LibIAPWS

Sangwook Ryu

Abstract

LibIAPWS is a C++ library to implement IAPWS [1] (International Association for the Properties of Water and Steam) formulation. It contains parametrizations for thermodynamic properties, including enthalpy, entropy, heat capacities and latent heat, as functions of temperature, pressure and mass density of H_2O. Coexisting phases among water vapor, liquid and ice are also computed and tabulated.

Contents

1 Library functions

1.1 R1-76 (2014) - `IAPWS::Lib76` class

IAPWS R1-76 [2] release provides parametrization for the surface tension σ of the interface between water vapor and liquid. There is one function `get_tension_surf`, which takes temperature T as input and returns the surface tension.

function	input	output
`get_tension_surf`	temperature T (K)	surface tension σ (J/m^2)

Table 1: Library functions in `IAPWS::Lib76` class

1.2 R6-95 (2018) - `IAPWS::Lib95` class

IAPWS R6-95 [3] release provides parametrization for the specific Helmholtz free energy f and thermodynamic quantities of water vapor and liquid. The mass density ρ and temperature T are considered as independent variables. The parametrized functions for single-phase are listed in Table 2.

function	input	output
`get_param_f`	mass density ρ (kg/m^3) temperature T (K)	specific Helmholtz free energy f (J/kg)
`get_param_g`	mass density ρ (kg/m^3) temperature T (K)	specific Gibbs free energy g (J/kg)
`get_param_pressure`	mass density ρ (kg/m^3) temperature T (K)	pressure p (Pa = N/m^2)
`get_param_dpress_drho`	mass density ρ (kg/m^3) temperature T (K)	derivative of pressure $\left.\frac{\partial p}{\partial \rho}\right\|_T$ (m^2/sec^2)
`get_param_erg_int`	mass density ρ (kg/m^3) temperature T (K)	specific internal energy u (J/kg)
`get_param_entropy`	mass density ρ (kg/m^3) temperature T (K)	specific entropy s (J/kg K)
`get_param_enthalpy`	mass density ρ (kg/m^3) temperature T (K)	specific enthalpy h (J/kg)
`get_param_heat_c_v`	mass density ρ (kg/m^3) temperature T (K)	specific isochoric heat capacity c_v (J/kg K)
`get_param_heat_c_p`	mass density ρ (kg/m^3) temperature T (K)	specific isobaric heat capacity c_p (J/kg K)
`get_param_speed_sound`	mass density ρ (kg/m^3) temperature T (K)	speed of sound w (m/sec)

Table 2: Library functions for single-phase thermodynamic quantities in `IAPWS::Lib95` class

In addition, one can utilize the parametrized Helmholtz free energy to obtain coexisting phases (saturation curve) between water vapor and liquid. In `IAPWS::Lib95` class, iterative root-finding algorithms are implemented to specify the saturation curve. (More details can be found in the following sections.) Table 3 presents functions to handle the tabulated saturation curve. One can call `make_tab_coex` to produce a table for saturation curve by specifying number of temperature bins and maximum temperature. The minimum temperature for tabulation is set to be that of the triple point $T_{\text{trip}} = 273.16\,\text{K}$. It is possible to expedite calculations in `make_tab_coex` function, by providing a function pointer to better initial guess for the saturation pressure. One option is to use `get_param4_sat_pressure` function in `IAPWS::Lib97` class and it is demonstrated in `make_tab_IAPWS95.cpp` source file. `export_tab_coex` and `import_tab_coex` functions can be used to write and read the tabulated saturation curve, respectively.

Once the tabulated saturation curve is produced or imported, one can call functions listed in Table 4 to obtain thermodynamic quantities on the saturation curve. The tabulated saturation curve is interpolated by means of natural cubic spline. All functions take temperature T (in K) as the input parameter.

function	input	output
`make_tab_coex`	number of temperature bins (`int`) maximum temperature (K) function pointer to initial guess for the saturation pressure (`double (*ptr)(double)`, default : `NULL`)	`void`
`export_tab_coex`	name of output file (`char *`)	`void`
`import_tab_coex`	name of input file (`char *`)	`void`

