INDEX

Class cSubstance

Objects of this class represent components in a mixture. The class stores the properties of the components required for application of the models.

Properties

AntA, AntB, AntC: parameters of the Antoine equation for pure component vapor pressure, ln P(mmHg) = AntA - AntB/(T (K) + AntC), scalars.

EoSParam: equation of state-specific parameters. Any number of EoSParam elements can be defined (EoSParam(1), EoSParam(2), etc.). Each EoSParam can be either a scalar or a matrix.

Hf: melting enthalpy at the triple point, scalar (J/mol).

MW: molecular weight, scalar (g/mol).

Name: component name or label, character string.

Pc: critical pressure, scalar (Pa).

Tc: critical temperature, scalar (K).

Tf: melting temperature at the triple point, scalar (K).

w: acentric factor, scalar (-).

Methods

This class has no associated methods.

List of files

cSubstance.m: constructor of the class.

display.m: represents cSubstance objects in the Matlab command window.

subsasgn.m and subsref.m: mutator and accessor of the class (implementation of dot/parenthesis syntax etc).

Example programs

SubstanceExample.m: Shows several ways of defining cSubstance objects as well as examples of retrieval and modification of the properties of cSubstance objects.

Class cMixture

Objects of this class represent a mixture of one/several components. The class stores cSubstance objects representing each component in the mixture, the molar fraction composition of the mixture, and interaction coefficients between components.

Properties

comp: components of the mixture, vector of cSubstance objects.

k: Interaction coefficient between components in a mixture.

molFrac: molar fraction composition of the mixture, vector of scalars (equivalent denomination: x).

massFrac: mass fraction composition of the mixture, vector of scalars. Read-only property, calculated from the molar fraction composition and the molar weight of components in the mixture.

MW: molecular weight of the mixture (g/mol). Read-only property, calculated from the molar fraction composition and the molar weight of components in the mixture.

numC: number of components in the mixture.

x: molar fraction composition of the mixture, vector of scalars (equivalent denomination: molFrac).

Methods

This class has no associated methods.

List of files

CheckMolFrac.m: ensures that the molar fraction vector is consistent (has the same number of elements as the number of components defined in the mixture, and the molar fraction of all components adds 1) and displays a warning if it is not. Private method, cannot be used directly.

cMixture.m: constructor of the class.

display.m: represents cMixture objects in the Matlab command window.

GetInterCoef.m: Implements dot/parenthesis syntax for reading interaction coefficients. Private method, cannot be used directly. Use the associated property "k" instead.

massFrac.m: Calculates the read-only property "massFrac" which returns mass fraction composition of the mixture calculated from the molar fraction composition and the molecular weight of components. Private method, cannot be used directly. Use the associated property "massFrac" instead.

MW.m: Calculates the read-only property "MW" which returns the molecular weight of the mixture calculated from the molar fraction composition and the molecular weight of components. Private method, cannot be used directly. Use the associated property "MW" instead.

SetInterCoef.m: Implements dot/parenthesis syntax for setting interaction coefficients. Private method, cannot be used directly. Use the associated property "k" instead.

subsasgn.m and subsref.m: mutator and accessor of the class (implementation of dot/parenthesis syntax etc).

Example programs

MixtureExample.m: Shows several ways of defining cMixture objects as well as examples of retrieval and modification of the properties of cMixture objects.

Class cIdealEoS

This class implements the calculation of phase equilibrium and thermodynamic properties of mixtures considering ideal behavior (ideal gas law + Raoult's law). This class provides the methods for calculating these properties, which are inherited by all EoS classes derived from it.

Properties

ID: identification tag of the equation of state.

Methods

See the following pages for a detailed description of each method.

BoilingTemp:Calculates the boiling point of a pure component at a certain pressure.

BubblePoint: Calculates the bubble point of a mixture (either the pressure or the temperature of formation of a gas bubble and its composition for a certain temperature or pressure and liquid composition).

compr: Calculates the compressibility coefficient of a fluid mixture.

Density: Calculates the density of a fluid mixture.

DewPoint: Calculates the dew point of a mixture (either the pressure or the temperature of a liquid droplet and its composition for a certain temperature or pressure and gas composition).

Enthalpy: Calculates the residual contribution to the enthalpy of a fluid mixture.

Flash: Performs an isothermal flash calculation.

fug: Calculates the fugacity coefficient of components in a mixture.

fugF: Calculates the fugacity coefficient of components in a fluid mixture.

fugS: Calculates the fugacity coefficient of components in a pure solid phase.

MultiFlash: Performs an isothermal flash calculation with multiple phases using a Gibbs minimization algorithm.

PxyDiagram: Calculates a pressure-composition phase equilibrium diagram of a binary mixture.

Solubility: Calculates the solubility of a solute in a fluid phase.

TxyDiagram: Calculates a pressure-composition phase equilibrium diagram of a binary mixture.

VaporPressure: Calculates the vapor pressure of a pure component at a certain temperature.

List of files

BoilingTemp.m: implements the calculation of boiling points of pure components.

BubblePoint.m: implements the calculation of bubble points of mixtures.

cIdealEoS.m: constructor of the class.

Density.m: implements the calculation of the density of mixtures.

derZ_derT.m: auxiliary function for the calculation of the residual contribution to the enthalpy.

DewPoint.m: implements the calculation of dew points of mixtures.

display.m: represents cIdealEoS objects in the Matlab command window.

Enthalpy.m: implements the calculation of the residual contribution to the enthalpy.

Flash.m: implements the calculation of an isothermal flash.

fug.m: implements the calculation of the fugacity of a mixture.

fugF.m: implements the calculation of the fugacity of a fluid mixture.

fugS.m: implements the calculation of the fugacity of a solid phase.

MultiFlash.m: implements the calculation of an isothermal flash with multiple phases.

obj_BTemp.m: objective function of the algorithm of calculation of pure component boiling temperatures.

obj_BubblePoint.m: objective function of the algorithm of calculation of bubble points.

obj_DewPoint.m: objective function of the algorithm of calculation of dew points.

obj_FlashMulti.m: objective function of the algorithm of calculation of isothermal flash with multiple phases.

obj_Pvap.m: objective function of the algorithm of calculation of pure component vapor pressures.

PxyDiagram.m: implements the calculation of a pressure-composition phase equilibrium diagram of a binary mixture.

RachfordRice.m: objective function of the algorithm of calculation of isothermal flash.

Solubility.m: implements the calculation of the solublity of a solid in a fluid mixture.

subsasgn.m and subsref.m: mutator and accessor of the class (implementation of dot/parenthesis syntax etc).

TxyDiagram.m: implements the calculation of a temperature-composition phase equilibrium diagram of a binary mixture.

VaporPressure.m: implements the calculation of the vapor pressure of pure components.

Example programs

See the following pages for specific examples for each method.

Method BoilingTemp

Function prototype

[T, flag, val, time, EoS] = BoilingTemp(EoS,comp,P,[guess_T],[options])

Purpose

Calculates the boiling temperature of a pure component at a certain pressure.

