component
sv   - Molar entropy of each component
vv   - Molar volume of each component
hl   - Molar enthalpy of each component
sl   - Molar entropy of each component
vl   - Molar volume of each component
dhvT - Molar enthalpy 1st temperature derivative of each component
dsvT - Molar entropy 1st temperature derivative of each component
dvvT - Molar volume 1st temperature derivative of each component
dhvP - Molar enthalpy 1st pressure derivative of each component
dsvP - Molar entropy 1st pressure derivative of each component
dsvP - Molar volume 1st pressure derivative of each component
dh1T - Molar enthalpy 1st temperature derivative of each component
ds1T - Molar entropy 1st temperature derivative of each component
dv1T - Molar volume 1st temperature derivative of each component
dh1P - Molar enthalpy 1st pressure derivative of each component
ds1P - Molar entropy 1st pressure derivative of each component
dhlP - Molar volume 1st pressure derivative of each component
d2hvT - Molar enthalpy 2nd temperature derivative of each component
d2svT - Molar entropy 2nd temperature derivative of each component
d2vvT - Molar volume 2nd temperature derivative of each component
d2hvP - Molar enthalpy 2nd pressure derivative of each component
d2svP - Molar entropy 2nd pressure derivative of each component
d2vvP - Molar volume 2nd pressure derivative of each component
d2hvTP - Molar enthalpy 2nd pressure and temperature derivative of each component
d2svTP - Molar entropy 2nd pressure and temperature derivative of each component
d2vvTP - Molar volume 2nd pressure and temperature derivative of each component
d2h1T - Molar enthalpy 2nd temperature derivative of each component
d2s1T - Molar entropy 2nd temperature derivative of each component
d2v1T - Molar volume 2nd temperature derivative of each component
d2h1P - Molar enthalpy 2nd pressure derivative of each component
d2s1P - Molar entropy 2nd pressure derivative of each component
d2v1P - Molar volume 2nd pressure derivative of each component
d2h1TP - Molar enthalpy 2nd pressure and temperature derivative of each component
d2s1TP - Molar entropy 2nd pressure and temperature derivative of each component
d2v1TP - Molar volume 2nd pressure and temperature derivative of each component

DEPENDENCIES:
PureIdVapHSV
PureResHSV

See also LoadParams
PureRealHSV MixRealHSV
PureRealVapHSV MixRealVapHSV
PureRealLiqHSV MixRealLiqHSV
PureFug MixFug
PureResHSV
PureResPhHSV MixResPhHSV
PureSolveEoS
PureSolvePhEoS MixSolvePhEoS
PureRealSatTemp
PureRealSatPres
PureEvalEoS MixEvalEoS
PureIdHSV MixIdHSV
PureIdVapHSV MixIdVapHSV
PureIdLiqHSV MixIdLiqHSV
PureParams MixParams
PureIdSatTemp
IdGasHeatCap
IdGasHeatCapInt
IdLiqVol
IdSatPres
SolvePolynomialNewton
12. PureRealVapHSV (Matlab/Mex)

Compute pure component vapor enthalpy, entropy and volume

SYNOPSIS:
[hv, sv, vv, ...]
   dhvT, dsvT, dvvT, dhvP, dsvP, dvvP, ...
   d2hvT, d2svT, d2vvT, d2hvP, d2svP, d2vvP, ...
   d2hvTP, d2svTP, d2vvTP] = ...
   PureRealVapHSV(T, P, params, tol, itmax)

DESCRIPTION:
Computes vapor enthalpy, entropy and volume of real pure components
   together with first and second order temperature and pressure derivatives.
   Derivatives are computed based on the number of output arguments.

   The enthalpy and entropy are computed from ideal and residual
   properties while the volume is obtained as the solution to a cubic
   equation of state

REQUIRED PARAMETERS:
   T      - Temperature [K]
   P      - Temperature [Pa]
   params - Vector with various parameters obtained by calling
            LoadParams

OPTIONAL PARAMETERS:
   tol    - Tolerance for Newton iterations
   itmax  - Maximum number of Newton iterations

RETURNS:
   hv      - Molar enthalpy of each component
   sv      - Molar entropy of each component
   vv      - Molar volume of each component
   dhvT    - Molar enthalpy 1st temperature derivative of each component
   dsvT    - Molar entropy 1st temperature derivative of each component
   dvvT    - Molar volume 1st temperature derivative of each component
   dhvP    - Molar enthalpy 1st pressure derivative of each component
   dsvP    - Molar entropy 1st pressure derivative of each component
   dsvP    - Molar volume 1st pressure derivative of each component
   d2hvT   - Molar enthalpy 2nd temperature derivative of each component
   d2svT   - Molar entropy 2nd temperature derivative of each component
   d2vvT   - Molar volume 2nd temperature derivative of each component
   d2hvP   - Molar enthalpy 2nd pressure derivative of each component
   d2svP   - Molar entropy 2nd pressure derivative of each component
   d2svP   - Molar volume 2nd pressure derivative of each component
   d2hvTP  - Molar enthalpy 2nd pressure and temperature derivative of each component
d2svTP  - Molar entropy 2nd pressure and temperature derivative of each component

d2vvTP  - Molar volume 2nd pressure and temperature derivative of each component

DEPENDENCIES:
  PureIdVapHSV
  PureResPhHSV

See also LoadParams
  PureRealHSV MixRealHSV
  PureRealVapHSV MixRealVapHSV
  PureRealLiqHSV MixRealLiqHSV
  PureFug MixFug
  PureResHSV MixResPhHSV
  PureSolveEoS MixSolvePhEoS
  PureRealSatTemp
  PureRealSatPres
  PureEvalEoS MixEvalEoS
  PureIdHSV MixIdHSV
  PureIdVapHSV MixIdVapHSV
  PureIdLiqHSV MixIdLiqHSV
  PureParams MixParams
  PureIdSatTemp
  IdGasHeatCap
  IdGasHeatCapInt
  IdLiqVol
  IdSatPres
  SolvePolynomiumNewton
13. PureRealLiqHSV (Matlab/Mex)

Compute pure component liquid enthalpy, entropy and volume

SYNOPSIS:

\[
\begin{align*}
\text{[hl, sl, vl, ...} \\
\text{dh1T, ds1T, dv1T, dh1P, ds1P, dv1P, ...} \\
\text{d2h1T, d2s1T, d2v1T, d2h1P, d2s1P, d2v1P, ...} \\
\text{d2h1TP, d2s1TP, d2v1TP]} = ...
\end{align*}
\]

DESCRIPTION:
Computes liquid enthalpy, entropy and volume of real pure components together with first and second order temperature and pressure derivatives. Derivatives are computed based on the number of output arguments.

The enthalpy and entropy are computed from ideal and residual properties while the volume is obtained as the solution to a cubic equation of state.

REQUIRED PARAMETERS:

\begin{align*}
\text{T} & \quad \text{Temperature [K]} \\
\text{P} & \quad \text{Temperature [Pa]} \\
\text{params} & \quad \text{Vector with various parameters obtained by calling LoadParams}
\end{align*}

OPTIONAL PARAMETERS:

\begin{align*}
\text{tol} & \quad \text{Tolerance for Newton iterations} \\
\text{itmax} & \quad \text{Maximum number of Newton iterations}
\end{align*}

RETURNS:

\begin{align*}
\text{hl} & \quad \text{Molar enthalpy of each component} \\
\text{sl} & \quad \text{Molar entropy of each component} \\
\text{vl} & \quad \text{Molar volume of each component} \\
\text{dh1T} & \quad \text{Molar enthalpy 1st temperature derivative of each component} \\
\text{ds1T} & \quad \text{Molar entropy 1st temperature derivative of each component} \\
\text{dv1T} & \quad \text{Molar volume 1st temperature derivative of each component} \\
\text{dh1P} & \quad \text{Molar enthalpy 1st pressure derivative of each component} \\
\text{ds1P} & \quad \text{Molar entropy 1st pressure derivative of each component} \\
\text{dv1P} & \quad \text{Molar volume 1st pressure derivative of each component} \\
\text{d2h1T} & \quad \text{Molar enthalpy 2nd temperature derivative of each component} \\
\text{d2s1T} & \quad \text{Molar entropy 2nd temperature derivative of each component} \\
\text{d2v1T} & \quad \text{Molar volume 2nd temperature derivative of each component} \\
\text{d2h1P} & \quad \text{Molar enthalpy 2nd pressure derivative of each component} \\
\text{d2s1P} & \quad \text{Molar entropy 2nd pressure derivative of each component} \\
\text{d2v1P} & \quad \text{Molar volume 2nd pressure derivative of each component} \\
\text{d2h1TP} & \quad \text{Molar enthalpy 2nd pressure and temperature derivative of each component}
\end{align*}
d2slTP - Molar entropy 2nd pressure and temperature derivative of each component

d2v1TP - Molar volume 2nd pressure and temperature derivative of each component

DEPENDENCIES:

PureIdVapHSV
PureResPhHSV

See also LoadParams

PureRealHSV MixRealHSV
PureRealVapHSV MixRealVapHSV
PureRealLiqHSV MixRealLiqHSV
PureFug MixFug
PureResHSV MixResHSV
PureResPhHSV MixResPhHSV

PureSolveEoS MixSolveEoS
PureSolvePhEoS MixSolvePhEoS

PureRealSatTemp
PureRealSatPres
PureEvalEoS MixEvalEoS

PureIdHSV MixIdHSV
PureIdVapHSV MixIdVapHSV
PureIdLiqHSV MixIdLiqHSV
PureParams MixParams

PureIdSatTemp
IdGasHeatCap
IdGasHeatCapInt
IdLiqVol
IdSatPres

SolvePolynomiumNewton
14. PureResHSV (Matlab/Mex)

Compute volume and residual enthalpy and entropy of pure components

SYNOPSIS:
[hv, sv, vv, hl, sl, vl, ...
dhvT, dsvT, dvvT, dhvP, dsvP, dvvP, ...
d1T, ds1T, dv1T, dh1P, ds1P, dv1P, ...
d2hvT, d2svT, d2vvT, d2hvP, d2svP, d2vvP, ...
d2hvTP, d2svTP, d2vvTP, ...
d2h1T, d2s1T, d2v1T, d2h1P, d2s1P, d2v1P, ...
d2h1TP, d2s1TP, d2v1TP] = ...
PureResHSV(T, P, params, tol, itmax)

DESCRIPTION:
Computes volume and residual enthalpy and entropy of a pure components
using a cubic equation of state. First and second order temperature and
pressure derivatives are computed based on the number of output arguments.
The residual enthalpy and entropy are obtained from the cubic equation
of state

\[ h^R(T, P) = RT (Z - 1) + \frac{1}{\epsilon - \sigma} \left( \frac{da}{dT} - a(T) \right)f(Z, B) \]
\[ s^R(T, P) = R \ln(Z - 1) + \frac{1}{\epsilon - \sigma} \frac{da}{dT} f(Z, B) \]

where

\[ f(Z, B) = \ln \left( \frac{Z + \epsilon B}{Z + \sigma B} \right) \]

The volume is obtained by solution of the cubic equation of state

\[ P = \frac{RT}{V - b} - \frac{a(T)}{(V^2 + 2Vb - b^2)} \]
\[ a(T) = \alpha(Tr, \omega_0) \psi (RTc)^2/Pc \]
\[ b = \Omega \frac{RTc}{Pc} \]
\[ \alpha(Tr, \omega_0) = (1 + m(\omega_0) \sqrt{1 - Tr})^2 \]
\[ m(\omega_0) = m_0 + m_1 \omega + m_2 \omega^2 \]
\[ Tr = T/Tc \]
\[ \psi = 0.45724 \]
\[ \Omega = 0.07779 \]

REQUIRED PARAMETERS:
T - Temperature [K]
P - Temperature [Pa]
params - Vector with various parameters obtained by calling
LoadParams

OPTIONAL PARAMETERS:
tol - Tolerance for Newton iterations
itmax  - Maximum number of Newton iterations

RETURNS:

hv  - Molar enthalpy of each component
sv  - Molar entropy of each component
vv  - Molar volume of each component
hl  - Molar enthalpy of each component
sl  - Molar entropy of each component
vl  - Molar volume of each component
dhvT - Molar enthalpy 1st temperature derivative of each component
dsvT - Molar entropy 1st temperature derivative of each component
dvvT - Molar volume 1st temperature derivative of each component
dhvP - Molar enthalpy 1st pressure derivative of each component
dsvP - Molar entropy 1st pressure derivative of each component
dvvP - Molar volume 1st pressure derivative of each component
dhlT - Molar enthalpy 1st temperature derivative of each component
dslT - Molar entropy 1st temperature derivative of each component
dvlT - Molar volume 1st temperature derivative of each component
dh1P - Molar enthalpy 1st pressure derivative of each component
ds1P - Molar entropy 1st pressure derivative of each component
dv1P - Molar volume 1st pressure derivative of each component
d2hvT - Molar enthalpy 2nd temperature derivative of each component
d2svT - Molar entropy 2nd temperature derivative of each component
d2vvT - Molar volume 2nd temperature derivative of each component
d2hvP - Molar enthalpy 2nd pressure derivative of each component
d2svP - Molar entropy 2nd pressure derivative of each component
d2vvP - Molar volume 2nd pressure derivative of each component
d2hvTP - Molar enthalpy 2nd pressure and temperature derivative of each component
d2svTP - Molar entropy 2nd pressure and temperature derivative of each component
d2vvTP - Molar volume 2nd pressure and temperature derivative of each component
d2hlT - Molar enthalpy 2nd temperature derivative of each component
d2slT - Molar entropy 2nd temperature derivative of each component
d2vlT - Molar volume 2nd temperature derivative of each component
d2hlP - Molar enthalpy 2nd pressure derivative of each component
d2slP - Molar entropy 2nd pressure derivative of each component
d2vlP - Molar volume 2nd pressure derivative of each component
d2hlTP - Molar enthalpy 2nd pressure and temperature derivative of each component
d2slTP - Molar entropy 2nd pressure and temperature derivative of each component
d2vlTP - Molar volume 2nd pressure and temperature derivative of each component

