

An open-source thermodynamic software library

Documentation pages for v. 1.1.4

Tobias K. S. Ritschel, 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	MixRealHSV (Matlab/Mex)	5
2	MixRealVapHSV (Matlab/Mex)	7
3	MixRealLiqHSV (Matlab/Mex)	9
4	MixResPhHSV (Matlab/Mex)	11
5	MixFug (Matlab/Mex)	13
6	MixSolvePhEoS (Matlab/Mex)	15
7	MixIdHSV (Matlab/Mex)	17
8	MixIdVapHSV (Matlab/Mex)	19
9	MixIdLiqHSV (Matlab/Mex)	21
10	PureRealHSV (Matlab/Mex)	23
11	PureRealVapHSV (Matlab/Mex)	25
12	PureRealLiqHSV (Matlab/Mex)	27
13	PureResHSV (Matlab/Mex)	29
14	PureResPhHSV (Matlab/Mex)	32
15	PureFug (Matlab/Mex)	34
16	PureSolveEoS (Matlab/Mex)	36
17	PureSolvePhEoS (Matlab/Mex)	38
18	PureIdVapHSV (Matlab/Mex)	40
19	PureIdLiqHSV (Matlab/Mex)	42
20	MixParams (Matlab/Mex)	44
21	PureParams (Matlab/Mex)	46
22	SolvePolynomiumCardano (Matlab/Mex)	47
23	SolvePolynomiumNewton (Matlab/Mex)	48
24	IdGasHeatCap (Matlab/Mex)	49
25	IdGasHeatCapInt (Matlab/Mex)	50

26	IdLiqVol (Matlab/Mex)	51
27	IdSatPres (Matlab/Mex)	52
28	MixRealHSV (C)	53
29	MixRealVapHSV (C)	55
30	MixRealLiqHSV (C)	57
31	MixResPhHSV (C)	59
32	MixSolvePhEoS (C)	61
33	MixIdHSV (C)	63
34	MixIdVapHSV (C)	65
35	MixIdLiqHSV (C)	67
36	PureRealHSV (C)	69
37	PureRealVapHSV (C)	71
38	PureRealLiqHSV (C)	73
39	PureResHSV (C)	75
40	PureResPhHSV (C)	77
41	PureSolveEoS (C)	79
42	PureSolvePhEoS (C)	81
43	PureIdHSV (C)	82
44	PureIdVapHSV (C)	83
45	PureIdLiqHSV (C)	84
46	MixParams (C)	85
47	PureParams (C)	86
48	SolvePolynomiumNewton (C)	87
49	IdGasHeatCap (C)	88
50	IdGasHeatCapInt (C)	89
51	IdLiqVol (C)	90

1. MixRealHSV (Matlab/Mex)

Compute enthalpy, entropy and volume of a real liquid mixture

SYNOPSIS:

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

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
```

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
```

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

PureIdHSV	PureRealHSV	PureSolveEoS
PureIdVapHSV	PureRealVapHSV	PureSolvePhEoS
PureIdLiqHSV	PureRealLiqHSV	PureResHSV
MixIdHSV	MixRealHSV	PureResPhHSV
MixIdVapHSV	MixRealVapHSV	MixSolvePhEoS
MixIdLiqHSV	MixRealLiqHSV	MixResPhHSV
IdLiqVol	MixParams	PureEvalEoS
IdSatPres	PureParams	MixEvalEoS
IdGasHeatCap	SolvePolynomiumNewton	
IdGasHeatCapInt		

2. MixRealVapHSV (Matlab/Mex)

Compute enthalpy, entropy and volume of a real vapor mixture

SYNOPSIS:

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

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]  
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:

MixIdVapHSV
MixResPhHSV

See also LoadParams

PureIdHSV
PureIdVapHSV
PureIdLiqHSV
MixIdHSV
MixIdVapHSV
MixIdLiqHSV
IdLiqVol
IdSatPres
IdGasHeatCap
IdGasHeatCapInt

PureRealHSV
PureRealVapHSV
PureRealLiqHSV
MixRealHSV
MixRealVapHSV
MixRealLiqHSV
MixParams
PureParams
SolvePolynomiumNewton

PureSolveEoS
PureSolvePhEoS
PureResHSV
PureResPhHSV
MixSolvePhEoS
MixResPhHSV
PureEvalEoS
MixEvalEoS

3. MixRealLiqHSV (Matlab/Mex)

Compute enthalpy, entropy and volume of a real liquid mixture

SYNOPSIS:

```
[H1, S1, V1, ...  
    dH1, dS1, dV1, ...  
    d2H1, d2S1, d2V1] = ...  
    MixRealLiqHSV(T, P, nl, params, params)
```

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]  
params     - Vector with various parameters obtained by calling  
             LoadParams
```

RETURNS:

```
H1         - Enthalpy of liquid phase  
S1         - Entropy of liquid phase  
V1         - Volume of liquid phase  
dH1        - First order derivatives of liquid enthalpy  
dS1        - First order derivatives of liquid entropy  
dV1        - First order derivatives of liquid volume  
d2H1       - Second order derivatives of liquid enthalpy  
d2S1       - Second order derivatives of liquid entropy  
d2V1       - Second order derivatives of liquid volume
```

DEPENDENCIES:

MixIdVapHSV
MixResPhHSV

See also LoadParams

PureIdHSV
PureIdVapHSV
PureIdLiqHSV
MixIdHSV
MixIdVapHSV
MixIdLiqHSV
IdLiqVol
IdSatPres
IdGasHeatCap
IdGasHeatCapInt

PureRealHSV
PureRealVapHSV
PureRealLiqHSV
MixRealHSV
MixRealVapHSV
MixRealLiqHSV
MixParams
PureParams
SolvePolynomiumNewton

PureSolveEoS
PureSolvePhEoS
PureResHSV
PureResPhHSV
MixSolvePhEoS
MixResPhHSV
PureEvalEoS
MixEvalEoS

4. 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

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

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

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

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

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

PureIdHSV	PureRealHSV	PureSolveEoS
PureIdVapHSV	PureRealVapHSV	PureSolvePhEoS
PureIdLiqHSV	PureRealLiqHSV	PureResHSV
MixIdHSV	MixRealHSV	PureResPhHSV
MixIdVapHSV	MixRealVapHSV	MixSolvePhEoS
MixIdLiqHSV	MixRealLiqHSV	MixResPhHSV
IdLiqVol	MixParams	PureEvalEoS
IdSatPres	PureParams	MixEvalEoS
IdGasHeatCap	SolvePolynomiumNewton	
IdGasHeatCapInt		

5. MixFug (Matlab/Mex)

Compute logarithmic fugacity coefficients of a real mixture

SYNOPSIS:

```
[lnphi, dlnphi, d2lnphi] = MixFug(T, P, n, phase, params, tol, 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

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

where

$$f(Z, B) = \ln((Z + \epsilon B) / (Z + \sigma B))$$

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:

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

RETURNS:

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

DEPENDENCIES:

MixSolveEoS

See also LoadParams

PureIdHSV	PureRealHSV	PureSolveEoS
PureIdVapHSV	PureRealVapHSV	PureSolvePhEoS
PureIdLiqHSV	PureRealLiqHSV	PureResHSV
MixIdHSV	MixRealHSV	PureResPhHSV
MixIdVapHSV	MixRealVapHSV	MixSolvePhEoS
MixIdLiqHSV	MixRealLiqHSV	MixResPhHSV
IdLiqVol	MixParams	PureEvalEoS
IdSatPres	PureParams	MixEvalEoS
IdGasHeatCap	SolvePolynomiumNewton	
IdGasHeatCapInt		

6. MixSolvePhEoS (Matlab/Mex)

Solve cubic equation of state for compressibility factor

SYNOPSIS:

```
[Z, ...  
  dZT, dZP, dZn, ...  
  d2ZT, d2ZP, d2ZTP, d2ZTn, d2ZPn, d2Zn] = ...  
  MixSolvePhEoS(T, P, n, phase, params, tol, itmax)
```

DESCRIPTION:

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

$$P = RT/(V - bm) - am(T)/((V + \epsilon bm)(V + \sigma bm))$$

for the compressibility factor. Other functions are

$$\begin{aligned} am(T, n) &= \sum_{i=1}^{N_C} \sum_{j=1}^{N_C} x_i x_j a_{ij}(T) \\ bm(n) &= \sum_{i=1}^{N_C} x_i b_{ij} \\ a_{ij}(T) &= (1 - k_{ij}) \sqrt{a_i(T) a_j(T)} \\ x_i &= n_i / \sum_{j=1}^{N_C} n_j \\ a_i(T) &= \alpha(\text{Tr}, \omega) \Psi (RT_c)^2 / P_c \\ b_i &= \Omega RT_c / P_c \\ \alpha(\text{Tr}, \omega) &= (1 + m(\omega) \sqrt{1 - \text{Tr}})^2 \\ m(\omega) &= m_0 + m_1 \omega + m_2 \omega^2 \\ \text{Tr} &= T / T_c \end{aligned}$$

REQUIRED PARAMETERS:

T - Temperature [K]
P - Pressure [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:

Z - Compressibility factor
dZT - First temperature derivative
dZP - First pressure derivative
dZn - First composition derivatives
d2ZT - Second temperature derivative
d2ZP - Second pressure derivative
d2ZTP - Second temperature and pressure derivative

d2ZTn - Second temperature and composition derivatives
d2ZPn - Second pressure and composition derivatives
d2Zn - Second composition derivatives

DEPENDENCIES:

MixParams
SolvePolynomiumNewton

See also LoadParams

PureIdHSV	PureRealHSV	PureSolveEoS
PureIdVapHSV	PureRealVapHSV	PureSolvePhEoS
PureIdLiqHSV	PureRealLiqHSV	PureResHSV
MixIdHSV	MixRealHSV	PureResPhHSV
MixIdVapHSV	MixRealVapHSV	MixSolvePhEoS
MixIdLiqHSV	MixRealLiqHSV	MixResPhHSV
IdLiqVol	MixParams	PureEvalEoS
IdSatPres	PureParams	MixEvalEoS
IdGasHeatCap	SolvePolynomiumNewton	
IdGasHeatCapInt		

7. MixIdHSV (Matlab/Mex)

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

SYNOPSIS:

```
[Hv, Sv, Vv, Hl, Sl, Vl, ...  
    dHv, dSv, dVv, dHl, dSl, dVl, ...  
    d2Hv, d2Sv, d2Vv, d2Hl, d2Sl, d2Vl] = ...  
    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
```

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:

IdGasHeatCap
 IdGasHeatCapInt
 IdLiqVol
 IdSatPres

See also LoadParams

PureIdHSV	PureRealHSV	PureSolveEoS
PureIdVapHSV	PureRealVapHSV	PureSolvePhEoS
PureIdLiqHSV	PureRealLiqHSV	PureResHSV
MixIdHSV	MixRealHSV	PureResPhHSV
MixIdVapHSV	MixRealVapHSV	MixSolvePhEoS
MixIdLiqHSV	MixRealLiqHSV	MixResPhHSV
IdLiqVol	MixParams	PureEvalEoS
IdSatPres	PureParams	MixEvalEoS
IdGasHeatCap	SolvePolynomiumNewton	
IdGasHeatCapInt		

