ThermoFluidProperties

Property Library for Humid Air Calculated as Ideal Mixture of Real Fluids

LibHuAir_Xiw

Prof. Hans-Joachim Kretzschmar
Dr. Sebastian Herrmann
Prof. Matthias Kunick

Property Functions

Functional Dependence	Function Name	Call from Fortran Program	Property or Function	Unit	Source or Algorithm	Information on page
$a = f(p,t,\xi_w)$	a_ptXiw_HuAir	= A_PTXIW_HUAIR(p,t,Xiw,succ)	Thermal diffusivity	m^2/s	[1-4], [6], [12], [14], [15]	3/2
$c_p = f(h,s,\xi_w)$	cp_hsXiw_HuAir	= CP_HSXIW_HUAIR(h,s,Xiw,succ)	Backward function: Isobaric heat capacity from enthalpy and entropy	$kJ/(kg \cdot K)$	[1-4], [13], [14]	3/3
$c_p = f(p,h,\xi_w)$	cp_phXiw_HuAir	= CP_PHXIW_HUAIR(p,h,Xiw,succ)	Backward function: Isobaric heat capacity from pressure and enthalpy	$kJ/(kg \cdot K)$	[1-4], [13], [14]	3/4
$c_p = f(p,s,\xi_w)$	cp_psXiw_HuAir	= CP_PSXIW_HUAIR(p,s,Xiw,succ)	Backward function: Isobaric heat capacity from pressure and entropy	$kJ/(kg \cdot K)$	[1-4], [13], [14]	3/5
$c_p = f(p,t,\xi_w)$	cp_ptXiw_HuAir	= CP_PTXIW_HUAIR(p,t,Xiw,succ)	Specific isobaric heat capacity	$kJ/(kg \cdot K)$	[1-4], [13], [14]	3/6
$c_p = f(t,s,\xi_w)$	cp_tsXiw_HuAir	= CP_TSXIW_HUAIR(t,s,Xiw,succ)	Backward function: Specific isobaric heat capacity from temperature and entropy	$kJ/(kg \cdot K)$	[1-4], [13], [14]	3/7
$c_v = f(p,t,\xi_w)$	cv_ptXiw_HuAir	= CV_PTXIW_HUAIR(p,t,Xiw,succ)	Specific isochoric heat capacity	$kJ/(kg \cdot K)$	[1-4], [13], [14]	3/8
$\eta = f(p,t,\xi_w)$	Eta_ptXiw_HuAir	= ETA_PTXIW_HUAIR(p,t,Xiw,succ)	Dynamic viscosity	$Pa \cdot s$	[7], [12], [15]	3/9
$h = f(p,s,\xi_w)$	h_psXiw_HuAir	= H_PSXIW_HUAIR(p,s,Xiw,succ)	Backward function: Specific enthalpy from pressure and entropy	kJ/kg	[1-4], [13], [14], [18], [19]	3/10
$h = f(p,t,\xi_w)$	h_ptXiw_HuAir	= H_PTXIW_HUAIR(p,t,Xiw,succ)	Specific enthalpy	kJ/kg	[1-4], [13], [14], [18], [19]	3/11
$h = f(t,s,\xi_w)$	h_tsXiw_HuAir	= H_TSXIW_HUAIR(t,s,Xiw,succ)	Backward function: Specific enthalpy from temperature and entropy	kJ/kg	[1-4], [13], [14], [18], [19]	3/12