Table 3: Tabulation of coexisting phases (saturation curve between vapor and liquid) in `IAPWS::Lib95` class

function	input	output
`get_coex_pressure`	temperature T (K)	saturation pressure p_{coex} (Pa)
`get_coex_mden_vap`	temperature T (K)	mass density of vapor ρ_{vap} (kg/m^3)
`get_coex_mden_liq`	temperature T (K)	mass density of liquid ρ_{liq} (kg/m^3)
`get_coex_enthalpy_vap`	temperature T (K)	specific enthalpy of vapor h_{vap} (J/kg)
`get_coex_enthalpy_liq`	temperature T (K)	specific enthalpy of liquid h_{liq} (J/kg)
`get_coex_entropy_vap`	temperature T (K)	specific entropy of vapor s_{vap} (J/kg K)
`get_coex_entropy_liq`	temperature T (K)	specific entropy of liquid s_{liq} (J/kg K)
`get_coex_heat_latent`	temperature T (K)	specific latent heat $h_{\mathrm{latent;vap\text{-}liq}}$ (J/kg)

Table 4: Library functions for coexisting phases (saturation curve between vapor and liquid) in `IAPWS::Lib95` class

1.3 R10-06 (2009) - `IAPWS::Lib06` class

IAPWS R10-06 [5] release provides parametrization for the specific Gibbs free energy and thermodynamic quantities of H_2O Ice Ih. Temperature T and pressure p are considered as independent variables. The parametrized functions for single-phase are listed in Table 5.

1.4 R12-08 (2008) - `IAPWS::Lib08V` class

IAPWS R12-08 [6] release provides parametrization for the shear viscosity μ as a function of the mass density ρ and temperature T. Functions to be used are listed in Table 10. Note that one also needs to have an instance of `IAPWS::Lib95` class to implement the critical enhancement. There is a class (pointer) variable `ptr_lib95_` to connect a `IAPWS::Lib95` object and one can call `set_ptr_lib95` function to set it up.

2 Derivation for thermodynamic quantities

2.1 From the Helmholtz free energy $f(\rho, T)$

If one has mass density ρ (equivalently specific volume $v = 1/\rho$) and temperature T as independent variables, thermodynamic quantities can be derived from the specific Helmholtz free energy f

$$\frac{f(\rho,T)}{RT} = \phi\left(\delta \equiv \frac{\rho}{\rho_*}, \tau \equiv \frac{T_*}{T}\right) \tag{1}$$

where R is the specific gas constant. In addition, dimensionless quantities δ and τ are introduced in terms of reference mass density ρ_* and temperature T_*, respectively. For instance, specific entropy s and pressure p can be given by first derivatives of the Helmholtz free energy.

$$s = -\left.\frac{\partial f}{\partial T}\right|_{\rho} = R\left(\tau\phi_\tau - \phi\right) \quad \text{where} \quad \phi_\tau \equiv \frac{\partial \phi}{\partial \tau} \tag{2}$$

$$p = -\left.\frac{\partial f}{\partial v}\right|_{T} = \rho^2 \left.\frac{\partial f}{\partial \rho}\right|_{T} = \rho RT\,\delta\phi_\delta \quad \text{where} \quad \phi_\delta \equiv \frac{\partial \phi}{\partial \delta} \tag{3}$$

The specific internal energy u and enthalpy h can be subsequently obtained as

$$u = f + Ts = RT\,\tau\phi_\tau \tag{4}$$

$$h = u + pv = RT\left(\tau\phi_\tau + \delta\phi_\delta\right) . \tag{5}$$

function	input	output
`get_param_g`	temperature T (K) pressure P (Pa)	specific Gibbs free energy g (J/kg)
`get_param_f`	temperature T (K) pressure P (Pa)	specific Helmholtz free energy f (J/kg)
`get_param_mdensity`	temperature T (K) pressure P (Pa)	mass density ρ (kg/m^3)
`get_param_entropy`	temperature T (K) pressure P (Pa)	specific entropy s (J/kg K)
`get_param_heat_c_p`	temperature T (K) pressure P (Pa)	specific isobaric heat capacity c_p (J/kg K)
`get_param_enthalpy`	temperature T (K) pressure P (Pa)	specific enthalpy h (J/kg)
`get_param_erg_int`	temperature T (K) pressure P (Pa)	specific internal energy u (J/kg)
`get_param_coeff_alpha`	temperature T (K) pressure P (Pa)	cubic expansion coefficient $\alpha = \frac{1}{v}\left.\frac{\partial v}{\partial T}\right\|_p$ (K^{-1})
`get_param_coeff_beta`	temperature T (K) pressure P (Pa)	pressure coefficient $\beta = \left.\frac{\partial p}{\partial T}\right\|_v$ (Pa/K)
`get_param_comp_kappa_T`	temperature T (K) pressure P (Pa)	isothermal compressibility $\kappa_T = -\frac{1}{v}\left.\frac{\partial v}{\partial p}\right\|_T$ (Pa^{-1})
`get_param_comp_kappa_s`	temperature T (K) pressure P (Pa)	isentropic compressibility $\kappa_s = -\frac{1}{v}\left.\frac{\partial v}{\partial p}\right\|_s$ (Pa^{-1})