Parameters and results

Parameters:

EoS: Equation of State used for calculations (cEOS object)

comp: Component (cSubstance object)

P: Pressure (Pa)

Optional parameters:

guess_T: initial guess of boiling point (K). If it is not provided, the initial guess is obtained using the Antoine equation using the 'Ant' parameters of the component

options: parameters of the fzero numerical resolution method (structure generated with "optimset")

Results:

T: Boiling temperature (K)

flag: flag returned by the fzero numerical resolution method:

- 1 FZERO found a zero X.
- 2 Calculations converged to single phase region
- -1 Algorithm terminated by output function.
- -3 NaN or Inf function value encountered during search for an interval containing a sign change.
- -4 Complex function value encountered during search for an interval containing a sign change.
- -5 FZERO may have converged to a singular point.

val: final value of the objective function obj_Pvap

time: time required for calculations

EoS: returns EoS used for calculations

Description

Calculates the boiling temperature of a pure component solving the condition of equality of fugacities between liquid and gas phase:

$$f^L = f^V \rightarrow \varphi^L = \varphi^V$$

This condition is implemented in the file obj_BTemp.m and is solved numerically in the file BoilingTemp.m using the Matlab fzero function.

Example programs

BoilingTemperatureExample.m shows several examples of application of the method.

Method BubblePoint

Function prototype

[T,P,y,K,flag,val,time,EoS] = BubblePoint(EoS,mix,T,P,guess_y,type,[options])

Purpose

Calculates the pressure or temperature of formation of the first bubble of gas at a certain temperature or pressure, as well as the composition of said gas.

Parameters and results

Parameters:

EoS: Equation of State used for calculations (cEOS object)

mix: mixture (cMixture object). Parameter mix.x or mox.molFrac must contain composition of the liquid phase

T: Temperature (K) (initial guess if type = 'T')

P: Pressure (Pa) (initial guess if type = 'P')

guess_y: Initial guess for composition of gas phase (molar fraction vector)

type: type of calculation

if type = 'T', then the bubble T is calculated for a given P

if type = 'P', then the bubble P is calculated for a given T

Optional parameters:

options: parameters of the fzero numerical resolution method (structure generated with "optimset")

Results:

T: Bubble point temperature (K)

P: Bubble point pressure (Pa)

y: Composition of the gas phase (molar fraction vector)

K: Distribution coefficients, K = y/x

flag: flag returned by the fzero numerical resolution method:

1 FZERO found a zero X.

2 Calculations converged to single phase region

-1 Algorithm terminated by output function.

-3 NaN or Inf function value encountered during search for an interval containing a sign change.

-4 Complex function value encountered during search for an interval containing a sign change.

-5 FZERO may have converged to a singular point.

val: final value of the objective function obj_BubblePointP

time: time required for calculations

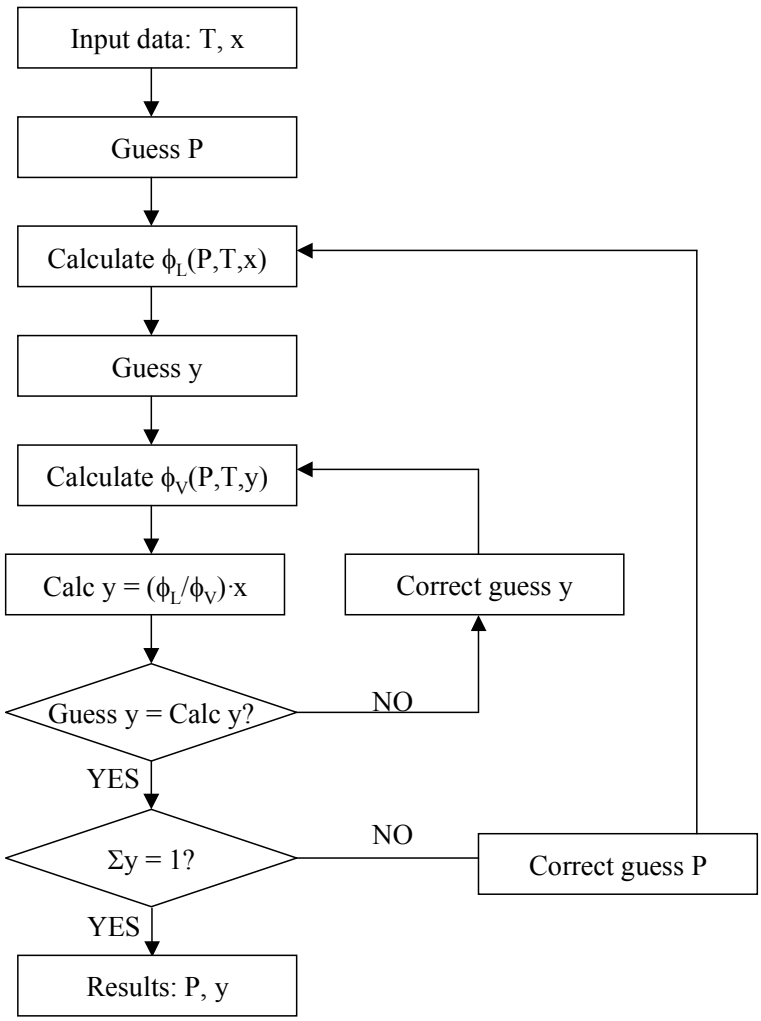
EoS: returns EoS used for calculations

Description

Calculates the bubble point of a mixture solving the condition of equality of fugacities of all components between the liquid and the gas phase:

$$f_i^L = f_i^v \rightarrow \varphi_i^L x_i = \varphi_i^v y_i$$

This condition is solved using the algorithm presented in the following figure:



Bubble point algorithm for specified temperature and liquid composition. The algorithm for a specified pressure and liquid composition is equivalent

The internal iteration loop (for convergence of gas composition) is implemented in the function obj_BubblePoint.m. The external iteration loop (convergence of pressure or temperature) is implemented in the function BubblePoint.m solving the condition of Σy_i-1 = 0 with the Matlab fzero function.

Example programs

BubblePointExample.m shows several examples of application of the method.

Method compr

Function prototype

[Z EoS] = compr(EoS,T,P,mix,phase,[varargin])

Purpose

Calculates the compressibility coefficient of a fluid mixture. Together with fugF, it is overloaded to define a new equation of state.

Parameters and results

Parameters:

EoS: Equation of state used for calculations

T: Temperature(K).

P: Pressure (Pa).

mix: cMixture object.

phase: set phase = 'liq' to get the coefficient of the liquid phase, phase = 'gas' to get the coefficient of the gas phase.

Optional Parameters:

May have additional optional parameters in specific equations of state (e.g. convergence settings).

Results:

Z: compresibility coefficient.

EoS: returns EoS used for calculations.

Description

Calculates the compressibility coefficient of a fluid mixture, according to the equation of state implemented.

Example programs

CompressibilityExample.m shows several examples of application of the method.