Functional Dependence	Function Name	Call from Fortran Program	Property or Function	Unit	Source or Algorithm	Information on page
$\kappa = f(p,s,\xi_w)$	Kappa_psXiw_HuAir	= KAPPA_PSXIW_HUAIR(p,s,Xiw,succ)	Backward function: Isentropic exponent from pressure and entropy	-	[1-4], [13], [14]	3/13
$\kappa = f(p,t,\xi_w)$	Kappa_ptXiw_HuAir	= KAPPA_PTXIW_HUAIR(p,t,Xiw,succ)	Isentropic exponent	-	[1-4], [13], [14]	3/14
$\lambda = f(p,t,\xi_w)$	Lambda_ptXiw_HuAir	= LAMBDA_PTXIW_HUAIR(p,t,Xiw,succ)	Thermal conductivity	$W/(m \cdot K)$	[6], [12], [15]	3/15
$\nu = f(p,t,\xi_w)$	Ny_ptXiw_HuAir	= NY_PTXIW_HUAIR(p,t,Xiw,succ)	Kinematic viscosity	m^2/s	[1-4], [7], [12], [14], [15]	3/16
$p = f(h,s,\xi_w)$	p_hsXiw_HuAir	= P_HSXIW_HUAIR(h,s,Xiw,succ)	Backward function: Pressure from enthalpy and entropy	bar	[1-4], [13], [14], [18], [19]	3/17
$p = f(t,s,\xi_w)$	p_tsXiw_HuAir	= P_TSXIW_HUAIR(t,s,Xiw,succ)	Backward function: Pressure from temperature and entropy	bar	[1-4], [13], [14], [18], [19]	3/18
$p_d = f(p,t,\xi_w)$	pd_ptXiw_HuAir	= PD_PTXIW_HUAIR(p,t,Xiw,succ)	Partial pressure of steam	bar	[1-4], [16], [17], [25], [26]	3/19
$p_{dsatt} = f(p,t)$	pdsatt_pt_HuAir	= PDSATT_PT_HUAIR(p,t,succ)	Saturation vapor pressure of water	bar	[1-4], [16], [17], [25], [26]	3/20
$\varphi = f(p,t,\xi_w)$	Phi_ptXiw_HuAir	= PHI_PTXIW_HUAIR(p,t,Xiw,succ)	Relative humidity	-	[1-4], [16], [17], [25], [26]	3/21
$p_l = f(p,t,\xi_w)$	pl_ptXiw_HuAir	= PL_PTXIW_HUAIR(p,t,Xiw,succ)	Partial pressure of air	bar	[1-4], [16], [17], [25], [26]	3/22
$Pr = f(p,t,\xi_w)$	Pr_ptXiw_HuAir	= PR_PTXIW_HUAIR(p,t,Xiw,succ)	PRANDTL-Number	-	[1-4], [6], [7], [12-15]	3/23
$\psi_l = f(\xi_w)$	Psil_Xiw_HuAir	= PSIL_XIW_HUAIR(Xiw,succ)	Mole fraction of air	kmol/kmol		3/24
$\psi_w = f(\xi_w)$	Psiw_Xiw_HuAir	= PSIW_XIW_HUAIR(Xiw,succ)	Mole fraction of water	kmol/kmol		3/25
$Region = f(h,s,\xi_w)$	Region_hsXiw_HuAir	= REGION_HSXIW_HUAIR(p,h,Xiw)	Region of state from enthalpy and entropy	-	[1-4], [14], [18], [19]	3/26
$Region = f(p,h,\xi_w)$	Region_phXiw_HuAir	= REGION_PHXIW_HUAIR(p,h,Xiw)	Region of state from pressure and enthalpy	-	[1-4], [14], [18], [19]	3/27