8. 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

PureIdHSV	PureRealHSV	PureSolveEoS
PureIdVapHSV	PureRealVapHSV	PureSolvePhEoS
PureIdLiqHSV	PureRealLiqHSV	PureResHSV
MixIdHSV	MixRealHSV	PureResPhHSV
MixIdVapHSV	MixRealVapHSV	MixSolvePhEoS
MixIdLiqHSV	MixRealLiqHSV	MixResPhHSV
IdLiqVol	MixParams	PureEvalEoS
IdSatPres	PureParams	MixEvalEoS
IdGasHeatCap	SolvePolynomiumNewton	
IdGasHeatCapInt		

9. MixIdLiqHSV (Matlab/Mex)

Compute enthalpy, entropy and volume of an ideal liquid mixture

SYNOPSIS:

```
[H1, S1, V1, ...  
    dH1, dS1, dV1, ...  
    d2H1, d2S1, d2V1] = ...  
    MixIdLiqHSV(T, P, n1, 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]  
n1         - mole numbers in liquid phase [kmol]  
params     - Vector with various parameters obtained by calling  
             LoadParams
```

RETURNS:

```
H1         - Enthalpy of liquid phase  
S1         - Entropy of liquid phase  
V1         - Volume of liquid phase  
dH1        - First order derivatives of liquid enthalpy  
dS1        - First order derivatives of liquid entropy  
dV1        - First order derivatives of liquid volume  
d2H1       - Second order derivatives of liquid enthalpy  
d2S1       - Second order derivatives of liquid entropy  
d2V1       - Second order derivatives of liquid volume
```

DEPENDENCIES:

MixIdVapHSV
IdLiqVol
IdSatPres

See also LoadParams

PureIdHSV
PureIdVapHSV
PureIdLiqHSV
MixIdHSV
MixIdVapHSV
MixIdLiqHSV
IdLiqVol
IdSatPres
IdGasHeatCap
IdGasHeatCapInt

PureRealHSV
PureRealVapHSV
PureRealLiqHSV
MixRealHSV
MixRealVapHSV
MixRealLiqHSV
MixParams
PureParams
SolvePolynomiumNewton

PureSolveEoS
PureSolvePhEoS
PureResHSV
PureResPhHSV
MixSolvePhEoS
MixResPhHSV
PureEvalEoS
MixEvalEoS

10. 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, ds1T, dv1T, dh1P, ds1P, dv1P, ...  
 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
 ds1P - 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

PureIdHSV	PureRealHSV	PureSolveEoS
PureIdVapHSV	PureRealVapHSV	PureSolvePhEoS
PureIdLiqHSV	PureRealLiqHSV	PureResHSV
MixIdHSV	MixRealHSV	PureResPhHSV
MixIdVapHSV	MixRealVapHSV	MixSolvePhEoS
MixIdLiqHSV	MixRealLiqHSV	MixResPhHSV
IdLiqVol	MixParams	PureEvalEoS
IdSatPres	PureParams	MixEvalEoS
IdGasHeatCap	SolvePolynomiumNewton	
IdGasHeatCapInt		

11. 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  
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

DEPENDENCIES:

PureIdVapHSV
PureResPhHSV

See also LoadParams

PureIdHSV	PureRealHSV	PureSolveEoS
PureIdVapHSV	PureRealVapHSV	PureSolvePhEoS
PureIdLiqHSV	PureRealLiqHSV	PureResHSV
MixIdHSV	MixRealHSV	PureResPhHSV
MixIdVapHSV	MixRealVapHSV	MixSolvePhEoS
MixIdLiqHSV	MixRealLiqHSV	MixResPhHSV
IdLiqVol	MixParams	PureEvalEoS
IdSatPres	PureParams	MixEvalEoS
IdGasHeatCap	SolvePolynomiumNewton	
IdGasHeatCapInt		

12. PureRealLiqHSV (Matlab/Mex)

Compute pure component liquid enthalpy, entropy and volume

SYNOPSIS:

```
[hl, sl, vl, ...  
    dh1T, ds1T, dv1T, dh1P, ds1P, dv1P, ...  
    d2h1T, d2s1T, d2v1T, d2h1P, d2s1P, d2v1P, ...  
    d2h1TP, d2s1TP, d2v1TP] = ...  
    PureRealLiqHSV(T, P, params, tol, itmax)
```