DEPENDENCIES:

PureParams
PureSolveEoS

See also LoadParams

PureRealHSV          MixRealHSV
PureRealVapHSV       MixRealVapHSV
34
PureRealLiqHSV  MixRealLiqHSV
PureFug        MixFug
PureResHSV     MixResHSV
PureResPhHSV   MixResPhHSV
PureSolveEoS   MixSolveEoS
PureSolvePhEoS MixSolvePhEoS
PureRealSatTemp MixRealSatTemp
PureEvalEoS    MixEvalEoS
PureIdHSV      MixIdHSV
PureIdVapHSV   MixIdVapHSV
PureIdLiqHSV   MixIdLiqHSV
PureParams     MixParams
PureIdSatTemp  MixIdSatTemp
IdGasHeatCap   MixIdGasHeatCap
IdGasHeatCapInt MixIdGasHeatCapInt
IdLiqVol       MixIdLiqVol
IdSatPres      MixIdSatPres
SolvePolynomiumNewton MixSolvePolynomiumNewton
15. PureResPhHSV (Matlab/Mex)

Compute volume and residual enthalpy and entropy of pure components

SYNOPSIS:
[h, s, v, ...
dhT, dsT, dvT, dhP, dsP, dvP, ...
d2hT, d2sT, d2vT, d2hP, d2sP, d2vP, ...
d2hTP, d2sTP, d2vTP] = ...
PureResPhHSV(T, P, phase, params, tol, itmax)

DESCRIPTION:
Computes volume and residual enthalpy and entropy of a pure components using a cubic equation of state. First and second order temperature and pressure derivatives are computed based on the number of output arguments. The residual enthalpy and entropy are obtained from the cubic equation of state

\[ h^R(T, P) = RT (Z - 1) + 1/(\epsilon - \sigma) b(T) \frac{dz}{dT} - a(T) f(Z, B) \]

\[ s^R(T, P) = R \ln(Z - 1) + 1/(\epsilon - \sigma) \frac{da}{dT} f(Z, B) \]

where

\[ f(Z, B) = \ln\left( \frac{Z + \epsilon B}{Z + \sigma B} \right) \]

The volume is obtained by solution of the cubic equation of state

\[ P = RT/(V - b) - a(T)/(V^2 + 2Vb - b^2) \]
\[ a(T) = \alpha(Tr, omega) \Psi (RTc)^2/Pc \]
\[ b = \Omega RTc/Pc \]
\[ \alpha(Tr, omega) = (1 + m(omega) \sqrt{1 - Tr})^2 \]
\[ m(omega) = m_0 + m_1 \omega + m_2 \omega^2 \]
\[ Tr = T/Tc \]
\[ \Psi = 0.45724 \]
\[ \Omega = 0.07779 \]

REQUIRED PARAMETERS:
- T - Temperature [K]
- P - Temperature [Pa]
- phase - 0: vapor, 1: liquid
- params - Vector with various parameters obtained by calling LoadParams

OPTIONAL PARAMETERS:
- tol - Tolerance for Newton iterations
- itmax - Maximum number of Newton iterations
RETURNS:

- **h** - Molar enthalpy of each component
- **s** - Molar entropy of each component
- **v** - Molar volume of each component
- **dhT** - Molar enthalpy 1st temperature derivative of each component
- **dsT** - Molar entropy 1st temperature derivative of each component
- **dvT** - Molar volume 1st temperature derivative of each component
- **dhP** - Molar enthalpy 1st pressure derivative of each component
- **dsP** - Molar entropy 1st pressure derivative of each component
- **dvP** - Molar volume 1st pressure derivative of each component
- **d2hT** - Molar enthalpy 2nd temperature derivative of each component
- **d2sT** - Molar entropy 2nd temperature derivative of each component
- **d2vT** - Molar volume 2nd temperature derivative of each component
- **d2hP** - Molar enthalpy 2nd pressure derivative of each component
- **d2sP** - Molar entropy 2nd pressure derivative of each component
- **d2vP** - Molar volume 2nd pressure derivative of each component
- **d2hTP** - Molar enthalpy 2nd pressure and temperature derivative of each component
- **d2sTP** - Molar entropy 2nd pressure and temperature derivative of each component
- **d2vTP** - Molar volume 2nd pressure and temperature derivative of each component

DEPENDENCIES:

- PureParams
- PureSolveEoS

See also LoadParams

PureRealHSV MixRealHSV
PureRealVapHSV MixRealVapHSV
PureRealLiqHSV MixRealLiqHSV
PureFug MixFug
PureResHSV MixResHSV
PureResPhHSV MixResPhHSV
PureSolveEoS MixSolveEoS
PureSolvePhEoS MixSolvePhEoS
PureRealSatTemp MixRealSatTemp
PureRealSatPres MixRealSatPres
PureEvalEoS MixEvalEoS
PureIdHSV MixIdHSV
PureIdVapHSV MixIdVapHSV
PureIdLiqHSV MixIdLiqHSV
PureParams MixParams
PureIdSatTemp MixIdSatTemp
IdGasHeatCap MixIdGasHeatCap
IdGasHeatCapInt MixIdGasHeatCapInt
IdLiqVol MixIdLiqVol
IdSatPres MixIdSatPres
SolvePolynomiumNewton
16. PureFug (Matlab/Mex)

Compute logarithmic fugacity coefficients of real components

SYNOPSIS:

```matlab
[lnphi, ...
dlnphiT, dlnphiP, ...
d2lnphiT, d2lnphiP, d2lnphiTP] = PureFug(T, P, phase, params, tol, itmax)
```

DESCRIPTION:
Computes fugacities of real components together with first and second order temperature and pressure derivatives. Derivatives are computed based on the number of output arguments. The fugacities are obtained from a cubic equation of state

\[
\ln \phi_i(T, P) = Z - 1 - \ln(Z - B) - 1/(\epsilon - \sigma) 1/(RT) a(T)
\]

where

\[
f(Z, B) = \ln \left( \frac{Z + \epsilon B}{Z + \sigma B} \right)
\]

REQUIRED PARAMETERS:
- `T` - Temperature [K]
- `P` - Pressure [Pa]
- `phase` - 0: vapor, 1: liquid
- `params` - Vector with various parameters obtained by calling `LoadParams`

OPTIONAL PARAMETERS:
- `tol` - Tolerance for Newton iterations
- `itmax` - Maximum number of Newton iterations

RETURNS:
- `lnphi` - Logarithmic fugacity coefficient
- `dlnphiT` - First order temperature derivatives
- `dlnphiP` - First order pressure derivatives
- `d2lnphiT` - Second order temperature derivatives
- `d2lnphiP` - Second order pressure derivatives
- `d2lnphiTP` - Second order temperature and pressure derivatives

DEPENDENCIES:
- `PureParams`
- `PureSolveEoS`

See also `LoadParams`
PureRealLiqHSV  MixRealLiqHSV
PureFug  MixFug
PureResHSV
PureResPhHSV  MixResPhHSV
PureSolveEoS
PureSolvePhEoS  MixSolvePhEoS
PureRealSatTemp
PureRealSatPres
PureEvalEoS  MixEvalEoS
PureIdHSV  MixIdHSV
PureIdVapHSV  MixIdVapHSV
PureIdLiqHSV  MixIdLiqHSV
PureParams  MixParams
PureIdSatTemp
IdGasHeatCap
IdGasHeatCapInt
IdLiqVol
IdSatPres
SolvePolynomiumNewton
17. PureSolveEoS (Matlab/Mex)

Solve cubic equation of state for vapor and liquid compressibility factors

SYNOPSIS:

\[ [Z_v, Z_l, \ldots, dZ_vT, dZ_vP, dZ_lT, dZ_lP, \ldots, d^2Z_vT, d^2Z_vP, d^2Z_vTP, d^2Z_lT, d^2Z_lP, d^2Z_lTP] = \ldots \]

PureSolveEoS(T, P, params, tol, itmax)

DESCRIPTION:

Solves a cubic equation of state

\[ P = \frac{RT}{V - b} - \frac{a(T)}{(V + \epsilon b)(V + \sigma b)} \]

for the compressibility factor. Other functions are

\[ a(T) = \alpha(Tr, \omega) \Psi \left( \frac{RT_c^2}{P_c} \right)^2 \]
\[ b = \Omega RT_c/P_c \]
\[ \alpha(Tr, \omega) = (1 + m(\omega)\sqrt{1 - Tr})^2 \]
\[ m(\omega) = m_0 + m_1 \omega + m_2 \omega^2 \]
\[ Tr = T/T_c \]

REQUIRED PARAMETERS:

- T - Temperature [K]
- P - Pressure [Pa]
- phase - 0: vapor, 1: liquid
- params - Vector with various parameters obtained by calling LoadParams

OPTIONAL PARAMETERS:

- tol - Tolerance for Newton iterations
- itmax - Maximum number of Newton iterations

RETURNS:

- Zv - Vapor compressibility factor
- Zl - Liquid compressibility factor
- dZvT - First temperature derivatives of compressibility factor
- dZvP - First pressure derivatives of compressibility factor
- dZlT - First temperature derivatives of compressibility factor
- dZlP - First pressure derivatives of compressibility factor
- d^2ZvT - Second temperature derivatives of compressibility factor
- d^2ZvP - Second pressure derivatives of compressibility factor
- d^2ZlT - Second temperature derivatives of compressibility factor
- d^2ZlP - Second pressure derivatives of compressibility factor
- d^2ZvTP - Second temperature and pressure derivatives of compressibility factor
- d^2ZlT - Second temperature derivatives of compressibility factor
- d^2ZlP - Second pressure derivatives of compressibility factor
- d^2ZlTP - Second temperature and pressure derivatives of compressibility factor
DEPENDENCIES:
  PureParams
  SolvePolynomiumNewton

See also LoadParams
  PureRealHSV   MixRealHSV
  PureRealVapHSV MixRealVapHSV
  PureRealLiqHSV MixRealLiqHSV
  PureFug       MixFug
  PureResHSV    MixResHSV
  PureResPhHSV  MixResPhHSV
  PureSolveEoS  MixSolveEoS
  PureSolvePhEoS MixSolvePhEoS
  PureRealSatTemp
  PureRealSatPres
  PureEvalEoS   MixEvalEoS
  PureIdHSV     MixIdHSV
  PureIdVapHSV  MixIdVapHSV
  PureIdLiqHSV  MixIdLiqHSV
  PureParams    MixParams
  PureIdSatTemp
  IdGasHeatCap
  IdGasHeatCapInt
  IdLiqVol
  IdSatPres
  SolvePolynomiumNewton
18. PureSolvePhEoS (Matlab/Mex)

Solve cubic equation of state for compressibility factor

SYNOPSIS:
\[ [Z, \ldots, dZT, dZP, \ldots, d2ZT, d2ZP, d2ZTP] = \ldots \]
\[ \text{PureSolvePhEoS}(T, P, \text{phase}, \text{params}, \text{tol}, \text{itmax}) \]

DESCRIPTION:
Solves a cubic equation of state
\[ P = \frac{RT}{V - b} - \frac{a(T)}{(V + \epsilon b)(V + \sigma b)} \]
for the compressibility factor. Other functions are
\[ a(T) = \alpha(Tr, \omega) \Psi \left( \frac{RTc}{Pc} \right)^2 \]
\[ b = \Omega RTc/Pc \]
\[ \alpha(Tr, \omega) = (1 + m(\omega) \sqrt{1 - Tr})^2 \]
\[ m(\omega) = m_0 + m_1 \omega + m_2 \omega^2 \]
\[ Tr = T/Tc \]

REQUIRED PARAMETERS:
\[ T \quad \text{Temperature [K]} \]
\[ P \quad \text{Pressure [Pa]} \]
\[ \text{phase} \quad 0: \text{vapor}, 1: \text{liquid} \]
\[ \text{params} \quad \text{Vector with various parameters obtained by calling} \]
\[ \quad \text{LoadParams} \]

OPTIONAL PARAMETERS:
\[ \text{tol} \quad \text{Tolerance for Newton iterations} \]
\[ \text{itmax} \quad \text{Maximum number of Newton iterations} \]

RETURNS:
\[ Z \quad \text{Compressibility factor} \]
\[ dZT \quad \text{First temperature derivatives of compressibility factor} \]
\[ dZP \quad \text{First pressure derivatives of compressibility factor} \]
\[ d2ZT \quad \text{Second temperature derivatives of compressibility factor} \]
\[ d2ZP \quad \text{Second pressure derivatives of compressibility factor} \]
\[ d2ZTP \quad \text{Second temperature and pressure derivatives of compressibility factor} \]

DEPENDENCIES:
\[ \text{PureParams} \]
\[ \text{SolvePolynomialNewton} \]

See also LoadParams
PureRealHSV  MixRealHSV
PureRealVapHSV  MixRealVapHSV
PureRealLiqHSV  MixRealLiqHSV
PureFug  MixFug
PureResHSV  MixResHSV
PureResPhHSV  MixResPhHSV
PureSolveEoS  MixSolveEoS
PureSolvePhEoS  MixSolvePhEoS
PureRealSatTemp  MixRealSatTemp
PureRealSatPres  MixRealSatPres
PureEvalEoS  MixEvalEoS
PureIdHSV  MixIdHSV
PureIdVapHSV  MixIdVapHSV
PureIdLiqHSV  MixIdLiqHSV
PureParams  MixParams
PureIdSatTemp  MixIdSatTemp
IdGasHeatCap  MixIdGasHeatCap
IdGasHeatCapInt  MixIdGasHeatCapInt
IdLiqVol  MixIdLiqVol
IdSatPres  MixIdSatPres
SolvePolynomiumNewton  MixSolvePolynomiumNewton
19. PureIdVapHSV (Matlab/Mex)

Compute pure component molar vapor enthalpy, entropy and volume

SYNOPSIS:
[hv, sv, vv, ...
dhvT, dsvT, dvvT, dsvP, dvvP, ...
d2hvT, d2svT, d2svP, d2vvp, d2vvTP] = ...
PureIdVapHSV(T, P, params)

DESCRIPTION:
Computes molar enthalpy, entropy and volume of a set of pure components using the ideal gas law and DIPPR correlations. First and second order temperature and pressure derivatives are computed based on the number of output arguments.