Table 5: Library functions for single-phase thermodynamic quantities in `IAPWS::Lib06` class

function	input	output
`make_tab_coex`	pointer to `IAPWS::Lib95` object (`IAPWS::Lib95 *ptr`) number of temperature bins (`int`) minimum temperature (K)	`void`
`export_tab_coex`	name of output file (`char *`)	`void`
`import_tab_coex`	name of input file (`char *`)	`void`

Table 6: Tabulation of coexisting phases (sublimation curve between vapor and ice) in `IAPWS::Lib06` class

The isochoric heat capacity c_v is given by the partial derivative of u with respect to T.

$$c_v = \left.\frac{\partial u}{\partial T}\right|_\rho = -R\,\tau^2\phi_{\tau\tau} \quad \text{where} \quad \phi_{\tau\tau} = \frac{\partial^2\phi}{\partial\tau^2} \tag{6}$$

The isobaric heat capacity c_p is defined as the partial derivative of h with respect to T, while the pressure is kept constant.

$$\begin{aligned}
c_p &= \left.\frac{\partial h}{\partial T}\right|_p \\
&= \lim_{\Delta T\to 0}\frac{h(\rho+\Delta\rho, T+\Delta T) - h(\rho, T)}{\Delta T} \quad \text{where} \quad \left.\frac{\partial p}{\partial \rho}\right|_T \Delta\rho + \left.\frac{\partial p}{\partial T}\right|_\rho \Delta T = 0\,. && (7) \\
&= \left.\frac{\partial h}{\partial T}\right|_\rho - \left.\frac{\partial h}{\partial \rho}\right|_T \frac{\partial p/\partial T|_\rho}{\partial p/\partial \rho|_T} && (8) \\
&= c_v + \frac{1}{\rho}\left.\frac{\partial p}{\partial T}\right|_\rho - \left(\left.\frac{\partial u}{\partial \rho}\right|_T + \frac{1}{\rho}\left.\frac{\partial p}{\partial \rho}\right|_T - \frac{p}{\rho^2}\right)\frac{\partial p/\partial T|_\rho}{\partial p/\partial \rho|_T} = c_v + \frac{1}{\rho}\left(\frac{p}{\rho} - \rho\left.\frac{\partial u}{\partial \rho}\right|_T\right)\frac{\partial p/\partial T|_\rho}{\partial p/\partial \rho|_T} && (9)
\end{aligned}$$

function	input	output
`get_coex_pressure`	temperature T (K)	sublimation pressure p_{coex} (Pa)
`get_coex_mden_vap`	temperature T (K)	mass density of vapor ρ_{vap} (kg/m^3)
`get_coex_mden_ice`	temperature T (K)	mass density of ice ρ_{ice} (kg/m^3)
`get_coex_enthalpy_vap`	temperature T (K)	specific enthalpy of vapor h_{vap} (J/kg)
`get_coex_enthalpy_ice`	temperature T (K)	specific enthalpy of ice h_{ice} (J/kg)
`get_coex_entropy_vap`	temperature T (K)	specific entropy of vapor s_{vap} (J/kg K)
`get_coex_entropy_ice`	temperature T (K)	specific entropy of ice s_{ice} (J/kg K)
`get_coex_heat_latent`	temperature T (K)	specific latent heat $h_{\text{latent;vap-ice}}$ (J/kg)

Table 7: Library functions for coexisting phases (sublimation curve between vapor and ice) in `IAPWS::Lib06` class

function	input	output
`make_tab_melt`	pointer to `IAPWS::Lib95` object (`IAPWS::Lib95 *ptr`) number of pressure bins (`int`) maximum pressure (Pa)	`void`
`export_tab_melt`	name of output file (`char *`)	`void`
`import_tab_melt`	name of input file (`char *`)	`void`