Method Density

Function prototype

[denMass,denMol,Z,EoS] = Density(EoS,mix,T,P,phase)

Purpose

Calculates the density of a fluid mixture

Parameters and results

Parameters:

EoS: Equation of State used for calculations

mix: mixture (cMixture object)

T: temperature (K)

P: pressure (Pa)

phase: set phase = 'liq' to calculate the density of a liquid phase or

phase = 'gas' to calculate the density of a gas phase

Results:

denMass: density (kg/m3)

denMol: molar density (mol/m3)

Z: compresibility coefficient

EoS: returns EoS used for calculations

Description

Calculates the density of a fluid mixture from its compressibility coefficient and mean molecular weight according to the equation:

$$\rho(kg/m^3) = \frac{P \cdot MW}{1000 \cdot Z \cdot R \cdot T}$$

Example programs

DensityExample.m shows several examples of application of the method.

Method DewPoint

Function prototype

[T,P,x,K,flag,val,time,EoS] = DewPoint(EoS,mix,T,P,guess_x,type,[options])

Purpose

Calculates the pressure or temperature of formation of the first droplet of liquid at a certain temperature or pressure, as well as the composition of said liquid.

Parameters and results

Parameters:

EoS: Equation of State used for calculations (cEOS object)

mix: mixture (cMixture object). Parameter mix.x or mox.molFrac must contain composition of the gas phase

T: Temperature (K) (initial guess if type = 'T')

P: Pressure (Pa) (initial guess if type = 'P')

guess_x: Initial guess for composition of liquid phase (molar fraction vector)

type: type of calculation

if type = 'T', then the dew T is calculated for a given P

if type = 'P', then the dew P is calculated for a given T

Optional parameters:

options: parameters of the fzero numerical resolution method (structure generated with "optimset")

Results:

P_or_T: if type = 'T', dew point temperature (K)

if type = 'P', dew point pressure (Pa)

x: Composition of the liquid phase (molar fraction vector)

K: Distribution coefficients, K = y/x

flag: flag returned by the fzero numerical resolution method:

1 FZERO found a zero X.

2 Calculations converged to single phase region

-1 Algorithm terminated by output function.

-3 NaN or Inf function value encountered during search for an interval containing a sign change.

-4 Complex function value encountered during search for an interval containing a sign change.

-5 FZERO may have converged to a singular point.

val: final value of the objective function obj_BubblePointP

time: time required for calculations

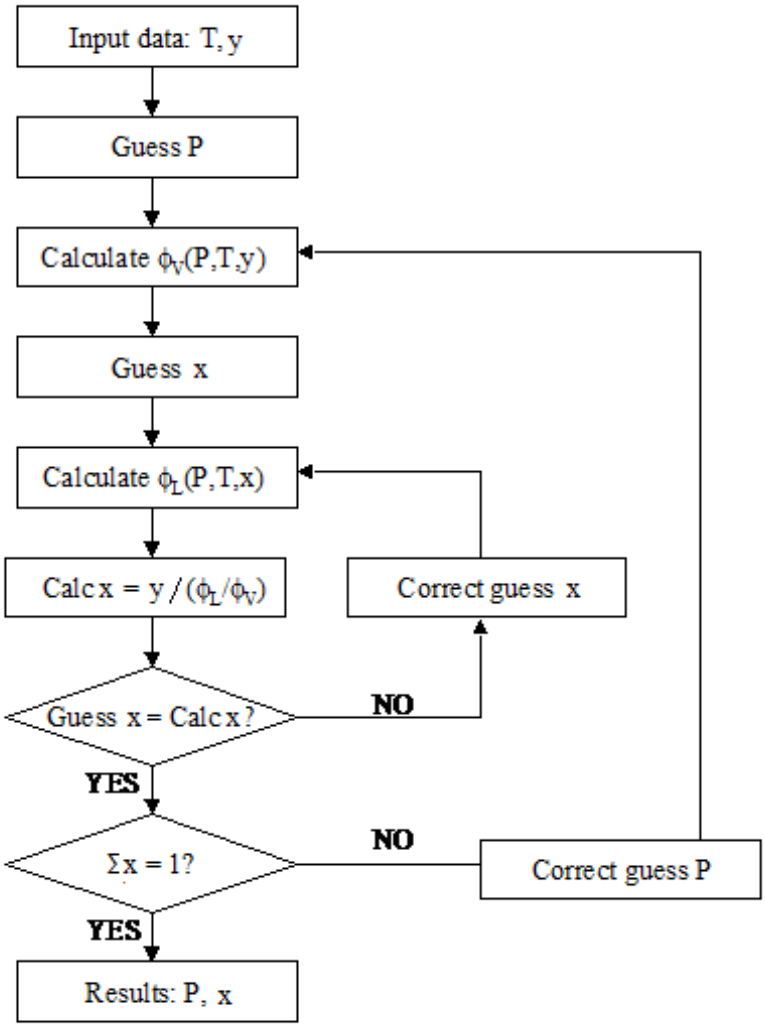
EoS: returns EoS used for calculations

Description

Calculates the dew point of a mixture solving the condition of equality of fugacities of all components between the liquid and the gas phase:

$$f_i^L = f_i^v \rightarrow \varphi_i^L x_i = \varphi_i^v y_i$$

This condition is solved using the algorithm presented in the following figure:



Dew point algorithm for specified temperature and gas composition. The algorithm for a specified pressure and gas composition is equivalent

The internal iteration loop (for convergence of liquid composition) is implemented in the function obj_DewPoint.m. The external iteration loop (convergence of pressure or temperature) is implemented in the function DewPoint.m solving the condition of Σx_i-1 = 0 with the Matlab fzero function.

Example programs

DewPointExample.m shows several examples of application of the method.

Method Enthalpy

Function prototype

[Hres,EoS] = Enthalpy(EoS,mix,T,P,phase)

Purpose

Calculates the residual contribution to the enthalpy of a mixture

Parameters and results

Parameters:

EoS: Equation of State used for calculations

mix: mixture (cMixture object)

T: temperature (K)

P: pressure (Pa)

phase: set phase = 'liq' to calculate the density of a liquid phase or

phase = 'gas' to calculate the density of a gas phase

Results:

Hres: residual enthalpy (J/mol)

EoS: returns EoS used for calculations

Description

Calculates the residual contribution to the enthalpy of a mixture. The total variation of enthalpy would be calculated taking into account the ideal gas contribution as follows:

$$\Delta H = \Delta H^{ideal-gas} - \Delta H^{residual}$$

The residual enthalpy is calculated numerically integrating the following expression:

$$\Delta H^{residual} = RT^2 \int_{P0}^{P} \left(\frac{\partial Z}{\partial T}\right)_P \frac{dP}{P}$$

The integrand of this equation is implemented in the function derZ_derT.m, and the integral is solved in Enthalpy.m using the Matlab quadl function.

For efficiency in the calculations, it is advisable to overload this function with an equation of state-specific analytical expression.

Example programs

EnthalpyExample.m shows several examples of application of the method.

Method Flash

Function prototype

[beta,x,y,K,val,time,EoS] = Flash(EoS,P,T,mix,guess_x,guess_y,guess_beta)

Purpose

Performs an isothermal flash calculation.