Functional Dependence	Function Name	Call from Fortran Program	Property or Function	Unit	Source or Algorithm	Information on page
$Region = \mathrm{f}(p,s,\xi_\mathrm{w})$	Region_psXiw_HuAir	= REGION_PSXIW_HUAIR(p,s,Xiw)	Region of state from pressure and entropy	-	[1-4], [14], [18], [19]	3/28
$Region = \mathrm{f}(p,T,\xi_\mathrm{w})$	Region_ptXiw_HuAir	= REGION_PTXIW_HUAIR(p,t,Xiw)	Region of state from pressure and temperature	-	[1-4], [14], [18], [19]	3/29
$Region = \mathrm{f}(t,s,\xi_\mathrm{w})$	Region_tsXiw_HuAir	= REGION_TSXIW_HUAIR(t,s,Xiw)	Region of state from temperature and entropy	-	[1-4], [14], [18], [19]	3/30
$\rho = \mathrm{f}(p,t,\xi_\mathrm{w})$	Rho_ptXiw_HuAir	= RHO_PTXIW_HUAIR(p,t,Xiw,succ)	Density	$\mathrm{kg/m^3}$	[1-4], [14], [18], [19]	3/31
$s = \mathrm{f}(p,h,\xi_\mathrm{w})$	s_phXiw_HuAir	= S_PHXIW_HUAIR(p,h,Xiw,succ)	Backward function: Entropy from pressure and enthalpy	$\mathrm{kJ/(kg \cdot K)}$	[1-4], [13], [14], [18], [19]	3/32
$s = \mathrm{f}(p,t,\xi_\mathrm{w})$	s_ptXiw_HuAir	= S_PTXIW_HUAIR(p,t,Xiw,succ)	Specific entropy	$\mathrm{kJ/(kg \cdot K)}$	[1-4], [13], [14], [18], [19]	3/33
$\sigma = \mathrm{f}(t)$	Sigma_t_HuAir	= SIGMA_T_HUAIR (t,succ)	Surface tension of water	N/m	[8]	3/34
$t = \mathrm{f}(h,s,\xi_\mathrm{w})$	t_hsXiw_HuAir	= T_HSXIW_HUAIR(h,s,Xiw,succ)	Backward function: Temperature from enthalpy and entropy	°C	[1-4], [13], [14], [18], [19]	3/35
$t = \mathrm{f}(p,h,\xi_\mathrm{w})$	t_phXiw_HuAir	= T_PHXIW_HUAIR(p,h,Xiw,succ)	Backward function: Temperature from pressure and enthalpy	°C	[1-4], [13], [14], [18], [19]	3/36
$t = \mathrm{f}(p,s,\xi_\mathrm{w})$	t_psXiw_HuAir	= T_PSXIW_HUAIR(p,s,Xiw,succ)	Backward function: Temperature from pressure and entropy	°C	[1-4], [13], [14], [18], [19]	3/37
$t = \mathrm{f}(p,t_\mathrm{f},\xi_\mathrm{w})$	t_ptfXiw_HuAir	= T_PTFXIW_HUAIR(p,tf,Xiw,succ)	Temperature	°C	[1-4], [13], [22]	3/38
$t_\mathrm{f} = \mathrm{f}(p,t,\xi_\mathrm{w})$	tf_ptXiw_HuAir	= TF_PTXIW_HUAIR(p,t,Xiw,succ)	Wet bulb temperature	°C	[1-4], [13], [22]	3/39
$t_\tau = \mathrm{f}(p,\xi_\mathrm{w})$	tTau_pXiw_HuAir	= TTAU_PXIW_HUAIR(p,Xiw,succ)	Dew point temperature	°C	[1-4], [16], [17]	3/40
$u = \mathrm{f}(p,t,\xi_\mathrm{w})$	u_ptXiw_HuAir	= U_PTXIW_HUAIR(p,t,Xiw,succ)	Internal energy	kJ/kg	[1-4], [13], [14], [18], [19]	3/41
$v = \mathrm{f}(h,s,\xi_\mathrm{w})$	v_hsXiw_HuAir	= V_HSXIW_HUAIR(h,s,Xiw,succ)	Backward function: Specific volume from enthalpy and entropy	$\mathrm{m^3/kg}$	[1-4], [13], [14], [18], [19]	3/42