Table 8: Tabulation of coexisting phases (melting curve between liquid and ice) in `IAPWS::Lib06` class

Derivatives of p and u can be written in terms of ϕ as following.

$$\rho \left.\frac{\partial u}{\partial \rho}\right|_T = RT\,\delta\tau\phi_{\delta\tau} \quad \text{where} \quad \phi_{\delta\tau} = \frac{\partial^2 \phi}{\partial\delta\,\partial\tau} \tag{10}$$

$$\frac{1}{\rho}\left.\frac{\partial p}{\partial T}\right|_\rho = R\left(\delta\phi_\delta - \delta\tau\phi_{\delta\tau}\right) \tag{11}$$

$$\left.\frac{\partial p}{\partial \rho}\right|_T = RT\left(2\,\delta\phi_\delta + \delta^2\phi_{\delta\delta}\right) \quad \text{where} \quad \phi_{\delta\delta} = \frac{\partial^2\phi}{\partial\delta^2} \tag{12}$$

which lead one to obtain

$$c_p = c_v + R\,\frac{\left(\delta\phi_\delta - \delta\tau\phi_{\delta\tau}\right)^2}{2\,\delta\phi_\delta + \delta^2\phi_{\delta\delta}}\,. \tag{13}$$

The speed of sound w can be computed by taking derivative of p with respect to ρ, while the specific entropy is kept constant.

$$\begin{aligned} w^2 &= \left.\frac{\partial p}{\partial \rho}\right|_s \\ &= \lim_{\Delta\rho\to 0}\frac{p(\rho+\Delta\rho, T+\Delta T) - p(\rho, T)}{\Delta\rho} \quad \text{where} \quad \left.\frac{\partial s}{\partial\rho}\right|_T \Delta\rho + \left.\frac{\partial s}{\partial T}\right|_\rho \Delta T = 0 \qquad (14) \\ &= \left.\frac{\partial p}{\partial\rho}\right|_T - \frac{1}{\rho}\left.\frac{\partial p}{\partial T}\right|_\rho \frac{\rho\,\partial s/\partial\rho|_T}{\partial s/\partial T|_\rho} \qquad (15) \end{aligned}$$

Derivatives of s can be written in terms of ϕ as following.

$$\rho\left.\frac{\partial s}{\partial\rho}\right|_T = -R\,\left(\delta\phi_\delta - \delta\tau\phi_{\delta\tau}\right) \tag{16}$$

$$\left.\frac{\partial s}{\partial T}\right|_\rho = -\frac{R}{T}\,\tau^2\phi_{\tau\tau} \tag{17}$$

which result in the following expression for w.

$$w^2 = RT\left[2\,\delta\phi_\delta + \delta^2\phi_{\delta\delta} - \frac{\left(\delta\phi_\delta - \delta\tau\phi_{\delta\tau}\right)^2}{\tau^2\phi_{\tau\tau}}\right] \tag{18}$$

function	input	output
get_melt_temperature	pressure p (Pa)	melting temperature T_{melt} (K)
get_melt_mden_liq	pressure p (Pa)	mass density of liquid ρ_{liq} (kg/m^3)
get_melt_mden_ice	pressure p (Pa)	mass density of ice ρ_{ice} (kg/m^3)
get_melt_enthalpy_liq	pressure p (Pa)	specific enthalpy of liquid h_{liq} (J/kg)
get_melt_enthalpy_ice	pressure p (Pa)	specific enthalpy of ice h_{ice} (J/kg)
get_melt_entropy_liq	pressure p (Pa)	specific entropy of liquid s_{liq} (J/kg K)
get_melt_entropy_ice	pressure p (Pa)	specific entropy of ice s_{ice} (J/kg K)
get_melt_heat_latent	pressure p (Pa)	specific latent heat $h_{\mathrm{latent;liq\text{-}ice}}$ (J/kg)

Table 9: Library functions for coexisting phases (melting curve between liquid and ice) in `IAPWS::Lib06` class

function	input	output
set_ptr_lib95	pointer to an instance of `Lib95` class (`IAPWS::Lib95 *ptr`)	`void`
get_param_viscosity	mass density ρ (kg/m^3) temperature T (K)	shear viscosity μ (Pa $\cdot$ sec)