Parameters and results

Parameters:

EoS: EoS used for calculations

P: Pressure (Pa)

T: Temperature (K)

mix: mixture (cMixture object)

mix.x must contain the global composition of the mixture

guess_x: initial guess for liquid composition (molar fraction vector)

guess_y: initial guess for gas composition (molar fraction vector)

guess_beta: initial guess for vaporized fraction

Results:

beta: vaporized fraction V/(V+L) (mol/mol)

x: liquid composition (molar fraction vector)

y: gas composition (molar fraction vector)

K: distribution coefficients K = y/x

val: final value of the objective function

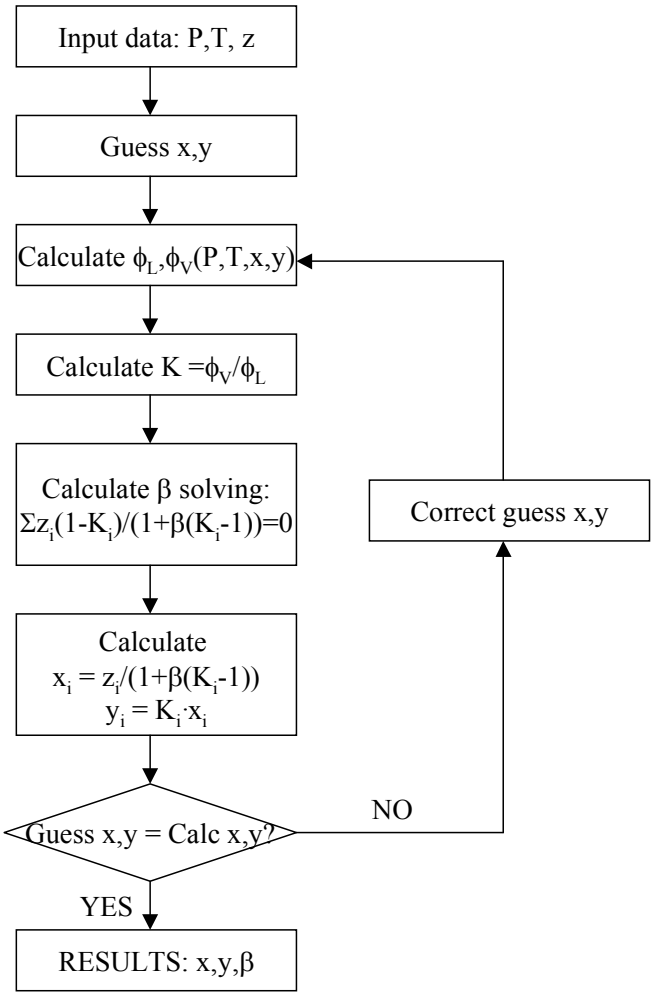
time: time required for calculations

EoS: returns EoS used for calculations

Description

Calculates an isothermal flash of a multicomponent mixture. The algorithm used for calculations is shown in the figure below.

The calculation of the vaporized fraction β by resolution of the mass balance is implemented in the function RachfordRice.m. The iterative loop for convergence of liquid and gas compositions is implemented in the function Flash.m



Flash algorithm

Example programs

FlashExample.m shows several examples of application of the method.

Method fug

Function prototype

[f, EoS] = fug(EoS,T,P,mix,phase,[varargin])

Purpose

Calculates the fugacity coefficients of components in a mixture

Parameters and results

Parameters:

EoS: Equation of state used for calculations

T: Temperature(K).

P: Pressure (Pa).

mix: cMixture object.

phase: set phase = 'liq' to calculate the fugacity of a liquid phase

phase = 'gas' to calculate the fugacity of a gas phase

phase = 'sol' to calculate the fugacity of a solid phase

Optional Parameters:

May have additional optional parameters in specific equations of state (e.g. convergence settings).

Results:

f: fugacity coefficients

EoS: returns EoS used for calculations.

Description

Calculates the fugacity coefficients of components in a mixture. For liquid or gas phases, this function invokes the function fugF.m to obtain the coefficients, while for solid phases it invokes the function fugS.m.

Method fugF

Function prototype

[f, EoS] = fugF(EoS,T,P,mix,phase,[varargin])

Purpose

Calculates the fugacity coefficients of components in a fluid mixture. Together with compr, it is overloaded to define a new equation of state.

Parameters and results

Parameters:

EoS: Equation of state used for calculations

T: Temperature(K).

P: Pressure (Pa).

mix: cMixture object.

phase: set phase = 'liq' to calculate the fugacity of a liquid phase

phase = 'gas' to calculate the fugacity of a gas phase

Optional Parameters:

May have additional optional parameters in specific equations of state (e.g. convergence settings).

Results:

f: fugacity coefficients

EoS: returns EoS used for calculations.

Description

Calculates the fugacity coefficients of a fluid mixture, according to the equation of state implemented.

Example programs

FugacityExample.m shows several examples of application of the method.

Method fugS

Function prototype

[f, EoS] = fugS(EoS,T,P,mix,phase,[varargin])

Purpose

Calculates the fugacity coefficients of components in a solid.

Parameters and results

Parameters:

EoS: Equation of state used for calculations

T: Temperature(K).

P: Pressure (Pa).

mix: cMixture object.

Optional Parameters:

May have additional optional parameters in specific equations of state (e.g. convergence settings).

Results:

f: fugacity coefficients

EoS: returns EoS used for calculations.

Description

Calculates the fugacity coefficients in a solid phase. The fugacity coefficients are calculated assuming that solid phases are composed by a pure component, and relating the fugacity of the solid phase to the fugacity of a reference, sub-cooled liquid according to the following equation:

$$ln\varphi_s = ln\varphi_L^0 + \frac{\Delta H_f}{R}\left(\frac{1}{T_f} - \frac{1}{T}\right)$$

Method MultiFlash

Function prototype

[beta, thita, x, K, phase, val, time, EoS] =

MultiFlash(EoS,P,T,mix,guess_x,guess_beta,phase)

Purpose

Calculates an isothermal flash with two or more phases in equilibrium.

Parameters and results

Parameters:

EoS: EoS used for calculations

P: Pressure (Pa)

T: Temperature (K)

mix: mixture (cMixture object)

mix.x must contain the global composition of the mixture

guess_x: initial guess for phase composition composition

(molar fraction matrix, with dimensions number of phases*number of components)

guess_beta: initial guess for phase fraction, vector length number of phases

phase: type of phase, cell vector with elements 'liq' for liquid phases,

'gas' for gas phases and 'sol' for solid phases (example: {'liq' 'liq' 'gas'})

Results:

beta: calculated phase fraction, vector length number of phases

thita: calculated phase stability, vector length number of phases

x: calculated phase composition (molar fraction matrix, with dimensions

number of phases*number of components)

K: distribution coefficients, matrix with dimensions number of phases*number of components

K(phase i, component j) = x(phase i, component j)/x(phase 1, component j)

phase: type of phase, cell vector with elements 'liq' for liquid phases

and 'gas' for gas phases (order in returned liquid may differ from

original order in parameter liquid)

val: final value of the objective function

time: time required for calculations

EoS: returns EoS used for calculations

Description

Calculates an isothermal flash with multiple phases by simultaneously solving for beta (fraction of each phase in the equilibrium) and thita (stability of each

phase). The calculation of beta and thita by minimization of Gibbs energy is implemented in the function obj_FlashMulti.m, while the function MultiFlash.m implements the iterative calculation of the composition of each phase.