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]  
params     - Vector with various parameters obtained by calling  
             LoadParams
```

OPTIONAL PARAMETERS:

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

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  
ds1P      - Molar volume 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  
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
PureResPhHSV

See also LoadParams

PureIdHSV	PureRealHSV	PureSolveEoS
PureIdVapHSV	PureRealVapHSV	PureSolvePhEoS
PureIdLiqHSV	PureRealLiqHSV	PureResHSV
MixIdHSV	MixRealHSV	PureResPhHSV
MixIdVapHSV	MixRealVapHSV	MixSolvePhEoS
MixIdLiqHSV	MixRealLiqHSV	MixResPhHSV
IdLiqVol	MixParams	PureEvalEoS
IdSatPres	PureParams	MixEvalEoS
IdGasHeatCap	SolvePolynomiumNewton	
IdGasHeatCapInt		

13. 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, ...  
 dh1T, 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

$$\begin{aligned}h^R(T, P) &= RT (Z - 1) + 1/((\epsilon - \sigma) b) (T da/dT - a(T)) f(Z, B) \\s^R(T, P) &= R \ln(Z - 1) + 1/((\epsilon - \sigma) b) da/dT f(Z, B)\end{aligned}$$

where

$$f(Z, B) = \ln((Z + \epsilon B) / (Z + \sigma B))$$

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

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

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
ds1P - 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:

PureParams
PureSolveEoS

See also LoadParams

PureIdHSV PureRealHSV PureSolveEoS
PureIdVapHSV PureRealVapHSV PureSolvePhEoS

PureIdLiqHSV	PureRealLiqHSV	PureResHSV
MixIdHSV	MixRealHSV	PureResPhHSV
MixIdVapHSV	MixRealVapHSV	MixSolvePhEoS
MixIdLiqHSV	MixRealLiqHSV	MixResPhHSV
IdLiqVol	MixParams	PureEvalEoS
IdSatPres	PureParams	MixEvalEoS
IdGasHeatCap	SolvePolynomiumNewton	
IdGasHeatCapInt		

14. 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

$$\begin{aligned}h^R(T, P) &= RT \ln(Z - 1) + 1/((\epsilon - \sigma) b) (T da/dT - a(T)) f(Z, B) \\s^R(T, P) &= R \ln(Z - 1) + 1/((\epsilon - \sigma) b) da/dT f(Z, B)\end{aligned}$$

where

$$f(Z, B) = \ln((Z + \epsilon B) / (Z + \sigma B))$$

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

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

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
dsP - 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

PureIdHSV	PureRealHSV	PureSolveEoS
PureIdVapHSV	PureRealVapHSV	PureSolvePhEoS
PureIdLiqHSV	PureRealLiqHSV	PureResHSV
MixIdHSV	MixRealHSV	PureResPhHSV
MixIdVapHSV	MixRealVapHSV	MixSolvePhEoS
MixIdLiqHSV	MixRealLiqHSV	MixResPhHSV
IdLiqVol	MixParams	PureEvalEoS
IdSatPres	PureParams	MixEvalEoS
IdGasHeatCap	SolvePolynomiumNewton	
IdGasHeatCapInt		

15. PureFug (Matlab/Mex)

Compute logarithmic fugacity coefficients of real components

SYNOPSIS:

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

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

$$\ln \phi_i(T, P) = Z - 1 - \ln(Z - B) - 1/(\epsilon - \sigma) \ln \left(\frac{Z + \epsilon B}{Z + \sigma B} \right)$$

where

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

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:

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

PureIdHSV	PureRealHSV	PureSolveEoS
PureIdVapHSV	PureRealVapHSV	PureSolvePhEoS

PureIdLiqHSV	PureRealLiqHSV	PureResHSV
MixIdHSV	MixRealHSV	PureResPhHSV
MixIdVapHSV	MixRealVapHSV	MixSolvePhEoS
MixIdLiqHSV	MixRealLiqHSV	MixResPhHSV
IdLiqVol	MixParams	PureEvalEoS
IdSatPres	PureParams	MixEvalEoS
IdGasHeatCap	SolvePolynomiumNewton	
IdGasHeatCapInt		

16. PureSolveEoS (Matlab/Mex)

Solve cubic equation of state for vapor and liquid compressibility factors

SYNOPSIS:

```
[Zv, Zl, ...  
 dZvT, dZvP, dZlT, dZlP, ...  
 d2ZvT, d2ZvP, d2ZvTP, d2ZlT, d2ZlP, d2ZlTP] = ...  
 PureSolveEoS(T, P, params, varargin)
```

DESCRIPTION:

Solves a cubic equation of state

$$P = RT/(V - b) - a(T)/((V + \epsilon b)(V + \sigma b))$$

for the compressibility factor. Other functions are

$$\begin{aligned} a(T) &= \alpha(\text{Tr}, \omega) \text{Psi} (RT_c)^2/P_c \\ b &= \Omega RT_c/P_c \\ \alpha(\text{Tr}, \omega) &= (1 + m(\omega)*\sqrt{1 - \text{Tr}})^2 \\ m(\omega) &= m_0 + m_1 \omega + m_2 \omega^2 \\ \text{Tr} &= T/T_c \end{aligned}$$

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
d2ZvT - Second temperature derivatives of compressibility factor
d2ZvP - Second pressure derivatives of compressibility factor
d2ZvTP - Second temperature and pressure derivatives of compressibility factor
d2ZlT - Second temperature derivatives of compressibility factor
d2ZlP - Second pressure derivatives of compressibility factor
d2ZlTP - Second temperature and pressure derivatives of compressibility factor

DEPENDENCIES:

PureParams
SolvePolynomiumNewton

See also LoadParams

PureIdHSV	PureRealHSV	PureSolveEoS
PureIdVapHSV	PureRealVapHSV	PureSolvePhEoS
PureIdLiqHSV	PureRealLiqHSV	PureResHSV
MixIdHSV	MixRealHSV	PureResPhHSV
MixIdVapHSV	MixRealVapHSV	MixSolvePhEoS
MixIdLiqHSV	MixRealLiqHSV	MixResPhHSV
IdLiqVol	MixParams	PureEvalEoS
IdSatPres	PureParams	MixEvalEoS
IdGasHeatCap	SolvePolynomiumNewton	
IdGasHeatCapInt		

17. PureSolvePhEoS (Matlab/Mex)

Solve cubic equation of state for compressibility factor

SYNOPSIS:

```
[Z, ...  
    dZT, dZP, ...  
    d2ZT, d2ZP, d2ZTP] = ...  
    PureSolvePhEoS(T, P, phase, params, tol, itmax)
```

DESCRIPTION:

Solves a cubic equation of state

$$P = RT/(V - b) - a(T)/((V + \epsilon b)(V + \sigma b))$$

for the compressibility factor. Other functions are

$$\begin{aligned} a(T) &= \alpha(\text{Tr}, \omega) \text{Psi} (RT_c)^2/P_c \\ b &= \Omega RT_c/P_c \\ \alpha(\text{Tr}, \omega) &= (1 + m(\omega)\sqrt{1 - \text{Tr}})^2 \\ m(\omega) &= m_0 + m_1 \omega + m_2 \omega^2 \\ \text{Tr} &= T/T_c \end{aligned}$$

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:

Z - Compressibility factor
dZT - First temperature derivatives of compressibility factor
dZP - First pressure derivatives of compressibility factor
d2ZT - Second temperature derivatives of compressibility factor
d2ZP - Second pressure derivatives of compressibility factor
d2ZTP - Second temperature and pressure derivatives of compressibility factor

DEPENDENCIES:

PureParams
SolvePolynomiumNewton

See also LoadParams

PureIdHSV	PureRealHSV	PureSolveEoS
PureIdVapHSV	PureRealVapHSV	PureSolvePhEoS
PureIdLiqHSV	PureRealLiqHSV	PureResHSV
MixIdHSV	MixRealHSV	PureResPhHSV
MixIdVapHSV	MixRealVapHSV	MixSolvePhEoS
MixIdLiqHSV	MixRealLiqHSV	MixResPhHSV
IdLiqVol	MixParams	PureEvalEoS
IdSatPres	PureParams	MixEvalEoS
IdGasHeatCap	SolvePolynomiumNewton	
IdGasHeatCapInt		

18. 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

```
PureIdHSV          PureRealHSV          PureSolveEoS  
PureIdVapHSV      PureRealVapHSV        PureSolvePhEoS
```

PureIdLiqHSV	PureRealLiqHSV	PureResHSV
MixIdHSV	MixRealHSV	PureResPhHSV
MixIdVapHSV	MixRealVapHSV	MixSolvePhEoS
MixIdLiqHSV	MixRealLiqHSV	MixResPhHSV
IdLiqVol	MixParams	PureEvalEoS
IdSatPres	PureParams	MixEvalEoS
IdGasHeatCap	SolvePolynomiumNewton	
IdGasHeatCapInt		

19. 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:

```
h1         - Molar enthalpy of each component  
s1         - Molar entropy of each component  
v1         - 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

```
PureIdHSV          PureRealHSV          PureSolveEoS
```

PureIdVapHSV	PureRealVapHSV	PureSolvePhEoS
PureIdLiqHSV	PureRealLiqHSV	PureResHSV
MixIdHSV	MixRealHSV	PureResPhHSV
MixIdVapHSV	MixRealVapHSV	MixSolvePhEoS
MixIdLiqHSV	MixRealLiqHSV	MixResPhHSV
IdLiqVol	MixParams	PureEvalEoS
IdSatPres	PureParams	MixEvalEoS
IdGasHeatCap	SolvePolynomiumNewton	
IdGasHeatCapInt		

20. MixParams (Matlab/Mex)

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

SYNOPSIS:

```
[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

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

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

PureIdVapHSV	PureRealVapHSV	PureSolveEoS
PureIdLiqHSV	PureRealLiqHSV	PureResHSV
MixIdVapHSV	MixRealVapHSV	MixSolveEoS
MixIdLiqHSV	MixRealLiqHSV	MixResHSV
IdLiqVol	MixParams	PureEvalEoS
IdSatPres	PureParams	MixEvalEoS
IdGasHeatCap	SolvePolynomiumNewton	
IdGasHeatCapInt		

21. 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 $i = 1, \dots, N_C$

```
a_i(T)           = alpha(Tr, omega) Psi (RTc)^2/Pc
b_i              = 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
```

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

PureIdVapHSV	PureRealVapHSV	PureSolveEoS
PureIdLiqHSV	PureRealLiqHSV	PureResHSV
MixIdVapHSV	MixRealVapHSV	MixSolveEoS
MixIdLiqHSV	MixRealLiqHSV	MixResHSV
IdLiqVol	MixParams	PureEvalEoS
IdSatPres	PureParams	MixEvalEoS
IdGasHeatCap	SolvePolynomiumNewton	
IdGasHeatCapInt		

22. 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

PureIdVapHSV	PureRealVapHSV	PureSolveEoS
PureIdLiqHSV	PureRealLiqHSV	PureResHSV
MixIdVapHSV	MixRealVapHSV	MixSolveEoS
MixIdLiqHSV	MixRealLiqHSV	MixResHSV
IdLiqVol	MixParams	PureEvalEoS
IdSatPres	PureParams	MixEvalEoS
IdGasHeatCap	SolvePolynomiumNewton	
IdGasHeatCapInt		

23. 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

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

REQUIRED PARAMETERS:

d2 - Quadratic coefficient
d1 - Linear coefficient
d0 - Constant coefficient

OPTIONAL PARAMETERS:

tol - Tolerance for stopping criteria
itmax - Maximum number of iterations

RETURNS:

Z - A single real root of the cubic equation

See also LoadParams

PureIdVapHSV	PureRealVapHSV	PureSolveEoS
PureIdLiqHSV	PureRealLiqHSV	PureResHSV
MixIdVapHSV	MixRealVapHSV	MixSolveEoS
MixIdLiqHSV	MixRealLiqHSV	MixResHSV
IdLiqVol	MixParams	PureEvalEoS
IdSatPres	PureParams	MixEvalEoS
IdGasHeatCap	SolvePolynomiumNewton	
IdGasHeatCapInt		

24. IdGasHeatCap (Matlab/Mex)

Compute ideal gas heat capacity

SYNOPSIS:

```
[cp, dcp] = IdGasHeatCap(T, params)
```

DESCRIPTION:

Computes ideal gas heat capacity using the DIPPR correlation

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

and the first order derivative

REQUIRED PARAMETERS:

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

RETURNS:

cp - Ideal gas heat capacity
dcp - First temperature derivative

See also LoadParams

PureIdVapHSV	PureRealVapHSV	PureSolveEoS
PureIdLiqHSV	PureRealLiqHSV	PureResHSV
MixIdVapHSV	MixRealVapHSV	MixSolveEoS
MixIdLiqHSV	MixRealLiqHSV	MixResHSV
IdLiqVol	MixParams	PureEvalEoS
IdSatPres	PureParams	MixEvalEoS
IdGasHeatCap	SolvePolynomiumNewton	
IdGasHeatCapInt		

25. IdGasHeatCapInt (Matlab/Mex)

Compute integrals of ideal gas heat capacity

SYNOPSIS:

```
[intcp, intcpT] = IdGasHeatCapInt(T, params)
```

DESCRIPTION:

Computes the following integrals of the ideal gas heat capacity

$$\begin{aligned} & \int_{T_0}^T c_{P,k}^{ig}(\tau) \, d\tau \\ & \int_{T_0}^T c_{P,k}^{ig}(\tau) / \tau \, d\tau \end{aligned}$$

where ideal gas heat capacity is defined by the DIPPR correlation

$$\begin{aligned} c_{P,k}^{ig}(T) = & A_k + B_k((C_k/T)/\sinh(C_k/T))^2 \\ & - D_k((E_k/T)/\cosh(E_k/T))^2 \end{aligned}$$

REQUIRED PARAMETERS:

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

RETURNS:

intcp - Integral of cp from T0 to T
intcpT - Integral of cp/T from T0 to T

See also LoadParams

PureIdVapHSV	PureRealVapHSV	PureSolveEoS
PureIdLiqHSV	PureRealLiqHSV	PureResHSV
MixIdVapHSV	MixRealVapHSV	MixSolveEoS
MixIdLiqHSV	MixRealLiqHSV	MixResHSV
IdLiqVol	MixParams	PureEvalEoS
IdSatPres	PureParams	MixEvalEoS
IdGasHeatCap	SolvePolynomiumNewton	
IdGasHeatCapInt		

26. IdLiqVol (Matlab/Mex)

Compute liquid volume

SYNOPSIS:

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

DESCRIPTION:

Computes liquid volume based on the DIPPR correlation

$$v1_k(T) = B_k^{(1 + (1 - T/C_k)^{D_k})/A_k}$$

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, [m³]
dv1 - First temperature derivative of liquid volumes
d2v1 - Second temperature derivative of liquid volumes
d3v1 - Third temperature derivative of liquid volumes

See also LoadParams

PureIdVapHSV	PureRealVapHSV	PureSolveEoS
PureIdLiqHSV	PureRealLiqHSV	PureResHSV
MixIdVapHSV	MixRealVapHSV	MixSolveEoS
MixIdLiqHSV	MixRealLiqHSV	MixResHSV
IdLiqVol	MixParams	PureEvalEoS
IdSatPres	PureParams	MixEvalEoS
IdGasHeatCap	SolvePolynomiumNewton	
IdGasHeatCapInt		

27. IdSatPres (Matlab/Mex)

Compute saturation pressure

SYNOPSIS:

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

DESCRIPTION:

Computes saturation pressure using the DIPPR correlation

$$\ln P_{\text{sat}_k} = A_k + B_k/T + C_k \ln(T) + D_k T^{E_k}$$

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

PureIdVapHSV	PureRealVapHSV	PureSolveEoS
PureIdLiqHSV	PureRealLiqHSV	PureResHSV
MixIdVapHSV	MixRealVapHSV	MixSolveEoS
MixIdLiqHSV	MixRealLiqHSV	MixResHSV
IdLiqVol	MixParams	PureEvalEoS
IdSatPres	PureParams	MixEvalEoS
IdGasHeatCap	SolvePolynomiumNewton	
IdGasHeatCapInt		

28. MixRealHSV (C)

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

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: $6*(NC + 2) + 6*(NC + 2) + 54*NC + 6*NC$ memory
nargout > 3: $6*(NC + 2) + 39*NC$ memory
else : $15*NC$ memory

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)

dH1 - 1st derivatives (must be size 2 + NC)
dS1 - 1st derivatives (must be size 2 + NC)
dV1 - 1st derivatives (must be size 2 + NC)
d2Hv - 2nd derivatives (must be size (2 + NC))
d2Sv - 2nd derivatives (must be size (2 + NC))
d2Vv - 2nd derivatives (must be size (2 + NC))
d2H1 - 2nd derivatives (must be size (2 + NC))
d2S1 - 2nd derivatives (must be size (2 + NC))
d2V1 - 2nd derivatives (must be size (2 + NC))

DEPENDENCIES:

PureIdVap
MixResPhHSV

29. MixRealVapHSV (C)

Compute enthalpy, entropy and volume of a real vapor mixture

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

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))
d2Sv - 2nd derivatives (must be size (2 + NC))
d2Vv - 2nd derivatives (must be size (2 + NC))

DEPENDENCIES:
 MixIdVapHSV
 MixResPhHSV

30. MixRealLiqHSV (C)

Compute enthalpy, entropy and volume of a real liquid mixture

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

RETURNS:

H1 - Liquid enthalpy (scalar)
S1 - Liquid entropy (scalar)
V1 - Liquid volume (scalar)
dH1 - 1st derivatives (must be size 2 + NC)
dS1 - 1st derivatives (must be size 2 + NC)
dV1 - 1st derivatives (must be size 2 + NC)
d2H1 - 2nd derivatives (must be size (2 + NC))
d2S1 - 2nd derivatives (must be size (2 + NC))
d2V1 - 2nd derivatives (must be size (2 + NC))

DEPENDENCIES:
 MixIdVapHSV
 MixResPhHSV

31. MixResPhHSV (C)

Compute volume and residual enthalpy and entropy of phase

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

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

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

$$\begin{aligned}a_m(T, n) &= \sum_{i=1}^{N_C} \sum_{j=1}^{N_C} x_i x_j a_{ij}(T) \\b_m(n) &= \sum_{i=1}^{N_C} x_i b_{ij} \\a_{ij}(T) &= (1 - k_{ij}) \sqrt{a_i(T) a_j(T)} \\x_i &= n_i / \sum_{j=1}^{N_C} n_j \\a_i(T) &= \alpha(\text{Tr}, \omega) \Psi_i (RT_c)^2 / P_c \\b_i &= \Omega_i RT_c / P_c \\\alpha(\text{Tr}, \omega) &= (1 + m(\omega) \sqrt{1 - \text{Tr}})^2 \\m(\omega) &= m_0 + m_1 \omega + m_2 \omega^2 \\\text{Tr} &= T / T_c\end{aligned}$$

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
 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))
 d2s - 2nd derivatives (must be size (2 + NC))
 d2v - 2nd derivatives (must be size (2 + NC))

DEPENDENCIES:

MixParams
 MixSolvePhEoS

32. MixSolvePhEoS (C)

Solve cubic equation of state for compressibility factor

DESCRIPTION:

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

$$P = RT/(V - bm) - am(T)/((V + \epsilon bm)(V + \sigma bm))$$