Table 10: Library functions for the shear viscosity in `IAPWS::Lib08V` class

2.2 From the Gibbs free energy $g(T, p)$

If one has pressure p and temperature T as independent variables, thermodynamic quantities can be derived from the specific Gibbs free energy g

$$\frac{g(T,p)}{RT} = \gamma\left(\tau \equiv \frac{T_*}{T}, \Pi \equiv \frac{p}{p_*}\right) \tag{19}$$

where we have a dimensionless quantity Π defined in terms of the reference pressure p_*. One can obtain the specific entropy s and specific volume v from first derivatives of the Gibbs free energy.

$$s = -\left.\frac{\partial g}{\partial T}\right|_p = R\left(\tau\gamma_\tau - \gamma\right) \quad \text{where} \quad \gamma_\tau \equiv \frac{\partial\gamma}{\partial\tau} \tag{20}$$

$$v = \left.\frac{\partial g}{\partial p}\right|_T = \frac{RT}{p}\,\Pi\gamma_\Pi \quad \text{where} \quad \gamma_\Pi \equiv \frac{\partial\gamma}{\partial\Pi} \tag{21}$$

which lead to the following expressions for the specific enthalpy and internal energy.

$$h = g + Ts = RT\,\tau\gamma_\tau \tag{22}$$

$$u = h - pv = RT\left(\tau\gamma_\tau - \Pi\gamma_\Pi\right)\,. \tag{23}$$

The isobaric heat capacity c_p is given by

$$c_p = \left.\frac{\partial h}{\partial T}\right|_p = -R\,\tau^2\gamma_{\tau\tau} \quad \text{where} \quad \gamma_{\tau\tau} = \frac{\partial^2\gamma}{\partial\tau^2}\,. \tag{24}$$

The isochoric heat capacity c_v is given by the partial derivative of u with respect to T, while the specific volume is kept constant.

$$\begin{aligned} c_v &= \left.\frac{\partial u}{\partial T}\right|_v \\ &= \lim_{\Delta T\to 0}\frac{u(T+\Delta T, p+\Delta p) - u(T,p)}{\Delta T} \quad \text{where} \quad \left.\frac{\partial v}{\partial T}\right|_p \Delta T + \left.\frac{\partial v}{\partial p}\right|_T \Delta p = 0\,. &(25) \\ &= \left.\frac{\partial u}{\partial T}\right|_p - \left.\frac{\partial u}{\partial p}\right|_T \frac{\partial v/\partial T|_p}{\partial v/\partial p|_T} &(26) \\ &= c_p - p\left.\frac{\partial v}{\partial T}\right|_p - \left(\left.\frac{\partial h}{\partial p}\right|_T - p\left.\frac{\partial v}{\partial p}\right|_T - v\right)\frac{\partial v/\partial T|_p}{\partial v/\partial p|_T} = c_p + \left(pv - p\left.\frac{\partial h}{\partial p}\right|_T\right)\frac{\partial v/\partial T|_p}{p\,\partial v/\partial p|_T} &(27) \end{aligned}$$

Derivatives of h and v can be written in terms of γ as following.

$$
\begin{aligned}
p \left.\frac{\partial h}{\partial p}\right|_T &= RT\,\Pi\tau\gamma_{\Pi\tau} \quad \text{where} \quad \gamma_{\Pi\tau} = \frac{\partial^2\gamma}{\partial\Pi\,\partial\tau} && (28)\\
\left.\frac{\partial v}{\partial T}\right|_p &= \frac{R}{p}\,\Pi\left(\gamma_\Pi - \tau\gamma_{\Pi\tau}\right) && (29)\\
p \left.\frac{\partial v}{\partial p}\right|_T &= \frac{RT}{p}\,\Pi^2\gamma_{\Pi\Pi} \quad \text{where} \quad \gamma_{\Pi\Pi} = \frac{\partial^2\gamma}{\partial\Pi^2} && (30)
\end{aligned}
$$

which result in the following expression for c_v.

$$
c_v = c_p + R\,\frac{\left(\gamma_\Pi - \tau\gamma_{\Pi\tau}\right)^2}{\gamma_{\Pi\Pi}} \tag{31}
$$

The speed of sound can be computed in a similar manner as one does with the Helmholtz free energy.