Example programs

MultiFlashExample.m shows several examples of application of the method.

Method PxyDiagram

Function prototype

[P,x,y,handle,EoS] = PxyDiagram(EoS,mix,T,guess_P,guess_y,[numP], [interval], [diagnostics])

Purpose

Calculates a Pressure-Composition phase equilibrium diagram of a binary mixture.

Parameters and results

Parameters:

EoS: Equation of State used for calculations

mix: mixture (cMixture object).

T: Temperature (K)

guess_P: initial guess of pressure for the first point of the diagram (Pa)

guess_y: initial guess of gas composition for the first point of the diagram (molar fraction)

Optional parameters (set [] to keep default value)

numP: number of points (default value 20)

interval: interval of calculation, vector [mol_frac_ini, mol_frac_end], molar fractions referred to first component of mixture (default value [0 1])

diagnostics: set diagnostics = 1 to get information after the calculation of each point of the diagram (default value = 0)

Results:

P: Calculated pressures (Pa)

x: Calculated liquid compositions (molar fraction)

y: Calculated gas compositions (molar fraction)

handle: Handle of the Pxy diagram

EoS: Returns EoS used for calculations

Description

Calculates a Pressure-Composition phase equilibrium diagram of a binary mixture. For this purpose, the method calculates a series of bubble points for pressure with specified temperature and composition (BubblePoint method), according to the values of the interval of calculation and the number of points in the diagram. If the interval encompasses the pure component conditions (x_1 = 0 or x_1 = 1), the corresponding point of the diagram is calculated with the VaporPressure method. The method uses the values of pressure and composition provided by the user as initial guesses for the calculation of the first point of the diagram. Subsequent points are obtained using the results of the last successful calculation as initial guess.

Example programs

PxyDiagramExample.m shows several examples of application of the method.

Method Solubility

Function prototype

[S,val,time,flag,EoS] = Solubility(EoS,mix,T,P,[guess_S])

Purpose

Calculates the solubility of a solid in a fluid phase.

Parameters and results

Parameters:

EoS: Equation of State used for calculations (cEOS object)

mix: mixture (cMixture object). Parameter mix.x or mox.molFrac must contain global composition of the mixture

T: Temperature (K)

P: Pressure (Pa)

Optional parameters:

guess_S: estimation of solubility, mol fraction vector (default value: global mixture composition)

Results:

S: composition of the fluid phase (molar fraction)

val: final value of the objective function obj_BubblePointP

time: time required for calculations

flag: = 1 if calculations finished correctly, = 0 if warnings appeared

EoS: returns EoS used for calculations

Description

Calculates the solubility of a solid phase in a fluid phase. The calculation is implemented performing a flash calculation considering a solid and a fluid phase, using the MultiFlash method, and returning the composition of the fluid phase in the equilibrium.

Method TxyDiagram

Function prototype

[T, x, y, handle, EoS] = TxyDiagram(EoS, mix, guess_T, P, guess_y, [numP], [interval], [diagnostics])

Purpose

Calculates a Temperature-Composition phase equilibrium diagram of a binary mixture.

Parameters and results

Parameters:

EoS: Equation of State used for calculations

mix: mixture (cMixture object)

guess_T: initial guess of temperature for the first point of the diagram (K)

P: Pressure (Pa)

guess_y: initial guess of gas composition for the first point of the

diagram (molar fraction)

Optional parameters (set [] to keep default value)

numP: number of points (default value 20)

interval: interval of calculation, vector [mol_frac_ini, mol_frac_end],

molar fractions referred to first component of mixture

(default value [0 1])

diagnostics: set diagnostics = 1 to get information after the calculation

of each point of the diagram (default value = 0)

Results:

T: Calculated temperatures (K)

x: Calculated liquid compositions (molar fraction)

y: Calculated gas compositions (molar fraction)

handle: Handle of the Txy diagram

EoS: Returns EoS used for calculations

Description

Calculates a Temperature-Composition phase equilibrium diagram of a binary mixture. For this purpose, the method calculates a series of bubble points for pressure with specified temperature and composition (BubblePoint method), according to the values of the interval of calculation and the number of points in the diagram. If the interval encompasses the pure component conditions ($x_1 = 0$ or $x_1 = 1$), the corresponding point of the diagram is calculated with the BoilingTemp method. The method uses the values of temperature and composition provided by the user as initial guesses for the calculation of the first point of the diagram. Subsequent points are obtained using the results of the last successful calculation as initial guess.

Example programs

TxyDiagramExample.m shows several examples of application of the method.

Method VaporPressure

Function prototype

[P, flag, val, time, EoS] = VaporPressure(EoS,comp,T,[guess_P],[options])

Purpose

Calculates the vapor pressure of a pure component at a certain temperature.

Parameters and results

Parameters:

EoS: Equation of State used for calculations (cEOS object)

comp: Component (cSubstance object)

T: Temperature (K)

Optional parameters:

guess_P: initial guess of vapor pressure (Pa). If it is not provided, the initial guess is obtained using the Antoine equation using the 'Ant' parameters of the component

options: parameters of the fzero numerical resolution method (structure generated with "optimset")

Results:

P: vapor pressure (Pa)

flag: flag returned by the fzero numerical resolution method:

1 FZERO found a zero X.

2 Calculations converged to single phase region

-1 Algorithm terminated by output function.

-3 NaN or Inf function value encountered during search for an interval containing a sign change.

-4 Complex function value encountered during search for an interval containing a sign change.

-5 FZERO may have converged to a singular point.

val: final value of the objective function obj_Pvap

time: time required for calculations

EoS: returns EoS used for calculations

Description

Calculates the vapor pressure of a pure component solving the condition of equality of fugacities between liquid and gas phase:

$$f^L = f^V \rightarrow \varphi^L = \varphi^V$$

This condition is implemented in the file obj_Pvap.m and is solved numerically in the file VaporPressure.m using the Matlab fzero function.

Example programs

VaporPressureExample.m shows several examples of application of the method.

Class cNewEoS

This class is provided as a template for the implementation of a new equation of state

Properties

This class has no associated properties.

Methods

This class has no associated methods.

List of files

cNewEoS.m: Constructor of the class

This file should be renamed using a more specific tag for the equation of state. The folder in which all the files of the class are stored should be renamed accordingly.

By default, it is considered that the class is derived from cIdealEoS. A more specific parent class should be defined if appropriate (line 7 of this file).

Specific properties required by the EoS should be defined in the base structure of this class (line 9 of this file).