Functional Dependence	Function Name	Call from Fortran Program	Property or Function	Unit	Source or Algorithm	Information on page
$v = f(p,h,\xi_w)$	v_phXiw_HuAir	= V_PHXIW_HUAIR(p,h,Xiw,succ)	Backward function: Specific volume from pressure and enthalpy	m^3/kg	[1-4], [13], [14], [18], [19]	3/43
$v = f(p,s,\xi_w)$	v_psXiw_HuAir	= V_PSXIW_HUAIR(p,s,Xiw,succ)	Backward function: Specific volume from pressure and entropy	m^3/kg	[1-4], [13], [14], [18], [19]	3/44
$v = f(p,t,\xi_w)$	v_ptXiw_HuAir	= V_PTXIW_HUAIR(p,t,Xiw,succ)	Specific volume	m^3/kg	[1-4], [14], [18], [19]	3/45
$v = f(t,s,\xi_w)$	v_tsXiw_HuAir	= V_TSXIW_HUAIR(t,s,Xiw,succ)	Backward function: Specific volume from temperature and entropy	m^3/kg	[1-4], [13], [14], [18], [19]	3/46
$w = f(p,t,\xi_w)$	w_ptXiw_HuAir	= W_PTXIW_HUAIR(p,t,Xiw,succ)	Isentropic speed of sound	m/s	[1-4], [13], [14]	3/47
$\xi_w = f(p,t,p_d)$	Xiw_ptpd_HuAir	= XIW_PTPD_HUAIR(p,t,pd,succ)	Mass fraction of water from partial pressure of steam	kg/kg	[1-4], [16], [17], [25], [26]	3/48
$\xi_w = f(p,t,\varphi)$	Xiw_ptPhi_HuAir	= XIW_PTPHI_HUAIR(p,t,Phi,succ)	Mass fraction of water from temperature and relative humidity	kg/kg	[1-4], [16], [17], [25], [26]	3/49
$\xi_w = f(p,t_\tau)$	Xiw_ptTau_HuAir	= XIW_PTTAU_HUAIR(p,tTau,succ)	Mass fraction of water from dew point temperature	kg/kg	[1-4], [16], [17], [25], [26]	3/50
$\xi_w = f(p,t,t_f)$	Xiw_pttf_HuAir	= XIW_PTTF_HUAIR(p,t,tf,succ)	Mass fraction of steam from temperature and wet bulb temperature	kg/kg	[1-4], [13], [14]	3/51
$\xi_{wf} = f(p,t,\xi_w)$	Xiwf_ptXiw_HuAir	= XIWF_PTXIW_HUAIR (p,t,Xiw,succ)	Mass fraction of liquid water	kg/kg	[1-4], [16], [17], [25], [26]	3/52
$\xi_{wsatt} = f(p,t)$	Xiwsatt_pt_HuAir	= XIWSATT_PT_HUAIR(p,t,succ)	Mass fraction steam of saturated air	kg/kg	[1-4], [16], [17], [25], [26]	3/53
$x_w = f(\xi_w)$	xw_Xiw_HuAir	= XW_XIW_HUAIR(Xiw,succ)	Humidity ratio (absolute humidity) from mass fraction of water	kg/kg_{Air}		3/54

Types of variables for function calls

All functions, except Region_...	REAL*8
All variables, except succ	REAL*8
Region_... , succ	INTEGER*4

Definition of the output value "succ":

succ	Meaning
0	Calculation not successful
1	Calculation successful

Definition of the region of state "Region":

Region	Meaning
0	Outside range of validity
1	Dry air
2	Unsaturated humid air
3	Liquid fog
4	Ice fog
5	Mixture of liquid fog and ice fog at 0 °C exactly
6	Pure water

Reference states:

Factor	Dry air	Water
Pressure	1.01325 bar	611.657 Pa
Temperature	0 °C	273.16 K
Enthalpy	0 kJ/kg	0.611783 J/kg
Internal energy	-78.37885533 kJ/kg	0 J/kg
Entropy	0.161802887 kJ/(kg K)	0 J/(kg K)

Composition of dry air (from *Lemmon* et al. [22], [23]):

Component		Mole fraction
Nitrogen	N_2	0.7812
Oxygen	O_2	0.2096
Argon	Ar	0.0092

Parameters

p - Total pressure in bar
t - Temperature in °C
Xiw - Mass fraction of water in kg water(steam)/kg humid air
succ - Output parameter: succ = 1 if calculation successful, or else succ = 0

Range of validity

Temperature t = - 143.15 °C ... 1726.85 °C
Pressure p = 6.112 mbar ... 1000 bar

Calculation algorithms

Unsaturated and saturated humid air $(0 \le Xi_w \le Xi_{ws})$:

Ideal mixture of dry air and steam

Dry air:

- v, h, u, s, c_p, c_v, κ, w from *Lemmon* et al. [14]
- λ, η from *Lemmon* et al. [15]

Steam:

- v, h, u, s, c_p, c_v, κ, w of steam from IAPWS-IF97 [1], [2], [3], [4]
- λ, η for $0°C \le t \le 800°C$ from IAPWS-85 [6], [7] (Mixture of volume fractions)
  for t < 0 °C and t > 800 °C from *Brandt* [12] (Mixture of volume fractions)