$$
\begin{aligned}
w^2 &= \left.\frac{\partial p}{\partial \rho}\right|_s \\
&= -v^2\left[\lim_{\Delta p\to 0}\frac{v(T+\Delta T, p+\Delta p) - v(T,p)}{\Delta p}\right]^{-1} \quad \text{where} \quad \left.\frac{\partial s}{\partial T}\right|_p \Delta T + \left.\frac{\partial s}{\partial p}\right|_T \Delta p = 0 && (32)\\
&= -v^2\left(\left.\frac{\partial v}{\partial p}\right|_T - \left.\frac{\partial v}{\partial T}\right|_p \frac{\partial s/\partial p|_T}{\partial s/\partial T|_p}\right)^{-1} = (pv)^2\left(pT\left.\frac{\partial v}{\partial T}\right|_p \frac{p\,\partial s/\partial p|_T}{T\,\partial s/\partial T|_p} - p^2\left.\frac{\partial v}{\partial p}\right|_T\right)^{-1} && (33)
\end{aligned}
$$

Derivatives of s can be written in terms of γ as following.

$$
\begin{aligned}
p\left.\frac{\partial s}{\partial p}\right|_T &= -R\,\Pi\left(\gamma_\Pi - \tau\gamma_{\Pi\tau}\right) && (34)\\
T\left.\frac{\partial s}{\partial T}\right|_p &= -R\,\tau^2\gamma_{\tau\tau} && (35)
\end{aligned}
$$

which lead to the following expression for the speed of sound.

$$
w^2 = RT\left[\frac{\gamma_\Pi^2}{(\gamma_\Pi - \tau\gamma_{\Pi\tau})^2/(\tau^2\gamma_{\tau\tau}) - \gamma_{\Pi\Pi}}\right] \tag{36}
$$

3 Phase diagram from R6-95 and R10-06

IAPWS R6-95 [3] release provides parametrization for the specific Helmholtz free energy f of water vapor and liquid for wide range of temperature T and mass density ρ. Thermodynamic quantities, such as pressure, internal energy, enthalpy, entropy and heat capacity, can be obtained from derivatives of the Helmholtz free energy. When a system undergoes first-order phase transition between water vapor and liquid, the saturation pressure p_{coex} remains constant while the mass density changes between ρ_{vap} and ρ_{liq}. Such coexisting phase can be determined by applying the Maxwell construction.

$$
\begin{aligned}
p_{\text{coex}}(T) &= p(\rho_{\text{vap}}, T) = p(\rho_{\text{liq}}, T) && (37)\\
p(\rho, T) &= -\left.\frac{\partial f}{\partial v}\right|_T = \rho^2\left.\frac{\partial f}{\partial \rho}\right|_T && (38)\\
f(\rho_{\text{liq}}, T) - f(\rho_{\text{vap}}, T) &= p_{\text{coex}}\left(v_{\text{vap}} - v_{\text{liq}}\right)\\
&= p_{\text{coex}}\left(\frac{1}{\rho_{\text{vap}}} - \frac{1}{\rho_{\text{liq}}}\right) \quad \text{where} \quad \text{specific volume } v = \frac{1}{\rho}\,. && (39)
\end{aligned}
$$

The specific latent heat h_{latent}, which is amount of heat required to transform unit mass of liquid water into vapor in this case, is given by difference in specific enthalpy or entropy.

$$
h_{\text{latent;vap-liq}}(T) = h(\rho_{\text{vap}}, T) - h(\rho_{\text{liq}}, T) = T\left[s(\rho_{\text{vap}}, T) - s(\rho_{\text{liq}}, T)\right] \tag{40}
$$

One can demonstrate that equation (40) is equivalent to what is given by the Clapeyron equation. Let us consider infinitesimally small deviations $\Delta\rho_{\rm vap}$, $\Delta\rho_{\rm liq}$ and ΔT in mass densities and temperature, respectively. Then left- and right-hand sides (LHS and RHS) of equation (39) become