A specific identification tag should be defined. If a class other than cIdealEoS is defined as base class, the identification tag should be assigned making reference to that base class instead of cIdealEoS (line 13 of this file)

compr.m: Calculation of the compressibility coefficient of the fluid mixture. The calculation of the compressibility coefficient with the equation of state should be implemented here to overload the method of the base class. This function can make reference to any number of auxiliary functions as required for the implementation. Any number of input parameters besides the standard parameters (temperature, pressure, mixture and phase) can be added using the 'varargin' feature. If done so, and these additional parameters are required for any calculation with the equation of state, notice that it may be necessary to overload the calculation methods of cIdealEoS (BoilingTemp, BubblePoint etc.) accordingly.

fugF.m: Calculation of the fugacity coefficients of components of the fluid mixture. The calculation of the fugacity coefficient with the equation of state should be implemented here to overload the method of the base class. This function can make reference to any number of auxiliary functions as required for the implementation. Any number of input parameters besides the standard parameters (temperature, pressure, mixture and phase) can be added using the 'varargin' feature. If done so, and these additional parameters are required for any calculation with the equation of state, notice that it may be necessary to overload the calculation methods of cIdealEoS (BoilingTemp, BubblePoint etc.) accordingly.

NOTICE: overloading compr.m and fugF.m with specific methods for the equation of state usually is the minimum requirement for implementing a new equation of state. However, any other method of the base class can be overloaded if required for implementation of the equation.

Class cGCEoS

Implements the Group-Contribution Equation of State (reference: Skjold-Jorgensen, S., 1984. Gas-solubility calculations. II. Application of a new group-contribution equation of state. Fluid Phase Equilibr. 16, 317-351). This class is derived from the base cIdealEoS class.

This class uses the following properties of cSubstance objects:

MW: molecular weight (g/mol)

Tc: Critical Temperature (K)

EoSParam(1) - dc - critical diameter (cm/mol)

EoSParam(2) - group decomposition

Properties

This class has no associated properties

Methods

ClearGroups: removes all groups defined.

compr: calculates the compressibility coefficient of a fluid mixture. Overloads the corresponding method of the base class.

FillDefaultGroups: Adds all group parameters defined in the following reference: Fornari, T., 2007. Revision and summary of the group contribution equation of state parameter table: Application to edible oil constituents. Fluid Phase Equilib. 262, 187-209. This method is executed automatically upon creating an object of this class.

fugF: calculates the fugacity coefficients of a fluid mixture. Overloads the corresponding method of the base class.

GetGroup: retrieves the parameters of a group.

GetkG: retrieves the interaction parameters between groups.

NumGroups: retrieves the number of groups defined in a cGCEoS object.

SetGroup: sets the parameters of a group.

SetHighMWParam: sets the parameters specifically correlated for mixtures with high molecular weight parameters, as defined in the following reference: Fornari, T., 2007. Revision and summary of the group contribution equation of state parameter table: Application to edible oil constituents. Fluid Phase Equilib. 262, 187-209

SetkG: sets the interaction parameters between groups.

List of files

cGCEoS.m: constructor of the class.

ClearGroups.m: implements the method for clearing defined groups.

compr.m: implements the calculation of compressibility coefficients of mixtures.

FillDefaultGroups.m: implements the addition of the default parameter table.

fugF.m: implements the calculation of fugacity coefficients of mixtures.

GetGroup.m: implements the retrieval of group parameters.

GetkG.m: implements the retrieval of group interaction parameters.

NumGroups.m: implements the retrieval of the number of defined groups.

obj_GC_EOS.m: objective function for the calculation of fugacity and compressibility coefficients with GC-EOS. It is used by methods compr and fugF, and solved numerically using the Matlab fsolve function.

SetGroup.m: implements the definition of group parameters.

SetHighMWParam.m: implements the addition of the parameter table for high molecular weight compounds.

SetkG: implements the definition of group interaction parameters.

Example programs

GCExample.m shows several examples of application of this equation of state.

Class cPCSAFTEoS

Implements the Perturbed-Chain SAFT Equation of State (reference: Gross, J., Sadowski, G., 2001. Perturbed-Chain SAFT: An equation of state based on a perturbation theory for chain molecules. Ind. Eng. Chem. Res. 40, 1244-1260.). This class is derived from the class cSAFTEoS.

This class uses the following properties of cSubstance objects:

MW: molecular weight (g/mol)

EoSParam(1) - m, number of segments per chain

EoSParam(2) - sigma, segment diameter (Angstrom)

EoSParam(3) - epsilon/k, depth of pair potential (K)

EoSParam(4) - Number of association sites (only for associating molecules)

EoSParam(5) - Effective association volume(only for associating molecules)

EoSParam(6) - e/k, association energy (K) (only for associating molecules)

This EoS uses the following interaction parameters of cMixture objects:

k1: Binary interaction parameter k

Properties

This class has no associated properties

Methods

fugF: calculates the fugacity coefficients of a fluid mixture. Overloads the corresponding method of the base class.

Helmholtz: calculates the Helmholtz free energy of a fluid mixture.

List of files

cPCSAFTEoS.m: constructor of the class.

fugF.m: implements the calculation of fugacity coefficients of mixtures.

HardSphereDiameter.m: calculates the temperature-dependant hard sphere diameter, auxiliary function used in compressibility and fugacity calculations.

Helmholtz.m: implements the calculation of the Helmholtz free energy of mixtures.

HelholtzDisp.m: calculates the dispersion contribution to the Helmholtz energy, auxiliary function used by the Helmholtz method.

HelmholtzHC.m: calculates the hard chain contribution to the Helmholtz energy, auxiliary function used by the Helmholtz method.

mu_Disp.m: calculates the dispersion contribution to the chemical potential, auxiliary function used by the fugF method.

mu_HC.m: calculates the hard chain contribution to the chemical potential, auxiliary function used by the fugF method.

obj_SAFT.m: auxiliary objective function for the calculation of fugacity and compressibility coefficients. It is called by the fugF and compr methods and solved numerically using the Matlab fsolve function.

Z_disp.m: calculates the dispersion contribution to the compressibility coefficient, auxiliary function used by the compr method.

Z_HC.m: calculates the hard chain contribution to the compressibility coefficient, auxiliary function used by the compr method.

Example programs

PCSAFTExample.m shows several examples of application of this equation of state.

Class cPHSCEoS

Implements the Perturbed--Hard-Sphere-Chain Equation of State (reference: Song, Y., Lambert, S. M., Prausnitz, J. M., 1994. A Perturbed Hard-Sphere-Chain equation of state for normal fluids and polymers. Ind. Eng. Chem. Res. 33, 1047-1057). This class is derived from the base cIdealEoS class.

This class uses the following properties of cSubstance objects:

MW: molecular weight (g/mol)

EoSParam(1) - V, characteristic volume (m^3/mol)

EoSParam(2) - A, characteristic surface area (m^2/mol)

EoSParam(3) - E, characteristic cohesive energy (J/mol)

This EoS uses the following interaction parameters of cMixture objects:

k1: Binary interaction parameter k

Properties

This class has no associated properties

Methods

compr: calculates the compressibility coefficient of a fluid mixture. Overloads the corresponding method of the base class.

fugF: calculates the fugacity coefficients of a fluid mixture. Overloads the corresponding method of the base class.

