**Content**

Structure of JSON database

Overview of implemented working pairs

Details of refrigerant equations

Details of Langmuir equation

Details of Toth equation

Details of Dual-Site-Sips equations

Details of Dubinin-Astakhov equation

**Structure of JSON-database**

**Generally:**

Database is an array and each element of the array is one working pair or one refrigerant, whereas each refrigerant or working pair is an object.

[{Refrigerant | Working pair},

{Refrigerant | Working pair},

…

{Refrigerant | Working pair}

]

**Refrigerants:**

Each refrigerant entry is an object that has the attributes ‘k’ and ‘v’. The attribute ‘k’ saves the name of the object, whereas the attribute ‘v’ saves coefficients for different equation of states. The equation of states are saved as objects that have attributes for the literature ‘\_c\_’, the name ‘\_e\_’ and the parameters ‘\_p\_’.

{“k”: {“\_as\_”: “dum\_sorbent”,

“\_rf\_”: “Name of refrigerant”,

“\_st\_”: “dum\_subtype”},

“v”: {“\_ep\_”: {“Equation1”: {“\_c\_”: “Literature”,

“\_e\_”: Name of equation”,

“\_p\_”: {“Parameter1”: Value,

“Parameter 2”: Value,

…},

….},

“\_r\_”: “Name of refrigerant”,

“\_s\_”: “dum\_sorbent”,

“\_t\_”: “dum\_subtype”}

}

**Working pairs:**

Each working pair has the same structure as a refrigerant entry and stores coefficients for different isotherm equations.

**Overview of implemented working pairs**

**Refrigerants:**

Table summarises implemented refrigerants and indicates if equation of states are implemented for the saturation pressure p\_sat(T) or for the saturated liquid density rho\_l(T).

|  |  |  |
| --- | --- | --- |
| **Refrigerant** | **EoS for p\_sat(T)** | **EoS for rho\_l(T)** |
| *R-134a* | X | X |
| *CO2* | X | X |
| *Methane* | X | X |
| *Water* | X | X |
| *Isobutane* | X | X |
| *Propane* | X | X |
| *R-410a* | X | Missing |
| *R-507a* | X | Missing |
| *Propylene* | Missing | Missing |

**Working pairs:**

Table summarises implemented working pairs.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Refrigerant** | **MOF** | **Carbon** | **Zeolite** | **Silicagel** |
| *R-134a* |  | maxsorb-iii  acf-a-20 |  |  |
| *CO2* |  | norit-rb1 | 5a  13x | X |
| *Methane* |  | norit-rb1 |  |  |
| *Water* |  |  | 5a  13x | X |
| *Isobutane* | cubtc |  |  |  |
| *Propane* | cubtc | maxsorb-iii |  |  |
| *R-410a* |  | maxsorb-iii |  |  |
| *R-507a* |  | acf-a-20 |  |  |
| *Propylene* | cubtc |  |  |  |

**Details of refrigerant equation**

**EoS p\_sat(T) and dp\_sat(T):**

For calculating the saturation pressure, the following generic function is implemented for all refrigerants:

with:

The derivative of the saturation pressure is implemented analytically.

**Parameters required from database:**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  | **…** | **…** |  |  |
| in K | in Pa | in - | in - | in - | in - | … | … | in - | in - |

**EoS rho\_l(T) and dp\_rho\_l\_dT(T):**

For calculating the saturated liquid density, the following generic function is implemented for all refrigerants:

with:

The derivative of the saturated liquid density is implemented analytically.

**Parameters required from database:**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  | **…** | **…** |  |  |
| in K | in kg/m³ | in - | in - | in - | in - | in - | … | … | in - | in - |

**Details of Langmuir equation**

**Equation w(p,T):**

Returns loading in kgref/kgsorbent and depends on pressure in Pa and temperature in K.

with:

**Equation p(w,T):**

Returns pressure in Pa and depends on loading in kgref/kgsorbent and temperature in K. Equation is analytical inverse of w(p,T).

**Equation T(p,w):**

Returns temperature in K and depends on pressure in Pa and loading in kgref/kgsorbent. Equation is analytical inverse of w(p,T).

**Equation dwdp(p,T):**

Returns derivative of loading with respect to pressure in kgref/kgsorbent/Pa and depends on pressure in Pa and temperature in K. Equation is analytical derivative of w(p,T) with respect to p.

**Equation dwdT(p,T):**

Returns derivative of loading with respect to temperature in kgref/kgsorbent/K and depends on pressure in Pa and temperature in K. Equation is analytical derivative of w(p,T) with respect to T.

**Parameters required from database:**

|  |  |  |
| --- | --- | --- |
|  |  |  |
| in J/mol | in 1/Pa | in kgref/kgsorbent |

**Toth equation**

**Equation w(p,T):**

Returns loading in kgref/kgsorbent and depends on pressure in Pa and temperature in K.

with: , ,

**Equation p(w,T):**

Returns pressure in Pa and depends on loading in kgref/kgsorbent and temperature in K. Equation is analytical inverse of w(p,T).

**Equation T(p,w):**

Returns temperature in K and depends on pressure in Pa and loading in kgref/kgsorbent. Equation is numerical inverse of w(p,T) using the Newton-Raphson method.