$$\begin{aligned}
\text{LHS} &= f(\rho_{\rm liq}+\Delta\rho_{\rm liq}, T+\Delta T) - f(\rho_{\rm vap}+\Delta\rho_{\rm vap}, T+\Delta T) \\
&= f(\rho_{\rm liq}, T) - f(\rho_{\rm vap}, T) \\
&\quad +\Delta T\left[\left.\frac{\partial f}{\partial T}\right|_{\rho}(\rho_{\rm liq},T) - \left.\frac{\partial f}{\partial T}\right|_{\rho}(\rho_{\rm vap},T)\right] + \Delta\rho_{\rm liq}\left.\frac{\partial f}{\partial \rho}\right|_{T}(\rho_{\rm liq},T) - \Delta\rho_{\rm vap}\left.\frac{\partial f}{\partial \rho}\right|_{T}(\rho_{\rm vap},T) \\
&= f(\rho_{\rm liq}, T) - f(\rho_{\rm vap}, T) + \Delta T\left[s(\rho_{\rm vap},T) - s(\rho_{\rm liq},T)\right] - p_{\rm coex}\left(\frac{\Delta\rho_{\rm vap}}{\rho_{\rm vap}^2} - \frac{\Delta\rho_{\rm liq}}{\rho_{\rm liq}^2}\right) \qquad (41) \\
\text{RHS} &= p_{\rm coex}\left(\frac{1}{\rho_{\rm vap}} - \frac{1}{\rho_{\rm liq}}\right) + \Delta T\,\frac{dp_{\rm coex}}{dT}\left(\frac{1}{\rho_{\rm vap}} - \frac{1}{\rho_{\rm liq}}\right) - p_{\rm coex}\left(\frac{\Delta\rho_{\rm vap}}{\rho_{\rm vap}^2} - \frac{\Delta\rho_{\rm liq}}{\rho_{\rm liq}^2}\right) \qquad (42)
\end{aligned}$$

By equating LHS and RHS, one obtains the Clapeyron equation for the latent heat.

$$h_{\text{latent;vap-liq}}(T) = \left(\frac{1}{\rho_{\rm vap}} - \frac{1}{\rho_{\rm liq}}\right) T\,\frac{dp_{\rm coex}}{dT} \qquad (43)$$

Once `LibIAPWS` library is built, one can run `make_tab_IAPWS95.exec` executable to obtain the coexistence curve (saturation curve) between water vapor and liquid. It produces a text file `tab_coex_IAPWS95.txt` which contains the tabulated coexistence curve as functions of temperature.

The phase diagram can be extended to incorporate the ice phase, by implementing IAPWS R10-06 [5] release. The specific Gibbs free energy g is parametrized as function of temperature T and pressure p, and coexistence curve can be obtained from continuity condition of the Gibbs free energy. The specific Gibbs free energy $g_{\rm fluid}$ for fluid (vapor or liquid) can be obtained from the parametrization in R6-95 release and it must match to $g_{\rm ice}$ provided in R10-06 release.

$$\begin{aligned}
g_{\rm ice}(T, p_{\rm coex}) &= g_{\rm fluid}(\rho_{\rm fluid}, T) \\
&= f_{\rm fluid}(\rho_{\rm fluid}, T) + \frac{p_{\rm coex}}{\rho_{\rm fluid}} \qquad (44)
\end{aligned}$$

For given temperature and pressure, mass density $\rho_{\rm fluid}$ of fluid is found by equation (38) and used to compute $g_{\rm fluid}$. The specific latent heat between fluid and ice can be found in a similar manner as equation (40).

$$h_{\text{latent;fluid-ice}}(T) = h_{\rm fluid}(\rho_{\rm fluid}, T) - h_{\rm ice}(T, p_{\rm coex}) = T\left[s_{\rm fluid}(\rho_{\rm fluid}, T) - s_{\rm ice}(T, p_{\rm coex})\right] \qquad (45)$$

After running `make_tab_IAPWS95.exec` to tabulate the saturation curve, one can run `make_tab_IAPWS06.exec` executable to obtain the coexistence curve between water fluid and ice. It produces two text file `tab_coex_IAPWS06.txt` and `tab_melt_IAPWS06.txt`. `tab_coex_IAPWS06.txt` provides the coexistence curve (sublimation curve) between water vapor and ice, while `tab_melt_IAPWS06.txt` contains the coexistence curve (melting curve) between water liquid and ice. The resulting phase diagram on the T-p plane is presented in Figure 1.

References

[1] `https://iapws.org`

[2] IAPWS, *Revised Release on Surface Tension of Ordinary Water Substance* (2014).