for the compressibility factor. Other functions are

$$\begin{aligned} am(T, n) &= \sum_{i=1}^{N_C} \sum_{j=1}^{N_C} x_i x_j a_{ij}(T) \\ bm(n) &= \sum_{i=1}^{N_C} x_i b_{ij} \\ a_{ij}(T) &= (1 - k_{ij}) \sqrt{a_i(T) a_j(T)} \\ x_i &= n_i / \sum_{j=1}^{N_C} n_j \\ a_i(T) &= \alpha(\text{Tr}, \omega) \Psi (RT_c)^2 / P_c \\ b_i &= \Omega RT_c / P_c \\ \alpha(\text{Tr}, \omega) &= (1 + m(\omega) \sqrt{1 - \text{Tr}})^2 \\ m(\omega) &= m_0 + m_1 \omega + m_2 \omega^2 \\ \text{Tr} &= T / T_c \end{aligned}$$

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*NC + 2*NC*NC memory
nargout > 1: 12*NC memory
else : 4*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 (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

33. MixIdHSV (C)

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

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]
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))

dS1 - First order derivatives (must have size (2 + NC))
dV1 - First order derivatives (must have size (2 + NC))
d2Hv - Second order derivatives (must have size (2 + NC))
d2Sv - Second order derivatives (must have size (2 + NC))
d2Vv - Second order derivatives (must have size (2 + NC))
d2H1 - Second order derivatives (must have size (2 + NC))
d2S1 - Second order derivatives (must have size (2 + NC))
d2V1 - Second order derivatives (must have size (2 + NC))

DEPENDENCIES:

IdLiqVol
IdSatPres
IdGasHeatCap
IdGasHeatCapInt

34. MixIdVapHSV (C)

Compute enthalpy, entropy and volume of an ideal vapor mixture

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))
d2Sv - Second order derivatives (must have size (2 + NC))
d2Vv - Second order derivatives (must have size (2 + NC))

DEPENDENCIES:

IdGasHeatCap

IdGasHeatCapInt

35. MixIdLiqHSV (C)

Compute enthalpy, entropy and volume of an ideal liquid mixture

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:

H1 - Enthalpy of liquid phase (scalar)
S1 - Entropy of liquid phase (scalar)
V1 - Volume of liquid phase (scalar)
dH1 - First order derivatives (must have size (2 + NC))
dS1 - First order derivatives (must have size (2 + NC))
dV1 - First order derivatives (must have size (2 + NC))
d2H1 - Second order derivatives (must have size (2 + NC))
d2S1 - Second order derivatives (must have size (2 + NC))
d2V1 - Second order derivatives (must have size (2 + NC))

DEPENDENCIES:

MixIdVapHSV
IdLiqVol
IdSatPres
IdGasHeatCap
IdGasHeatCapInt

36. PureRealHSV (C)

Compute pure component vapor and liquid enthalpy, entropy and volume

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)
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)
ds1P - 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)
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:

PureIdVapHSV
PureResHSV

37. PureRealVapHSV (C)

Compute pure component vapor enthalpy, entropy and volume

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

38. PureRealLiqHSV (C)

Compute pure component liquid enthalpy, entropy and volume

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:

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)
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)
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)
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

39. PureResHSV (C)

Compute volume and residual enthalpy and entropy of pure components

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

$$\begin{aligned}h^R(T, P) &= RT (Z - 1) + 1/((\epsilon - \sigma) b) (T da/dT - a(T)) f(Z, B) \\s^R(T, P) &= R \ln(Z - 1) + 1/((\epsilon - \sigma) b) da/dT f(Z, B)\end{aligned}$$

where

$$f(Z, B) = \ln((Z + \epsilon B) / (Z + \sigma B))$$

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

$$\begin{aligned}P &= RT/(V - b) - a(T)/(V^2 + 2Vb - b^2) \\a(T) &= \alpha(\text{Tr}, \omega) \text{Psi} (RT_c)^2/P_c \\b &= \Omega RT_c/P_c \\ \alpha(\text{Tr}, \omega) &= (1 + m(\omega) \sqrt{1 - \text{Tr}})^2 \\ m(\omega) &= m_0 + m_1 \omega + m_2 \omega^2 \\ \text{Tr} &= T/T_c \\ \text{Psi} &= 0.45724 \\ \Omega &= 0.07779\end{aligned}$$

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

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)
dvlT	- 1st temperature derivative (must be size NC)
dhlP	- 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)
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)
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

40. PureResPhHSV (C)

Compute volume and residual enthalpy and entropy of pure components

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

$$\begin{aligned}h^R(T, P) &= RT \ln(Z - 1) + 1/((\epsilon - \sigma) b) (T da/dT - a(T)) f(Z, B) \\s^R(T, P) &= R \ln(Z - 1) + 1/((\epsilon - \sigma) b) da/dT f(Z, B)\end{aligned}$$

where

$$f(Z, B) = \ln((Z + \epsilon B) / (Z + \sigma B))$$

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

$$\begin{aligned}P &= RT/(V - b) - a(T)/(V^2 + 2Vb - b^2) \\a(T) &= \alpha(\text{Tr}, \omega) \text{Psi} (RT_c)^2/P_c \\b &= \Omega RT_c/P_c \\ \alpha(\text{Tr}, \omega) &= (1 + m(\omega)\sqrt{1 - \text{Tr}})^2 \\ m(\omega) &= m_0 + m_1 \omega + m_2 \omega^2 \\ \text{Tr} &= T/T_c \\ \text{Psi} &= 0.45724 \\ \Omega &= 0.07779\end{aligned}$$