Certain derivatives are not returned by this routine because they are zero (e.g. pressure derivative of enthalpy d/dP h^v)

REQUIRED PARAMETERS:
T - Temperature [K]
P - Temperature [Pa]
params - Vector with various parameters obtained by calling LoadParams

RETURNS:
hv - Molar enthalpy of each component
sv - Molar entropy of each component
vv - Molar volume of each component
dhvT - Molar enthalpy 1st temperature derivative of each component
dsvT - Molar entropy 1st temperature derivative of each component
dvvT - Molar volume 1st temperature derivative of each component
dsvP - Molar entropy 1st pressure derivative of each component
dvvP - Molar volume 1st pressure derivative of each component
d2hvT - Molar enthalpy 2nd temperature derivative of each component
d2svT - Molar entropy 2nd temperature derivative of each component
d2svP - Molar entropy 2nd pressure derivative of each component
d2vvP - Molar volume 2nd pressure derivative of each component
d2vvTP - Molar volume 2nd pressure and temperature derivative of each component

DEPENDENCIES:
IdGasHeatCap
IdGasHeatCapInt

See also LoadParams
PureRealHSV MixRealHSV
PureRealVapHSV MixRealVapHSV
<table>
<thead>
<tr>
<th>PureRealLiqHSV</th>
<th>MixRealLiqHSV</th>
</tr>
</thead>
<tbody>
<tr>
<td>PureFug</td>
<td>MixFug</td>
</tr>
<tr>
<td>PureResHSV</td>
<td>MixResHSV</td>
</tr>
<tr>
<td>PureResPhHSV</td>
<td>MixResPhHSV</td>
</tr>
<tr>
<td>PureSolveEoS</td>
<td>MixSolvePhEoS</td>
</tr>
<tr>
<td>PureRealSatTemp</td>
<td>MixRealSatTemp</td>
</tr>
<tr>
<td>PureRealSatPres</td>
<td>MixRealSatPres</td>
</tr>
<tr>
<td>PureEvalEoS</td>
<td>MixEvalEoS</td>
</tr>
<tr>
<td>PureIdHSV</td>
<td>MixIdHSV</td>
</tr>
<tr>
<td>PureIdVapHSV</td>
<td>MixIdVapHSV</td>
</tr>
<tr>
<td>PureIdLiqHSV</td>
<td>MixIdLiqHSV</td>
</tr>
<tr>
<td>PureParams</td>
<td>MixParams</td>
</tr>
<tr>
<td>PureIdSatTemp</td>
<td>MixIdSatTemp</td>
</tr>
<tr>
<td>IdGasHeatCap</td>
<td></td>
</tr>
<tr>
<td>IdGasHeatCapInt</td>
<td></td>
</tr>
<tr>
<td>IdLiqVol</td>
<td></td>
</tr>
<tr>
<td>IdSatPres</td>
<td></td>
</tr>
<tr>
<td>SolvePolynomialNewton</td>
<td></td>
</tr>
</tbody>
</table>

45
20. PureIdLiqHSV (Matlab/Mex)

Compute pure component molar liquid enthalpy, entropy and volume

SYNOPSIS:
[h1, s1, v1, ... dh1T, ds1T, dv1T, dh1P, ds1P, ... d2h1T, d2s1T, d2v1T, d2h1TP, d2s1TP] = ...
PureIdLiqHSV(T, P, params)

DESCRIPTION:
Computes molar enthalpy, entropy and volume of a set of pure components together with first and second order temperature and pressure derivatives. Derivatives are computed based on the number of output arguments.

Certain derivatives are not returned by this routine because they are zero (e.g. pressure derivative of volume d/dP v^l = 0)

REQUIRED PARAMETERS:
T - Temperature [K]
P - Temperature [Pa]
params - Vector with various parameters obtained by calling
LoadParams

RETURNS:
hl - Molar enthalpy of each component
sl - Molar entropy of each component
vl - Molar volume of each component
dh1T - Molar enthalpy 1st temperature derivative of each component
ds1T - Molar entropy 1st temperature derivative of each component
dv1T - Molar volume 1st temperature derivative of each component
dh1P - Molar enthalpy 1st pressure derivative of each component
ds1P - Molar entropy 1st pressure derivative of each component
d2h1T - Molar enthalpy 2nd temperature derivative of each component
d2s1T - Molar entropy 2nd temperature derivative of each component
d2v1T - Molar volume 2nd temperature derivative of each component
d2h1TP - Molar enthalpy 2nd pressure and temperature derivative of each component
d2s1TP - Molar entropy 2nd pressure and temperature derivative of each component

DEPENDENCIES:
PureIdVapHSV
IdLiqVol
IdSatPres

See also LoadParams
PureRealHSV MixRealHSV
PureRealVapHSV  MixRealVapHSV
PureRealLiqHSV  MixRealLiqHSV
PureFug        MixFug
PureResHSV     MixResHSV
PureResPhHSV   MixResPhHSV
PureSolveEoS   MixSolveEoS
PureSolvePhEoS MixSolvePhEoS
PureRealSatTemp MixRealSatTemp
PureRealSatPres MixRealSatPres
PureEvalEoS    MixEvalEoS
PureIdHSV      MixIdHSV
PureIdVapHSV   MixIdVapHSV
PureIdLiqHSV   MixIdLiqHSV
PureParams     MixParams
PureIdSatTemp  MixIdSatTemp
IdGasHeatCap   MixIdGasHeatCap
IdGasHeatCapInt MixIdGasHeatCapInt
IdLiqVol       MixIdLiqVol
IdSatPres      MixIdSatPres
SolvePolynomiunNewton MixSolvePolynomiunNewton
21. MixParams (Matlab/Mex)

Compute van der Waals mixing rules $a_m(T, n)$ and $b_m(n)$

SYNOPSIS:

```matlab
[am, bm, ...
  damT, damn, dbmn, ...
  d2amT, d2amTn, d2amn, d2bmn, ...
  d3amTn2, d3amT, d3amT2n] = ...
  MixParams(T, n, params)
```

DESCRIPTION:
Computes the van der Waals mixing parameters $a_m(T, n)$ and $b_m(n)$ based on the pure component properties $a_i(T)$ and $b_i$

\[
am(T, n) = \sum_{i=1}^{N_C} \sum_{j=1}^{N_C} x_i x_j a_{ij}(T) \\
bm(n) = \sum_{i=1}^{N_C} x_i b_{ij} \\
a_{ij}(T) = (1 - k_{ij}) \sqrt{a_i(T) a_j(T)} \\
x_i = \frac{n_i}{\sum_{j=1}^{N_C} n_j}
\]

The computations of derivatives are only computed if requested as output.

REQUIRED PARAMETERS:
- $T$ - Temperature [K]
- $n$ - mole numbers [kmol]
- params - Vector with various parameters obtained by calling LoadParams

RETURNS:
- $am$ - Mixing rule parameter
- $bm$ - Mixing rule parameter
- $damT$ - First order temperature derivative of $am$
- $damn$ - First order composition derivatives of $am$
- $dbmn$ - First order composition derivative of $bm$
- $d2amT$ - Second order temperature derivative of $am$
- $d2amTn$ - Second order temperature and composition derivatives of $am$
- $d2amn$ - Second order composition derivatives of $am$
- $d2bmn$ - Second order composition derivatives of $bm$
- $d3amT$ - Third order temperature derivative of $am$
- $d3amT2n$ - Third order temperature (x2) and composition derivatives of $am$
- $d3amTn2$ - Third order temperature and composition (x2) derivatives of $am$

DEPENDENCIES:
PureParams

See also LoadParams
PureRealHSV \hspace{1cm} MixRealHSV
PureRealVapHSV \hspace{1cm} MixRealVapHSV
PureRealLiqHSV \hspace{1cm} MixRealLiqHSV
PureFug \hspace{1cm} MixFug
PureResHSV \hspace{1cm} MixResHSV
PureResPhHSV \hspace{1cm} MixResPhHSV
PureSolveEoS \hspace{1cm} MixSolveEoS
PureSolvePhEoS \hspace{1cm} MixSolvePhEoS
PureRealSatTemp \hspace{1cm} MixRealSatTemp
PureRealSatPres \hspace{1cm} MixRealSatPres
PureEvalEoS \hspace{1cm} MixEvalEoS
PureIdHSV \hspace{1cm} MixIdHSV
PureIdVapHSV \hspace{1cm} MixIdVapHSV
PureIdLiqHSV \hspace{1cm} MixIdLiqHSV
PureParams \hspace{1cm} MixParams
PureIdSatTemp \hspace{1cm} MixIdSatTemp
IdGasHeatCap \hspace{1cm} MixIdGasHeatCap
IdGasHeatCapInt \hspace{1cm} MixIdGasHeatCapInt
IdLiqVol \hspace{1cm} MixIdLiqVol
IdSatPres \hspace{1cm} MixIdSatPres
SolvePolynomialNewton
22. PureParams (Matlab/Mex)

Compute pure component parameters $a_i(T)$ and $b_i$

SYNOPSIS:
\[ [a, b, daT, d2aT, d3aT] = PureParams(T, params) \]

DESCRIPTION:
Computes the pure component parameters $a_i(T)$ and $b_i$ for $i = 1, \ldots, N_C$

\[
\begin{align*}
a_i(T) & = \alpha(Tr, \omega) \Psi \frac{(RTc)^2}{Pc} \\
b_i & = \Omega \frac{RTc}{Pc} \\
\alpha(Tr, \omega) & = (1 + m(\omega) \sqrt{1 - Tr})^2 \\
m(\omega) & = m_0 + m_1 \omega + m_2 \omega^2 \\
Tr & = \frac{T}{Tc}
\end{align*}
\]

The computations of derivatives are only computed if requested as output.