[3] IAPWS, *Revised Release on the IAPWS Formulation 1995 for the Thermodynamic Properties of Ordinary Water Substance for General and Scientific Use* (2018).

[4] IAPWS, *Revised Release on the IAPWS Industrial Formulation 1997 for the Thermodynamic Properties of Water and Steam* (2012).

[5] IAPWS, *Revised Release on the Equation of State 2006 for H_2O Ice Ih* (2009).

[6] IAPWS, *Release on the IAPWS Formulation 2008 for the Viscosity of Ordinary Water Substance* (2008).

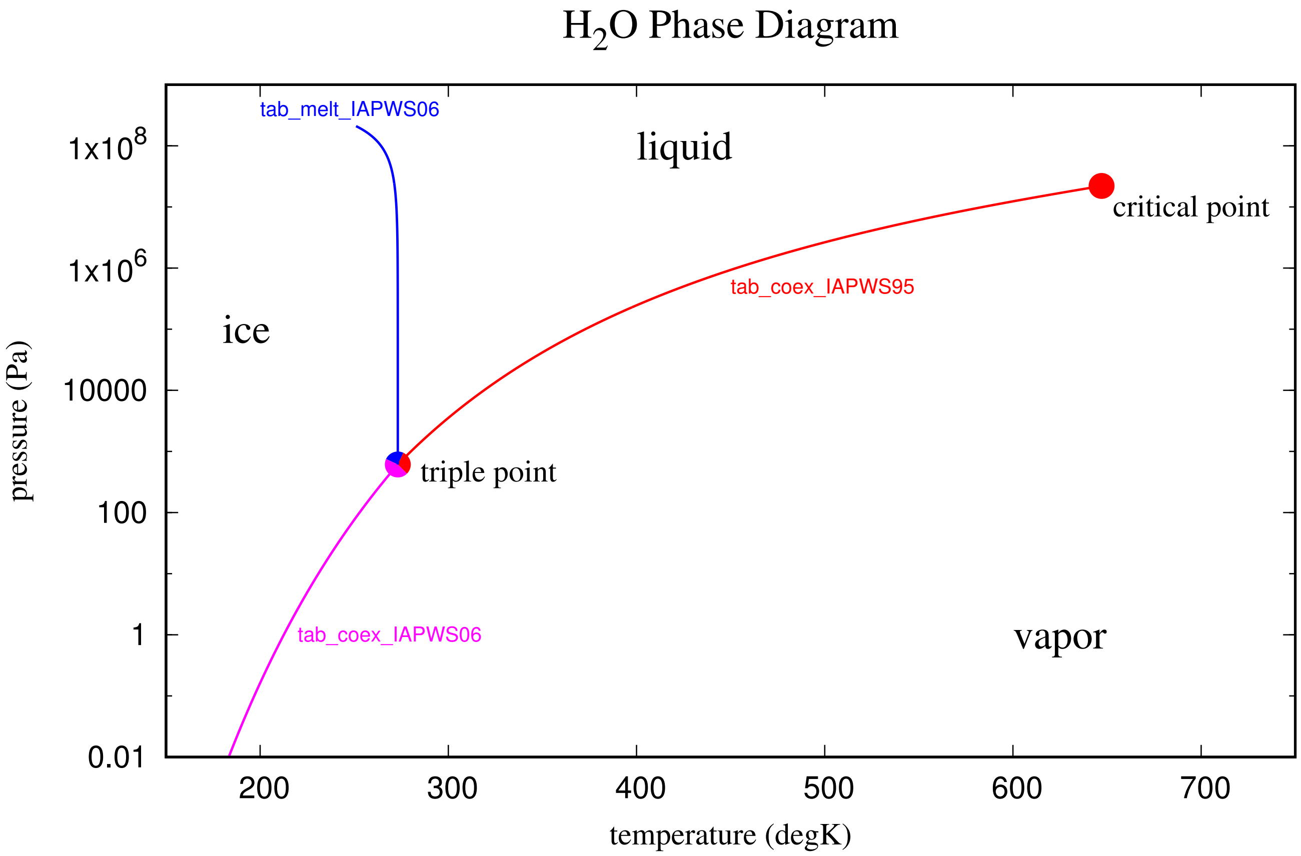

Figure 1: Phase diagram of H_2O derived from IAPWS R6-95 and R10-06 releases.