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)
dsP - 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

41. PureSolveEoS (C)

Solve cubic equation of state for compressibility factor

DESCRIPTION:

Solves a cubic equation of state

$$P = RT/(V - b) - a(T)/((V + \epsilon b)(V + \sigma b))$$

for the compressibility factor. Other functions are

$$\begin{aligned} a(T) &= \alpha(\text{Tr}, \omega) \text{Psi} (RT_c)^2/P_c \\ b &= \Omega RT_c/P_c \\ \alpha(\text{Tr}, \omega) &= (1 + m(\omega)*\sqrt{1 - \text{Tr}})^2 \\ m(\omega) &= m_0 + m_1 \omega + m_2 \omega^2 \\ \text{Tr} &= T/T_c \end{aligned}$$

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:

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)
d2ZlT - 2nd temperature derivatives (must be size NC)
d2ZlP - 2nd pressure derivatives (must be size NC)
d2ZlTP - 2nd temperature and pressure derivatives (must be size NC)

DEPENDENCIES:

PureParams

SolvePolynomiumNewton

42. PureSolvePhEoS (C)

Solve cubic equation of state for compressibility factor

DESCRIPTION:

Solves a cubic equation of state

$$P = RT/(V - b) - a(T)/((V + \epsilon b)(V + \sigma b))$$

for the compressibility factor. Other functions are

$$\begin{aligned} a(T) &= \alpha(\text{Tr}, \omega) \text{Psi} (RT_c)^2/P_c \\ b &= \Omega RT_c/P_c \\ \alpha(\text{Tr}, \omega) &= (1 + m(\omega)*\sqrt{1 - \text{Tr}})^2 \\ m(\omega) &= m_0 + m_1 \omega + m_2 \omega^2 \\ \text{Tr} &= T/T_c \end{aligned}$$

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

43. PureIdHSV (C)

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

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:

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)
dh1T - 1st temperature derivative (must be size NC)
dslT - 1st temperature derivative (must be size NC)
dvlT - 1st temperature derivative (must be size NC)
dh1P - 1st pressure derivative (must be size NC)
dslP - 1st pressure derivative (must be size NC)
d2h1T - 2nd temperature derivative (must be size NC)
d2slT - 2nd temperature derivative (must be size NC)
d2vlT - 2nd temperature derivative (must be size NC)
d2h1TP - 2nd pressure and temperature derivative (must be size NC)
d2slTP - 2nd pressure and temperature derivative (must be size NC)

DEPENDENCIES:

IdLiqVol
IdSatPres
IdGasHeatCap
IdGasHeatCapInt

44. PureIdVapHSV (C)

Compute pure component molar vapor enthalpy, entropy and volume

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 pressure 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

45. PureIdLiqHSV (C)

Compute pure component molar liquid enthalpy, entropy and volume

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)
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)
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. MixParams (C)

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

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

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

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)
d2amn - 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*NC)

DEPENDENCIES:

PureParams

47. PureParams (C)

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

DESCRIPTION:

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

$$\begin{aligned} a_i(T) &= \alpha(\text{Tr}, \omega) \Psi_i (RT_c)^2/P_c \\ b_i &= \Omega_i RT_c/P_c \\ \alpha(\text{Tr}, \omega) &= (1 + m(\omega) \sqrt{1 - \text{Tr}})^2 \\ m(\omega) &= m_0 + m_1 \omega + m_2 \omega^2 \\ \text{Tr} &= T/T_c \end{aligned}$$

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)

48. SolvePolynomiumNewton (C)

Solve cubic equation iteratively

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:

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)

49. IdGasHeatCap (C)

Compute ideal gas heat capacity

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
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)

50. IdGasHeatCapInt (C)

Compute integrals of ideal gas heat capacity

DESCRIPTION:

Computes the following integrals of the ideal gas heat capacity

$$\begin{aligned} & \int_{T_0}^T c_{P,k}^{ig}(\tau) \, d\tau \\ & \int_{T_0}^T c_{P,k}^{ig}(\tau) / \tau \, d\tau \end{aligned}$$

where ideal gas heat capacity is defined by the DIPPR correlation

$$\begin{aligned} c_{P,k}^{ig}(T) = & A_k + B_k((C_k/T)/\sinh(C_k/T))^2 \\ & - D_k((E_k/T)/\cosh(E_k/T))^2 \end{aligned}$$

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)

51. IdLiqVol (C)

Compute liquid volume

DESCRIPTION:

Computes liquid volume based on the DIPPR correlation

$$v_{l,k}(T) = B_k^{1 + (1 - T/C_k)^{D_k}}/A_k$$

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:

v1 - Liquid volumes for each component, [m³], (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)

52. IdSatPres (C)

Compute saturation pressure

DESCRIPTION:

Computes saturation pressure using the DIPPR correlation

$$\ln \text{Psat}_k = A_k + B_k/T + C_k \ln(T) + D_k T^E_k$$

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)