REQUIRED PARAMETERS:
- $T$ - Temperature [K]
- $params$ - Vector with various parameters obtained by calling LoadParams

RETURNS:
- $a$ - Pure component parameter
- $b$ - Pure component parameter
- $daT$ - First order temperature derivative of $a$
- $d2aT$ - Second order temperature derivative of $a$
- $d3aT$ - Third order temperature derivative of $a$

See also LoadParams
- PureRealHSV MixRealHSV
- PureRealVapHSV MixRealVapHSV
- PureRealLiqHSV MixRealLiqHSV
- PureFug MixFug
- PureResHSV MixResHSV
- PureResPhHSV MixResPhHSV
- Pure SolveEoS MixSolveEoS
- Pure RealSatTemp MixRealSatTemp
- Pure RealSatPres MixRealSatPres
- Pure EvalEoS MixEvalEoS
- Pure IdHSV MixIdHSV
- Pure IdVapHSV MixIdVapHSV
- Pure IdLiqHSV MixIdLiqHSV
- PureParams MixParams

50
PureIdSatTemp
IdGasHeatCap
IdGasHeatCapInt
IdLiqVol
IdSatPres
SolvePolynomiumNewton
23. SolvePolynomiumCardano (Matlab/Mex)

Solve cubic equation analytically

SYNOPSIS:

roots = SolvePolynomiumCardano(d2, d1, d0)

DESCRIPTION:

Solves the cubic equations

\[ Z^3 + d2 \, Z^2 + d1 \, Z + d0 = 0 \]

using Cardano’s formula

REQUIRED PARAMETERS:

d2  - Quadratic coefficient

d1  - Linear coefficient

d0  - Constant coefficient

RETURNS:

roots  - The real root(s) of the cubic equation

See also: LoadParams

PureRealHSV  MixRealHSV
PureRealVapHSV MixRealVapHSV
PureRealLiqHSV MixRealLiqHSV
PureFug  MixFug
PureResHSV
PureResPhHSV MixResPhHSV
PureSolveEoS
PureSolvePhEoS MixSolvePhEoS
PureRealSatTemp
PureRealSatPres
PureEvalEoS MixEvalEoS
PureIdHSV MixIdHSV
PureIdVapHSV MixIdVapHSV
PureIdLiqHSV MixIdLiqHSV
PureParams MixParams
PureIdSatTemp
IdGasHeatCap
IdGasHeatCapInt
IdLiqVol
IdSatPres
SolvePolynomiumNewton
SolvePolynomiumNewton (Matlab/Mex)

Solve cubic equation iteratively

SYNOPSIS:
Z = SolvePolynomiumNewton(d2, d1, d0, Z0, tol, itmax)

DESCRIPTION:
Solves the cubic equation

\[ q(Z) = Z^3 + d2 Z^2 + d1 Z + d0 = 0 \]

using an iterative Newton approach. The approach is terminated when

\[ |q(Z)| < tol \quad \text{and} \quad |\Delta Z| = |q(Z)/q'(Z)| < tol. \]

REQUIRED PARAMETERS:
\begin{align*}
d2 & \quad \text{Quadratic coefficient} \\
d1 & \quad \text{Linear coefficient} \\
d0 & \quad \text{Constant coefficient}
\end{align*}

OPTIONAL PARAMETERS:
\begin{align*}
tol & \quad \text{Tolerance for stopping criteria} \\
\text{itmax} & \quad \text{Maximum number of iterations}
\end{align*}

RETURNS:
\[ Z \quad \text{A single real root of the cubic equation} \]

See also LoadParams
PureRealHSV \quad MixRealHSV
PureRealVapHSV \quad MixRealVapHSV
PureRealLiqHSV \quad MixRealLiqHSV
PureFug \quad MixFug
PureResHSV \quad MixResHSV
PureResPhHSV \quad MixResPhHSV
PureSolveEoS \quad MixSolvePhEoS
PureSolvePhEoS \quad MixSolvePhEoS
PureRealSatTemp \quad MixParams
PureRealSatPres \quad MixParams
PureEvalEoS \quad MixEvalEoS
PureIdHSV \quad MixIdHSV
PureIdVapHSV \quad MixIdVapHSV
PureIdLiqHSV \quad MixIdLiqHSV
PureParams \quad MixParams
IdGasHeatCap
IdGasHeatCapInt
IdLiqVol
IdSatPres
SolvePolynomiumNewton
25. IdGasHeatCap (Matlab/Mex)

Compute ideal gas heat capacity

SYNOPSIS:
\[ [cp, dcp] = \text{IdGasHeatCap}(T, \text{params}) \]

DESCRIPTION:
Computes ideal gas heat capacity using the DIPPR correlation
\[
c_P,k^{ig}(T) = A_k + B_k((C_k/T)/\sinh(C_k/T))^2 + D_k((E_k/T)/\cosh(E_k/T))^2
\]
and the first order derivative

REQUIRED PARAMETERS:
T - Temperature, [K]
params - Vector with various parameters obtained by calling LoadParams

RETURNS:
\[ \begin{align*}
cp & \quad \text{Ideal gas heat capacity} \\
dcp & \quad \text{First temperature derivative}
\end{align*} \]

See also LoadParams
PureRealHSV MixRealHSV
PureRealVapHSV MixRealVapHSV
PureRealLiqHSV MixRealLiqHSV
PureFug MixFug
PureResHSV MixResHSV
PureResPhHSV MixResPhHSV
PureSolveEoS MixSolveEoS
PureSolvePhEoS MixSolvePhEoS
PureRealSatTemp MixRealSatTemp
PureRealSatPres MixRealSatPres
PureEvalEoS MixEvalEoS
PureIdHSV MixIdHSV
PureIdVapHSV MixIdVapHSV
PureIdLiqHSV MixIdLiqHSV
PureParams MixParams
PureIdSatTemp MixIdSatTemp
IdGasHeatCap MixIdGasHeatCap
IdGasHeatCapInt MixIdGasHeatCapInt
IdLiqVol MixIdLiqVol
IdSatPres MixIdSatPres
SolvePolynomiumNewton MixSolvePolynomiumNewton
26. IdGasHeatCapInt (Matlab/Mex)

Compute integrals of ideal gas heat capacity

SYNOPSIS:
\[ \text{[intcp, intcpT] = IdGasHeatCapInt(T, params)} \]

DESCRIPTION:
Computes the following integrals of the ideal gas heat capacity
\[
\int_{T_0}^{T} c_{P,k}^{ig}(\tau) \, d\tau \\
\int_{T_0}^{T} c_{P,k}^{ig}(\tau) / \tau \, d\tau
\]
where ideal gas heat capacity is defined by the DIPPR correlation
\[
c_{P,k}^{ig}(T) = A_k + B_k((C_k/T)/\sinh(C_k/T))^2 \\
- D_k((E_k/T)/\cosh(E_k/T))^2
\]

REQUIRED PARAMETERS:
- \( T \) - Temperature, [K]
- \( \text{params} \) - Vector with various parameters obtained by calling \( \text{LoadParams} \)

RETURNS:
- \( \text{intcp} \) - Integral of \( c_p \) from \( T_0 \) to \( T \)
- \( \text{intcpT} \) - Integral of \( c_p/T \) from \( T_0 \) to \( T \)

See also \( \text{LoadParams} \)

PureRealHSV MixRealHSV
PureRealVapHSV MixRealVapHSV
PureRealLiqHSV MixRealLiqHSV
PureFug MixFug
PureResHSV MixResPhHSV
PureSolveEoS MixSolvePhEoS
PureRealSatTemp
PureRealSatPres
PureEvalEoS MixEvalEoS
PureIdHSV MixIdHSV
PureIdVapHSV MixIdVapHSV
PureIdLiqHSV MixIdLiqHSV
PureParams MixParams
PureIdSatTemp
IdGasHeatCap
IdGasHeatCapInt
IdLiqVol
IdSatPres
SolvePolynomialNewton
27. IdLiqVol (Matlab/Mex)

Compute liquid volume

SYNOPSIS:
[v1, dv1, d2v1, d3v1] = IdLiqVol(T, params)

DESCRIPTION:
Computes liquid volume based on the DIPPR correlation

\[ v_{1,k}(T) = B_k^{(1 + (1 - T/C_k)^D_k)/A_k} \]

The first, second and third order temperature derivatives are computed depending on the number of output arguments.

NOTE:
The DIPPR correlation is not limited to ideal substances as the name suggests but in this thermodynamic library it is only applied in the computation of ideal liquid properties.

REQUIRED PARAMETERS:
T - Temperature [K]
params - Vector with various parameters obtained by calling LoadParams

RETURNS:
v1 - Vector of liquid volumes for each component, [m^3]
dv1 - First temperature derivative of liquid volumes
d2v1 - Second temperature derivative of liquid volumes
d3v1 - Third temperature derivative of liquid volumes

See also LoadParams
PureRealHSV MixRealHSV
PureRealVapHSV MixRealVapHSV
PureRealLiqHSV MixRealLiqHSV
PureFug MixFug
PureResHSV MixResHSV
PureResPhHSV MixResPhHSV
PureSolveEoS MixSolveEoS
PureRealSatTemp MixRealSatTemp
PureRealSatPres MixRealSatPres
PureEvalEoS MixEvalEoS
PureIdHSV MixIdHSV
PureIdVapHSV MixIdVapHSV
PureIdLiqHSV MixIdLiqHSV
PureParams MixParams

58
PureIdSatTemp
IdGasHeatCap
IdGasHeatCapInt
IdLiqVol
IdSatPres
SolvePolynomiumNewton
28. **IdSatPres (Matlab/Mex)**

Compute saturation pressure

**SYNOPSIS:**

\[
[Psat, dPsat, d2Psat, d3Psat] = IdSatPres(T, params)
\]

**DESCRIPTION:**

Computes saturation pressure using the DIPPR correlation

\[
\ln Psat_k = A_k + B_k/T + C_k \ln(T) + D_k T^{E_k}
\]

The first, second and third order temperature derivatives are computed depending on the number of output arguments.

**NOTE:**

The DIPPR correlation is not limited to ideal substances as the name suggests but in this thermodynamic library it is only applied in the computation of ideal liquid properties.

**REQUIRED PARAMETERS:**

- **T** - Temperature [K]
- **params** - Vector with various parameters obtained by calling `LoadParams`

**RETURNS:**

- **Psat** - Vector of vapor pressures for each component [Pa]
- **dPsat** - First temperature derivative of vapor pressures
- **d2Psat** - Second temperature derivative of vapor pressures
- **d3Psat** - Third temperature derivative of vapor pressures

See also `LoadParams`

- `PureRealHSV`
- `MixRealHSV`
- `PureRealVapHSV`
- `MixRealVapHSV`
- `PureRealLiqHSV`
- `MixRealLiqHSV`
- `PureFug`
- `MixFug`
- `PureResHSV`
- `MixResHSV`
- `PureSolveEoS`
- `MixSolvePhEoS`
- `PureRealSatTemp`
- `MixRealSatTemp`
- `PureEvalEoS`
- `MixEvalEoS`
- `PureIdHSV`
- `MixIdHSV`
- `PureIdVapHSV`
- `MixIdVapHSV`
- `PureIdLiqHSV`
- `MixIdLiqHSV`
- `PureParams`
- `MixParams`
PureIdSatTemp
IdGasHeatCap
IdGasHeatCapInt
IdLiqVol
IdSatPres
SolvePolynomiumNewton
29. **LoadParams (C)**

Load thermodynamic and equation of state parameters

**SYNOPSIS:**

```c
LoadParams(comp, NC, EOS, k, PathToData, p);
```

**DESCRIPTION:**

Loads parameters from the DIPPR database together with Peng-Robinson or Soave-Redlich-Kwong equation of state parameters.

**REQUIRED PARAMETERS:**

- `comp` - Array containing IDs of components (must have size NC)
- `NC` - Number of components (length of comp)
- `EOS` - Equation of state (1: PR, 0: SRK, -1: none)
- `k` - Binary interaction parameters (must have size NC*NC if EOS >= 0)
- `PathToData` - Path to file with DIPPR data (default: NULL - uses data/DIPPRdata.dat)

**RETURNS:**

- `p` - Array with thermodynamic and equation of state parameters (The function NoPar(int NC) returns the size that p should have on entry to LoadParams)
30. MixRealHSV (C)

Compute vapor and liquid enthalpy, entropy and volume of a real mixture

SYNOPSIS:
MixRealHSV(T, P, nv, nl, p, tol, itmax, nargout, memaux,
   Hv, Sv, Vv, Hl, Sl, Vl,
   dHv, dSv, dVV, dHl, dSl, dVl,
   d2Hv, d2Sv, d2VV, d2Hl, d2Sl, d2Vl);

DESCRIPTION:
Computes enthalpy, entropy and volume of a real mixture together with first and second order temperature and pressure derivatives. Derivatives are computed based on the number of output arguments.

The output is formatted such that for the first order derivatives

   Element 1 : temperature derivative
   Element 2 : pressure derivative
   Element 3+: composition derivatives

and for the symmetric second order derivatives

   Element (1, 1) : temperature derivative
   Element (2, 1) : temperature and pressure derivative
   Element (2, 2) : pressure derivative
   Elements (3+, 1) : temperature and composition derivatives
   Elements (3+, 2) : pressure and composition derivatives
   Elements (3+, 3+) : composition derivatives

REQUIRED PARAMETERS:
T - Temperature [K]
P - Temperature [Pa]
nv - mole numbers in vapor phase [kmol]
nl - mole numbers in liquid phase [kmol]
p - Vector with various parameters obtained by calling LoadParams
nargout - Number of output arguments
memaux - Auxiliary memory must be size
   nargout > 6: 6*(NC + 2)ˆ2 + 6*(NC + 2) + 54*NC + 6*NCˆ2 memory
   nargout > 3: 6*(NC + 2) + 39*NC memory
   else : 15*NC memory

OPTIONAL PARAMETERS:
tol - Tolerance for Newton iterations (-1 for default)
itmax - Maximum number of Newton iterations (-1 for default)
RETURNS:
Hv - Vapor enthalpy (scalar)
Sv - Vapor entropy (scalar)
Vv - Vapor volume (scalar)
Hl - Liquid enthalpy (scalar)
Sl - Liquid entropy (scalar)
Vl - Liquid volume (scalar)
dHv - 1st derivatives (must be size 2 + NC)
dSv - 1st derivatives (must be size 2 + NC)
dVv - 1st derivatives (must be size 2 + NC)
dHl - 1st derivatives (must be size 2 + NC)
dSl - 1st derivatives (must be size 2 + NC)
dVl - 1st derivatives (must be size 2 + NC)
d2Hv - 2nd derivatives (must be size (2 + NC)^2)
d2Sv - 2nd derivatives (must be size (2 + NC)^2)
d2Vv - 2nd derivatives (must be size (2 + NC)^2)
d2Hl - 2nd derivatives (must be size (2 + NC)^2)
d2Sl - 2nd derivatives (must be size (2 + NC)^2)
d2Vl - 2nd derivatives (must be size (2 + NC)^2)

DEPENDENCIES:
PureIdVap
MixResPhHSV
31. MixRealVapHSV (C)

Compute enthalpy, entropy and volume of a real vapor mixture

SYNOPSIS:
MixRealVapHSV(T, P, nv, p, tol, itmax, nargout, memaux,
        Hv, Sv, Vv,
        dHv, dSv, dVv,
        d2Hv, d2Sv, d2Vv);

DESCRIPTION:
Computes enthalpy, entropy and volume of a real vapor mixture together
with first and second order temperature and pressure derivatives. Derivatives are computed based on the number of output arguments.