Supersaturated humid air (liquid fog or ice fog)

Liquid fog $(Xi_w > Xi_{wsatt})$ and $t \ge 0°C$

Ideal mixture of saturated humid air and water liquid

- saturated humid air as specified above
- v, h, u, s, κ, w of liquid drops from IAPWS-IF97 [1], [2], [3], [4]
- λ, η of liquid drops from IAPWS-85, IAPWS-08 [6], [7] (Mixture of volume fractions)

Ice fog $(Xi_w > Xi_{wsatt})$ and $t < 0°C$

Ideal mixture of saturated humid air and water ice

- saturated humid air as specified above

- v, h, s of ice crystals from IAPWS-06 [18], [19]
- λ of ice crystals as non varying value
- η, κ, w of saturated humid air

$Xi_{wsatt}(p,t)$ from saturation pressure $p_{dsatt}(p,t)$ of water in mixtures of gases

$p_{dsatt}(p,t)$ is the saturation vapor pressure from $p_{dsatt}(p,t) = f(p,t) \cdot p_s(t)$

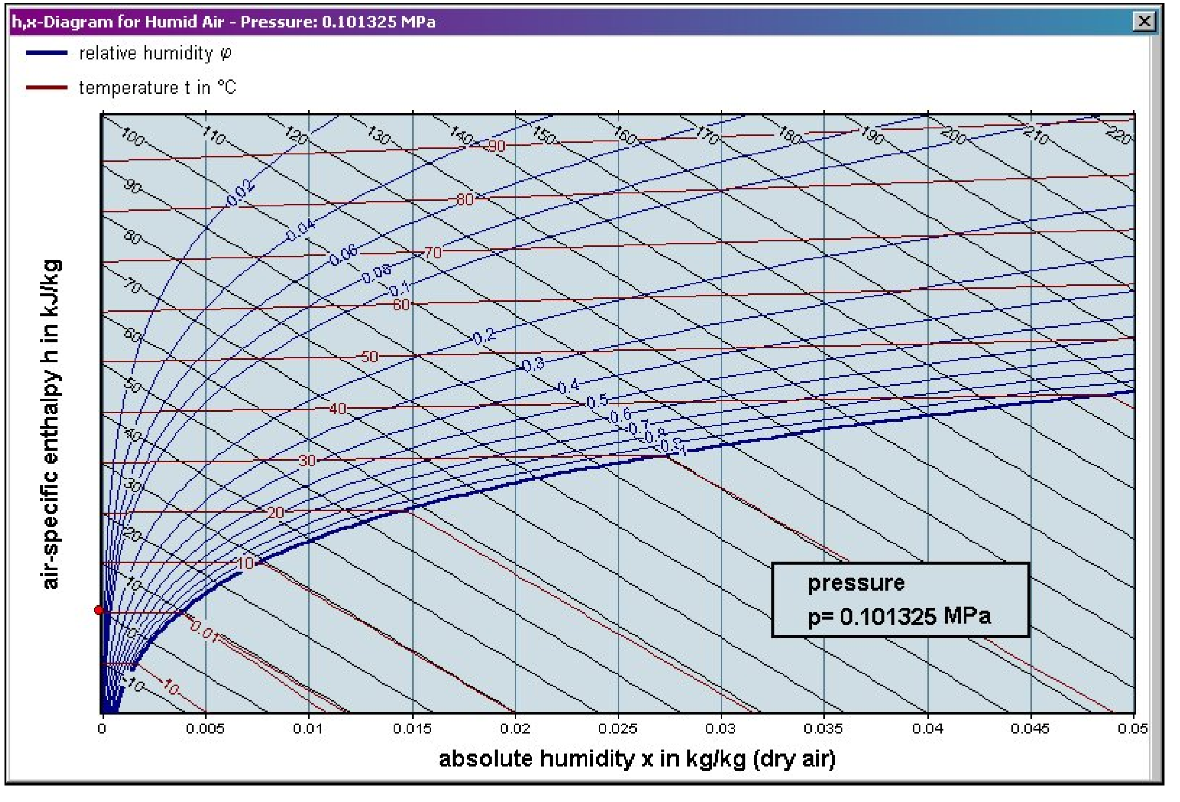
- $f(p,t)$ from Herrmann et al. [25], [26],
- $p_s(t)$ for T ≥ 0.01 °C from IAPWS - IF97 [1], [2], [3], [4],
- $p_s(t)$ for T < 0 °C from IAPWS-08 [16], [17].