List of files

cPHSCEoS.m: constructor of the class.

compr.m: implements the calculation of compressibility coefficients of mixtures.

fugF.m: implements the calculation of fugacity coefficients of mixtures.

obj_PHSC.m: auxiliary objective function for the calculation of fugacity and compressibility coefficients. It is called by the fugF and compr methods and solved numerically using the Matlab fsolve function

Example programs

PHSCExample.m shows several examples of application of this equation of state.

Class cPRBMEoS

Implements the Peng-Robinson Equation of State with Boston-Mathias alpha function (reference: Boston, J. F., Mathias, P. M., 1980. Phase Equilibria in a Third-Generation Process Simulator. 2nd International Conference on Phase Equilibria and Fluid Properties in the Chemical Industry, Berlin, Germany.). This class is derived from the class cPREoS.

This class uses the following properties of cSubstance objects:

MW: molecular weight (g/mol)

Tc: Critical Temperature (K)

Pc: Critical Pressure (Pa)

w: Acentric factor

This EoS uses the following interaction parameters of cMixture objects:

k1: Binary interaction parameter k (quadratic mixing rules)

k2: Binary interaction parameter l (quadratic mixing rules)

Properties

This class has no associated properties.

Methods

This class has no associated methods.

List of files

alpha_function.m: calculates the Boston-Mathias alpha function, auxiliary function used by compr and fugF methods.

cPRBMEoS.m: constructor of the class.

Example programs

PRBMExample.m shows several examples of application of this equation of state.

Class cPREoS

Implements the standard Peng-Robinson Equation of State with quadratic mixing rules (reference: Peng, D. Y., Robinson, D. B., 1976. A new two-constant equation of state. Ind. Eng. Chem. Fundam. 15, 59-64.). This class is derived from the base cIdealEoS class.

This class uses the following properties of cSubstance objects:

MW: molecular weight (g/mol)

Tc: Critical Temperature (K)

Pc: Critical Pressure (Pa)

w: Acentric factor

This EoS uses the following interaction parameters of cMixture objects:

k1: Binary interaction parameter k (quadratic mixing rules)

k2: Binary interaction parameter l (quadratic mixing rules)

Properties

This class has no associated properties.

Methods

compr: calculates the compressibility coefficient of a fluid mixture. Overloads the corresponding method of the base class.

fugF: calculates the fugacity coefficients of a fluid mixture. Overloads the corresponding method of the base class.

Enthalpy: calculates the residual enthalpy of a fluid mixture. Overloads the corresponding method of the base class.

List of files

alpha_function.m: calculates the standard alpha function, auxiliary function used by compr and fugF methods.

compr.m: implements the calculation of compressibility coefficients of mixtures.

cPREoS.m: constructor of the class.

fugF.m: implements the calculation of fugacity coefficients of mixtures.

Enthalpy.m: implements the calculation of the residual enthalpy of mixtures.

Example programs

PRExample.m shows several examples of application of this equation of state.

Class cPRSVEoS

Implements the Peng-Robinson Equation of State with Stryjek-Vera alpha function (reference: Stryjek, R., Vera, J. H., 1986. PRSV: An improved Peng-Robinson equation of state for pure compounds and mixtures. Can. J. Chem. Eng. 64, 323-333.). This class is derived from the class cPREoS.

This class uses the following properties of cSubstance objects:

MW: molecular weight (g/mol)

Tc: Critical Temperature (K)

Pc: Critical Pressure (Pa)

w: Acentric factor

EoSParam(1) - kappa (alpha function parameter)

This EoS uses the following interaction parameters of cMixture objects:

k1: Binary interaction parameter k (quadratic mixing rules)

k2: Binary interaction parameter l (quadratic mixing rules)

Properties

This class has no associated properties.

Methods

This class has no associated methods.

List of files

alpha_function.m: calculates the Stryjek-Vera alpha function, auxiliary function used by compr and fugF methods.

cPRSVEoS.m: constructor of the class.

Example programs

PRSVExample.m shows several examples of application of this equation of state.

Class cPRWSEoS

Implements the Peng-Robinson Equation of State with Wong-Sandler mixing rules (reference: Wong, S. S. H., Sandler, S. I., 1992. A Theoretically correct mixing rule for cubic equations of state. AIChE J. 38, 671-680). This class is derived from the base cIdealEoS class.

This class uses the following properties of cSubstance objects:

MW: molecular weight (g/mol)

Tc: Critical Temperature (K)

Pc: Critical Pressure (Pa)

w: Acentric factor

This EoS uses the following interaction parameters of cMixture objects:

k1: Binary interaction parameter k (quadratic mixing rules)

k2: Non-randomness parameter

k3: Binary interaction parameter tau

Properties

This class has no associated properties.

Methods

compr: calculates the compressibility coefficient of a fluid mixture. Overloads the corresponding method of the base class.

fugF: calculates the fugacity coefficients of a fluid mixture. Overloads the corresponding method of the base class.

List of files

alpha_function.m: calculates the standard alpha function, auxiliary function used by compr and fugF methods.

compr.m: implements the calculation of compressibility coefficients of mixtures.

cPRWSEoS.m: constructor of the class.

fugF.m: implements the calculation of fugacity coefficients of mixtures.

Example programs

PRWSExample.m shows several examples of application of this equation of state.

Class cPSRKEoS

Implements the Predictive Soave-Redlich-Kwong equation of state (reference: Holderbaum, T., Gmehling, J. , 1991. PSRK: A group contribution equation of state based on UNIFAC. Fluid Phase Equilibr. 70, 251-265). This class is derived from the base cIdealEoS class.

This class uses the following properties of cSubstance objects:

MW: molecular weight (g/mol)

Tc: Critical Temperature (K)

Pc: Critical Pressure (Pa)

w: Acentric factor

EoSParam(1) - c1 (alpha function parameter)

EoSParam(2) - c2 (alpha function parameter)

EoSParam(3) - c3 (alpha function parameter)

EoSParam(4) - Group decomposition

Properties

This class has no associated properties.

Methods

ClearGroups: removes all groups defined.

compr: calculates the compressibility coefficient of a fluid mixture. Overloads the corresponding method of the base class.

FillDefaultGroups: Adds all group parameters defined in the following reference: Horstmann, S., Jabloniec, A., Krafczyk, J., Fischer, K., Gmehling, J. PSRK group contribution of state: comprehensive revision and extension IV, including critical constants and a-function parameters for 1000 components. Fluid Phase Equilibr. 227 (2005) 157-164). This method is executed automatically upon creating an object of this class.

fugF: calculates the fugacity coefficients of a fluid mixture. Overloads the corresponding method of the base class.

GetGroup: retrieves the parameters of a group.

GetkG: retrieves the interaction parameters between groups.

NumG: retrieves the number of groups defined in a cGCEoS object.

SetGroup: sets the parameters of a group.

SetkG: sets the interaction parameters between groups.

List of files

cPSRKEoS.m: constructor of the class.

ClearGroups.m: implements the method for clearing defined groups.

compr.m: implements the calculation of compressibility coefficients of mixtures.

FillDefaultGroups.m: implements the addition of the default parameter table.

fugF.m: implements the calculation of fugacity coefficients of mixtures.

GetGroup.m: implements the retrieval of group parameters.