The output is formatted such that for the first order derivatives

Element 1 : temperature derivative
Element 2 : pressure derivative
Element 3+: composition derivatives

and for the symmetric second order derivatives

Element (1, 1) : temperature derivative
Element (2, 1) : temperature and pressure derivative
Element (2, 2) : pressure derivative
Elements (3+, 1) : temperature and composition derivatives
Elements (3+, 2) : pressure and composition derivatives
Elements (3+, 3+): composition derivatives

REQUIRED PARAMETERS:
T - Temperature [K]
P - Temperature [Pa]
nv - mole numbers in vapor phase [kmol]
p - Vector with various parameters obtained by calling LoadParams
nargout - Number of output arguments
memaux - Auxiliary memory must be size
        nargout > 6: 6*(NC + 2)^2 + 6*(NC + 2) + 46*NC + 6*NC^2 memory
        nargout > 3: 6*(NC + 2) + 35*NC memory
        else : 15*NC memory

OPTIONAL PARAMETERS:
tol - Tolerance for Newton iterations (-1 for default)
itmax - Maximum number of Newton iterations (-1 for default)

RETURNS:
Hv  - Vapor enthalpy (scalar)
Sv  - Vapor entropy (scalar)
Vv  - Vapor volume (scalar)
dHv - 1st derivatives (must be size 2 + NC)
dSv - 1st derivatives (must be size 2 + NC)
dVv - 1st derivatives (must be size 2 + NC)
d2Hv - 2nd derivatives (must be size (2 + NC)^2)
d2Sv - 2nd derivatives (must be size (2 + NC)^2)
d2Vv - 2nd derivatives (must be size (2 + NC)^2)

DEPENDENCIES:
  MixIdVapHSV
  MixResPhHSV
32. MixRealLiqHSV (C)

Compute enthalpy, entropy and volume of a real liquid mixture

SYNOPSIS:
MixRealliqHSV(T, P, nl, p, tol, itmax, nargout, memaux,
Hl, Sl, Vl,
dHl, dSl, dVl,
d2Hl, d2Sl, d2Vl);

DESCRIPTION:
Computes enthalpy, entropy and volume of a real liquid mixture together
with first and second order temperature and pressure derivatives.
Derivatives are computed based on the number of output arguments.

The output is formatted such that for the first order derivatives

Element 1 : temperature derivative
Element 2 : pressure derivative
Element 3+: composition derivatives

and for the symmetric second order derivatives

Element (1, 1) : temperature derivative
Element (2, 1) : temperature and pressure derivative
Element (2, 2) : pressure derivative
Elements (3+, 1) : temperature and composition derivatives
Elements (3+, 2) : pressure and composition derivatives
Elements (3+, 3+): composition derivatives

REQUIRED PARAMETERS:
T - Temperature [K]
P - Temperature [Pa]
nl - mole numbers in liquid phase [kmol]
p - Vector with various parameters obtained by calling LoadParams
nargout - Number of output arguments
memaux - Auxiliary memory must be size

nargout > 6: 6*(NC + 2)^2 + 6*(NC + 2) + 46*NC + 6*NC^2 memory
nargout > 3: 6*(NC + 2) + 35*NC memory
else : 15*NC memory

OPTIONAL PARAMETERS:
tol - Tolerance for Newton iterations (-1 for default)
itmax - Maximum number of Newton iterations (-1 for default)

RETURNS:
Hl - Liquid enthalpy (scalar)
S1 - Liquid entropy (scalar)
Vl - Liquid volume (scalar)
dHl - 1st derivatives (must be size 2 + NC)
dS1 - 1st derivatives (must be size 2 + NC)
dVl - 1st derivatives (must be size 2 + NC)
d2Hl - 2nd derivatives (must be size (2 + NC)^2)
d2S1 - 2nd derivatives (must be size (2 + NC)^2)
d2Vl - 2nd derivatives (must be size (2 + NC)^2)

DEPENDENCIES:
MixIdVapHSV
MixResPhHSV
33. MixResPhHSV (C)

Compute volume and residual enthalpy and entropy of phase

SYNOPSIS:
MixResPhHSV(T, P, n, phase, p, tol, itmax, nargout, memaux,
h, s, v,
dh, ds, dv,
d2h, d2s, d2v);

DESCRIPTION:
Computes volume and residual enthalpy and entropy of a real mixture together with
first and second order temperature and pressure derivatives. Derivatives
are computed based on the number of output arguments. The enthalpy and
entropy are obtained from a cubic equation of state

\[ h^R(T, P) = RT (Z - 1) + 1/(\epsilon - \sigma) \frac{b_m}{(T dam/dT - a_m(T))f(Z, B)} \]
\[ s^R(T, P) = R \ln(Z - 1) + 1/(\epsilon - \sigma) \frac{dam/dT}{bm} f(Z, B) \]

where

\[ f(Z, B) = \ln\left( \frac{Z + \epsilon B}{Z + \sigma B} \right) \]

The output is formatted such that for the first order derivatives

Element 1 : temperature derivative
Element 2 : pressure derivative
Element 3+: composition derivatives

and for the symmetric second order derivatives

Element (1, 1) : temperature derivative
Element (2, 1) : temperature and pressure derivative

69
Element (2, 2) : pressure derivative
Elements (3+, 1) : temperature and composition derivatives
Elements (3+, 2) : pressure and composition derivatives
Elements (3+, 3+) : composition derivatives

REQUIRED PARAMETERS:
T - Temperature [K]
P - Temperature [Pa]
n - mole numbers [kmol]
phase - 0: vapor, 1: liquid
p - Vector with various parameters obtained by calling LoadParams
nargout - Number of output arguments
memaux - Auxiliary memory must be size
    nargout > 6: 39*NC + 6*NC*NC memory
    nargout > 3: 29*NC memory
    else : 10*NC memory

OPTIONAL PARAMETERS:
tol - Tolerance for Newton iterations (-1 for default)
itmax - Maximum number of Newton iterations (-1 for default)

RETURNS:
h - Residual enthalpy (scalar)
s - Residual entropy (scalar)
v - Volume (scalar)
dh - 1st derivatives (must be size 2 + NC)
ds - 1st derivatives (must be size 2 + NC)
dv - 1st derivatives (must be size 2 + NC)
d2h - 2nd derivatives (must be size (2 + NC)^2)
d2s - 2nd derivatives (must be size (2 + NC)^2)
d2v - 2nd derivatives (must be size (2 + NC)^2)

DEPENDENCIES:
MixParams
MixSolvePhEoS
34. MixSolvePhEoS (C)

Solve cubic equation of state for compressibility factor

SYNOPSIS:
MixSolvePhEoS(T, P, n, phase, p, tol, itmax, nargout, memaux,
Z,
dZT, dZP, dZn,
d2ZT, d2ZP,
d2ZTP, d2ZTn, d2ZPn, d2Zn);

DESCRIPTION:
Solves a cubic equation of state and the quadratic van der Waals mixing rules

\[ P = \frac{RT}{V - bm} - \frac{am(T)}{(V + \epsilon bm)(V + \sigma bm)} \]

for the compressibility factor. Other functions are

\[ am(T, n) = \sum_{i=1}^{N_C} \sum_{j=1}^{N_C} x_i x_j a_{ij}(T) \]
\[ bm(n) = \sum_{i=1}^{N_C} x_i b_{ij} \]
\[ a_{ij}(T) = (1 - k_{ij}) \sqrt{a_i(T) a_j(T)} \]
\[ x_i = \frac{n_i}{\sum_{j=1}^{N_C} n_j} \]
\[ a_i(T) = \alpha(Tr, omega) \Psi (RTc)^2/Pc \]
\[ b_i = \Omega RTc/Pc \]
\[ \alpha(Tr, omega) = (1 + m(omega) \times \sqrt{1 - Tr})^2 \]
\[ m(omega) = m_0 + m_1 omega + m_2 omega^2 \]
\[ Tr = T/Tc \]

REQUIRED PARAMETERS:
T  - Temperature [K]
P  - Pressure [Pa]
n  - mole numbers [kmol]
phase - 0: vapor, 1: liquid
p  - Vector with various parameters obtained by calling LoadParams
nargout - Number of output arguments
memaux - Auxiliary memory must be size

nargout > 4: 16*N_C + 2*N_C*N_C memory
nargout > 1: 12*N_C memory
else : 4*N_C memory

OPTIONAL PARAMETERS:
tol  - Tolerance for Newton iterations (-1 for default)
itmax - Maximum number of Newton iterations (-1 for default)

RETURNS:
Z    - Compressibility factor (scalar)
dZT  - 1st temperature derivative (scalar)
dZP  - 1st pressure derivative (scalar)
dZn  - 1st composition derivatives (must be size NC)
d2ZT - 2nd temperature derivative (scalar)
d2ZP - 2nd pressure derivative (scalar)
d2ZTP - 2nd temperature and pressure derivative (scalar)
d2ZTn - 2nd temperature and composition derivatives (must be size NC)
d2ZPn - 2nd pressure and composition derivatives (must be size NC)
d2Zn  - 2nd composition derivatives (must be size NC*NC)

DEPENDENCIES:
MixParams
SolvePolynomiumNewton
35. MixIdHSV (C)

Compute vapor and liquid enthalpy, entropy and volume of an ideal mixture

SYNOPSIS:
MixIdHSV(T, P, nv, nl, p, nargout, memaux,
   Hv, Sv, Vv, Hl, Sl, Vl,
   dHv, dSv, dVv, dHl, dSl, dVl,
   d2Hv, d2Sv, d2Vv, d2Hl, d2Sl, d2Vl);

DESCRIPTION:
Computes enthalpy, entropy and volume of an ideal mixture together
with first and second order temperature and pressure derivatives.
Derivatives are computed based on the number of output arguments.

The output is formatted such that for the first order derivatives

   Element 1 : temperature derivative
   Element 2 : pressure derivative
   Element 3+: composition derivatives

and for the symmetric second order derivatives

   Element (1, 1) : temperature derivative
   Element (2, 1) : temperature and pressure derivative
   Element (2, 2) : pressure derivative
   Elements (3+, 1) : temperature and composition derivatives
   Elements (3+, 2) : pressure and composition derivatives
   Elements (3+, 3+): composition derivatives

REQUIRED PARAMETERS:
T   - Temperature [K]
P   - Temperature [Pa]
nv  - mole numbers in vapor phase [kmol]
nl  - mole numbers in liquid phase [kmol]
p   - Vector with various parameters obtained by calling LoadParams
nargout - Number of output arguments
memaux - Auxiliary memory must be size
   nargout > 12: 32*NC memory
   nargout > 6: 29*NC memory
   else : 17*NC memory

RETURNS:
Hv   - Enthalpy of vapor phase (scalar)
Sv   - Entropy of vapor phase (scalar)
Vv   - Volume of vapor phase (scalar)
Hl - Enthalpy of liquid phase (scalar)
Sl - Entropy of liquid phase (scalar)
Vl - Volume of liquid phase (scalar)
dHv - First order derivatives (must have size (2 + NC))
dSv - First order derivatives (must have size (2 + NC))
dVv - First order derivatives (must have size (2 + NC))
dHl - First order derivatives (must have size (2 + NC))
dSl - First order derivatives (must have size (2 + NC))
dVl - First order derivatives (must have size (2 + NC))
d2Hv - Second order derivatives (must have size (2 + NC)^2)
d2Sv - Second order derivatives (must have size (2 + NC)^2)
d2Vv - Second order derivatives (must have size (2 + NC)^2)
d2Hl - Second order derivatives (must have size (2 + NC)^2)
d2Sl - Second order derivatives (must have size (2 + NC)^2)
d2Vl - Second order derivatives (must have size (2 + NC)^2)

DEPENDENCIES:
IdLiqVol
IdSatPres
IdGasHeatCap
IdGasHeatCapInt
36. MixIdVapHSV (C)

Compute enthalpy, entropy and volume of an ideal vapor mixture

SYNOPSIS:
MixIdVapHSV(T, P, nv, p, nargout, memaux,
    Hv, Sv, Vv,
    dHv, dSv, dVv,
    d2Hv, d2Sv, d2Vv);

DESCRIPTION:
Computes enthalpy, entropy and volume of an ideal vapor mixture based on the ideal gas law and DIPPR correlations. First and second order temperature and pressure derivatives are computed based on the number of output arguments.