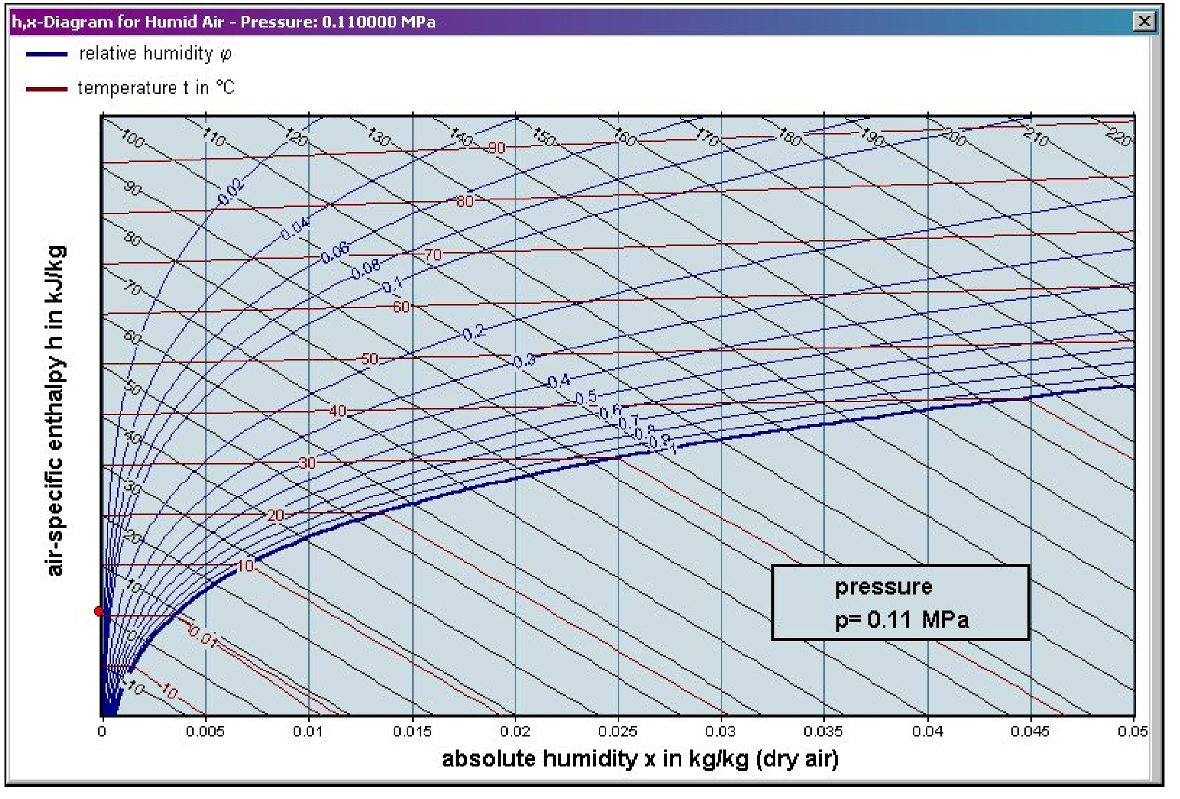
Thermodynamic Diagrams

FluidEXL*Graphics* enables the user to represent the calculated properties in the following thermodynamic diagrams:

- h,x-Diagram p = 0.101325 MPa
- h,x-Diagram p = 0.11 MPa

The diagrams, in which the calculated point will be displayed, are shown below.




h,x-Diagram for Humid Air - Pressure: 0.110000 MPa
relative humidity φ
temperature t in °C
air-specific enthalpy h in kJ/kg
absolute humidity x in kg/kg (dry air)
pressure
p= 0.11 MPa