GetkG.m: implements the retrieval of group interaction parameters.

NumG.m: implements the retrieval of the number of defined groups.

SetGroup.m: implements the definition of group parameters.

SetkG: implements the definition of group interaction parameters.

UNIFAC.m: implements the calculation of excess Gibbs energy with UNIFAC, auxiliary function used by compr and fugF methods.

Example programs

PSRKExample.m shows several examples of application of this equation of state.

Class cPTEoS

Implements the Patel-Teja equation of state (reference: Patel, N. C., Teja, A. S., 1982. A new cubic equation of state for fluids and fluid mixtures. Chem. Eng. Sci. 37, 463-473). This class is derived from the base cIdealEoS class.

This class uses the following properties of cSubstance objects:

MW: molecular weight (g/mol)

Tc: Critical Temperature (K)

Pc: Critical Pressure (Pa)

EoSParam(1) - zc (Critical compressibility coefficient)

EoSParam(2) - F (alpha function parameter)

This EoS uses the following interaction parameters of cMixture objects:

k1: Binary interaction parameter k (quadratic mixing rules)

k2: Binary interaction parameter l (quadratic mixing rules)

k3: Binary interaction parameter m (quadratic mixing rules)

Properties

This class has no associated properties.

Methods

compr: calculates the compressibility coefficient of a fluid mixture. Overloads the corresponding method of the base class.

fugF: calculates the fugacity coefficients of a fluid mixture. Overloads the corresponding method of the base class.

List of files

alpha_function.m: calculates the standard alpha function, auxiliary function used by compr and fugF methods.

compr.m: implements the calculation of compressibility coefficients of mixtures.

cPTEoS.m: constructor of the class.

fugF.m: implements the calculation of fugacity coefficients of mixtures.

Example programs

PTExample.m shows several examples of application of this equation of state.

Class cSAFTEoS

Implements the SAFT Equation of State (reference: Chapman, W. G., Gubbins, K. E., Jackson, G., Radosz, M., 1990. New reference equation of state for associating liquids. Ind. Eng. Chem. Res. 29, 1709-1721). This class is derived from the base cIdealEoS class.

This class uses the following properties of cSubstance objects:

MW: molecular weight (g/mol)

EoSParam(1) - m, number of segments per chain

EoSParam(2) - sigma, segment diameter (Angstrom)

EoSParam(3) - epsilon/k, depth of pair potential (K)

EoSParam(4) - Number of association sites (only for associating molecules)

EoSParam(5) - Effective association volume(only for associating molecules)

EoSParam(6) - e/k, association energy (K) (only for associating molecules)

This EoS uses the following interaction parameters of cMixture objects:

k1: Binary interaction parameter k

Properties

This class has no associated properties

Methods

compr: calculates the compressibility coefficients of a fluid mixture. Overloads the corresponding method of the base class.

fugF: calculates the fugacity coefficients of a fluid mixture. Overloads the corresponding method of the base class.

Helmholtz: calculates the Helmholtz free energy of a fluid mixture.

List of files

compr.m: implements the calculation of compressibility coefficients of mixtures.

cSAFTEoS.m: constructor of the class.

fugF.m: implements the calculation of fugacity coefficients of mixtures.

HardSphereDiameter.m: calculates the temperature-dependant hard sphere diameter, auxiliary function used in compressibility and fugacity calculations.

Helmholtz.m: implements the calculation of the Helmholtz free energy of mixtures.

HelholtzAss.m: calculates the association contribution to the Helmholtz energy, auxiliary function used by the Helmholtz method.

HelmholtzChain.m: calculates the chain contribution to the Helmholtz energy, auxiliary function used by the Helmholtz method.

HelmholtzSeg.m: calculates the segment contribution to the Helmholtz energy, auxiliary function used by the Helmholtz method.

mu_Ass.m: calculates the association contribution to the chemical potential, auxiliary function used by the fugF method.

mu_Chain.m: calculates the chain contribution to the chemical potential, auxiliary function used by the fugF method.

mu_Seg.m: calculates the segment contribution to the chemical potential, auxiliary function used by the fugF method.

obj_HelmholtzAss.m: calculates the fraction of non-associated sites, auxiliary function used by fugF, Helmholtz and compr methods

obj_muAss.m: auxiliary function for the calculation of the association contribution to the chemical potential.

obj_SAFT.m: auxiliary objective function for the calculation of fugacity and compressibility coefficients. It is called by the fugF and compr methods and solved numerically using the Matlab fsolve function.

Z_ass.m: calculates the association contribution to the compressibility coefficient, auxiliary function used by the compr method.

Z_chain.m: calculates the chain contribution to the compressibility coefficient, auxiliary function used by the compr method.

Z_seg.m: calculates the segment contribution to the compressibility coefficient, auxiliary function used by the compr method.

Example programs

SAFTExample.m shows several examples of application of this equation of state.

Class cSRKEoS

Implements the standard Soave-Redlich-Kwong Equation of State with quadratic mixing rules (reference: Soave, G., 1972. Equilibrium constants from a modified Redlich-Kwong equation of state. Chem. Eng. Sci. 27, 1197-1203.). This class is derived from the base cIdealEoS class.

This class uses the following properties of cSubstance objects:

MW: molecular weight (g/mol)

Tc: Critical Temperature (K)

Pc: Critical Pressure (Pa)

w: Acentric factor

This EoS uses the following interaction parameters of cMixture objects:

k1: Binary interaction parameter k (quadratic mixing rules)

k2: Binary interaction parameter l (quadratic mixing rules)

Properties

This class has no associated properties.

Methods

compr: calculates the compressibility coefficient of a fluid mixture. Overloads the corresponding method of the base class.

fugF: calculates the fugacity coefficients of a fluid mixture. Overloads the corresponding method of the base class.

List of files

alpha_function.m: calculates the standard alpha function, auxiliary function used by compr and fugF methods.

compr.m: implements the calculation of compressibility coefficients of mixtures.

cPREoS.m: constructor of the class.

fugF.m: implements the calculation of fugacity coefficients of mixtures.

Example programs

SRKExample.m shows several examples of application of this equation of state.

Class cVPTEoS

Implements the Valderrama-Patel-Teja equation of state (reference: Soave, G., 1972. Equilibrium constants from a modified Redlich-Kwong equation of state. Chem. Eng. Sci. 27, 1197-1203.). This class is derived from the class cPTEoS.

This class uses the following properties of cSubstance objects:

MW: molecular weight (g/mol)

Tc: Critical Temperature (K)

Pc: Critical Pressure (Pa)

w: Acentric factor

EoSParam(1) - zc

This EoS uses the following interaction parameters of cMixture objects:

k1: Binary interaction parameter k (quadratic mixing rules)

k2: Binary interaction parameter l (quadratic mixing rules)

k3: Binary interaction parameter m (quadratic mixing rules)

Properties

This class has no associated properties.

Methods

This class has no associated methods.

List of files

alpha_function.m: calculates the Valderrama alpha function, auxiliary function used by compr and fugF methods..

cVPTEoS.m: constructor of the class.

Example programs

VPTExample.m shows several examples of application of this equation of state.