**Equation dwdp(p,T):**

Returns derivative of loading with respect to pressure in kgref/kgsorbent/Pa and depends on pressure in Pa and temperature in K. Equation is analytical derivative of w(p,T) with respect to p.

**Equation dwdT(p,T):**

Returns derivative of loading with respect to temperature in kgref/kgsorbent/K and depends on pressure in Pa and temperature in K. Equation is analytical derivative of w(p,T) with respect to T.

**Parameters required from database:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  | **r** |  |
| in 1/Pa | in K | in kgref/kgsorbent | in - | in K | in - | in kgref/kgsorbent |

**Dual-Site-Sips equation**

**Equation w(p,T):**

Returns loading in kgref/kgsorbent and depends on pressure in Pa and temperature in K.

with:

**Equation p(w,T):**

Returns pressure in Pa and depends on loading in kgref/kgsorbent and temperature in K. Equation is numerical inverse of w(p,T) using the Newton-Raphson method.

**Equation T(p,w):**

Returns temperature in K and depends on pressure in Pa and loading in kgref/kgsorbent. Equation is numerical inverse of w(p,T) using the Newton-Raphson method.

**Equation dwdp(p,T):**

Returns derivative of loading with respect to pressure in kgref/kgsorbent/Pa and depends on pressure in Pa and temperature in K. Equation is analytical derivative of w(p,T) with respect to p.

**Equation dwdT(p,T):**

Returns derivative of loading with respect to temperature in kgref/kgsorbent/K and depends on pressure in Pa and temperature in K. Equation is analytical derivative of w(p,T) with respect to T.

**Parameters required from database:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **bA,0** | **bB,0** | **QA,st** | **QB,st** | **ηA** | **ηB** | **wA** | **wB** | **T0** |
| in 1/Pa | in 1/Pa | in J/mol | in J/mol | in - | in - | in kgref/kgsorbent | in kgref/kgsorbent | in K |

**Dubinin-Astakhov equation**

**Equation W(A,rho):**

Returns volumetric loading in m³/kg and depends on adsorption potential in J/mol and saturated liquid density of adsorpt in kg/m³.

with:,

**Equation A(W,rho):**

Returns adsorption potential in J/mol and depends on volumetric loading in m³/kg and saturated liquid density of adsorpt in kg/m³. Equation is analytical inverse of W(A,rho).

**Equation w(p,T,psat(T),rho(T)):**

Returns loading in kgref/kgsorbent and depends on pressure in Pa, temperature in K, saturation pressure in Pa and saturated liquid density of adsorpt in kg/m³.

with: ,

**Equation p(w,T,psat(T),rho(T)):**

Returns pressure in Pa and depends on loading in kgref/kgsorbent, temperature in K, saturation pressure in Pa and saturated liquid density of adsorpt in kg/m³. Equation is analytical inverse of w(p,T,psat(T),rho(T)).

**Equation T(p,w,func\_psat(T), func\_rho(T), func\_dpsat(T)\_dT, func\_drho(T)\_dT):**

Returns temperature in K and depends on pressure in Pa, loading in kgref/kgsorbent, temperature in K and function pointers that calculate saturation pressure in Pa, saturated liquid density of adsorpt in kg/m³, derivative of saturation pressure in Pa/K and derivative of saturated liquid density of adsorpt with respect to temperature in kg/m³/K. Equation is numerical inverse of w(p,T,psat(T),rho(T)) using the Newton-Raphson method.

**Equation dWdA(A,rho):**

Returns derivative of volumetric loading with respect to adsorption potential in J m³/mol/kg and depends on adsorption potential in J/mol and saturated liquid density of adsorpt in kg/m³. Equation is analytical derivative of W(A,rho) with respect to A.

**Equation dAdW(W,rho):**

Returns derivative of adsorption potential with respect to volumetric loading in J/mol and depends on volumetric loading in m³/kg and saturated liquid density of adsorpt in kg/m³. Equation is analytical derivative of A(W,rho) with respect to W.

**Equation dwdp(p,T,psat(T),rho(T)):**

Returns derivative of loading with respect to pressure in kgref/kgsorbent/Pa and depends on pressure in Pa, temperature in K, saturation pressure in Pa and saturated liquid density of adsorpt in kg/m³. Equation is analytical derivative of w(p,T) with respect to p.

**Equation dwdT(p,T,psat(T),rho(T),dpsat(T)\_dT,drho(T)\_dT):**

Returns derivative of loading with respect to temperature in kgref/kgsorbent/K and depends on pressure in Pa, temperature in K, saturation pressure in Pa and saturated liquid density of adsorpt in kg/m³, derivative of saturation pressure in Pa/K and derivative of saturated liquid density of adsorpt with respect to temperature in kg/m³/K. Equation is analytical derivative of w(p,T) with respect to T.

**Further equations:**

Additionally, functions are implemented for w(p,T), p(w,T), T(p,w), dwdp(p,T) and dwdT(p,T). However, these functions use the equation of state that is implemented in the C-library to calculate the saturation pressure, the saturated liquid pressure and its derivatives. Consequently, the user cannot chose the equation of state when using these functions.

**Parameters required from database:**

|  |  |  |  |
| --- | --- | --- | --- |
| **E** | **n** | **W0** | **Flag** |
| in J/mol | in - | in m³/kg | in - |