The output is formatted such that for the first order derivatives

    Element 1 : temperature derivative
    Element 2 : pressure derivative
    Element 3+: composition derivatives

and for the symmetric second order derivatives

    Element (1, 1) : temperature derivative
    Element (2, 1) : temperature and pressure derivative
    Element (2, 2) : pressure derivative
    Elements (3+, 1) : temperature and composition derivatives
    Elements (3+, 2) : pressure and composition derivatives
    Elements (3+, 3+): composition derivatives

REQUIRED PARAMETERS:
T      - Temperature [K]
P      - Temperature [Pa]
nv     - mole numbers in vapor phase [kmol]
p      - Vector with various parameters obtained by calling LoadParams
nargout - Number of output arguments
memaux - Auxiliary memory must be size
    nargout > 6: 7*NC memory
    nargout > 3: 6*NC memory
    else : 5*NC memory

RETURNS:
Hv     - Enthalpy of vapor phase (scalar)
Sv     - Entropy of vapor phase (scalar)
Vv     - Volume of vapor phase (scalar)
dHv - First order derivatives (must have size (2 + NC))
dSv - First order derivatives (must have size (2 + NC))
dVv - First order derivatives (must have size (2 + NC))
d2Hv - Second order derivatives (must have size (2 + NC)^2)
d2Sv - Second order derivatives (must have size (2 + NC)^2)
d2Vv - Second order derivatives (must have size (2 + NC)^2)

DEPENDENCIES:
   IdGasHeatCap
   IdGasHeatCapInt
37. MixIdLiqHSV (C)

Compute enthalpy, entropy and volume of an ideal liquid mixture

SYNOPSIS:
MixIdLiqHSV(T, P, nl, p, nargout, memaux,
    Hl, Sl, Vl,
    dHl, dSl, dVl,
    d2Hl, d2Sl, d2Vl);

DESCRIPTION:
Computes enthalpy, entropy and volume of an ideal liquid mixture together with first and second order temperature and pressure derivatives. Derivatives are computed based on the number of output arguments.

The output is formatted such that for the first order derivatives

Element 1 : temperature derivative
Element 2 : pressure derivative
Element 3+: composition derivatives

and for the symmetric second order derivatives

Element (1, 1) : temperature derivative
Element (2, 1) : temperature and pressure derivative
Element (2, 2) : pressure derivative
Elements (3+, 1) : temperature and composition derivatives
Elements (3+, 2) : pressure and composition derivatives
Elements (3+, 3+) : composition derivatives

REQUIRED PARAMETERS:
T - Temperature [K]
P - Temperature [Pa]
nl - mole numbers in liquid phase [kmol]
p - Vector with various parameters obtained by calling LoadParams
nargout - Number of output arguments
memaux - Auxiliary memory must be size
    nargout > 6: 31*NC memory
    nargout > 2: 28*NC memory
    else : 16*NC memory

RETURNS:
Hl - Enthalpy of liquid phase (scalar)
Sl - Entropy of liquid phase (scalar)
Vl - Volume of liquid phase (scalar)
dHl - First order derivatives (must have size (2 + NC))
dS_l - First order derivatives (must have size \( (2 + NC) \))
dV_l - First order derivatives (must have size \( (2 + NC) \))
d2H_l - Second order derivatives (must have size \( (2 + NC)^2 \))
d2S_l - Second order derivatives (must have size \( (2 + NC)^2 \))
d2V_l - Second order derivatives (must have size \( (2 + NC)^2 \))

DEPENDENCIES:
  MixIdVapHSV
  IdLiqVol
  IdSatPres
  IdGasHeatCap
  IdGasHeatCapInt
38. PureRealHSV (C)

Compute pure component vapor and liquid enthalpy, entropy and volume

SYNOPSIS:

```c
PureRealHSV(T, P, p, tol, itmax, nargout, memaux,
    hv, sv, vv, hl, sl, vl,
    dhvT, dsvT, dvvT, dhvP, dsvP, dvvP,
    dh1T, dsl1T, dv1T, dh1P, dsl1P, dv1P,
    d2hvT, d2svT, d2vvT, d2hvP, d2svP, d2vvP,
    d2h1T, d2s1T, d2v1T, d2h1P, d2s1P, d2v1P);
```

DESCRIPTION:
Computes liquid enthalpy, entropy and volume of real pure components
together with first and second order temperature and pressure derivatives.
Derivatives are computed based on the number of output arguments.

The enthalpy and entropy are computed from ideal and residual
properties while the volume is obtained as the solution to a cubic
equation of state

REQUIRED PARAMETERS:

- `T` - Temperature [K]
- `P` - Temperature [Pa]
- `p` - Vector with various parameters obtained by calling LoadParams
- `nargout` - Number of output arguments
- `memaux` - Auxiliary memory must be size
  - `nargout > 9`: 50*NC memory
  - `nargout > 3`: 31*NC memory
  - else: 9*NC memory

OPTIONAL PARAMETERS:

- `tol` - Tolerance for Newton iterations (-1 for default)
- `itmax` - Maximum number of Newton iterations (-1 for default)

RETURNS:

- `hv` - Molar enthalpy of each component (must be size NC)
- `sv` - Molar entropy of each component (must be size NC)
- `vv` - Molar volume of each component (must be size NC)
- `hl` - Molar enthalpy of each component (must be size NC)
- `sl` - Molar entropy of each component (must be size NC)
- `vl` - Molar volume of each component (must be size NC)
- `dhvT` - 1st temperature derivative (must be size NC)
- `dsvT` - 1st temperature derivative (must be size NC)
- `dvvT` - 1st temperature derivative (must be size NC)
- `dhvP` - 1st pressure derivative (must be size NC)
- `dsvP` - 1st pressure derivative (must be size NC)
- `dvvP` - 1st pressure derivative (must be size NC)
dsvP  - 1st pressure derivative (must be size NC)
dsvP  - 1st pressure derivative (must be size NC)
dh1T  - 1st temperature derivative (must be size NC)
ds1T  - 1st temperature derivative (must be size NC)
dv1T  - 1st temperature derivative (must be size NC)
dh1P  - 1st pressure derivative (must be size NC)
ds1P  - 1st pressure derivative (must be size NC)
dslP  - 1st pressure derivative (must be size NC)
d2hvT  - 2nd temperature derivative (must be size NC)
d2svT  - 2nd temperature derivative (must be size NC)
d2vvT  - 2nd temperature derivative (must be size NC)
d2hvP  - 2nd pressure derivative (must be size NC)
d2svP  - 2nd pressure derivative (must be size NC)
d2vvP  - 2nd pressure derivative (must be size NC)
d2hvTP - 2nd pressure and temperature derivative (must be size NC)
d2svTP - 2nd pressure and temperature derivative (must be size NC)
d2vvTP - 2nd pressure and temperature derivative (must be size NC)
d2hlT  - 2nd temperature derivative (must be size NC)
d2slT  - 2nd temperature derivative (must be size NC)
d2vlT  - 2nd temperature derivative (must be size NC)
d2h1P  - 2nd pressure derivative (must be size NC)
d2s1P  - 2nd pressure derivative (must be size NC)
d2v1P  - 2nd pressure derivative (must be size NC)
d2h1TP - 2nd pressure and temperature derivative (must be size NC)
d2s1TP - 2nd pressure and temperature derivative (must be size NC)
d2v1TP - 2nd pressure and temperature derivative (must be size NC)

DEPENDENCIES:
  PureIdVapHSV
  PureResHSV
39. PureRealVapHSV (C)

Compute pure component vapor enthalpy, entropy and volume

SYNOPSIS:

PureRealVapHSV(T, P, p, tol, itmax, nargout, memaux,  
    hv, sv, vv,  
    dhvT, dsvT, dvvT, dhvP, dsvP, dvvP,  
    d2hvT, d2svT, d2vvT, d2hvP, d2svP, d2vvP,  
    d2hvTP, d2svTP, d2vvTP);

DESCRIPTION:

Computes vapor enthalpy, entropy and volume of real pure components  
together with first and second order temperature and pressure derivatives.  
Derivatives are computed based on the number of output arguments.

The enthalpy and entropy are computed from ideal and residual  
properties while the volume is obtained as the solution to a cubic  
equation of state

REQUIRED PARAMETERS:

T - Temperature [K]  
P - Temperature [Pa]  
p - Vector with various parameters obtained by calling LoadParams  
nargout - Number of output arguments  
memaux - Auxiliary memory must be size  
    nargout > 9: 40*NC memory  
    nargout > 3: 26*NC memory  
    else : 13*NC memory

OPTIONAL PARAMETERS:

tol - Tolerance for Newton iterations (-1 for default)  
itmax - Maximum number of Newton iterations (-1 for default)

RETURNS:

hv - Molar enthalpy of each component (must be size NC)  
sv - Molar entropy of each component (must be size NC)  
vv - Molar volume of each component (must be size NC)  
dhvT - 1st temperature derivative (must be size NC)  
dsvT - 1st temperature derivative (must be size NC)  
dvvT - 1st temperature derivative (must be size NC)  
dhvP - 1st pressure derivative (must be size NC)  
dsvP - 1st pressure derivative (must be size NC)  
dsvP - 1st pressure derivative (must be size NC)  
d2hvT - 2nd temperature derivative (must be size NC)  
d2svT - 2nd temperature derivative (must be size NC)  
d2vvT - 2nd temperature derivative (must be size NC)
d2hvP - 2nd pressure derivative (must be size NC)
d2svP - 2nd pressure derivative (must be size NC)
d2vvP - 2nd pressure derivative (must be size NC)
d2hvTP - 2nd pressure and temperature derivative (must be size NC)
d2svTP - 2nd pressure and temperature derivative (must be size NC)
d2vvTP - 2nd pressure and temperature derivative (must be size NC)

DEPENDENCIES:
  PureIdVapHSV
  PureResPhHSV
40. PureRealLiqHSV (C)

Compute pure component liquid enthalpy, entropy and volume

SYNOPSIS:

PureRealLiqHSV(T, P, p, tol, itmax, nargout, memaux,
  h1, s1, vl,
  dh1T, ds1T, dv1T, dh1P, ds1P, dv1P,
  d2h1T, d2s1T, d2v1T, d2h1P, d2s1P, d2v1P, d2h1TP, d2s1TP, d2v1TP);

DESCRIPTION:

Computes liquid enthalpy, entropy and volume of real pure components
together with first and second order temperature and pressure derivatives.
Derivatives are computed based on the number of output arguments.

The enthalpy and entropy are computed from ideal and residual
properties while the volume is obtained as the solution to a cubic
equation of state

REQUIRED PARAMETERS:

T - Temperature [K]
P - Temperature [Pa]
p - Vector with various parameters obtained by calling LoadParams
nargout - Number of output arguments
memaux - Auxiliary memory must be size
  nargout > 9: 40*NC memory
  nargout > 3: 26*NC memory
  else : 13*NC memory

OPTIONAL PARAMETERS:

tol - Tolerance for Newton iterations (-1 for default)
itmax - Maximum number of Newton iterations (-1 for default)

RETURNS:

h1 - Molar enthalpy of each component (must be size NC)
s1 - Molar entropy of each component (must be size NC)
vl - Molar volume of each component (must be size NC)
dh1T - 1st temperature derivative (must be size NC)
ds1T - 1st temperature derivative (must be size NC)
dv1T - 1st temperature derivative (must be size NC)
dh1P - 1st pressure derivative (must be size NC)
ds1P - 1st pressure derivative (must be size NC)
dv1P - 1st pressure derivative (must be size NC)
d2h1T - 2nd temperature derivative (must be size NC)
d2s1T - 2nd temperature derivative (must be size NC)
d2v1T - 2nd temperature derivative (must be size NC)
d2h1P - 2nd pressure derivative (must be size NC)

d2s1P - 2nd pressure derivative (must be size NC)

d2v1P - 2nd pressure derivative (must be size NC)

d2h1TP - 2nd pressure and temperature derivative (must be size NC)

d2s1TP - 2nd pressure and temperature derivative (must be size NC)

d2v1TP - 2nd pressure and temperature derivative (must be size NC)

DEPENDENCIES:

PureIdVapHSV
PureResPhHSV
41. PureResHSV (C)

Compute volume and residual enthalpy and entropy of pure components

SYNOPSIS:

```
PureResHSV(T, P, p, tol, itmax, nargout, memaux, 
    hv, sv, vv, hl, sl, vl, 
    dhvT, dsvT, dvvT, dhvP, dsvP, dvvP, 
    dh1T, dsl1T, dvl1T, dh1P, dsl1P, dvl1P, 
    d2hvT, d2svT, d2vvT, d2hvP, d2svP, d2vvP, d2hvTP, d2svTP, d2vvTP, 
    d2h1T, d2sl1T, d2v1T, d2h1P, d2sl1P, d2v1P, d2h1TP, d2sl1TP, d2v1TP);
```

DESCRIPTION:

Computes volume and residual enthalpy and entropy of a pure component using a cubic equation of state. First and second order temperature and pressure derivatives are computed based on the number of output arguments. The residual enthalpy and entropy are obtained from the cubic equation of state

\[ h^R(T, P) = RT \left( Z - 1 \right) + \frac{1}{(\epsilon - \sigma)} \left( \frac{d}{dT} f(Z, B) \right) \]

\[ s^R(T, P) = R \ln(Z - 1) + \frac{1}{(\epsilon - \sigma)} \left( \frac{d}{dT} f(Z, B) \right) \]

where

\[ f(Z, B) = \ln\left( \frac{Z + \epsilon B}{Z + \sigma B} \right) \]

The volume is obtained by solution of the cubic equation of state

\[ P = \frac{RT}{V - b} - \frac{a(T)}{(V^2 + 2Vb - b^2)} \]

\[ a(T) = \alpha(Tr, \omega) \Psi \left( \frac{RTc}{Pc} \right)^2 \]

\[ b = \Omega \frac{RTc}{Pc} \]

\[ \alpha(Tr, \omega) = (1 + m(\omega) \sqrt{1 - Tr})^2 \]

\[ m(\omega) = m_0 + m_1 \omega + m_2 \omega^2 \]

\[ Tr = T/Tc \]

\[ \Psi = 0.45724 \]

\[ \Omega = 0.07779 \]

REQUIRED PARAMETERS:

- T: Temperature [K]
- P: Temperature [Pa]
- p: Vector with various parameters obtained by calling LoadParams
- nargout: Number of output arguments
- memaux: Auxiliary memory must be size
  - nargout > 9: 21*NC memory
  - nargout > 3: 13*NC memory
  - else: 7*NC memory

85
OPTIONAL PARAMETERS:

- tol - Tolerance for Newton iterations (-1 for default)
- itmax - Maximum number of Newton iterations (-1 for default)

RETURNS:

- hv - Molar enthalpy of each component
- sv - Molar entropy of each component
- vv - Molar volume of each component
- hl - Molar enthalpy of each component
- sl - Molar entropy of each component
- vl - Molar volume of each component
- dhvT - 1st temperature derivative (must be size NC)
- dsvT - 1st temperature derivative (must be size NC)
- dvvT - 1st temperature derivative (must be size NC)
- dhvP - 1st pressure derivative (must be size NC)
- dsvP - 1st pressure derivative (must be size NC)
- dsvP - 1st pressure derivative (must be size NC)
- dhlT - 1st temperature derivative (must be size NC)
- dslT - 1st temperature derivative (must be size NC)
- dv1T - 1st temperature derivative (must be size NC)
- dh1P - 1st pressure derivative (must be size NC)
- dslP - 1st pressure derivative (must be size NC)
- dslP - 1st pressure derivative (must be size NC)
- d2hvT - 2nd temperature derivative (must be size NC)
- d2svT - 2nd temperature derivative (must be size NC)
- d2vvT - 2nd temperature derivative (must be size NC)
- d2hvT - 2nd pressure derivative (must be size NC)
- d2svT - 2nd pressure derivative (must be size NC)
- d2vvT - 2nd pressure derivative (must be size NC)
- d2hvTP - 2nd pressure and temperature derivative (must be size NC)
- d2svTP - 2nd pressure and temperature derivative (must be size NC)
- d2vvTP - 2nd pressure and temperature derivative (must be size NC)
- d2hlT - 2nd temperature derivative (must be size NC)
- d2slT - 2nd temperature derivative (must be size NC)
- d2vlT - 2nd temperature derivative (must be size NC)
- d2hlP - 2nd pressure derivative (must be size NC)
- d2slP - 2nd pressure derivative (must be size NC)
- d2vlP - 2nd pressure derivative (must be size NC)
- d2hlTP - 2nd pressure and temperature derivative (must be size NC)
- d2slTP - 2nd pressure and temperature derivative (must be size NC)
- d2vlTP - 2nd pressure and temperature derivative (must be size NC)

DEPENDENCIES:

- PureParams
- PureSolveEoS
42. PureResPhHSV (C)

Compute volume and residual enthalpy and entropy of pure components

SYNOPSIS:
```
PureResPhHSV(T, P, phase, p, tol, itmax, nargout, memaux,
   h, s, v,
   dhT, dsT, dvT, dhP, dsP, dvP,
   d2hT, d2sT, d2vT, d2hP, d2sP, d2vP, d2hTP, d2sTP, d2vTP);
```

DESCRIPTION:
Computes volume and residual enthalpy and entropy of a pure components using a cubic equation of state. First and second order temperature and pressure derivatives are computed based on the number of output arguments. The residual enthalpy and entropy are obtained from the cubic equation of state

\[
h^R(T, P) = RT \ (Z - 1) + 1/((\epsilon/\sigma) b) (T \ da/dT - a(T)) \ f(Z, B)
\]

\[
s^R(T, P) = R \ln (Z - 1) + 1/((\epsilon/\sigma) b) \ \ da/dT \ f(Z, B)
\]

where

\[
f(Z, B) = \ln \left( \frac{Z + \epsilon B}{Z + \sigma B} \right)
\]

The volume is obtained by solution of the cubic equation of state

\[
P = \frac{RT}{V - b} - \frac{a(T) + 2Vb - b^2}{(V^2 + 2Vb - b^2)}
\]

\[
a(T) = \alpha(Tr, \omega) \ \Psi \ (RTc)^2/Pc
\]

\[
b = \Omega \ RTc/Pc
\]

\[
\alpha(Tr, \omega) = (1 + m(\omega) \sqrt{1 - Tr})^2
\]

\[
m(\omega) = m_0 + m_1 \ \omega + m_2 \ \omega^2
\]

\[
Tr = T/Tc
\]

\[
\Psi = 0.45724
\]

\[
\Omega = 0.07779
\]

REQUIRED PARAMETERS:

| T  | - Temperature [K] |
| P  | - Temperature [Pa] |
| phase | - 0: vapor, 1: liquid |
| p  | - Vector with various parameters obtained by calling LoadParams |
| nargout | - Number of output arguments |
| memaux | - Auxiliary memory must be size |

nargout > 9: 15*NC memory
nargout > 3: 10*NC memory
else : 6*NC memory
OPTIONAL PARAMETERS:

tol - Tolerance for Newton iterations (-1 for default)
itmax - Maximum number of Newton iterations (-1 for default)

RETURNS:

h - Molar enthalpy of each component
s - Molar entropy of each component
v - Molar volume of each component
dhT - 1st temperature derivative (must be size NC)
dsT - 1st temperature derivative (must be size NC)
dvT - 1st temperature derivative (must be size NC)
dhP - 1st pressure derivative (must be size NC)
dsP - 1st pressure derivative (must be size NC)
dvP - 1st pressure derivative (must be size NC)
d2hT - 2nd temperature derivative (must be size NC)
d2sT - 2nd temperature derivative (must be size NC)
d2vT - 2nd temperature derivative (must be size NC)
d2hP - 2nd pressure derivative (must be size NC)
d2sP - 2nd pressure derivative (must be size NC)
d2vP - 2nd pressure derivative (must be size NC)
d2hTP - 2nd pressure and temperature derivative (must be size NC)
d2sTP - 2nd pressure and temperature derivative (must be size NC)
d2vTP - 2nd pressure and temperature derivative (must be size NC)

DEPENDENCIES:

PureParams
PureSolvePhEoS
Solve cubic equation of state for compressibility factor

SYNOPSIS:
PureSolveEoS(T, P, p, tol, itmax, nargout, memaux, 
Zv, Zl 
dZvT, dZvP, dZlT, dZlP, 
d2ZvT, d2ZvP, d2ZlT, d2ZlP, d2ZlTP);

DESCRIPTION:
Solves a cubic equation of state

\[ P = \frac{RT}{V - b} - \frac{a(T)}{(V + \epsilon b)(V + \sigma b)} \]

for the compressibility factor. Other functions are

\[ a(T) = \alpha(Tr, \omega) \Psi \left( \frac{RTc}{Pc} \right)^2 \]
\[ b = \Omega \frac{RTc}{Pc} \]
\[ \alpha(Tr, \omega) = (1 + m(\omega)\sqrt{1 - Tr})^2 \]
\[ m(\omega) = m_0 + m_1 \omega + m_2 \omega^2 \]
\[ Tr = \frac{T}{Tc} \]

REQUIRED PARAMETERS:
T - Temperature [K]
P - Pressure [Pa]
p - Vector with various parameters obtained by calling LoadParams
nargout - Number of output arguments
memaux - Auxiliary memory must be size
  nargout > 3: 4*NC memory
  nargout > 1: 3*NC memory
  else : 2*NC memory

OPTIONAL PARAMETERS:
tol - Tolerance for Newton iterations (-1 for default)
itmax - Maximum number of Newton iterations (-1 for default)

RETURNS:
Zv - Compressibility factor
Zl - Compressibility factor
dZvT - 1st temperature derivatives (must be size NC)
dZvP - 1st pressure derivatives (must be size NC)
dZlT - 1st temperature derivatives (must be size NC)
dZlP - 1st pressure derivatives (must be size NC)
d2ZvT - 2nd temperature derivatives (must be size NC)
d2ZvP - 2nd pressure derivatives (must be size NC)
d2ZvTP  - 2nd temperature and pressure derivatives (must be size NC)
d2Z1T  - 2nd temperature derivatives (must be size NC)
d2Z1P  - 2nd pressure derivatives (must be size NC)
d2Z1TP - 2nd temperature and pressure derivatives (must be size NC)

DEPENDENCIES:
  PureParams
  SolvePolynomiumNewton
44. PureSolvePhEoS (C)

Solve cubic equation of state for compressibility factor

SYNOPSIS:
PureSolvePhEoS(T, P, phase, p, tol, itmax, nargout, memaux,
Z, dZT, dZP, d2ZT, d2ZP, d2ZTP);

DESCRIPTION:
Solves a cubic equation of state

\[ P = \frac{RT}{V - b} - \frac{a(T)}{(V + \epsilon b)(V + \sigma b)} \]

for the compressibility factor. Other functions are

\[ a(T) = \alpha(Tr, \omega) \Psi \left( \frac{RTc}{Pc} \right)^2 \]
\[ b = \Omega \frac{RTc}{Pc} \]
\[ \alpha(Tr, \omega) = (1 + m(\omega) \sqrt{1 - Tr})^2 \]
\[ m(\omega) = m_0 + m_1 \omega + m_2 \omega^2 \]
\[ Tr = \frac{T}{Tc} \]

REQUIRED PARAMETERS:
- T - Temperature [K]
- P - Pressure [Pa]
- phase - 0: vapor, 1: liquid
- p - Vector with various parameters obtained by calling LoadParams
- nargout - Number of output arguments
- memaux - Auxiliary memory must be size
  - nargout > 3: 4*NC memory
  - nargout > 1: 3*NC memory
  - else: 2*NC memory

OPTIONAL PARAMETERS:
- tol - Tolerance for Newton iterations (-1 for default)
- itmax - Maximum number of Newton iterations (-1 for default)

RETURNS:
- Z - Compressibility factor
- dZT - 1st temperature derivatives (must be size NC)
- dZP - 1st pressure derivatives (must be size NC)
- d2ZT - 2nd temperature derivatives (must be size NC)
- d2ZP - 2nd pressure derivatives (must be size NC)
- d2ZTP - 2nd temperature and pressure derivatives (must be size NC)
DEPENDENCIES:
    PureParams
    SolvePolynomiumNewton
45. PureIdHSV (C)

Compute pure component molar vapor and liquid enthalpy, entropy and volume

SYNOPSIS:
PureIdHSV(T, P, p, nargout, memaux,
    hv, sv, vv, hl, sl, vl,
    dhvT, dsvT, dvvT,
    dsvP, dvvP
    dh1T, ds1T, dv1T,
    dh1P, ds1P,
    d2hvT, d2svT,
        d2svP, d2vvP,
        d2vvP,
    d2hlT, d2slT, d2vlT,
    d2hlP, d2slP);

DESCRIPTION:
Computes molar enthalpy, entropy and volume of a set of pure components together with first and second order temperature and pressure derivatives. Derivatives are computed based on the number of output arguments.

