Short user guide for Matlab class thermo

Class thermo

Objects of the thermo class are suited for efficient and accurate calculation of thermodynamic variables and calculations. It requires a model of the molar Helmholtz .free energy as a function of temperature and molar volume. At present such models are available for H_2 and CO_2.

The user starts by creating a `thermo` object:

```
>> th = thermo('H2');
```
or
```
>> th = thermo('CO2');
```

`thermo` reads in general parameters for the given species as well as parameters for the thermodynamic model. Next the thermodynamic state of object must be initialised by specifying the temperature and either molar volume or pressure. Examples:

```
>> th.Tvcalc(300,0.5)

>> th.Tpcalc(300,250e5)
```

`th.Tvcalc` uses the model in `H2helmholtz.m` or `CO2helmholtz.m` to calculate the molar Helmholtz free energy and its partial derivatives with respect to temperature T and molar volume v. It then populates the object with those derivatives as well as a set of thermodynamic properties such as molar enthalpy (`th.h`) and speed of sound (`th.c`). For a full list of properties, type

```
>> th.properties
```

Since the model is explicit in T and v, `th.Tvcalc` is fast and robust.

`th.Tpcalc` uses Newton's method to iterate on v to solve $p(T, v) = p$. Example for H_2:

```
>> th.Tpcalc(300,250e5)
>> fprintf('Speed of sound : %4.0f m/s\n',th.c)
Speed of sound : 1537 m/s
```

`th.Tpcalc` is quite fast and is robust except near the two-phase region. Here one should specify a relevant initial value for the molar volume.

A full list of methods available from the thermodynamic object is shown by typing:

```
>> th.methods
```

Saturation properties

The parameter files for H_2 and CO_2 include model parameters for the saturation pressure as a function of temperature. This model is compatible with the Helmholtz model. This model is implemented in two methods of the thermo class:

```
>> [p,dpdT] = th.psat(T);
```

This returns the saturation pressure and the derivative with respect to T. Note that the temperature must be between the triple point temperature and the critical temperature . Using T outside this region will result in an error.

The second method is:

```
>> [ps,liq,vap] = th.saturation(T);
```

This returns property structs for the saturated liquid and vapour states, making it easy to access any thermodynamic variable. For instance, the liquid density is given by

```
>> rho = th.Mw/liq.v
```

Units

SI units are used throughout; with the exception that kmol is used instead of mol. Intrinsic variables are on a per kmol basis. Properties per kg are found by dividing by th.Mw.

Matlab applications

A small number of application cases are shipped together with the thermodynamic model:

`demo_fill_gas_tank.m:`
Pressure and temperature histories when filling of a H_2 gas tank at a constant rate.

`demo_gas_tank_system.m:`
Pressure and temperature histories when filling gas from one tank into another.

`demo_tank_leak.m:`
Leak from the vapour phase of a liquid H_2 tank. Demonstrates vapour/liquid equilibrium and both sonic (choked) and subsonic flow.

`H2twophase_region.m, CO2twophase_region.m:`
Plots the two-phase envelope and selected isotherms in a v-p diagram.