Certain derivatives are not returned by this routine because they are zero (e.g. pressure derivative of volume d/dP v^l = 0)

REQUIRED PARAMETERS:
T   - Temperature [K]
P   - Temperature [Pa]
p   - Vector with various parameters obtained by calling LoadParams
nargout - Number of output arguments
memaux - Auxiliary memory must be size
    - nargout > 16: 19*NC memory
    - nargout > 6: 17*NC memory
    - else : 11*NC memory

RETURNS:
hv - Molar enthalpy of each component (must be size NC)
sv - Molar entropy of each component (must be size NC)
vv - Molar volume of each component (must be size NC)
hl - Molar enthalpy of each component (must be size NC)
sl - Molar entropy of each component (must be size NC)
vl - Molar volume of each component (must be size NC)
dhvT - 1st temperature derivative (must be size NC)
dsvT - 1st temperature derivative (must be size NC)
dvvT - 1st temperature derivative (must be size NC)
dsvP - 1st pressure derivative (must be size NC)
dvvP - 1st pressure derivative (must be size NC)
dhlT - 1st temperature derivative (must be size NC)
dslT - 1st temperature derivative (must be size NC)
dvlT - 1st temperature derivative (must be size NC)
dhlP - 1st pressure derivative (must be size NC)
dslP - 1st pressure derivative (must be size NC)
d2hvT - 2nd temperature derivative (must be size NC)
d2svT - 2nd temperature derivative (must be size NC)
d2svP - 2nd pressure derivative (must be size NC)
d2vvP - 2nd pressure derivative (must be size NC)
d2vvTP - 2nd pressure and temperature derivative (must be size NC)
d2h1T - 2nd temperature derivative (must be size NC)
d2s1T - 2nd temperature derivative (must be size NC)
d2v1T - 2nd temperature derivative (must be size NC)
d2h1TP - 2nd pressure and temperature derivative (must be size NC)
d2s1TP - 2nd pressure and temperature derivative (must be size NC)

DEPENDENCIES:
  IdLiqVol
  IdSatPres
  IdGasHeatCap
  IdGasHeatCapInt
46. PureIdVapHSV (C)

Compute pure component molar vapor enthalpy, entropy and volume

SYNOPSIS:

```
PureIdVapHSV(T, P, p, nargout, memaux,
    hv, sv, vv,
    dhvT, dsvT, dvvT,
    dsvP, dvvP
    d2hvT, d2svT,
    d2svP, d2vvP,
    d2vvTP);
```

DESCRIPTION:
Computes molar enthalpy, entropy and volume of a set of pure components using the ideal gas law and DIPPR correlations. First and second order temperature and pressure derivatives are computed based on the number of output arguments.

Certain derivatives are not returned by this routine because they are zero (e.g. pressure derivative of enthalpy d/dP h−v)

REQUIRED PARAMETERS:

- **T** - Temperature [K]
- **P** - Temperature [Pa]
- **p** - Vector with various parameters obtained by calling LoadParams
- **nargout** - Number of output arguments
- **memaux** - Auxiliary memory must be size
  - nargout > 8: 4*NC memory
  - nargout > 3: 3*NC memory
  - else : 2*NC memory

RETURNS:

- **hv** - Molar enthalpy of each component (must be size NC)
- **sv** - Molar entropy of each component (must be size NC)
- **vv** - Molar volume of each component (must be size NC)
- **dhvT** - 1st temperature derivative (must be size NC)
- **dsvT** - 1st temperature derivative (must be size NC)
- **dvvT** - 1st temperature derivative (must be size NC)
- **dsvP** - 1st pressure derivative (must be size NC)
- **dvvP** - 1st pressure derivative (must be size NC)
- **d2hvT** - 2nd temperature derivative (must be size NC)
- **d2svT** - 2nd temperature derivative (must be size NC)
- **d2svP** - 2nd temperature derivative (must be size NC)
- **d2vvP** - 2nd pressure derivative (must be size NC)
- **d2vvTP** - 2nd pressure and temperature derivative (must be size NC)
DEPENDENCIES:
   IdGasHeatCap
   IdGasHeatCapInt
47. PureIdLiqHSV (C)

Compute pure component molar liquid enthalpy, entropy and volume

SYNOPSIS:

 PureIdLiqHSV(T, P, p, nargout, memaux,  
       hl, sl, vl,  
       dh1T, ds1T, dv1T,  
       dh1P, ds1P,  
       d2h1T, d2s1T, d2v1T,  
       d2h1TP, d2s1TP);

DESCRIPTION:

Computes molar enthalpy, entropy and volume of a set of pure components together with first and second order temperature and pressure derivatives. Derivatives are computed based on the number of output arguments.

Certain derivatives are not returned by this routine because they are zero (e.g. pressure derivative of volume d/dP v^l = 0)

REQUIRED PARAMETERS:

T  - Temperature [K]  
P  - Temperature [Pa]  
p  - Vector with various parameters obtained by calling LoadParams  
nargout - Number of output arguments  
memaux - Auxiliary memory must be size  
           nargout > 8: 19*NC memory  
           nargout > 3: 17*NC memory  
           else : 11*NC memory

RETURNS:

hl  - Molar enthalpy of each component (must be size NC)  
sl  - Molar entropy of each component (must be size NC)  
v1  - Molar volume of each component (must be size NC)  
dh1T  - 1st temperature derivative (must be size NC)  
ds1T  - 1st temperature derivative (must be size NC)  
dv1T  - 1st temperature derivative (must be size NC)  
dh1P  - 1st pressure derivative (must be size NC)  
ds1P  - 1st pressure derivative (must be size NC)  
d2h1T  - 2nd temperature derivative (must be size NC)  
d2s1T  - 2nd temperature derivative (must be size NC)  
d2v1T  - 2nd temperature derivative (must be size NC)  
d2h1TP  - 2nd pressure and temperature derivative (must be size NC)  
d2s1TP  - 2nd pressure and temperature derivative (must be size NC)
DEPENDENCIES:
    IdLiqVol
    IdSatPres
    IdGasHeatCap
    IdGasHeatCapInt
Compute van der Waals mixing rules $a_m(T, n)$ and $b_m(n)$

SYNOPSIS:

MixParams(T, n, p, nargout, memaux,
        am, bm,
        damT, damn, dbmn,
        d2amT, d2amTn, d2amnn, d2bmn,
        d3amT, d3amT2n, d3amTn2);

DESCRIPTION:

Computes the van der Waals mixing parameters $a_m(T, n)$ and $b_m(n)$ based on the pure component properties $a_i(T)$ and $b_i$

\[
\begin{align*}
    a_m(T, n) &= \sum_{i=1}^{N_C} \sum_{j=1}^{N_C} x_i x_j a_{ij}(T) \\
    b_m(n) &= \sum_{i=1}^{N_C} x_i b_{ij} \\
    a_{ij}(T) &= (1 - k_{ij}) \sqrt{a_i(T) a_j(T)} \\
    x_i &= \frac{n_i}{\sum_{j=1}^{N_C} n_j}
\end{align*}
\]

The computations of derivatives are only computed if requested as output.

REQUIRED PARAMETERS:

- $T$ - Temperature [K]
- $n$ - Mole numbers [kmol]
- $p$ - Vector with various parameters obtained by calling LoadParams
- $nargout$ - Number of output arguments
- $memaux$ - Auxiliary memory must be size
  - $nargout > 9$: 10*NC memory
  - $nargout > 5$: 8*NC memory
  - $nargout > 2$: 6*NC memory
  - else: 4*NC memory

RETURNS:

- $am$ - Mixing rule parameter (scalar)
- $bm$ - Mixing rule parameter (scalar)
- $damT$ - 1st temperature derivative (scalar)
- $damn$ - 1st composition derivatives (must be size NC)
- $dbmn$ - 1st composition derivative of (must be size NC)
- $d2amT$ - 2nd temperature derivative (scalar)
- $d2amTn$ - 2nd temperature and composition derivatives (must be size NC)
- $d2amnn$ - 2nd composition derivatives (must be size NC*NC)
- $d2bmn$ - 2nd composition derivatives of (must be size NC*NC)
- $d3amT$ - 3rd temperature derivative (scalar)
- $d3amT2n$ - 3rd temperature (x2) and composition derivatives (must be size NC)
d3amTn2 - 3rd temperature and composition (x2) derivatives (must be size NC\times NC)

DEPENDENCIES:
  PureParams
49. PureParams (C)

Compute pure component parameters $a_i(T)$ and $b_i$

SYNOPSIS:
 PureParams(T, p, nargout, memaux, 
 a, b,  
daT, d2aT, d3aT);

DESCRIPTION:
 Computes the pure component parameters $a_i(T)$ and $b_i$ for $i = 1,\ldots,N_C$

\begin{align*}
a_i(T) &= \alpha(\text{Tr}, \omega) \Psi \frac{(RT_c)^2}{P_c} \\
b_i &= \Omega \frac{RT_c}{P_c} \\
\alpha(\text{Tr}, \omega) &= (1 + m(\omega) \sqrt{1 - \text{Tr}})^2 \\
m(\omega) &= m_0 + m_1 \omega + m_2 \omega^2 \\
\text{Tr} &= \frac{T}{T_c}
\end{align*}

The computations of derivatives are only computed if requested as output.

REQUIRED PARAMETERS:
 T - Temperature [K]
 p - Vector with various parameters obtained by calling LoadParams
 nargout - Number of output arguments

RETURNS:
 a - Pure component parameter (must have size NC)
b - Pure component parameter (must have size NC)
daT - 1st temperature derivative (must have size NC)
d2aT - 2nd temperature derivative (must have size NC)
d3aT - 3rd temperature derivative (must have size NC)
50. SolvePolynomiumNewton (C)

Solve cubic equation iteratively

SYNOPSIS:
SolvePolynomiumNewton(d2, d1, d0, tol, itmax, Z);

DESCRIPTION:
Solves the cubic equation

\[ q(Z) = Z^3 + d2 \cdot Z^2 + d1 \cdot Z + d0 = 0 \]

using an iterative Newton approach. The approach is terminated when

\[ |q(Z)| < tol \quad \text{and} \quad |\Delta Z| = |q(Z)/q'(Z)| < tol. \]

REQUIRED PARAMETERS:
- \( d2 \) - Quadratic coefficient (scalar)
- \( d1 \) - Linear coefficient (scalar)
- \( d0 \) - Constant coefficient (scalar)
- \( tol \) - Tolerance for stopping criteria (scalar)
- \( itmax \) - Maximum number of iterations (scalar)
- \( Z \) - On input: initial guess for \( Z \) (scalar)

RETURNS:
- \( Z \) - A single real root of the cubic equation (scalar)
51. **IdGasHeatCap (C)**

Compute ideal gas heat capacity

**SYNOPSIS:**

```
IdGasHeatCap(T, p, nargout, 
    cp, dcp);
```

**DESCRIPTION:**

Computes ideal gas heat capacity using the DIPPR correlation

\[
c_{P,k}^{ig}(T) = A_k + B_k \left( \frac{C_k}{T} \right)^{2} \sinh \left( \frac{C_k}{T} \right) \\
    + D_k \left( \frac{E_k}{T} \right)^{2} \cosh \left( \frac{E_k}{T} \right)
\]

and the first order derivative

**REQUIRED PARAMETERS:**

- **T** - Temperature, [K]
- **params** - Vector with various parameters obtained by calling LoadParams
- **nargout** - Number of output arguments

**RETURNS:**

- **cp** - Ideal gas heat capacity of each component (must have size NC)
- **dcp** - First temperature derivative (must have size NC)
52. IdGasHeatCapInt (C)

Compute integrals of ideal gas heat capacity

SYNOPSIS:
IdGasHeatCapInt(T, p, nargout, intcp, intcpT);

DESCRIPTION:
Computes the following integrals of the ideal gas heat capacity

\[ \int_{T_0}^{T} c_{P,k}^ig(tau) \, dtau \]
\[ \int_{T_0}^{T} c_{P,k}^ig(tau) / tau \, dtau \]

where ideal gas heat capacity is defined by the DIPPR correlation

\[ c_{P,k}^ig(T) = A_k + B_k((C_k/T)/\sinh(C_k/T))^2 \]
\[ - D_k((E_k/T)/\cosh(E_k/T))^2 \]

REQUIRED PARAMETERS:
T - Temperature, [K]
p - Vector with various parameters obtained by calling LoadParams
nargout - Number of output arguments

RETURNS:
intcp - Integral of cp from T0 to T (must have size NC)
intcpT - Integral of cp/T from T0 to T (must have size NC)
53. IdLiqVol (C)

Compute liquid volume

SYNOPSIS:
IdLiqVol(T, p, nargout,
  vl, dvl, d2vl, d3vl);

DESCRIPTION:
Computes liquid volume based on the DIPPR correlation

\[ v_{1,k}(T) = B_k^{1 + (1 - T/C_k)^D_k}/A_k \]

The first, second and third order temperature derivatives are computed depending on the number of output arguments.

NOTE:
The DIPPR correlation is not limited to ideal substances as the name suggests but in this thermodynamic library it is only applied in the computation of ideal liquid properties.

REQUIRED PARAMETERS:
  T - Temperature [K]
  p - Vector with various parameters obtained by calling LoadParams
  nargout - Number of output arguments

RETURNS:
  vl - Liquid volumes for each component, [m^3], (must have size NC)
  dvl - First temperature derivative (must have size NC)
  d2vl - Second temperature derivative (must have size NC)
  d3vl - Third temperature derivative (must have size NC)
54. IdSatPres (C)

Compute saturation pressure

SYNOPSIS:

IdSatPres(T, p, nargout,
Psat, dPsat, d2Psat, d3Psat);

DESCRIPTION:
Computes saturation pressure using the DIPPR correlation

\[ \ln Psat_k = A_k + B_k/T + C_k \ln(T) + D_k T^E_k \]

The first, second and third order temperature derivatives are computed depending on the number of output arguments.

NOTE:
The DIPPR correlation is not limited to ideal substances as the name suggests but in this thermodynamic library it is only applied in the computation of ideal liquid properties.

REQUIRED PARAMETERS:

- T - Temperature [K]
- p - Vector with various parameters obtained by calling LoadParams
- nargout - Number of output arguments

RETURNS:

- Psat - Vapor pressures for each component, [Pa], (must have size NC)
- dPsat - First temperature derivative (must have size NC)
- d2Psat - Second temperature derivative (must have size NC)
- d3Psat - Third temperature derivative (must have size NC)